# Online Difficulty Filtering for Reasoning Oriented Reinforcement Learning

Sanghwan Bae[1*] Jiwoo Hong[2*†] Min Young Lee[1] Hanbyul Kim[1] JeongYeon Nam[3†] Donghyun Kwak[1]

NAVER Cloud[1] KAIST AI[2] TwelveLabs[3]

baaesh10@gmail.com, jiwoo_hong@kaist.ac.kr

## Abstract

Recent advances in reinforcement learning with verifiable rewards (RLVR) show that large language models enhance their reasoning abilities when trained with verifiable signals. However, due to reward sparsity, effectiveness depends heavily on selecting samples of appropriate difficulty. In this work, we present a formal analysis of online difficulty-aware filtering and establish its theoretical foundations. We show that expected policy improvement is lower-bounded by the variance of task-level success probabilities, implying that selecting tasks of intermediate difficulty maximizes learning efficiency. Building on this, we demonstrate that balanced filtering maximizes this lower bound, leading to superior performance and sample efficiency. Evaluations across multiple math reasoning benchmarks validate that balanced filtering consistently enhances convergence speed and final performance, achieving up to +12% gains in less than half the training steps of standard GRPO. By extending our analysis to various reward distributions, we provide a principled foundation for future RLVR curriculum strategies, confirmed through both theoretical analysis and extensive empirical results.

## 1 Introduction

Reinforcement learning (RL) has become a key training paradigm for training large language models (LLMs) to further enhance their generational capabilities (Ouyang et al., 2022; Touvron et al., 2023; Yang et al., 2024a; Walsh et al., 2025), often posed as *post-training*. Specifically, reinforcement learning with verifiable rewards (RLVR) that have a discrete set of correct answers for domains like math reasoning is emerging as a new application of RL (OpenAI et al., 2024; Lambert et al., 2025a; Guo et al., 2025). Despite its effectiveness, the training efficiency is the main bottleneck in RL for LLMs, which recent works try to overcome either by hardware optimization (Mei et al., 2025; Noukhovitch et al., 2025) or algorithmic solutions (Lee and Lim, 2024; Ahmadian et al., 2024).

Efficient learning, *i.e.*, achieving the optimal performance with less data, has long been studied in the education domain, where theories such as the Zone of Proximal Development (Cole, 1978; Tzannetos et al., 2023, ZPD) emphasize that learning is most efficient when tasks are *neither too easy nor too hard*, but instead fall within a learner's optimal challenge zone. This has motivated a variety of strategies in general language modeling (Platanios et al., 2019; Maharana and Bansal, 2022; Xie et al., 2023). When the idea of filtering the data with intermediate difficulty is applied to RLVR, it can be used for progressively introducing harder problems (Team et al., 2025) or filtering examples based on the pre-defined proxies (Muennighoff et al., 2025; Ye et al., 2025; Yang et al., 2025). Especially, by setting the proxy as the training policy's capability, we can harness the online learning in RLVR for difficulty-aware efficient learning (Cui et al., 2025). Despite applying difficulty-aware data curation can be found in recent works for RLVR, they often lack detailed analysis on *how the theory of ZPD can be linked to online reinforcement learning algorithms*, *i.e.*, theoretical foundation of online difficulty-aware filtering in RLVR.

In this work, we establish a theoretical foundation for online difficulty-aware filtering in RLVR through identifying the sample reward variance as the lower bound of the reversed KL divergence between the initial policy and the optimal policy, thereby being an effective proxy for filtration. With a novel asynchronous sampling strategy that replaces filtered-out items with parallel rollouts and retains the batch size, we build a theoretical and empirical background a reliable filtering strategy for RLVR. Our contributions are as follows:

[*] Equal contribution.

[†] Work was done while the authors were at NAVER Cloud.

*Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics*
*Volume 1: Long Papers*, pages 700–719
March 24-29, 2026 ©2026 Association for Computational Linguistics

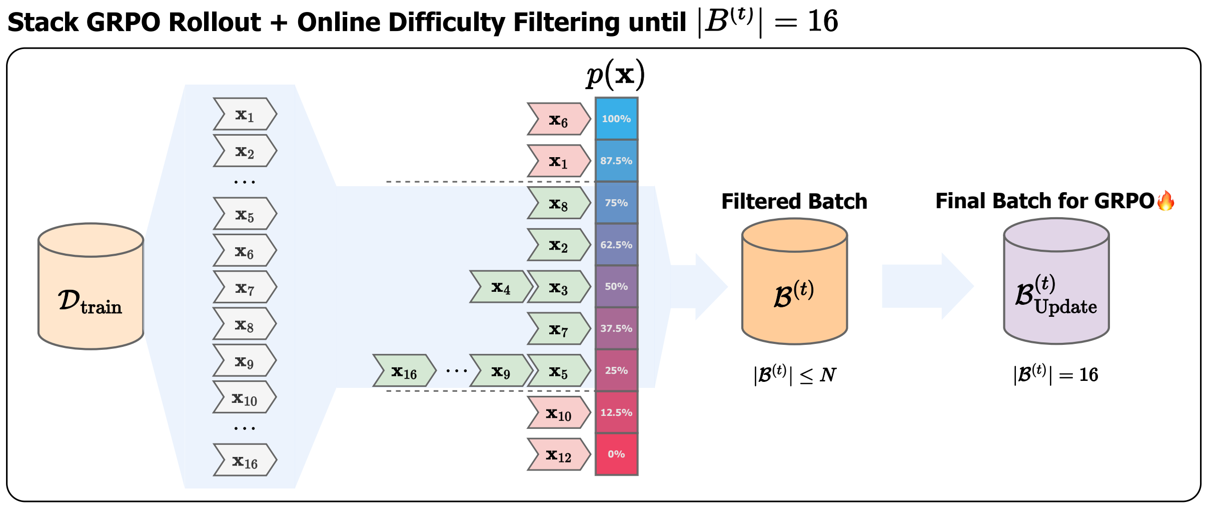


Figure 1: **Balanced online difficulty filtering** for maximizing the effectiveness of GRPO. With $G$ rollouts for each prompt $\mathbf{x}$, we measure the pass rate $p(\mathbf{x})$ as the average accuracy and filter them by predefined thresholds: *e.g.*, $0.25 \leq p(\mathbf{x}) \leq 0.75$ in this case. We recursively stack filtered prompts until the train batch size meets the fixed size $N$. We elaborate on the asynchronous implementation in Appendix B.

1. **Theoretical assessment of prompt-level learnability** (§3): We prove that the learning signal for the prompt, given the initial policy, can be approximated via sample reward variance (Proposition 3.1), implying why the prompts with extremely low or high pass rate should be filtered (Remarks 3.3 and 3.2).
2. **Empirical scalability and generalizability of balanced online difficulty filtering** (§5): With 3B and 7B models, we empirically validate the effectiveness of the balanced online difficulty filtering across five math reasoning benchmarks with varying levels, *e.g.*, $+10\%$ on AIME and $+4.2\%$ in average for 3B and $+12\%$ on AMC and $+4.5\%$ in average for 7B.
3. **Sample efficiency of balanced online difficulty filtering** (§6): We show that the filtering in fact decreases the amount of training data and time in achieving the optimal performance, using less than half of gradient updates to outperform the plain GRPO with filtering.
4. **Generalizability of sample reward variance as learnability proxy** (§6.3): We provide a general proof that wider range of reward distributions, *e.g.*, Gaussian (Corollary 6.2) or Multinomial (Corollary 6.3), can also enjoy the sample reward variance as the learnability proxy.

## 2 Preliminaries

**Reinforcement learning in language models.** Given the training policy $\pi_\theta$ initialized from the reference policy $\pi_{\text{init}}$, reinforcement learning (RL) in language model environment optimizes $\pi_\theta$ to maximize the reward assessed by the reward function $r$ (Christiano et al., 2017; Ziegler et al., 2020):

$$\max_\theta \mathbb{E}_{\mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}\left[r(\mathbf{x},\mathbf{y})\right] - \beta\mathbb{D}_{\text{KL}}\left(\pi_\theta \| \pi_{\text{init}}\right), \quad (1)$$

penalizing excessive divergence of $\pi_\theta$ with hyperparameter $\beta$ for the input and output token sequences $\mathbf{y} = \{y_i\}_{i=1}^K$ and $\mathbf{x} = \{x_i\}_{i=1}^M$. The policy gradient methods like REINFORCE (Williams, 1992) or PPO (Schulman et al., 2017) are often applied, defining *token-level* reward with the per-token divergence as a final reward (Ziegler et al., 2020; Huang et al., 2024):

$$r(\mathbf{x},\mathbf{y}) - \beta \log \frac{\pi_\theta(\mathbf{y}|\mathbf{x})}{\pi_{\text{init}}(\mathbf{y}|\mathbf{x})}. \quad (2)$$

The corresponding optimal policy $\pi^*$ is well known to be defined with respect to $\pi_{\text{init}}$ as (Korbak et al., 2022; Go et al., 2023; Rafailov et al., 2023),

$$\pi^*(\mathbf{y}|\mathbf{x}) = Z(\mathbf{x})\pi_{\text{init}}(\mathbf{y}|\mathbf{x})e^{\frac{1}{\beta}r(\mathbf{x},\mathbf{y})}, \quad (3)$$

where $Z(\mathbf{x})$ is the partition function that normalizes the action probability given $\mathbf{x}$.

**Group relative policy optimization.** Unlike PPO, recent works exclude parameterized value models (Ahmadian et al., 2024; Kazemnejad et al., 2024; Wu et al., 2024), including group relative policy optimization (Shao et al., 2024, GRPO). GRPO leverages the PPO-style clipped surrogate objective but calculates the policy gradient by weighting the log-likelihood of each trajectory with its advantage, thus removing the need for a critic (Vojnovic and Yun, 2025; Mroueh, 2025). For each prompt, $G$ sampled responses and their reward $r_i$ is used to calculate the advantage $\hat{A}_i$:

$$\hat{A}_i = \frac{r_i - \mathrm{mean}(r_1, \ldots, r_G)}{\mathrm{std}(r_1, \ldots, r_G)}, \tag{4}$$

where $\mathrm{mean}(\cdot)$ and $\mathrm{std}(\cdot)$ are the average and standard deviation of the input values. The effectiveness of GRPO is especially highlighted in the tasks with verifiable reward stipulated through the binary reward functions (Lambert et al., 2024; Guo et al., 2025; Wei et al., 2025b):

$$r_{\text{acc}}(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 & \text{if output is correct} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

## 3 Learnability in GRPO and Online Difficulty Filtering

We study the *learnability* of prompts in reinforcement learning with language model environments under binary rewards. Our analysis shows that prompts that are either trivially easy or impossibly hard yield zero divergence and thus no learning signal, while intermediate prompts with higher reward variance maximize the effective gradient information. These results are formalized in Proposition 3.1, which motivates a **balanced online difficulty filtering** strategy (§3.3-§3.4) for optimizing GRPO training.

### 3.1 Background

The optimal value function and the partition function in the soft RL setting (Schulman et al., 2018; Richemond et al., 2024) are defined as:

$$V^*(\mathbf{x}) := \beta \log \mathbb{E}_{\mathbf{y}\sim\pi_{\text{init}}(\cdot|\mathbf{x})}\left[e^{\frac{1}{\beta}r(\mathbf{x},\mathbf{y})}\right] \tag{6}$$

$$Z(\mathbf{x}) = \exp\left(\frac{1}{\beta}V^*(\mathbf{x})\right). \tag{7}$$

Using $V^*(\mathbf{x})$ in equation 3, the log ratio between $\pi_{\text{init}}$ and $\pi^*$ can be expressed as:

$$\log\frac{\pi^*(\mathbf{y}|\mathbf{x})}{\pi_{\text{init}}(\mathbf{y}|\mathbf{x})} = \frac{1}{\beta}\Big(r(\mathbf{x},\mathbf{y}) - V^*(\mathbf{x})\Big). \tag{8}$$

Taking the expectation with respect to $\pi_{\text{init}}$ yields:

$$\begin{aligned}&\mathbb{E}_{\mathbf{y}\sim\pi_{\text{init}}(\cdot|\mathbf{x})}\left[\log\frac{\pi^*(\mathbf{y}|\mathbf{x})}{\pi_{\text{init}}(\mathbf{y}|\mathbf{x})}\right] \\ &= \frac{1}{\beta}\mathbb{E}_{\mathbf{y}\sim\pi_{\text{init}}(\cdot|\mathbf{x})}\left[r(\mathbf{x},\mathbf{y})\right] - \frac{1}{\beta}V^*(\mathbf{x}),\end{aligned} \tag{9}$$

where the right-hand side (RHS) is a soft-RL variant of the advantage function scaled by $\beta^{-1}$ (Haarnoja et al., 2017; Schulman et al., 2018), as $\mathbb{E}_{\pi_{\text{init}}}[r(\mathbf{x},\mathbf{y})]$ can be interpreted as a Q-function. And the left-hand side (LHS) corresponds to the negative reverse KL divergence between $\pi_{\text{init}}$ and $\pi^*$ (Rafailov et al., 2024):

$$\begin{aligned}&\mathbb{D}_{\text{KL}}\left(\pi_{\text{init}}(\mathbf{y}|\mathbf{x})|\pi^*(\mathbf{y}\parallel\mathbf{x})\right) \\ &= -\mathbb{E}_{\mathbf{y}\sim\pi_{\text{init}}(\cdot|\mathbf{x})}\left[\log\frac{\pi^*(\mathbf{y}|\mathbf{x})}{\pi_{\text{init}}(\mathbf{y}|\mathbf{x})}\right].\end{aligned} \tag{10}$$

**Learnability in binary reward case.** For the binary reward $r_{\text{acc}}$, the reward distribution is Bernoulli with parameter $p(\mathbf{x})$ for prompt $\mathbf{x}$, policy $\pi$, and $\mathbf{y} \sim \pi(\cdot|\mathbf{x})$, which we refer as "*pass rate*":

$$p(\mathbf{x}) = \mathbb{E}_{\pi_{\text{init}}}\left[r_{\text{acc}}(\mathbf{x},\mathbf{y})\right], \tag{11}$$

and variance $p(\mathbf{x})(1 - p(\mathbf{x}))$. Here, we categorize the prompts into five categories:

1. **Absolute-hard** ($\mathbf{x}_{\text{Hard}},\ p(\mathbf{x}_{\text{Hard}}) = 0$)
2. **Soft-hard** ($\mathbf{x}_{\text{hard}}, p(\mathbf{x}_{\text{hard}}) = \epsilon$)
3. **Intermediate** ($\mathbf{x}_{\text{inter}},\ \epsilon \leq p(\mathbf{x}_{\text{inter}}) \leq 1-\epsilon$)
4. **Soft-easy** ($\mathbf{x}_{\text{easy}},\ p(\mathbf{x}_{\text{easy}}) = 1-\epsilon$)
5. **Absolute-easy** ($\mathbf{x}_{\text{Easy}},\ p(\mathbf{x}_{\text{Easy}}) = 1$)

where $\epsilon$ is a small positive constant satisfying $0 \ll \epsilon < 0.5$. The variance is zero if and only if $p(\mathbf{x}) = 0$ or $p(\mathbf{x}) = 1$, corresponding to *absolute hard* and *absolute easy* prompts, respectively.

### 3.2 Prompt-level learnability: theoretical analysis

We analyze the learnability of prompts under the soft reinforcement learning formulation defined in §3.1. Let the binary verifiable reward $r_{\text{acc}}(x, y) \in \{0, 1\}$ follow a Bernoulli distribution,

$$p(x) := \mathbb{E}_{y\sim\pi_{\text{init}}(\cdot|x)}[r_{\text{acc}}(x,y)], \tag{12}$$

which we refer to as the *pass rate* of prompt $x$. Given the definitions of the value function and log-ratio, the expected log ratio between $\pi_{\text{init}}$ and the soft-optimal policy $\pi^*$ can be written as

$$\begin{aligned}&\mathbb{E}_{y\sim\pi_{\text{init}}(\cdot|x)}\left[\log\frac{\pi^*(y|x)}{\pi_{\text{init}}(y|x)}\right] \\ &= \frac{p(x)}{\beta} - \log\Big((1-p(x)) + p(x)e^{1/\beta}\Big),\end{aligned} \tag{13}$$

and its negative corresponds to the reverse KL divergence $\mathbb{D}_{\text{KL}}\left(\pi_{\text{init}}(\cdot|x)\parallel\pi^*(\cdot|x)\right)$ defined in equation 9.

**Proposition 3.1** (Variance-controlled separation and degeneracy)**.** *For any prompt $x$ and temperature $\beta > 0$, the reverse KL divergence between the initial and optimal policies, $\pi_{\text{init}}$ and $\pi^*$, satisfies*

$$\mathbb{D}_{\text{KL}}\left(\pi_{\text{init}}(\cdot|x)\|\pi^*(\cdot|x)\right) \;\geq\; \frac{p(x)\big(1-p(x)\big)}{2\beta^2}, \tag{14}$$

*with the bound maximized at $p(x) = \frac{1}{2}$. In particular, for absolute-hard or absolute-easy prompts where $p(x) \in \{0, 1\}$, the divergence vanishes and $\pi_{\text{init}}$ is already optimal:*

$$\mathbb{D}_{\text{KL}}\left(\pi_{\text{init}}(\cdot|x)\,\|\,\pi^*(\cdot|x)\right) = 0. \tag{15}$$

*Proof sketch.* Starting from equation 13 and expanding $\log\left((1-p)+pe^{1/\beta}\right)$ with $\log(1+\epsilon) \geq \epsilon - \frac{\epsilon^2}{2}$ for $\epsilon = p(x)(1/\beta + 1/(2\beta^2))$, we obtain

$$\mathbb{E}_{y\sim\pi_{\text{init}}}\left[\log\frac{\pi^*(y|x)}{\pi_{\text{init}}(y|x)}\right] \leq -\frac{p(x)\big(1-p(x)\big)}{2\beta^2}. \tag{16}$$

Since the reverse KL divergence is the negative of this expectation equation 9, the inequality in equation 14 follows. When $p(x) \in \{0, 1\}$, both the expected reward and the soft value function coincide ($V^*(x) = r_{\text{acc}}(x, y)$), yielding zero divergence. The complete proof is provided in Appendix C. □

**Remark 3.2** (Learnability in absolute prompts: No learning signal at extremes)**.** *For absolute-hard or absolute-easy prompts ($p(x) \in \{0, 1\}$), the advantage term in GRPO equation 4 becomes zero for all rollouts, indicating that such prompts provide no gradient signal during policy optimization.*

**Remark 3.3** (Learnability in soft prompts: Maximal learnability band)**.** *The lower bound in equation 14 is proportional to the Bernoulli variance $p(x)(1 - p(x))$, which attains its maximum at $p(x) = 0.5$. Hence, prompts whose pass rates lie in the intermediate region $\epsilon \leq p(x) \leq 1-\epsilon$ provide the strongest learning signal, while soft-hard ($p(x) \approx \epsilon$) and soft-easy ($p(x) \approx 1 - \epsilon$) prompts contribute only marginally.*

Proposition 3.1 unifies both extreme and intermediate cases of prompt learnability. When $p(x)$ approaches the extremes, the reverse KL divergence $\mathbb{D}_{\text{KL}}\left(\pi_{\text{init}}\|\pi^*\right)$ tends to zero, implying no separation between policies and thus no effective update. Conversely, when $p(x) \approx 0.5$, the reward variance, *i.e.*, learnability, is maximized. These insights directly motivate the filtering criterion in our online curriculum, which selectively retains prompts within the intermediate difficulty band to ensure the highest learning efficiency.

### 3.3 Method: online difficulty filtering with fixed batch size

From this vein, it is reasonable to comprise the input prompt set with *intermediate* difficulty. Furthermore, balanced difficulty in the prompt set encourages balanced model updates for penalizing bad trajectories and reinforcing good trajectories in GRPO (Mroueh, 2025).

We analyze an online difficulty filtering approach that ensures a fixed batch size throughout training for a reasoning-oriented agent. Unlike static curricula with predefined difficulty orderings in problems (Yang et al., 2024c; Team et al., 2025; Li et al., 2025), our approach dynamically assesses difficulty *on the fly* in each training step and applies difficulty filtering logic following the theoretical insights studied in §3. We describe the detailed process in Algorithm 1 and the high-level illustration of the algorithm in Figure 4 in Appendix B.

**Online difficulty filtering with sample success rate for learnability.** First, we fill the batch $\mathcal{B}^{(t)}$ of the training step $t$ with filtered examples by measuring the success rate $p(\mathbf{x})$, equation 11, of each prompt $\mathbf{x}$ using sampled rollouts with size of $G$ as in equation 25. With the predefined difficulty threshold $T_{\text{Low}}$ and $T_{\text{High}}$, we asynchronously filter and fill the batch to meet the fixed batch size.

**Ensuring fixed batch size with asynchronous sampling and efficient batching.** While we showed that online difficulty filtering could maximize learnability in GRPO, naive filtering could result in inconsistent training batch size, leading to training instability and degraded performance (Li et al., 2022). For this reason, we ensure the fixed batch size to $|\mathcal{B}| = N$ for the batch $\mathcal{B}^{(t)}$ at the training step $t$ as in equation 25. We extend technical details in Appendix B.

### 3.4 Difficulty filtering strategies

Based on the literature study in Appendix A, we experiment two different difficulty filtering strategies, **balanced** and **skewed** difficulty filtering:

1. **Balanced difficulty filtering**: We set the thresholds to be symmetric to the success rate of 0.5: *e.g.*, $T_{\text{High}} = 0.8$ and $T_{\text{Low}} = 0.2$.

2. **Skewed difficulty filtering**: We set asymmetric thresholds, only filtering either easy or hard prompts: *e.g.*, $T_{\text{High}} = 0.6$ and $T_{\text{Low}} = 0$.

**Algorithm 1** Iterative GRPO with Online Difficulty Filtering

```
Require: Initial policy model $\pi_{\text{init}}$; Reward $r$; Prompts queue $\mathcal{Q}$; Pass rate thresholds $T_{\text{Low}}, T_{\text{High}}$; Batch size $N$; Group size $G$; $r_{\text{acc}}$ equation 5; Visit count $\text{vc}(\mathbf{x})$.
$\mathcal{P}_{\text{active}}$: The set of examples currently undergoing asynchronous rollout.
$C_{\text{max}}$: The maximum number of examples that can be processed concurrently.
function $f_{async}(\mathbf{x})$
    $\{\mathbf{y}_i\}_{i=1}^G \sim \pi_\theta(\cdot \mid \mathbf{x})$
    if $T_{\text{Low}} \leq \frac{1}{G}\sum_{i=1}^G r_{acc}(\mathbf{x}, \mathbf{y}_i) \leq T_{\text{High}}$ then
        $\mathcal{B}^{(t)} \leftarrow \mathcal{B}^{(t)} \cup \{(\mathbf{x}, \{\mathbf{y}_i\}_{i=1}^G, \{r(\mathbf{x}, \mathbf{y}_i)\}_{i=1}^G)\}$
    $\text{vc}(\mathbf{x}) \leftarrow \text{vc}(\mathbf{x}) + 1$
Initialize policy model $\pi_\theta \leftarrow \pi_{\text{init}}$
Initialize visit count $\text{vc}(\mathbf{x}) \leftarrow 0$ for all $\mathbf{x} \in \mathcal{D}$
for iteration $= 1, \ldots, I$ do
    Initialize reference model $\pi_{\text{ref}} \leftarrow \pi_\theta$
    for step $= 1, \ldots, M$ do
        Initialize $\mathcal{B}^{(t)} \leftarrow \varnothing$, $\mathcal{P}_{\text{active}} \leftarrow \varnothing$
        Sort examples by visit count $\mathcal{Q} \leftarrow \text{sort}_{\text{vc}}(\mathcal{D})$
        while $|\mathcal{B}^{(t)}| < N$ do
            if $|\mathcal{P}_{\text{active}}| < C_{\text{max}}$ then
                $\mathbf{x} \leftarrow \text{nextExample}(\mathcal{Q})$
                $\mathcal{P}_{\text{active}} \leftarrow \mathcal{P}_{\text{active}} \cup f_{async}(\mathbf{x})$
        Cancel $\mathcal{P}_{\text{active}}$
        Compute $\hat{A}_i$ for $\mathbf{y}_i$ in $\mathcal{B}^{(t)}$ through group relative advantage estimation equation 4.
        Update the policy model $\pi_\theta$ by maximizing the GRPO objective.
Output $\pi_\theta$
```

We test if incorporating either side of extreme pass rate cases can boost the performance of online difficulty filtering, even though the learnability for either side has the same lower bound as in §3.2.

# 4 Experiments

## 4.1 Experimental Setup

**Supervised fine-tuning.** Before reinforcement learning with verifiable rewards (RLVR) experiments, we fine-tune Qwen2.5-3B base (Yang et al., 2024b) as a cold start, following Guo et al. (2025). Specifically, we curate 1.1K verified problem-solution pairs, with math problems sampled from NuminaMath (Li et al., 2024) and solutions distilled from DeepSeek-R1 (Guo et al., 2025).

**Reinforcement learning with verifiable rewards.** For RLVR, we employ GRPO on top of the SFT checkpoint. In each training step, the model generates 16 rollouts for 16 prompts drawn from NuminaMath problems and receives a reward based on their correctness. We leave out 1,024 problems as a validation set. We also add a format reward and a language reward as in Guo et al. (2025). Additional training details are reported in the Appendix D.

## 4.2 Experimental design

**Different strategies in online difficulty filtering.** Along with the plain GRPO without any prompt filtering, we test the online difficulty filtering with two different strategies introduced in §3.4: *i.e.*, balanced and skewed filtering. For the balanced setting, we test $(T_{\text{Low}}, T_{\text{High}}) \in \{(0, 1), (0.1, 0.9), (0.2, 0.8), (0.3, 0.7), (0.4, 0.6)\}$. For a skewed setting, we sweep $T_{\text{Low}}$ in $\{0, 0.2, 0.4\}$ when $T_{\text{High}} = 1$ and $T_{\text{High}}$ in $\{0.6, 0.8, 1\}$ when $T_{\text{Low}} = 0$.

**Comparison against existing offline filtering methods.** We mainly compare two offline difficulty filtering methods with our approach: offline data curation (Yang et al., 2024c; Cui et al., 2025; Muennighoff et al., 2025; Ye et al., 2025) and offline scheduling (Team et al., 2025; Li et al., 2025). Offline data curation refers to the strategy that filters the problems by their difficulty before training, and offline scheduling additionally orders the training batches accordingly. For both offline strategies, we used Qwen2.5-7B-Instruct (Yang et al., 2024b) or our SFT model as the difficulty proxies.

**Evaluation Benchmarks.** We evaluate pass@1 across math reasoning benchmarks of varying difficulty levels: MATH500 (Hendrycks et al., 2021), AIME (Li et al., 2024), AMC (Li et al., 2024), MinervaMath (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024) (See Appendix E).

Table 1: Five math reasoning benchmark evaluation results with Qwen2.5-3B. "Minerva." and "Olympiad." refer to MinervaMath and OlympiadBench. "External" and "Initial" in offline filtering indicate using Qwen2.5-7B-Instruct and our SFT model as a difficulty proxy for filtering. $p(\mathbf{x})$ equation 11 is the pass rate, the average correctness of rollouts. The highest and the second highest scores in each benchmark are highlighted with **bold** and underline.

| Method | Difficulty Filter | MATH500 | AIME | AMC | Minerva. | Olympiad. | Avg. |
|---|---|---|---|---|---|---|---|
| **SFT** | - | 49.8 | 0.0 | 20.5 | 13.2 | 17.3 | 20.2 |
| **GRPO w/ Offline Filtering** | **Curation** | | | | | | |
| | External model | 59.6 | 6.6 | 27.7 | 24.3 | 23.9 | 28.4 |
| | Initial model | 55.6 | 10.0 | 28.9 | 18.8 | 18.2 | 26.3 |
| | **Schedule** | | | | | | |
| | External model | 57.8 | 10.0 | 28.9 | 20.6 | 21.5 | 27.8 |
| | Initial model | 57.0 | 3.3 | 28.9 | 19.1 | 24.9 | 26.7 |
| **GRPO w/ Online Filtering (*Ours*)** | **Plain** | | | | | | |
| | $0 \leq p(\mathbf{x}) \leq 1$ | 57.2 | 3.3 | 30.1 | 18.7 | 22.2 | 26.3 |
| | **Skewed** | | | | | | |
| | $0 < p(\mathbf{x}) \leq 1$ | 57.0 | 0.0 | 26.5 | 19.8 | 21.4 | 24.9 |
| | $0.2 < p(\mathbf{x}) \leq 1$ | 60.4 | 0.0 | 27.7 | 17.2 | 24.5 | 25.9 |
| | $0.4 < p(\mathbf{x}) \leq 1$ | 55.8 | 0.0 | 21.7 | 19.9 | 21.6 | 23.8 |
| | $0 \leq p(\mathbf{x}) < 1.0$ | 55.4 | 3.3 | 22.8 | 19.8 | 19.8 | 24.2 |
| | $0 \leq p(\mathbf{x}) < 0.8$ | 56.2 | 0.0 | 28.9 | 17.2 | 21.7 | 24.8 |
| | $0 \leq p(\mathbf{x}) < 0.6$ | 56.2 | 3.3 | 26.5 | 21.3 | 21.6 | 25.8 |
| | **Balanced** | | | | | | |
| | $0 < p(\mathbf{x}) < 1$ | 60.8 | 3.3 | 31.3 | 18.0 | **27.3** | 27.3 |
| | $0.1 < p(\mathbf{x}) < 0.9$ | 58.8 | **13.3** | 25.3 | 22.4 | 22.2 | 28.4 |
| | $0.2 < p(\mathbf{x}) < 0.8$ | 62.2 | 10.0 | 30.1 | 20.5 | 26.3 | 29.8 |
| | $0.3 < p(\mathbf{x}) < 0.7$ | **64.6** | 6.6 | 28.9 | **25.4** | 24.7 | **30.1** |
| | $0.4 < p(\mathbf{x}) < 0.6$ | 60.2 | 6.6 | **32.8** | 25.0 | 24.9 | 29.9 |

## 5 Results

We first compare different online filtering strategies in §5.1 and expand to existing offline difficulty filtering methods in §5.2.

### 5.1 Balanced and skewed filtering

**Balanced online difficulty filtering consistently outperforms plain GRPO.** In Table 1, balanced filtering ("Balanced") outperforms the plain GRPO ("Plain") on the average score of five challenging math reasoning benchmarks in all five threshold choices. While fine-tuning the SFT checkpoint with plain GRPO without filtering reaches an average score of 26.3%, balanced filtering achieves over 30%, with overall improvements across the benchmarks. For instance, balanced filtering achieved up to 10% point improvement in AIME, which is the most difficult benchmark as shown through the accuracy in Table 1. This supports our theoretical analysis in §3, as online difficulty filtering enhances the effectiveness of GRPO training compared to the plain version without any filtering.

**Progressively stricter threshold in balanced filtering incrementally improves performance.** By tightening the pass rate threshold $(T_{\text{Low}}, T_{\text{High}})$ for balanced filtering in Table 1, the average score of five benchmarks starts from 27.3% in $(0, 1)$, gradually increasing until over 30% in $(0.3, 0.7)$. Furthermore, simply removing examples in $(0, 1)$ that do not contribute to learning in GRPO results in a slight improvement over the baseline, aligning with Remark 3.2, *i.e.*, $\hat{A}$ is zero for $(0, 1)$. This result suggests that excluding ineffective examples improves both performance and training efficiency by focusing updates on meaningful data. These observations are further supported by the difficulty-level analysis using MATH500 that provides five different levels in Appendix F, which shows consistent gains across different levels.

**Skewed online difficulty filtering is less effective than plain GRPO.** While skewed filtering ("Skewed") in Table 1 improves average per-

Table 2: Five math reasoning benchmark evaluations with Qwen2.5-7B. The notations follow that of Table 1.

| Method | Difficulty Filter | MATH500 | AIME | AMC | Minerva | Olympiad | Avg. |
|---|---|---|---|---|---|---|---|
| **SFT** | – | 72.6 | 12.1 | 34.9 | 32.0 | 35.1 | 37.3 |
| **GRPO w/ Online Filtering (Ours)** | **Plain** | | | | | | |
| | $0 \leq p(\mathbf{x}) \leq 1$ | 75.0 | 12.3 | 42.2 | 32.7 | 35.7 | 39.6 |
| | **Balanced** | | | | | | |
| | $0 < p(\mathbf{x}) < 1$ | 75.0 | 13.1 | 41.0 | 33.5 | 36.9 | 39.9 |
| | $0.3 < p(\mathbf{x}) < 0.7$ | **75.8** | **15.0** | **47.0** | **33.8** | **37.6** | **41.8** |

formance up to 5.7% over the SFT checkpoint, plain GRPO with 26.3% outperforms skewed filtering consistently in every threshold choice, which achieves around 24.9% to 25.9%. Overall, maximizing the expected *learnability* in GRPO enhances learning in complex reasoning tasks. As discussed in §3.3, balanced filtering emerges as the best choice since it balances between penalizing and reinforcing diverse explorations.

### 5.2 Offline and online filtering

We apply the offline difficulty filtering with implementations from previous works (Yang et al., 2024b), with balanced threshold $(T_{\text{Low}}, T_{\text{High}}) = (0.2, 0.8)$ following the results in §5.1. While both offline curation ("Curation") and offline scheduling ("Schedule") in Table 1 show marginal improvements over plain GRPO with a maximum 2.1% improvement, balanced online difficulty filtering consistently outperforms offline methods. Within offline methods, using an external difficulty assessment proxy ("External model") exceeded the case using the SFT checkpoint ("initial model") on average, but with varying results by benchmark.

### 5.3 Scalability of balanced online filtering

We adopt 7B scale model within the same Qwen2.5 family to confirm the scalability of the proposed method. In Table 2, stricter filtering thresholds $(0.3 < p(\mathbf{x}) < 0.7)$ yield the strongest performance with 3% and 5% increase in AIME and AMC, respectively. Overall, the ascending trend in Table 2 aligns with the 3B cases, demonstrating the scalability of online difficulty filtering.

## 6 Analysis

### 6.1 Learning dynamics analysis

In Figure 2, we collect the exact batches curated through balanced online difficulty filtering with $(T_{\text{Low}}, T_{\text{High}}) = (0.2, 0.8)$ and measure the "difficulty" that each model perceives through $1 - p(\mathbf{x})$ for four checkpoints: before, during, and after GRPO, along with the external proxy Qwen2.5-7B-Instruct. As anticipated, the checkpoint evaluated during GRPO maintains an average difficulty of around 0.5, dynamically providing suitably challenging examples throughout the training process. However, both before and after GRPO checkpoints perceive incremental difficulty increases across the curated batches, indicating that the training examples become objectively more challenging over time. Moreover, the external proxy model consistently perceives lower difficulty relative to the initial model but higher difficulty than the final trained model ("After GRPO").

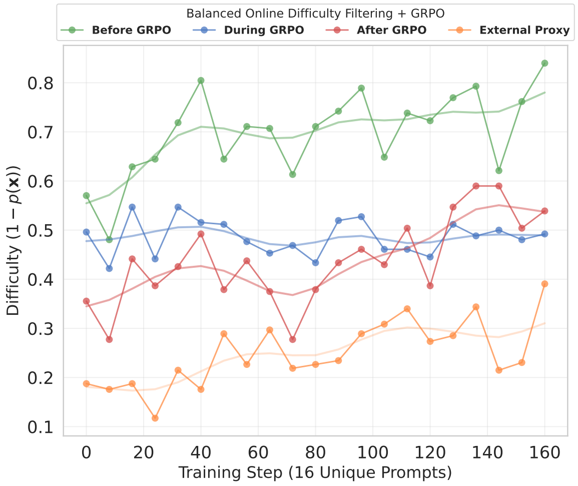
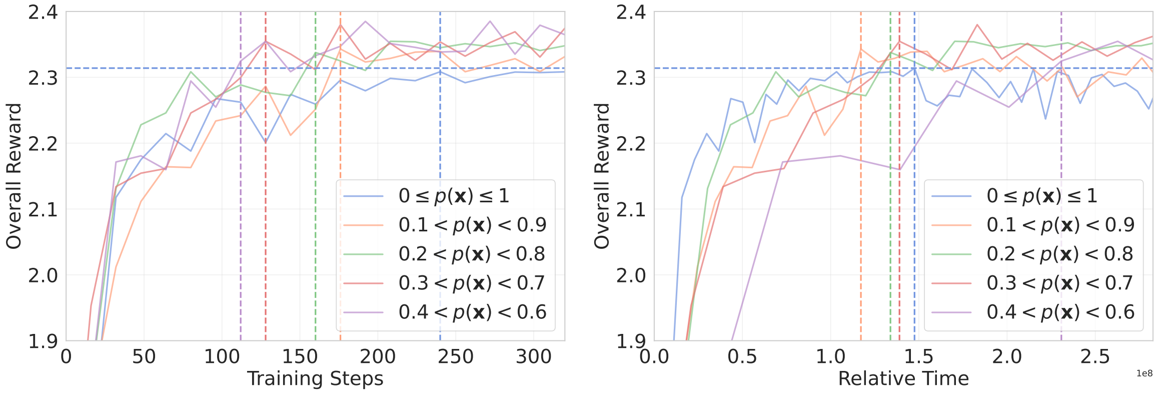


Figure 2: Perceived difficulty per batch curated through balanced online filtering. Defining "difficulty" as $1 - p(\mathbf{x})$, a greater difficulty implies lower sample accuracy.

This observation, with the results in Table 1, shows that offline difficulty filtering with external proxies can provide partially meaningful difficulty assessments while not being perfectly aligned to the training model's capability, shown through marginal improvements in Table 1 compared to plain GRPO. However, the advantage of the balanced online difficulty filtering is still evident in better performance and efficiency.



(a) Validation Accuracy over Training Steps

(b) Validation Accuracy over Training Time

Figure 3: Validation reward as a function of step (3a) and relative time (3b). The horizontal dashed line indicates the maximum reward achieved by plain GRPO, and the vertical dashed lines indicate when GRPO with each threshold surpasses the plain GRPO's maximum reward.

## 6.2 Training efficiency analysis

Figure 3 illustrates the progression of the reward in the validation set, plotted against both the training steps (3a) and the training time on the wall clock (3b). As shown in Figure 3a, models trained with balanced online difficulty filtering consistently outperform the plain GRPO ($0 \leq p(\mathbf{x}) \leq 1$) in fewer training steps. For instance, using $0.4 < p(\mathbf{x}) < 0.6$ achieved the highest overall reward and took the fewest gradient updates to outperform the plain GRPO within less than 50% of steps in Figure 3a. This suggests that by filtering out less informative examples, the average learnability within each batch increases, allowing faster learning progress. Interestingly, Figure 3b shows that this benefit carries over even when measured by wall-clock time by exceeding plain GRPO's maximum reward in less training time. However, we also observe that overly aggressive filtering, such as in the case of the $0.4 < p(\mathbf{x}) < 0.6$ setting, can require significantly more rollouts to fill a batch, leading to longer training times overall. These results suggest that online filtering can enable more efficient learning even in real-world settings.

## 6.3 Theoretical generalizability

We extend the learnability view of difficulty filtering in reinforcement learning for language models to a broader class of reward distributions. We begin by expressing the reverse KL divergence between the initial policy $\pi_{\text{init}}(\cdot)$ and the optimal policy $\pi^*(\cdot)$ as the cumulant generating function (CGF) of a *centered* reward. We report the detailed proofs for the proposition and corollaries in Appendix G.

**Proposition 6.1** (Reverse KL as CGF of centered reward)**.** *Fix a prompt $x$ and temperature $\beta > 0$. Let $\mu(x) := \mathbb{E}_{y\sim\pi_{\text{init}}(\cdot|x)}[r(x, y)]$. Then*

$$\begin{aligned}
&\mathbb{D}_{\text{KL}}\big(\pi_{\text{init}}(\cdot|x) \,\|\, \pi^*(\cdot|x)\big) \\
&= \log\, \mathbb{E}_{y\sim\pi_{\text{init}}(\cdot|x)}\left[\exp\left(\frac{r(x,y)-\mu(x)}{\beta}\right)\right] \\
&= K_{r(x,\cdot)-\mu(x)}\left(\tfrac{1}{\beta}\right),
\end{aligned} \tag{17}$$

*where $K_Z(t) := \log \mathbb{E}[\exp(tZ)]$ denotes the CGF of a random variable $Z$.*

*Proof sketch.* Starting from the soft value $V^*(x) = \beta \log \mathbb{E}[e^{r(x,y)/\beta}]$ and the identity in equation 9,

$$\begin{aligned}
\mathbb{D}_{\text{KL}}\big(\pi_{\text{init}} \| \pi^*\big) &= -\mathbb{E}\left[\log \frac{\pi^*}{\pi_{\text{init}}}\right] \\
&= \log \mathbb{E}\left[e^{r/\beta}\right] - \frac{1}{\beta}\,\mathbb{E}[r] \\
&= \log \mathbb{E}\left[e^{(r-\mu)/\beta}\right],
\end{aligned} \tag{18}$$

and the right-hand side is precisely the CGF of $r - \mu$ at $t = \frac{1}{\beta}$. □

Proposition 6.1 highlights that learnability quantified by the reverse KL divergence is governed by the *fluctuations* of the reward around its mean. The variance term used as a proxy in §3 is the leading component of this control. This CGF view naturally generalizes to two practical settings: (1) *Gaussian rewards*, representing continuous rewards from neural evaluators (Lambert et al., 2025b; Wen et al., 2025); and (2) *multinomial rewards*, representing combinations of verifiable objectives (Guo et al., 2025; Gunjal et al., 2025).

**Corollary 6.2** (Gaussian rewards). *If $r_G(x,y) \sim \mathcal{N}(\mu_G(x), \sigma_G^2(x))$ under $\pi_{\text{init}}(\cdot|x)$, then*

$$\mathbb{D}_{\text{KL}}\big(\pi_{\text{init}}(\cdot|x) \,\|\, \pi^*(\cdot|x)\big) \;=\; \frac{\sigma_G^2(x)}{2\,\beta^2}. \qquad (19)$$

For Gaussian rewards, the reverse KL divergence, *i.e.*, the learnability, is *exactly proportional* to the reward variance. Consequently, the sample variance from rollouts provides an unbiased and consistent empirical estimate of learnability. Let the sample variance

$$\widehat{\sigma_G^2}(x) \;=\; \frac{1}{n-1}\sum_{i=1}^{n}\big(r_i - \bar{r}\big)^2, \qquad (20)$$

where $r_i$ are $n$ independent reward samples. Since $\mathbb{E}[\widehat{\sigma_G^2}(x)] = \sigma_G^2(x)$,

$$\mathbb{E}\left[\frac{1}{2\beta^2}\,\widehat{\sigma_G^2}(x)\right] = \mathbb{D}_{\text{KL}}\big(\pi_{\text{init}}(\cdot|x) \,\|\, \pi^*(\cdot|x)\big). \qquad (21)$$

Thus, the sample reward variance with sufficient size of rollouts can be a reliable proxy for learnability, even in the Gaussian reward setting, such as the classifier reward models (Lambert et al., 2025b).

**Corollary 6.3** (Multinomial rewards). *Suppose $r_M(x,y) \in \{0,1,\ldots,N\}$ with variance $\sigma^2(x)$ and cumulants $\kappa_k(x)$. Then, for $t = \frac{1}{\beta}$,*

$$\begin{aligned}&\mathbb{D}_{\text{KL}}\big(\pi_{\text{init}}(\cdot|x) \,\|\, \pi^*(\cdot|x)\big) = K_{r_M-\mu}(t)\\ &= \frac{\sigma^2(x)}{2}\,t^2 + \frac{\kappa_3(x)}{6}\,t^3 + \frac{\kappa_4(x)}{24}\,t^4 + \cdots \\ &\geq \frac{\sigma^2(x)}{2\,\beta^2} - \mathcal{O}\left(\frac{1}{\beta^3}\right),\end{aligned} \qquad (22)$$

*and in particular the variance $\frac{\sigma^2(x)}{2\beta^2}$ provides a tight second-order lower control of the reverse KL.*

Here, the leading variance term is the dominant contribution to learnability, while the higher cumulants $\kappa_3, \kappa_4, \ldots$ account for skewness and tail behavior typical of multi-objective or rubric-based rewards. The binary case is a special instance with $N{=}1$. If $r_B(x,y) \in \{0,1\}$, then

$$\text{Var}[r_B(x,y)] \;=\; p(x)\big(1-p(x)\big), \qquad (23)$$

and substituting into the series above recovers the bound used in §3:

$$\mathbb{D}_{\text{KL}}\big(\pi_{\text{init}}(\cdot|x) \,\|\, \pi^*(\cdot|x)\big) \;\geq\; \frac{p(x)\big(1-p(x)\big)}{2\,\beta^2}. \qquad (24)$$

This expands the applicability of the learnability viewpoint for the online difficulty filtering to the real-world setting, where multiple verifiable rewards are engaged, *e.g.*, format rewards and accuracy rewards in reasoning (Guo et al., 2025; Luo et al., 2025; Wei et al., 2025a).

We further extend the principled discussion on algorithm-agnostic generalizability of the balanced online difficulty filtering in Appendix H.

## 7 Conclusion

In this work, we established the theoretical foundations of online difficulty-aware filtering in reinforcement learning with verifiable rewards (RLVR). Our analysis showed that tasks with intermediate difficulty maximize the lower bound of policy improvement, providing a principled explanation for the effectiveness of difficulty-based data curation, which was heuristically adopted in previous works. Through extensive ablation studies across multiple reasoning benchmarks, we verified these theoretical insights and demonstrated consistent gains in both sample efficiency and performance. These results highlight the importance of theoretically grounded difficulty control in RLVR, offering a unified perspective that connects empirical heuristics with formal learning principles.

## Limitations

Our work provides both theoretical and empirical guidelines for online difficulty filtering in reasoning-oriented reinforcement learning for language models. While our theoretical analysis can be applied to any verifiable task, as we have shown through the proposition and corollaries, our empirical validation was conducted solely on math reasoning tasks. We leave the exploration of diverse verifiable tasks, such as coding and scientific reasoning, for future work. Furthermore, we plan to investigate the broader applicability of our method to larger scales and wider model families.

## References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267.

Jiangjie Chen, Qianyu He, Siyu Yuan, Aili Chen, Zhicheng Cai, Weinan Dai, Hongli Yu, Qiying Yu, Xuefeng Li, Jiaze Chen, Hao Zhou, and Mingxuan Wang. 2025a. Enigmata: Scaling logical reasoning in large language models with synthetic verifiable puzzles. *Preprint*, arXiv:2505.19914.

Yongchao Chen, Yueying Liu, Junwei Zhou, Yilun Hao, Jingquan Wang, Yang Zhang, Na Li, and Chuchu Fan. 2025b. R1-code-interpreter: Llms reason with code via supervised and multi-stage reinforcement learning. *Preprint*, arXiv:2505.21668.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Michael Cole. 1978. *Mind in society: Development of higher psychological processes*. Harvard university press.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, and 1 others. 2025. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*.

Cade Daniel, Chen Shen, Eric Liang, and Richard Liaw. 2023. How continuous batching enables 23x throughput in llm inference while reducing p50 latency.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. 2018. Automatic goal generation for reinforcement learning agents. In *International conference on machine learning*, pages 1515–1528. PMLR.

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. 2023. Aligning language models with preferences through f-divergence minimization. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. 2025. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *Preprint*, arXiv:2507.17746.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. 2017. Reinforcement learning with deep energy-based policies. In *ICML*, pages 1352–1361.

Alexander Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. 2024. Teaching large language models to reason with reinforcement learning. In *AI for Math Workshop @ ICML 2024*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. 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)*, pages 3828–3850.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Zhenyu Hou, Xin Lv, Rui Lu, Jiajie Zhang, Yujiang Li, Zijun Yao, Juanzi Li, Jie Tang, and Yuxiao Dong. 2025. Advancing language model reasoning through reinforcement learning and inference scaling. *arXiv preprint arXiv:2501.11651*.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. 2025. Reinforce++: An efficient rlhf algorithm with robustness to both prompt and reward models. *Preprint*, arXiv:2501.03262.

Shengyi Huang, Michael Noukhovitch, Arian Hosseini, Kashif Rasul, Weixun Wang, and Lewis Tunstall. 2024. The n+ implementation details of RLHF with PPO: A case study on TL;DR summarization. In *First Conference on Language Modeling*.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. 2024. VinePPO: Accurate credit assignment in RL for LLM mathematical reasoning. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. 2022. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. In *Advances in Neural Information Processing Systems*, volume 35, pages 16203–16220. Curran Associates, Inc.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2025. Training language models to self-correct via reinforcement learning. In *The Thirteenth International Conference on Learning Representations*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, and 4 others. 2025a. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2025b. RewardBench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1755–1797, Albuquerque, New Mexico. Association for Computational Linguistics.

Bruce W Lee, Hyunsoo Cho, and Kang Min Yoo. 2024. Instruction tuning with human curriculum. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1281–1309.

Changhun Lee and Chiehyeon Lim. 2024. Towards Pareto-efficient RLHF: Paying attention to a few high-reward samples with reward dropout. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8335–8349, Miami, Florida, USA. Association for Computational Linguistics.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Conglong Li, Minjia Zhang, and Yuxiong He. 2022. The stability-efficiency dilemma: Investigating sequence length warmup for training GPT models. In *Advances in Neural Information Processing Systems*.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13:9.

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

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2. Notion Blog.

Adyasha Maharana and Mohit Bansal. 2022. On curriculum learning for commonsense reasoning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 983–992, Seattle, United States. Association for Computational Linguistics.

Zhiyu Mei, Wei Fu, Kaiwei Li, Guangju Wang, Huanchen Zhang, and Yi Wu. 2025. Real: Efficient RLHF training of large language models with parameter reallocation. In *Eighth Conference on Machine Learning and Systems*.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, and 1 others. 2025. Mm-eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning. *arXiv preprint arXiv:2503.07365*.

Youssef Mroueh. 2025. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *Preprint*, arXiv:2503.06639.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

Marwa Naïr, Kamel Yamani, Lynda Said Lhadj, and Riyadh Baghdadi. 2024. Curriculum learning for small code language models. In *62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024*, pages 531–542. Association for Computational Linguistics (ACL).

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

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard

Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 77 others. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. 2025. Generalizing verifiable instruction following. *Preprint*, arXiv:2507.02833.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. 2024. From $r$ to $q^*$: Your language model is secretly a q-function. In *First Conference on Language Modeling*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Pierre Harvey Richemond, Yunhao Tang, Daniel Guo, Daniele Calandriello, Mohammad Gheshlaghi Azar, Rafael Rafailov, Bernardo Avila Pires, Eugene Tarassov, Lucas Spangher, Will Ellsworth, Aliaksei Severyn, Jonathan Mallinson, Lior Shani, Gil Shamir, Rishabh Joshi, Tianqi Liu, Remi Munos, and Bilal Piot. 2024. Offline regularised reinforcement learning for large language models alignment. *Preprint*, arXiv:2405.19107.

John Schulman, Xi Chen, and Pieter Abbeel. 2018. Equivalence between policy gradients and soft q-learning. *Preprint*, arXiv:1704.06440.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Georgios Tzannetos, Bárbara Gomes Ribeiro, Parameswaran Kamalaruban, and Adish Singla. 2023. Proximal curriculum for reinforcement learning agents. *arXiv preprint arXiv:2304.12877*.

Milan Vojnovic and Se-Young Yun. 2025. What is the alignment objective of grpo? *Preprint*, arXiv:2502.18548.

Evan Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, and 23 others. 2025. 2 OLMo 2 furious (COLM's version). In *Second Conference on Language Modeling*.

Chenxing Wei, Jiarui Yu, Ying Tiffany He, Hande Dong, Yao Shu, and Fei Yu. 2025a. Redit: Reward dithering for improved LLM policy optimization. In *2nd Workshop on Models of Human Feedback for AI Alignment*.

Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I. Wang. 2025b. Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution. *Preprint*, arXiv:2502.18449.

Xueru Wen, Jie Lou, Yaojie Lu, Hongyu Lin, XingYu, Xinyu Lu, Ben He, Xianpei Han, Debing Zhang, and Le Sun. 2025. Rethinking reward model evaluation: Are we barking up the wrong tree? In *The Thirteenth International Conference on Learning Representations*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.

Tianhao Wu, Banghua Zhu, Ruoyu Zhang, Zhaojin Wen, Kannan Ramchandran, and Jiantao Jiao. 2024. Pairwise proximal policy optimization: Language model alignment with comparative RL. In *First Conference on Language Modeling*.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. 2023. Data selection for language models via importance resampling. In *Advances in Neural Information Processing Systems*, volume 36, pages 34201–34227. Curran Associates, Inc.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024a. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024b. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024c. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, and 1 others. 2025. Sglang: Efficient execution of structured language model programs. *Advances in Neural Information Processing Systems*, 37:62557–62583.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *Preprint*, arXiv:1909.08593.

## A Related Works

**Reinforcement learning with verifiable rewards.** Recent advancements demonstrate significant reasoning improvements in LLMs through RL (Havrilla et al., 2024; OpenAI et al., 2024; Lambert et al., 2024; Guo et al., 2025; Walsh et al., 2025; Kumar et al., 2025). OpenAI o1 (OpenAI et al., 2024) initially reported that increasing the compute during RL training and inference improves reasoning performance. DeepSeek R1 (Guo et al., 2025) further found that, in reinforcement learning with verifiable rewards (RLVR), longer responses correlate with better reasoning. Concurrent studies (Team et al., 2025; Hou et al., 2025; Luo et al., 2025) employed algorithms, such as GRPO (Shao et al., 2024) or RLOO (Ahmadian et al., 2024), relying on advantage estimation via sampling rather than PPO-like value networks. Hou et al. (2025) further found that training efficiency improved with increased sampling in RLOO, invoking the need for more sample-efficient training strategies in RLVR.

**Difficulty-based curriculum learning.** Curriculum learning has been widely adopted in fine-tuning LLMs to improve training efficiency (Lee et al., 2024; Naïr et al., 2024; Team et al., 2025; Cui et al., 2025). Static curricula, *i.e.*, offline data curation with a predetermined task difficulty, have been effective in multiple domains: instruction-tuning (Lee et al., 2024) and coding (Naïr et al., 2024; Team et al., 2025; Li et al., 2025) to name a few. In RLVR, Team et al. (2025) employs a static difficulty-based curriculum, assigning tasks at fixed difficulty levels to ensure efficient progression. Similarly, Li et al. (2025) selects a high-impact subset of training data based on a "learning impact measure". Meantime, adaptive curricula dynamically adjust task difficulty based on the learners' progress, addressing the limitations of static curricula (Florensa et al., 2018; Cui et al., 2025). Specifically, Cui et al. (2025) applied adaptive filtering in reasoning and reported an empirical advantage in reducing reward variance. However, Meng et al. (2025) observed that such dynamic exclusion of examples may destabilize training, as it causes fluctuations in the effective batch size.

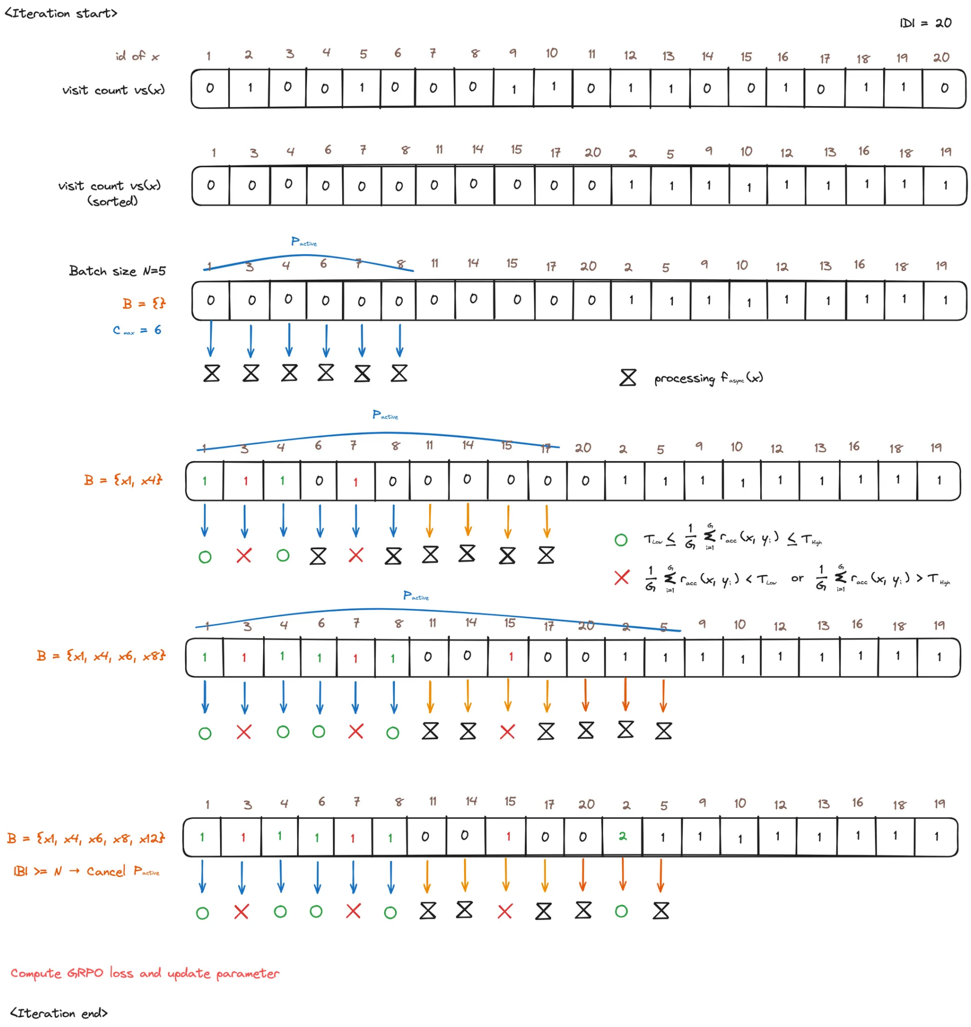
These works introduce adaptive filtering heuristics for RLVR but do not characterize *when* and *why* such filtering helps or how it interacts with KL-regularized RL objectives. In contrast, our work (1) derives a reverse-KL lower bound that identifies reward variance as the key learnability proxy, (2) shows that this directly motivates intermediate-difficulty selection, and (3) proposes an asynchronous fixed-batch implementation that avoids the batch-size instability.

## B Asynchronous Implementation of Online Difficulty Filtering

We provide a detailed diagram depicting the practical implementation of the online difficulty filtering, especially with the asynchronous setting (Noukhovitch et al., 2025). The formal expression of filling the batch $\mathcal{B}^{(t)}$ for the balanced online difficulty filtering is:

$$\mathcal{B}^{(t)} = \left\{ \left(\mathbf{x}, \{\mathbf{y}_i, r_{\text{acc}}(\mathbf{x}, \mathbf{y}_i)\}_{i=1}^{G}\right) \mid T_{\text{Low}} \leq \frac{1}{G} \sum_{i=1}^{G} r_{\text{acc}}(\mathbf{x}, \mathbf{y}_i) \leq T_{\text{High}},\ \mathbf{y}_i \sim \pi_{\theta_t}(\cdot|\mathbf{x}) \right\}. \tag{25}$$

Here, the sample mean of $r_{acc}(\mathbf{x}, \mathbf{y}_i)$ is an unbiased estimate of $\mathbb{E}_{\mathbf{y} \sim \pi_{\theta_t}(\cdot|\mathbf{x})} \left[r_{\text{acc}}(\mathbf{x}, \mathbf{y})\right]$. Rollouts for each prompt are sampled asynchronously and in parallel, enabling continuous batching of prompts and rollouts (Daniel et al., 2023; Kwon et al., 2023; Noukhovitch et al., 2025). Each prompt's visit count, $\text{vc}(\mathbf{x})$, is incremented after generating $G$ rollouts, ensuring it isn't re-processed in the same iteration. Moreover, the active rollout process $\mathcal{P}_{\text{active}}$ is halted once the batch capacity is reached, allowing prompt training with the collected data. This sampling-based framework is compatible with Monte Carlo methods such as RLOO (Ahmadian et al., 2024) and VinePPO (Kazemnejad et al., 2024).


<Iteration start>
|D| = 20
id of x
visit count vs(x)
visit count vs(x) (sorted)
Batch size N=5
B = {}
C max = 6
P active
processing f async (x)
B = {x1, x4}
$T_{Low} \leq \frac{1}{G} \sum_{i=1}^{G} r_{acc}(x, y_i) \leq T_{High}$
$\frac{1}{G} \sum_{i=1}^{G} r_{acc}(x, y_i) < T_{Low}$ or $\frac{1}{G} \sum_{i=1}^{G} r_{acc}(x, y_i) > T_{High}$
B = {x1, x4, x6, x8}
B = {x1, x4, x6, x8, x12}
|B| >= N → Cancel P active
Compute GRPO loss and update parameter
<Iteration end>

| id of x | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| visit count vs(x) | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |

| | 1 | 3 | 4 | 6 | 7 | 8 | 11 | 14 | 15 | 17 | 20 | 2 | 5 | 9 | 10 | 12 | 13 | 16 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| visit count vs(x) (sorted) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| B = {} | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| B = {x1, x4} | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| B = {x1, x4, x6, x8} | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| B = {x1, x4, x6, x8, x12} | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Figure 4: Illustration of the rollout process in the proposed algorithm with online difficulty filtering. Each iteration begins by sorting the dataset based on the visit count $\text{vc}(\mathbf{x})$ of each example $\mathbf{x}$. A batch of unvisited or least-visited prompts is selected, respecting a predefined concurrency limit $C_{\text{max}}$. The asynchronous function $f_{async}$ samples responses from the current policy and evaluates them using the accuracy reward $r_{acc}$. Prompts with a pass rate within the accepted range $[T_{\text{Low}}, T_{\text{High}}]$ are added to the training batch. Once the batch $\mathcal{B}$ reaches the target size $N$, any remaining asynchronous jobs in $\mathcal{P}_{\text{active}}$ are canceled. The policy is then updated using the GRPO loss computed over the collected batch.

## C Learnability in Soft Prompts

We prove the variance-controlled lower bound in Proposition 3.1.

*Proof.* Throughout, let $r_{\text{acc}}(x, y) \in \{0, 1\}$ denote a binary verifiable reward with

$$P\big(r_{\text{acc}}(x,y) = 1\big) = p(x) \;\text{ and }\; P\big(r_{\text{acc}}(x,y) = 0\big) = 1 - p(x), \tag{26}$$

and assume $\beta > 0$ with $1/\beta \ll 1$ so that second-order Taylor expansions are valid. Expectations are taken with respect to $y \sim \pi_{\text{init}}(\cdot \mid x)$ unless noted.

**Step 1: Soft value under Bernoulli rewards.** Define the random variable

$$Y \;=\; \exp\Big(\tfrac{1}{\beta}\, r_{\text{acc}}(x,y)\Big) \;=\; \begin{cases} 1, & r_{\text{acc}}(x,y) = 0, \\ \exp(1/\beta), & r_{\text{acc}}(x,y) = 1. \end{cases} \tag{27}$$

Then

$$\mathbb{E}[Y] \;=\; (1 - p(x)) + p(x)\, e^{1/\beta}. \tag{28}$$

Using the soft value definition, we obtain

$$V^*(x) \;=\; \beta \,\log\Big((1 - p(x)) + p(x)\, e^{1/\beta}\Big). \tag{29}$$

**Step 2: Expected log-ratio.** From the identity in equation 9, substituting $\mathbb{E}[r_{\text{acc}}(x,y)] = p(x)$ and equation 29 gives

$$\mathbb{E}\left[\log \frac{\pi^*(y \mid x)}{\pi_{\text{init}}(y \mid x)}\right] \;=\; \frac{p(x)}{\beta} - \log\Big((1 - p(x)) + p(x)\, e^{1/\beta}\Big). \tag{30}$$

**Step 3: Second-order expansion and lower bound.** Write $e^{1/\beta} = 1 + \frac{1}{\beta} + \frac{1}{2\beta^2} + \mathcal{O}(\beta^{-3})$. Then

$$(1 - p(x)) + p(x)\, e^{1/\beta} \;=\; 1 + p(x)\left(\frac{1}{\beta} + \frac{1}{2\beta^2}\right) + \mathcal{O}\left(\frac{1}{\beta^3}\right). \tag{31}$$

Using $\log(1+\epsilon) \ge \epsilon - \frac{\epsilon^2}{2}$ with $\epsilon = p(x)\left(\frac{1}{\beta} + \frac{1}{2\beta^2}\right)$ yields

$$\log\Big((1 - p(x)) + p(x)\, e^{1/\beta}\Big) \;\ge\; \frac{p(x)}{\beta} + \frac{p(x)\big(1 - p(x)\big)}{2\beta^2} + \mathcal{O}\left(\frac{1}{\beta^3}\right). \tag{32}$$

Subtracting equation 32 from equation 30 gives

$$\mathbb{E}\left[\log \frac{\pi^*(y \mid x)}{\pi_{\text{init}}(y \mid x)}\right] \;\le\; -\frac{p(x)\big(1 - p(x)\big)}{2\beta^2} + \mathcal{O}\left(\frac{1}{\beta^3}\right). \tag{33}$$

By the reverse KL identity in equation 9,

$$D_{\text{KL}}\big(\pi_{\text{init}}(\cdot \mid x) \,\|\, \pi^*(\cdot \mid x)\big) \;=\; -\mathbb{E}\left[\log \frac{\pi^*(y \mid x)}{\pi_{\text{init}}(y \mid x)}\right] \;\ge\; \frac{p(x)\big(1 - p(x)\big)}{2\beta^2} + \mathcal{O}\left(\frac{1}{\beta^3}\right). \tag{34}$$

Dropping higher-order terms establishes the stated bound:

$$D_{\text{KL}}\big(\pi_{\text{init}}(\cdot \mid x) \,\|\, \pi^*(\cdot \mid x)\big) \;\ge\; \frac{p(x)\big(1 - p(x)\big)}{2\beta^2}. \tag{35}$$

**Extremal cases.** If $p(x) \in \{0, 1\}$, then $r_{\text{acc}}(x, y)$ is almost surely constant under $\pi_{\text{init}}$, so $V^*(x) = \mathbb{E}[r_{\text{acc}}(x, y)]$ and equation 30 evaluates to 0. Hence

$$D_{\text{KL}}\big(\pi_{\text{init}}(\cdot \mid x) \,\|\, \pi^*(\cdot \mid x)\big) = 0, \tag{36}$$

which matches Proposition 3.1 at the endpoints.

□

## D Training Configurations

All experiments are built on the Qwen2.5-3B base model (Yang et al., 2024b). We integrate DeepSpeed ZeRO-3 (Rajbhandari et al., 2020) optimization in our training pipeline to handle memory and computation efficiently. Both the SFT and RLVR stages are conducted on a distributed setup of 8×NVIDIA A100 (80GB) GPUs.

**Training Data Curation** For SFT, we sample problems from the NuminaMath dataset (Li et al., 2024) and generate solutions using DeepSeek-R1 (Guo et al., 2025). Only samples with verifiably correct solutions are retained, and we stop once approximately 1,000 such problem-solution pairs are collected. The final SFT dataset contains 1,107 filtered problems. For RLVR, we adopt a subset of the public dataset used in Cui et al. (2025)[1]. We specifically use only the math domain problems. This dataset provides a diverse pool of challenging prompts.

**Supervised fine-tuning** We use a learning rate of $5 \times 10^{-6}$ and fine-tune it for 5 epochs. The learning rate schedule is linear, with the first 25 steps used for warm-up. We use a batch size of 21.

**Reinforcement learning** We utilize the SGLang (Zheng et al., 2025) framework to accelerate parallel rollout generation, enabling efficient sampling of multiple reasoning trajectories. Training is run for 256 steps, with empirical performance gains saturating after roughly 128 steps. Each update uses 16 sampled rollouts with 16 distinct prompts per batch, followed by a one-step policy update per rollout.

**Reward design** To guide the model toward producing responses aligned with the DeepSeek R1 format, we introduce a **format reward** based on five constraints: (1) the response must begin with a '<think>' tag, (2) the '<think>' section must be properly closed with a '</think>' tag, (3) the '<think>' section must be non-empty, (4) the summary section following '</think>' must also be non-empty, and (5) the response must terminate with an eot token. Each constraint contributes 0.2 points, resulting in a maximum format reward of 1.0. In addition, we implement a **language reward** to reduce language mixing, especially given that all prompts during training and evaluation are in English. This reward was computed as the ratio of characters in the response that are alphabetic, symbolic (e.g., mathematical symbols), or whitespace, and ranged from 0 to 1. Lastly, we define an **accuracy reward**, assigning a score of 1.0 for correct answers and 0.0 for incorrect ones. The total reward is the sum of these three components—format, language, and accuracy—yielding a final reward score between 0 and 3.

## E Evaluation Benchmarks

We employ five different challenging math reasoning benchmarks:

- **MATH500** (Hendrycks et al., 2021) consists of 500 problems sampled from Lightman et al. (2023), maintaining topic and difficulty balance.
- **AIME** (Li et al., 2024, American Invitational Mathematics Examination) uses 30 problems from the 2024 official competition, while **AMC** (Li et al., 2024, American Mathematics Competitions) includes 40 problems from the 2023 official competition. Both benchmarks consist of contest-level advanced mathematical problems.
- **MinervaMath** (Lewkowycz et al., 2022) evaluates quantitative reasoning with complex mathematical problems at an undergraduate or Olympiad level.

[1] https://huggingface.co/datasets/PRIME-RL/Eurus-2-RL-Data

- **OlympiadBench** (He et al., 2024) includes 674 open-ended text-only competition problems from a broader set of 8,476 Olympiad and entrance exam questions, specifically using the “OE_TO_maths_en_COMP” subset.

Inference is conducted via SGLang (Zheng et al., 2025) with top-$p$ set to 0.95, temperature set to 0.6, and the maximum number of output tokens limited to 8,192.

## F Difficulty-Aware Performance Analysis

To further understand the effect of our method, we analyze performance variations based on difficulty levels.

**Benchmark-Level Difficulty Spectrum** As discussed in § 4, our benchmark suite spans a wide difficulty range. This is reflected in the SFT checkpoint performance of Qwen2.5-3B, which ranges from 0.0% to 49.8% as shown in Table 1. We order the benchmarks in ascending difficulty according to SFT performance: AIME (0.0%), MinervaMath (13.2%), OlympiadBench (17.3%), AMC (20.5%), and MATH500 (49.8%). From this perspective, we observe two trends:

- Narrowing the difficulty threshold (i.e., tighter filtering range) generally improves performance, especially on challenging tasks like MinervaMath and AIME.

- Harder benchmarks benefit more from filtering. For instance, AIME shows more than a 300% relative improvement over SFT, and MinervaMath improves by 35%.

**Difficulty-Level Breakdown within MATH500** We also analyze performance by difficulty levels in the MATH500 benchmark. Table 3 shows that balanced filtering GRPO outperforms plain GRPO across most difficulty levels, especially on harder ones (Level 3–5).

| **Difficulty** | **Plain** $(0 \leq p(\mathbf{x}) \leq 1\ )$ | **w/ Online Filtering** $(0 < p(\mathbf{x}) < 1)$ | **w/ Online Filtering** $(0.3 < p(\mathbf{x}) < 0.7)$ |
|---|---|---|---|
| Level 1 | **88.37** | **88.37** | 83.72 |
| Level 2 | 78.89 | **83.33** | **83.33** |
| Level 3 | 71.43 | 70.48 | **79.05** |
| Level 4 | 47.66 | 50.78 | **55.47** |
| Level 5 | 30.60 | **32.84** | 32.09 |

Table 3: Accuracy (%) of GRPO-trained models on MATH500 by difficulty level. The highest score for each level is in **bold**.

## G General rewards: CGF derivations and bounds

We provide full proofs for Proposition 6.1, Corollary 6.2, and Corollary 6.3.

### G.1 Proof of Proposition 6.1.

*Proof.* Fix $x$ and write expectations over $y \sim \pi_{\text{init}}(\cdot|x)$. By the definition of the soft value,

$$V^*(x) \;=\; \beta \,\log \mathbb{E}\Big[\exp\Big(\tfrac{1}{\beta}\, r(x,y)\Big)\Big]\,. \tag{37}$$

From equation 9,

$$\mathbb{E}\left[\log \frac{\pi^*(y|x)}{\pi_{\text{init}}(y|x)}\right] \;=\; \frac{1}{\beta}\,\mathbb{E}[r(x,y)] \;-\; \frac{1}{\beta}\,V^*(x) \;=\; \frac{1}{\beta}\,\mu(x) \;-\; \log \mathbb{E}\Big[e^{r/\beta}\Big]\,. \tag{38}$$

Multiplying both sides by $-1$ gives

$$D_{\mathrm{KL}}\big(\pi_{\text{init}} \| \pi^*\big) \;=\; -\,\mathbb{E}\left[\log \frac{\pi^*}{\pi_{\text{init}}}\right] \;=\; \log \mathbb{E}\Big[e^{r/\beta}\Big] \;-\; \frac{1}{\beta}\,\mu(x). \tag{39}$$

Factor out the mean inside the exponential:

$$\log \mathbb{E}\Big[e^{r/\beta}\Big] \;=\; \log \mathbb{E}\Big[e^{(r-\mu)/\beta}\, e^{\mu/\beta}\Big] \;=\; \frac{\mu}{\beta} \;+\; \log \mathbb{E}\Big[e^{(r-\mu)/\beta}\Big]. \tag{40}$$

Substitute into the previous line to obtain

$$D_{\mathrm{KL}}\big(\pi_{\text{init}} \| \pi^*\big) \;=\; \log \mathbb{E}\Big[\exp\Big(\tfrac{r-\mu}{\beta}\Big)\Big] \;=\; K_{r-\mu}\Big(\tfrac{1}{\beta}\Big), \tag{41}$$

which is the desired identity. □

### G.2 Proof of Corollary 6.2.

*Proof.* Let $Z := r_G - \mu_G$. Under $\pi_{\text{init}}$, $Z \sim \mathcal{N}(0, \sigma_G^2)$, whose moment generating function is $\mathbb{E}[e^{tZ}] = \exp(\frac{1}{2}\sigma_G^2 t^2)$. Therefore the CGF is $K_Z(t) = \frac{1}{2}\sigma_G^2 t^2$. Applying Proposition 6.1 with $t = \frac{1}{\beta}$ yields

$$D_{\mathrm{KL}}\big(\pi_{\text{init}} \| \pi^*\big) \;=\; K_Z\Big(\tfrac{1}{\beta}\Big) \;=\; \frac{\sigma_G^2}{2\,\beta^2}. \tag{42}$$

□

### G.3 Proof of Corollary 6.3.

*Proof.* Let $Z := r_M - \mu$. Write the CGF Taylor series around $t = 0$:

$$K_Z(t) \;=\; \sum_{k=2}^{\infty} \frac{\kappa_k}{k!}\, t^k \;=\; \frac{\kappa_2}{2}\, t^2 \;+\; \frac{\kappa_3}{6}\, t^3 \;+\; \frac{\kappa_4}{24}\, t^4 \;+\; \cdots, \tag{43}$$

where $\kappa_2 = \sigma^2 = \mathrm{Var}(Z)$ and $\kappa_k$ are the cumulants of $Z$. Because $r_M$ is bounded (multinomial or $[a, b]$), all moments of $Z$ are finite and thus all $\kappa_k$ are finite. For $t = \frac{1}{\beta}$,

$$K_Z\Big(\tfrac{1}{\beta}\Big) \;=\; \frac{\sigma^2}{2\,\beta^2} \;+\; \frac{\kappa_3}{6\,\beta^3} \;+\; \frac{\kappa_4}{24\,\beta^4} \;+\; \cdots. \tag{44}$$

Hence,

$$K_Z\Big(\tfrac{1}{\beta}\Big) \;\geq\; \frac{\sigma^2}{2\,\beta^2} \;-\; \left|\frac{\kappa_3}{6\,\beta^3}\right| \;-\; \left|\frac{\kappa_4}{24\,\beta^4}\right| \;-\; \cdots, \tag{45}$$

which shows that the variance term dominates at second order and provides a lower control up to $\mathcal{O}(\beta^{-3})$. In particular, for sufficiently large $\beta$ (or small $t$), we obtain

$$K_Z\Big(\tfrac{1}{\beta}\Big) \;\geq\; \frac{\sigma^2}{2\,\beta^2} \;-\; \mathcal{O}\Big(\frac{1}{\beta^3}\Big). \tag{46}$$

Finally, apply Proposition 6.1 to translate this bound to the reverse KL, completing the proof. □

## H Discussion: Native Generalizability of Online Difficulty Filtering in RL for LLMs

The concept of online difficulty filtering in reinforcement learning for language models is **inherently versatile across verifiable tasks and reward formulations**, even beyond the empirical scope of this paper. Recent work in reinforcement learning with verifiable rewards (RLVR) demonstrates its applicability to diverse domains, including logic puzzles (Chen et al., 2025a), medical reasoning (Gunjal et al., 2025), complex instruction-following (Pyatkin et al., 2025), and coding (Chen et al., 2025b). Our theoretical framework builds on the standard KL-regularized or soft reinforcement learning formulation (Schulman et al., 2018; Richemond et al., 2024), which optimizes the objective

$$\max_{\pi_\theta} \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[\,r(x,y)\,] - \beta\,\mathrm{D}_{\mathrm{KL}}\big(\pi_\theta(\cdot|x)\,\|\,\pi_{\text{init}}(\cdot|x)\big), \tag{47}$$

yielding the Boltzmann-rational optimal policy relative to the reference $\pi_{\text{init}}$. Modern post-training algorithms such as PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), RLOO, and REINFORCE++ (Hu et al., 2025) can all be interpreted as optimizing surrogates of this KL-regularized objective, differing primarily in how the policy gradient is estimated or regularized. Hence, the theoretical mechanism behind online difficulty filtering, namely that the reverse KL divergence between $\pi_{\text{init}}$ and $\pi^*$ is determined by the cumulant generating function of the centered reward and dominated by its variance, extends directly to these algorithms as long as a KL penalty to the reference is preserved.

The generalized analysis in §6.3 further shows that this relationship holds exactly for Gaussian rewards and forms a tight second-order lower bound for multinomial or bounded rewards. Since continuous neural reward models are approximately Gaussian and rubric-based verifiable rewards are discrete and bounded, these two cases jointly cover most practical reward systems in RLHF and RLVR. Accordingly, the same variance-guided filtering principle applies regardless of task or reward type: prompts with intermediate success probabilities maximize the effective learning signal, while trivially easy or hard ones contribute negligible gradients. This makes balanced online difficulty filtering a theoretically grounded and natively general component for PPO-, GRPO-, or RLOO-style alignment training across diverse verifiable tasks.