Efficient Reinforcement Learning for LLM Driven by Policy-Intrinsic Cues

Anonymous ACL submission

Abstract

Reinforcement learning with verifiable rewards (RLVR) has improved the reasoning ability of large language models, yet training remains costly because many rollouts contribute little to policy improvement. Existing data selection methods rely on external, static signals (e.g., pass rates, human-defined difficulty) that fail to align with the dynamic nature of policy evolution and lack unified theoretical justification. We propose an information-centric efficient selection framework that decomposes ideal data selection into two independently controllable dimensions, namely input-side control and rollout-side control, both instantiated using only policy-intrinsic information at near-zero cost. Under this framework, we present PREPO[1], which uses prompt perplexity as a measure of policy–input adaptation rather than task difficulty to dynamically schedule prompt selection, and relative entropy as gradient information gain to weight rollouts by their contribution to policy uncertainty reduction. On Qwen and Llama models, PREPO achieves competitive results on mathematical reasoning benchmarks with up to $3\times$ fewer rollouts than baselines, demonstrating that principled, theory-driven data selection can substantially improve RLVR efficiency.

1 Introduction

Reinforcement learning (RL) has become central in improving the reasoning capabilities of large language models (LLMs) by optimizing self-generated rollouts (Guo et al., 2025; Team et al., 2025; Chen et al., 2025a). Recent advances in reinforcement learning with verifiable reward (RLVR) demonstrate that it is a simple yet effective method for scaling reasoning performance (Shao et al., 2024; Yu et al., 2025). However, applying RLVR to models that generate long reasoning traces incurs substantial computational overhead in the rollout stage, significantly hampering training efficiency and becoming the primary training bottleneck (Zhong et al., 2024).

A natural remedy is to select training data more carefully, pruning uninformative prompts or rollouts while preserving those that drive learning. Prior studies have explored several directions, including using pass rate as a difficulty indicator (Zhang et al., 2025; Albalak et al., 2025), applying human-defined criteria such as specific domains or topics (Chen et al., 2025b), and relying on auxiliary models or embedding techniques to characterize data semantics. However, these approaches share a fundamental limitation: they judge data value from external, static perspectives that fail to align with the dynamic, non-stationary nature of policy evolution during RL training. Pass rates require multiple rounds of costly sampling without reducing the overall computation; human criteria introduce subjective bias and limited applicability across models; auxiliary methods add both computational overhead and distribution mismatch. More critically, none of these approaches provides a unified theoretical account of why certain data is valuable for a given policy at a given training stage. We therefore ask

> *Can we characterize training data value purely from the policy's intrinsic information structure, in a way that is (a) theoretically grounded, (b) computationally neutral, and (c) dynamically aligned with the policy state?*

In this work, we answer affirmatively by establishing an *information-centric efficient selection framework* for RLVR. We formalize the ideal data selection problem and decompose it into two fundamental, independently controllable dimensions: (1) input-side control, which determines which prompts should be selected before rollout, and (2)

[1]We will release our data and code upon acceptance.

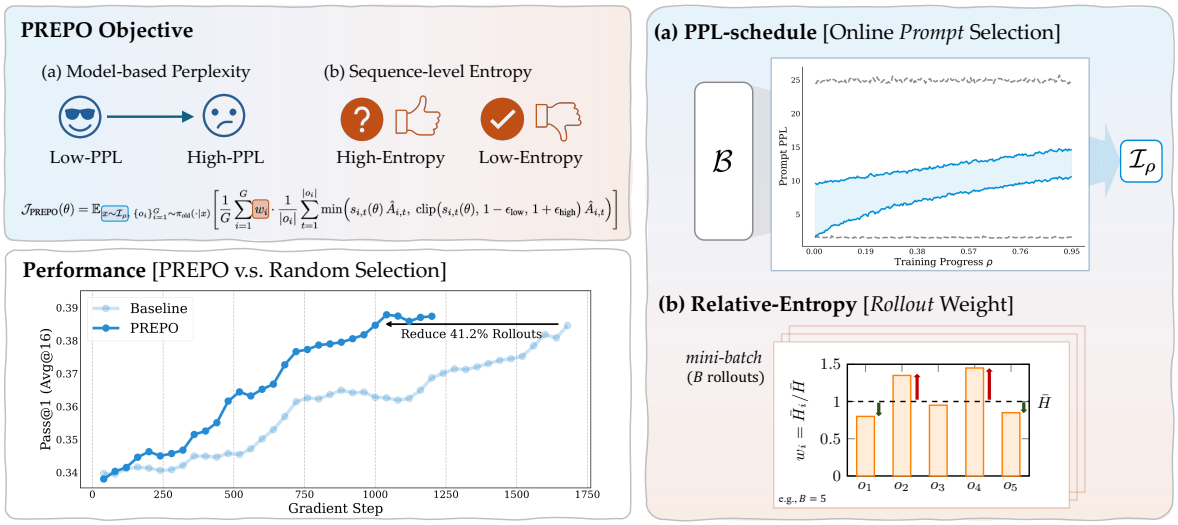


Figure 1: **Overview of PREPO.** The PREPO objective integrates perplexity-based schedule learning and sequence-level entropy weighting into a unified optimization scheme. On Qwen2.5-Math-7B, PREPO achieves higher performance while requiring only 41.2% of the rollouts used by random selection. (a) PPL-schedule actively selects prompts according to model-based perplexity, starting with lower-PPL prompts *("adapted" to the model)* and progressively introducing higher-PPL ones *("less adapted")* as training progresses. (b) Relative-Entropy Weighting adjusts rollout contributions by comparing each sequence's entropy against the batch average, amplifying high-entropy rollouts *("informative attempts")* and downweighting low-entropy ones *("redundant responses")*.

rollout-side control, which determines how generated sequences should be weighted during optimization. We instantiate both dimensions using only policy-intrinsic information. For the first dimension, we use prompt perplexity as a measure of policy–input adaptation that captures how well the current policy models a given input, and schedule prompt selection from high-adaptation to low-adaptation as training progresses. For the second dimension, we use sequence-level relative entropy as a measure of gradient information gain, amplifying rollouts that contribute more to policy uncertainty reduction.

Together, these form Perplexity-schedule with Relative-Entropy Policy Optimization (PREPO), a unified method derived from a single theoretical principle rather than assembled from ad-hoc components. Both signals are computed from quantities already available during standard RL training, incurring almost no additional cost.

Across Qwen and Llama models, PREPO surpasses existing data-pruning baselines and remains competitive with the full-data baseline, while reducing rollout usage by more than 40% (see Fig. 1 for Qwen2.5-Math-7B). Our contributions are as follows. First, we establish an information-centric unified framework that formalizes efficient data selection for RLVR as two independently controllable dimensions. Second, we propose PREPO as a principled instantiation of this framework, using only policy-intrinsic signals at zero additional cost. Third, extensive experiments on multiple models demonstrate that PREPO achieves up to 3× rollout reduction while maintaining competitive performance.

2 Related Work

Data efficiency in RLVR. A growing body of work has explored data efficiency for RLVR. Offline methods pre-compute sample quality via difficulty estimation or reward models (Qu et al., 2025; Zhang et al., 2025; Chen et al., 2025b; Kamalloo et al., 2025), but their static scores cannot track the evolving policy. Online difficulty filtering (Bae et al., 2025) and predictive prompt allocation (Qu et al., 2025) adapt selection during training, yet still depend on pass-rate signals that require extra rollouts. Stage-wise difficulty schedules (Zhang et al., 2025; Chen et al., 2025b) share with PREPO the intuition of matching data to the model's current competence, but rely on externally defined difficulty rather than a policy-intrinsic measure. Gradient-informed (Kamalloo et al., 2025; Chen et al., 2025c) and advantage-based (Wang and Guofeng, 2025) selectors are closer to our goal of measuring per-sample learning value, though they incur additional computation to obtain the signal. Down-sampling strategies (Li et al., 2025) and replay buffer designs (Liu et al., 2025) reduce roll-

out redundancy on the output side but leave input selection unaddressed. PREPO differs from all the above by scoring prompts with perplexity read off the existing forward pass, achieving input selection at zero rollout overhead while simultaneously weighting rollouts on the output side.

Entropy mechanism in RLVR. The entropy of a policy's output distribution indicates how much randomness remains in its decisions, and incorporating it into RL objectives helps maintain diverse exploration. Recent RLVR studies reveal its particular importance for reasoning LLMs. Cui et al. (2025) identify rapid entropy collapse as a failure mode and propose covariance-based updates to slow its decay, motivating our use of entropy as a positive learning signal rather than merely a regularizer. Wang et al. (2025) show that high-entropy "forking tokens," though rare, account for most reasoning gains; PREPO lifts this insight from the token level to the sequence level by weighting entire rollouts by their relative entropy. Cheng et al. (2025) incorporates an entropy term into the advantage function to encourage exploration, but introduces extra hyperparameters; our relative-entropy weighting is parameter-free and scale-invariant. Truncated importance-sampling methods such as CISPO (Chen et al., 2025a) address a complementary axis of policy-gradient optimization. PREPO unifies input-side and rollout-side selection from a single policy-intrinsic principle, filling the gap left by methods that address only one dimension.

3 PREPO: An Information-Centric Framework for Efficient RLVR

We present PREPO, grounded in an information-centric framework for efficient data selection in RLVR. We first show that any prompt or rollout selection implicitly modifies the training objective, then identify two control variables within this modified objective that can be set using policy-intrinsic signals at zero additional cost.

3.1 Data Selection as Objective Modification

Consider the standard GRPO objective over a candidate prompt batch $\mathcal{B} = \{x_i\}_{i=1}^N$, where G rollouts are sampled per prompt.

$$\mathcal{J}_{\text{GRPO}}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{G}\sum_{j=1}^{G} L_{ij}(\theta), \qquad (1)$$

where $L_{ij}(\theta)$ is the clipped surrogate loss for rollout o_j on prompt x_i. This objective implicitly assumes uniform sampling over prompts and uniform weighting over rollouts. Any data selection method, whether based on pass rates, difficulty scores, gradient signals, or our approach, modifies this objective by changing the sampling distribution and weighting scheme. We make this explicit by writing the general modified objective

$$\mathcal{J}(\theta;\, p,\, \alpha) = \mathbb{E}_{x\sim p(x)}\, \mathbb{E}_{\{o_j\}\sim\pi_{\text{old}}(\cdot|x)} \left[\sum_{j=1}^{G} \alpha_j(x, o_j)\cdot L_j(\theta)\right] \qquad (2)$$

where $p(x)$ is the prompt selection distribution and $\alpha_j(x, o_j)$ is the rollout weight function. GRPO corresponds to $p = \text{Uniform}(\mathcal{B})$ and $\alpha_j = 1/G$. Different data selection methods correspond to different instantiations of (p, α):

- **Pass-rate filtering**: $p(x) \propto \mathbb{1}[\hat{r}(x) \in (\delta_l, \delta_h)]$, $\alpha_j = 1/G$, where $\hat{r}(x)$ is the estimated pass rate and $\delta_l, \delta_h \in [0, 1]$ are lower and upper thresholds defining the retained difficulty window. Requires rollouts to estimate $\hat{r}$.
- **Gradient-based selection**: $p(x) \propto \|\nabla_\theta L(x)\|$, $\alpha_j = 1/G$. Requires additional forward-backward passes.
- **PREPO**: $p = \text{Uniform}(\mathcal{I}_\rho)$, $\alpha_j = w_j/G$. Both determined from the existing forward pass.

The compression rate $\gamma = K/N \in (0, 1]$ governs how selective p is, with $\gamma = 1$ recovering the full batch. A method is *data-efficient* when it achieves comparable policy improvement per training step at $\gamma \ll 1$.

3.2 Two Control Variables

The modified objective in Eq. 2 exposes two control variables that act at different stages of the training pipeline,

- $p(x)$ acts **before rollout generation**, determining which prompts receive rollouts and directly controlling computation cost.
- α_j acts **after rollout generation**, determining how much each completed rollout contributes to the gradient and controlling signal quality.

These two variables are operationally decoupled, since p is determined before rollouts are generated and α is determined after. In this work, we instantiate both using policy-intrinsic information. We use the policy's surprisal on the input to determine p and the policy's entropy during generation to determine α. Both are available from quantities already computed in the standard training loop, introducing no additional rollout or evaluation cost. This realization directly addresses the three criteria raised in Section 1: the framework is *theoretically*

grounded in the modified-objective decomposition (formalized by Propositions 1 and 2), *computationally neutral* because both signals reuse existing forward-pass quantities, and *dynamically aligned* with the policy state because p and α are recomputed from π_θ at every training step.

3.3 Input Selection via Policy–Input Adaptation

We now specify $p(x)$ by designing a prompt selection mechanism based on the policy's intrinsic adaptation to each input.

Prompt scoring. For a prompt $x_i = (x_{i,1}, \ldots, x_{i,|x_i|})$, the per-token average surprisal under π_θ is

$$h_{\pi_\theta}(x_i) = -\frac{1}{|x_i|} \sum_{t=1}^{|x_i|} \log \pi_\theta(x_{i,t} \mid x_{i,<t}), \quad (3)$$

and its exponential form gives the perplexity, which serves as the adaptation score[2]

$$\mathrm{PPL}_{\pi_\theta}(x_i) = \exp\big(h_{\pi_\theta}(x_i)\big). \quad (4)$$

This quantity measures how well the current policy adapts to the prompt. Since π_θ evolves throughout training, $\mathrm{PPL}_{\pi_\theta}(x_i)$ is recomputed at each step and the ranking over $\mathcal{B}$ is dynamic. A prompt at quantile q at step t may sit at a different quantile at $t+1$.

Why PPL predicts learning signal. In GRPO with binary reward $r \in \{0, 1\}$, a prompt contributes non-zero gradients only when its rollout group has mixed outcomes, so the effective per-prompt learning signal is proportional to $\mathrm{Var}_o[r(x, o)] = \mu(x)(1-\mu(x))$, where $\mu(x)$ is the pass rate. Empirically, PPL exhibits a statistically significant negative correlation with $\mu(x)$ across diverse policies (Appendix A.1), reflecting how well the current policy already adapts to the input. Unlike static difficulty scores, this signal moves with the policy and tracks the boundary between confident and uncertain inputs. Prompts whose PPL sits near this boundary fall where $\mu(1-\mu)$ is largest, making PPL a zero-cost, policy-dependent indicator of which prompts carry non-trivial learning signal without requiring rollouts to estimate pass rates.

[2]We use h for surprisal (a deterministic function of the observed token sequence) and H for Shannon entropy (an expectation over the vocabulary). These are distinct quantities: h depends on the actual tokens in x, while H averages over all possible continuations.

Dynamic scheduling. Rather than selecting a fixed PPL quantile, we traverse the PPL range along a schedule that tracks the policy's evolving capability boundary. The schedule reflects a bias-variance tradeoff. Always selecting the highest-PPL prompts would maximize exploration but produce high-variance gradients that slow convergence, while always selecting the lowest-PPL prompts would yield stable gradients but starve the policy of new challenges. A progressive schedule balances this by starting with low-PPL prompts for stable early learning and shifting toward higher PPL as the policy matures.

Let $\rho = \mathrm{step}/\mathrm{total\ steps} \in [0, 1]$ denote the normalized training progress, and let σ be the permutation that sorts $\mathcal{B}$ by ascending $\mathrm{PPL}_{\pi_\theta}(x_i)$. We define the selected sub-batch as

$$\mathcal{I}_\rho = \{\, \sigma(j) : l(\rho) \le j \le l(\rho) + K - 1 \,\}, \quad (5)$$

where the starting index follows a linear schedule

$$l(\rho) = \lfloor \rho \cdot (N - K) \rfloor, \quad (6)$$

so that $\mathcal{I}_\rho$ shifts smoothly from low-PPL to high-PPL prompts. Nonlinear variants[3] can also be employed. This sets $p = \mathrm{Uniform}(\mathcal{I}_\rho)$ in Eq. 2.

Proposition 1 (Compression Bound)**.** *Let $\mathcal{B}$ be a batch of N prompts with per-sample loss bounded by $|L| \le M$. For any subset $\mathcal{I} \subseteq \mathcal{B}$ of size γN, the deviation from the full-batch objective satisfies*

$$\big|\, \mathbb{E}_{x \in \mathcal{I}}[L] - \mathbb{E}_{x \in \mathcal{B}}[L] \,\big| \;\le\; 2M(1-\gamma). \quad (7)$$

This bound holds for any selection method and shrinks linearly with γ, ensuring that compression itself is safe (proof in Appendix B). PREPO's structural advantage over random selection lies not in a tighter worst-case bound but in preferentially retaining high-signal prompts: the linear schedule traverses the full PPL range over T training steps so that each prompt enters $\mathcal{I}_\rho$ on average $T\gamma$ times, matching the coverage of random selection while ordering traversal by informativeness.

3.4 Rollout Weighting via Sequence Entropy

We now specify α_j by designing a rollout weighting mechanism based on the policy's uncertainty during generation.

[3]A general family: $l(\rho) = \lfloor \rho^\alpha \cdot (N - K) \rfloor$, $\alpha \in \mathbb{R}^+$.

Sequence-level entropy. For a rollout o_i generated from prompt x, the token-level entropy is $H_t = -\sum_{v \in \mathcal{V}} \pi_\theta(v \mid o_{<t}, x) \log \pi_\theta(v \mid o_{<t}, x)$. The sequence-level average is

$$\bar{H}_i = \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} H_t. \tag{8}$$

This measures how uncertain the policy was during generation. High-entropy rollouts correspond to sequences where the policy faced genuinely uncertain decisions, while low-entropy rollouts correspond to confident, potentially redundant generation.

Relative information weight. For scale-invariance, we normalize against the batch mean

$$\bar{H} = \frac{1}{B} \sum_{k=1}^{B} \bar{H}_k, \tag{9}$$

$$w_i = \frac{\bar{H}_i}{\bar{H}}. \tag{10}$$

Setting $\alpha_i = w_i/G$ in Eq. 2 amplifies rollouts with above-average uncertainty and attenuates those that are relatively redundant.

Why entropy predicts rollout informativeness. The policy gradient for a rollout is scaled by its advantage $\hat{A}$ and importance ratio $s(\theta)$. High-entropy rollouts indicate decision points where the policy distribution is spread across alternatives, and these are precisely the tokens where the gradient has the most room to reshape the distribution. Low-entropy rollouts reinforce already-confident predictions with diminishing returns. By up-weighting high-entropy rollouts, we allocate gradient budget to where it is most effective.

Proposition 2 (Information Amplification)**.** *Let $w_i = \bar{H}_i/\bar{H}$ be the relative-entropy weights defined in Eq. 10. Then*

1. ***Conservation.*** *$w_i \geq 0$ and $\frac{1}{G}\sum_{i=1}^{G} w_i = 1$.*
2. ***Reduction to GRPO.*** *When $\bar{H}_i = \bar{H}\ \forall i$, we have $w_i = 1$ and the objective recovers standard GRPO.*
3. ***Bounded concentration.*** *$w_i \leq M_H/\bar{H}$ whenever $\bar{H}_i \leq M_H$.*
4. ***Signal amplification.*** *The weighted mean of rollout entropies satisfies*

$$\frac{1}{G}\sum_{i=1}^{G} w_i \bar{H}_i = \bar{H}\left(1 + \mathrm{CV}^2(\bar{H}_i)\right), \tag{11}$$

where CV *is the coefficient of variation.*

The weighting amplifies the effective information signal by the factor $1+\mathrm{CV}^2 \geq 1$, growing with the dispersion of rollout informativeness within the batch (proof in Appendix B).

3.5 PREPO Objective

The PPL-based scheduling (Section 3.3) decides which prompts enter each training step, while the relative-entropy weighting (Section 3.4) modulates how much each rollout influences the gradient. Embedding both into the GRPO objective gives the unified PREPO objective

$$\mathcal{J}_{\text{PREPO}}(\theta) = \mathbb{E}_{x \sim \mathcal{I}_\rho,\, \{o_i\}_{i=1}^{G} \sim \pi_{\text{old}}(\cdot|x)} \left[\frac{1}{G}\sum_{i=1}^{G} w_i \cdot \frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \min\Big(s_{i,t}(\theta)\, \hat{A}_{i,t}, \; \text{clip}\big(s_{i,t}(\theta),\, 1-\epsilon_{\text{low}},\, 1+\epsilon_{\text{high}}\big)\, \hat{A}_{i,t} \Big) \right] \tag{12}$$

with $s_{i,t}(\theta) = \pi_\theta(o_{i,t} \mid x, o_{i,<t}) \,/\, \pi_{\text{old}}(o_{i,t} \mid x, o_{i,<t})$ and $\hat{A}_{i,t} = (r_i - \bar{r}) \,/\, \sigma_r$, $\bar{r}$ and σ_r being the group mean and standard deviation of rewards. This instantiates Eq. 2 with $p = \mathrm{Uniform}(\mathcal{I}_\rho)$ and $\alpha_i = w_i/G$.

3.6 Discussion

Scope of the variance argument. The link from PPL to learning signal in Section 3.3 assumes a binary reward and the group-relative advantage estimator in GRPO. For shaped or continuous rewards, $\mathrm{Var}_o[r]$ replaces $\mu(1-\mu)$, but the qualitative claim that high-variance rollout outcomes concentrate near intermediate PPL still applies. Both intrinsic signals also presume a non-degenerate policy. When π_θ is near-uniform at the start of training or near-collapsed after entropy collapse, PPL and entropy lose discriminating power and PREPO degrades toward uniform sampling and unit weighting.

Why both signals are needed. Although PPL and sequence entropy are read from the same policy, they capture different quantities at different stages, surprisal over observed input tokens before rollout, and uncertainty over the vocabulary during generation. PPL identifies prompts likely to produce mixed-outcome rollouts, while entropy identifies, among the rollouts that materialize, which ones carry exploration signal worth amplifying. The two are non-redundant because the decomposition in Eq. 2 separates pre-rollout selection from post-

rollout weighting, so a strong signal at one stage cannot substitute for the other.

4 Experiments

4.1 Setups

We benchmark PREPO against three baselines: Random selection, Dynamic Sampling (DS) (Yu et al., 2025), and GRESO (Zheng et al., 2025) (detailed in Appendix F). Our evaluation covers multiple models: Qwen2.5-7B (Team, 2024), Qwen2.5-Math-1.5B and Qwen2.5-Math-7B (Yang et al., 2024), Qwen3-4B (non-thinking) (Yang et al., 2025), and Llama3.1-8B (Dubey et al., 2024). For training data, we use DAPO-Math-17K (Yu et al., 2025) and MATH500 (Lightman et al., 2023a) with a total of 18,417 training samples.

Training and Evaluation. All models are trained using the `verl` (Sheng et al., 2025), with `vLLM` (Kwon et al., 2023) employed for rollout generation to ensure efficient inference. For the Qwen models, we evaluate them on four benchmarks, including AIME25 (Art of Problem Solving, 2025) (30 samples), AIME24 (Art of Problem Solving, 2024) (30 samples), MATH500 (Lightman et al., 2023b) (500 samples), and OlympiadBench (He et al., 2024) (8,476 samples), which cover a diverse range of mathematical reasoning challenges. The Llama model is evaluated on MATH500 (Lightman et al., 2023b) and GSM8K (Cobbe et al., 2021) (1,319 samples). We evaluate all models using *pass@1 (avg16)*, i.e., the accuracy of the top-1 response averaged over 16 generations, with temperature 1. We evaluate models every 50 training steps and report the best average performance on all benchmarks.

Experiment Configuration. For the Qwen2.5-Math models, we use a maximum context length of 4096 tokens, matching their supported limit. For Qwen3-4B and Llama3.1-8B, we set the context length to 32,768 tokens. Rollouts are generated with temperature as 1 using `vLLM`, producing 8 responses per prompt. For PREPO, Random, and GRESO, we adopt an online selection ratio of $K/N = 20\%$ at each training step, where the candidate batch size (N) is 1280 and the actual batch size (K) is fixed at 256. For GRESO, we set the targeted zero-variance percentage as 50%. For DS, the candidate batch size is 384. The mini-batch size (B) is 64 for all experiments. For all experiments, we set the clipping thresholds to $\epsilon_{\text{low}} = 0.2$ by default, with a larger $\epsilon_{\text{high}} = 0.28$ as the upper bound. The actor model is optimized with AdamW using a constant learning rate of 1×10^{-6}, momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and a weight decay of 0.01. Following Yu et al. (2025), we omit the KL-divergence regularization term. Training is applied only to the actor parameters and parallelized with Fully Sharded Data Parallel. All experiments are conducted on 32 GPUs.

4.2 Results

PREPO achieves consistent two- to three-fold rollout reduction. As shown in Table 1 below, PREPO substantially lowers rollout usage relative to all baselines while maintaining or surpassing accuracy. On Qwen2.5-7B, PREPO reduces rollouts from over 1M (DS: 1,040K; Random: 716K; GRESO: 680K) to 304K. On Qwen2.5-Math-1.5B, it cuts the budget from several million (DS: 3.6M; Random: 3.0M; GRESO: 2.5M) down to 1.1M. Similarly, PREPO reduces rollouts to 540K on Qwen2.5-Math-7B (vs. 1.66M for DS, 905K for Random, 654K for GRESO) and to 348K on Qwen3-4B (vs. 688K, 553K, and 472K, respectively).

Dynamic sampling and random selection are inefficient. Although DS sometimes yields accuracy gains, it consistently demands the largest rollout budget. For instance, on Qwen2.5-Math-1.5B, DS requires 3.6M rollouts to reach an average score of 34.10, whereas PREPO attains a higher 36.23 with only 1.1M rollouts. This confirms that DS wastes computation by discarding uninformative prompts only after rollouts are generated. Random selection occasionally performs comparably to or even better than DS, yet its rollout cost remains high. On Qwen2.5-Math-7B, for example, random selection consumes 905K rollouts to achieve 39.45, while PREPO surpasses it with 39.59 using just 540K rollouts. Overall, PREPO delivers both higher accuracy and lower cost, while also producing a more diverse set of problems than online random selection (see Appendix L.6).

GRESO improves efficiency but lags behind PREPO. GRESO reduces rollout demand by pre-filtering uninformative prompts, making it more efficient than DS and Random. However, its accuracy often falls short of PREPO. For instance, on Qwen3-4B, GRESO achieves 69.89 accuracy with 472K rollouts, while PREPO reaches 75.99 with only 348K rollouts. This suggests that PREPO's intrinsic exploration signals provide a more effective

Table 1: Performance comparison (%) on Qwen models. *For each model, the top row reports the base model's performance. Best results are highlighted in bold or underlined. K = thousand, M = million.*

Method	AIME25	AIME24	MATH	Olympiad	Avg ↑	# Rollouts ↓
Qwen2.5-7B	1.25	4.17	72.26	33.09	27.69	–
+ DS	7.92	17.55	75.50	38.91	34.97	1040K
+ Random	6.98	16.41	75.70	38.47	34.39	716K
+ GRESO	9.22	10.83	76.65	42.07	34.59	680K
+ PREPO (Ours)	10.21	16.09	76.30	39.85	**35.61**	**304K**
Qwen2.5-Math-1.5B	3.54	10.21	55.76	27.41	24.23	–
+ DS	10.83	25.83	76.40	23.33	34.10	3.6M
+ Random	20.00	16.67	76.25	30.50	35.86	3.0M
+ GRESO	15.38	20.00	76.65	24.17	34.16	2.5M
+ PREPO (Ours)	20.00	16.67	76.25	32.00	**36.23**	**1.1M**
Qwen2.5-Math-7B	9.17	20.80	72.26	39.56	35.45	–
+ DS	13.33	33.33	81.35	30.17	39.55	1664K
+ Random	10.00	26.67	77.80	43.26	39.45	905K
+ GRESO	18.33	25.83	77.80	26.83	37.46	654K
+ PREPO (Ours)	12.81	26.15	77.85	41.58	**39.59**	**540K**
Qwen3-4B	30.00	53.33	94.10	52.67	57.53	–
+ DS	63.33	66.67	95.10	58.00	70.78	688K
+ Random	60.00	70.00	96.00	59.33	71.33	553K
+ GRESO	56.67	69.17	96.40	57.33	69.89	472K
+ PREPO (Ours)	66.67	80.00	96.60	60.67	**75.99**	**348K**

alternative to heuristic rollout filtering.

PREPO generalizes across model architectures. On Llama3.1-8B (see Table 2), PREPO once again achieves the strongest results, reaching an average of 36.55 with just 115K rollouts. In contrast, DS requires nearly five times as many rollouts, with substantially lower accuracy. This confirms that PREPO generalizes effectively across model families and scales, providing a consistent advantage in both performance and efficiency.

Table 2: Performance comparison (%) on Llama. *For each model, the top row reports the base model's performance. Best results are highlighted in bold or underlined. K = thousand.*

Method	GSM8K	MATH	Avg ↑	# Rollouts ↓
Llama3.1-8B	9.53	6.05	7.79	–
+ DS	39.50	17.00	28.25	553K
+ Random	46.63	14.60	30.61	266K
+ GRESO	41.77	16.80	29.29	273K
+ PREPO (Ours)	51.10	21.81	**36.55**	**115K**

PREPO could be more effective when perplexity distributions are concentrated. We found that PREPO achieves large improvement when perplexity values are relatively compact across Qwen3-4B, and LLaMA3.1-8B. As shown in Table 3, the normalized standard deviation of those models remains below one, indicating less dispersed distributions that make perplexity-based filtering more reliable. In these cases, PREPO can better exploit the PPL-schedule to reduce rollouts.

Table 3: Normalized standard deviation of prompt perplexity (std/mean) across models. *Values above one indicate more dispersed distributions.*

Qwen2.5-7B	Qwen3-4B	LLaMA3.1-8B	Qwen2.5-Math-7B	Qwen2.5-Math-1.5B
0.73	0.65	0.75	1.23	1.02

Training Dynamics of PREPO versus the Random Baseline. Compared with random selection, PREPO sustains higher entropy loss and gradient norm without instability, while reducing the proportion of zero-advantage rollouts (Figures 9 and 10 in Appendix I). Average prompt length decreases steadily and average response length stays longer, suggesting an adaptive shift to harder prompts and more deliberate generation as training progresses.

4.3 Ablation Study

In this section, we conduct the ablation analysis to isolate the contribution of each component in PREPO. In addition to the linear PPL-schedule, we evaluate four variants: (1) linear PPL-schedule,

(2) random filtering with relative-entropy weighting, (3) linear PPL-schedule combined with an entropy-regularization loss, and (4) a nonlinear PPL-schedule. The nonlinear variant follows the general form $l(\rho) = \lfloor \rho^{\alpha}(N - K) \rfloor$ with $\alpha = 0.5$, which produces a schedule that shifts more quickly toward higher-PPL prompts.

Across the four Qwen models and Llama3.1-8B, PREPO achieves the highest average performance, and variants that remove or modify either component show consistent drops, indicating that both the perplexity-based schedule and the relative-entropy weighting contribute to PREPO's effectiveness. The nonlinear schedule performs competitively but remains slightly below the linear schedule, suggesting that the linear form is already a stable and effective design choice. Full per-benchmark results are reported in Appendix I (Tables 5 and 6).

4.4 Alternative Selection Ratio

Figure 2 and Table 8 (see Appendix I) show the effect of varying the online batch selection ratio (K/N) with N fixed for PREPO.

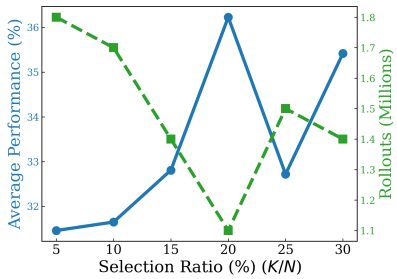


Figure 2: Performance across different selection ratios K/N.

The 20% ratio achieves the best average performance while also requiring the fewest rollouts. A 30% ratio is competitive on some benchmarks but less efficient, while 25% does not improve overall accuracy. Smaller ratios (15%, 10%, 5%) degrade performance and consume more rollouts. Overall, 20% strikes the best balance between accuracy and efficiency, and is adopted as the default in our experiments.

Further analysis of the relative-entropy weight distribution and effective batch size (Appendix J) confirms that the weighting redistributes gradient mass without inflating the effective batch size.

5 Conclusion and Future Work

This work establishes an information-centric framework for efficient data selection in RLVR, grounded in the principle that training data value should be characterized purely from the policy's intrinsic information structure. By decomposing the selection problem into two fundamental dimensions, namely input-side control via policy–input adaptation and rollout-side control via gradient information gain, we derive PREPO as a theoretically consistent instantiation that operates at near-zero additional cost.

Beyond the empirical efficiency gains of up to $3\times$ rollout reduction while maintaining competitive performance, PREPO demonstrates that RLVR can be guided by interpretable, dynamic, policy-intrinsic cues rather than static heuristics or auxiliary models.

The framework is model-agnostic, task-agnostic, and extensible. Future work may incorporate additional intrinsic signals, explore alternative scheduling functions, or combine information-centric selection with system-level optimizations. We hope this work provides a principled and reusable foundation for efficient reinforcement learning with large language models.

Appendix roadmap. Appendices D–E analyze prompt properties and training dynamics. Appendices F–H provide baseline definitions, implementation details, and evaluation procedures. Appendix I reports additional results, including ablations and sensitivity analyses. Appendix L discusses common concerns such as problem diversity and memorization risk. Appendix K provides additional evidence that PPL functions as a genuine policy-intrinsic signal, controlling for prompt length, structural complexity, and language.

6 Limitations

This study has several limitations that should be acknowledged. (1) response lengths were restricted to 32K tokens, leaving the applicability of PREPO to models generating substantially longer outputs an open question; and (2) the evaluation was limited to mathematical reasoning tasks, while its effectiveness in other domains remains to be explored.

References

Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh,

Chase Blagden, Violet Xiang, Dakota Mahan, and Nick Haber. 2025. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models. *Preprint*, arXiv:2502.17387.

Art of Problem Solving. 2024. 2024 aime i. https://artofproblemsolving.com/wiki/index.php/2024_AIME_I. Accessed: 2025-06-09.

Art of Problem Solving. 2025. 2025 aime i. https://artofproblemsolving.com/wiki/index.php/2025_AIME_I. Accessed: 2025-06-09.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. 2025. Online difficulty filtering for reasoning oriented reinforcement learning. *arXiv preprint arXiv:2504.03380*.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, Chengjun Xiao, Chengyu Du, Chi Zhang, Chu Qiao, Chunhao Zhang, Chunhui Du, Congchao Guo, Da Chen, Deming Ding, and 80 others. 2025a. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *ArXiv*, abs/2506.13585.

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

Xinjie Chen, Minpeng Liao, Guoxin Chen, Chengxi Li, and Biao Fu. 2025c. From data-centric to sample-centric: Enhancing llm reasoning via progressive optimization. *arXiv preprint arXiv:2507.06573*.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. 2025. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, and 1 others. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

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*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng 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. *arXiv preprint arXiv:2402.14008*.

Ehsan Kamalloo, Shipeng Li, Shikun Li, Zhiqin Yang, Xinghua Zhang, Gaode Chen, Xiaobo Xia, Hengyu Liu, and Zhe Peng. 2025. Learnalign: Reasoning data selection for reinforcement learning in llms based on improved gradient alignment. *arXiv preprint arXiv:2506.11480*.

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

Kai Li, Xinggao Fan, and X. Liu. 2025. Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning. *arXiv preprint arXiv:2504.13818*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023a. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

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

X Liu and 1 others. 2025. Improving data efficiency for llm reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. *arXiv preprint arXiv:2506.05316*.

Yun Qu, Qi Wang, Yixiu Mao, Vincent Tao Hu, Björn Ommer, and Xiangyang Ji. 2025. Can prompt difficulty be online predicted for accelerating rl finetuning of reasoning models? *arXiv preprint arXiv:2507.04632*.

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

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

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*.

Qwen Team. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, and 1 others. 2025. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*.

Zhenting Wang and Cui Guofeng. 2025. Dump: Automated distribution-level data selection for rl. *arXiv preprint arXiv:2504.09710*.

Mingqi Wu, Zhihao Zhang, Qiaole Dong, Zhiheng Xi, Jun Zhao, Senjie Jin, Xiaoran Fan, Yuhao Zhou, Yanwei Fu, Qin Liu, and 1 others. 2025. Reasoning or memorization? unreliable results of reinforcement learning due to data contamination. *arXiv preprint arXiv:2507.10532*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Enci Zhang, Xingang Yan, Wei Lin, Tianxiang Zhang, and Qianchun Lu. 2025. Learning like humans: Advancing llm reasoning capabilities via adaptive difficulty curriculum learning and expert-guided self-reformulation. *arXiv preprint arXiv:2505.08364*.

Haizhong Zheng, Yang Zhou, Brian R Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. 2025. Act only when it pays: Efficient reinforcement learning for llm reasoning via selective rollouts. *arXiv preprint arXiv:2506.02177*.

Yinmin Zhong, Zili Zhang, Bingyang Wu, Shengyu Liu, Yukun Chen, Changyi Wan, Hanpeng Hu, Lei Xia, Ranchen Ming, Yibo Zhu, and 1 others. 2024. Optimizing rlhf training for large language models with stage fusion. *arXiv preprint arXiv:2409.13221*.

Appendices

A Empirical Validation of Policy-Intrinsic Signals **11**
- A.1 PPL Distinguishes Prompts by Learning Value 11
- A.2 Training Dynamics of Different Adaptation Levels 11
- A.3 Comparison with Random Sampling 11

B Proofs of Propositions **12**

C Disclaimer on LLM Usage **13**

D HIGH-PPL and LOW-PPL Prompts **13**

E More Training Dynamics of LOW-PPL and HIGH-PPL Prompts **13**

F Description of Baseline Methods **15**

G Code Implementation **15**

H Details of Evaluation Framework **15**

I Additional Experimental Results **15**

J Properties of Relative-Entropy Weights **16**

K Additional Evidence for PPL as a Policy-Intrinsic Signal **17**
- K.1 Controlling for Prompt Length . . 17
- K.2 Language × Length Controlled Analysis 17
- K.3 Structural Complexity Analysis . . 17
- K.4 Gradient Contribution by PPL Decile 17

L Discussion **19**
- L.1 Do the Results Generalize to Other Training Datasets? 19
- L.2 What Does the PPL-schedule Contribute During Training? 19
- L.3 What Does Relative Entropy Bring to Training? 19
- L.4 Does Prompt Perplexity Change During Training? 19
- L.5 What Is the Time Cost of PREPO? 19
- L.6 How Does PREPO Affect Problem Diversity? 19
- L.7 How Does PREPO Compare to Training Without Filtering? 19
- L.8 Does PREPO Lead to Memorization of Training Data? 21

L.9 How Sensitive Is PREPO to Extreme Entropy Values? 21

A Empirical Validation of Policy-Intrinsic Signals

This section provides empirical evidence that perplexity serves as a meaningful, non-trivial policy-intrinsic signal, supporting the theoretical framework in Section 3.2.

A.1 PPL Distinguishes Prompts by Learning Value

We examine prompt perplexity on the DAPO-Math-17K dataset (Yu et al., 2025) (17,917 samples). Table 4 shows that prompt PPL exhibits a statistically significant negative correlation with passrate@16 across all tested models, confirming that PPL captures meaningful variation in how the policy interacts with different inputs.

Table 4: Correlation between prompt PPL and passrate@16. (*** $p < 0.001$, ** $p < 0.05$)

Qwen2.5-7B	Qwen2.5-Math-7B	Qwen2.5-Math-1.5B	Qwen3-4B	LLama3.1-8B
-0.233^{***}	-0.183^{**}	-0.186^{***}	-0.169^{***}	-0.199^{***}

A.2 Training Dynamics of Different Adaptation Levels

We compare prompts from the lowest 20% (LOW-PPL) and highest 20% (HIGH-PPL) of the PPL distribution (see Appendix D for examples). Figure 3 illustrates their training dynamics on Qwen2.5-Math-7B.

The two groups exhibit complementary behavior. LOW-PPL prompts drive rapid reward improvement and validation accuracy gains in early training, but also lead to faster entropy collapse. In contrast, HIGH-PPL prompts sustain higher entropy and lower zero-advantage ratio, yielding stronger performance in later stages.

These trends are consistent across other Qwen models (Appendix E). For Llama3.1-8B (Figure 8), a similar pattern appears, though overall performance remains limited as the validation dataset is too challenging for Llama models.

A.3 Comparison with Random Sampling

To confirm that PPL-based grouping offers value beyond chance, we compare with a random 20% subset. As shown in Figure 4, the random group consistently falls between the LOW-PPL and HIGH-PPL groups across all metrics, demonstrating that PPL provides a non-trivial signal for data characterization.

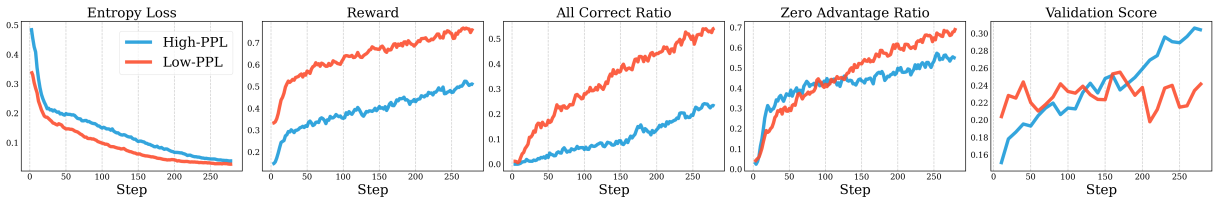


Figure 3: Training dynamics of LOW-PPL vs. HIGH-PPL prompts on Qwen2.5-Math-7B. *(a)* HIGH-PPL *prompts maintain higher entropy. (b)* LOW-PPL *prompts yield faster reward gains. (c)* LOW-PPL *prompts reach higher all-correct ratios earlier. (d)* LOW-PPL *prompts show higher zero-advantage ratios in later stages. (e)* HIGH-PPL *prompts eventually achieve stronger performance.*

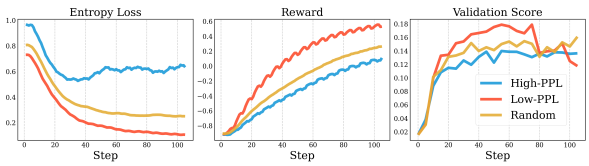


Figure 4: Comparison among LOW-PPL, HIGH-PPL, and Random subsets.

B Proofs of Propositions

Proof of Proposition 1. Let $\mu_{\mathcal{B}}$ denote the uniform distribution over $\mathcal{B}$ and $\mu_{\mathcal{I}}$ the uniform distribution over $\mathcal{I}$. The total variation distance between them is

$$\mathrm{TV}(\mu_{\mathcal{B}}, \mu_{\mathcal{I}}) = \frac{1}{2} \sum_{x \in \mathcal{B}} \left| \frac{1}{N} - \frac{\mathbb{1}[x \in \mathcal{I}]}{\gamma N} \right|.$$

For $x \in \mathcal{I}$ (of which there are γN), each term contributes $\left|\frac{1}{N} - \frac{1}{\gamma N}\right| = \frac{1-\gamma}{\gamma N}$, totaling $\gamma N \cdot \frac{1-\gamma}{\gamma N} = 1 - \gamma$. For $x \notin \mathcal{I}$ (of which there are $(1-\gamma)N$), each term contributes $\frac{1}{N}$, totaling $1 - \gamma$. Hence $\mathrm{TV}(\mu_{\mathcal{B}}, \mu_{\mathcal{I}}) = \frac{1}{2} \cdot 2(1-\gamma) = 1 - \gamma$.

By the variational characterization of total variation, for any measurable function f with $|f| \leq M$,

$$|\mathbb{E}_{\mu_{\mathcal{I}}}[f] - \mathbb{E}_{\mu_{\mathcal{B}}}[f]| \leq 2M \cdot \mathrm{TV}(\mu_{\mathcal{B}}, \mu_{\mathcal{I}}) = 2M(1-\gamma).$$

Setting $f = L$ completes the proof. □

Proof of Proposition 2. We prove each property in turn.

(1) Conservation. Since $\bar{H}_i \geq 0$ (entropy is non-negative), $w_i = \bar{H}_i / \bar{H} \geq 0$. For the sum,

$$\frac{1}{G} \sum_{i=1}^{G} w_i = \frac{1}{G} \sum_{i=1}^{G} \frac{\bar{H}_i}{\bar{H}} = \frac{1}{\bar{H}} \cdot \frac{1}{G} \sum_{i=1}^{G} \bar{H}_i = \frac{\bar{H}}{\bar{H}} = 1.$$

(2) Reduction to GRPO. If $\bar{H}_i = \bar{H}$ for all i, then $w_i = \bar{H}/\bar{H} = 1$ for all i, and $\alpha_i = w_i/G = 1/G$, recovering the standard GRPO objective.

(3) Bounded concentration. By definition, $w_i = \bar{H}_i / \bar{H}$. If $\bar{H}_i \leq M_H$, then $w_i \leq M_H / \bar{H}$.

(4) Signal amplification.

$$\frac{1}{G} \sum_{i=1}^{G} w_i \bar{H}_i = \frac{1}{G} \sum_{i=1}^{G} \frac{\bar{H}_i^2}{\bar{H}} = \frac{1}{\bar{H}} \cdot \frac{1}{G} \sum_{i=1}^{G} \bar{H}_i^2.$$

Using the identity $\frac{1}{G} \sum_i \bar{H}_i^2 = \mathrm{Var}(\bar{H}_i) + \bar{H}^2$ and $\mathrm{CV}^2 = \mathrm{Var}(\bar{H}_i)/\bar{H}^2$,

$$\frac{1}{\bar{H}} \left(\mathrm{Var}(\bar{H}_i) + \bar{H}^2 \right) = \bar{H} \left(\frac{\mathrm{Var}(\bar{H}_i)}{\bar{H}^2} + 1 \right) = \bar{H}(1 + \mathrm{CV}^2).$$

Since $\mathrm{CV}^2 \geq 0$, the amplification factor $1 + \mathrm{CV}^2 \geq 1$, with equality if and only if all rollouts share the same entropy. □

C Disclaimer on LLM Usage

The use of LLMs is permitted as a general-purpose assistance tool. In this work, LLMs were employed solely for grammar correction and sentence rephrasing. Their role was restricted to improving clarity and style; therefore, they are not considered contributors to the research.

D HIGH-PPL and LOW-PPL Prompts

As illustrated in Figure 5, HIGH-PPL prompts exhibit a greater prevalence of non-English characters relative to LOW-PPL prompts. This pattern is consistently observed across both the Qwen2.5-series and Llama model families.

Example of LOW-PPL Problems

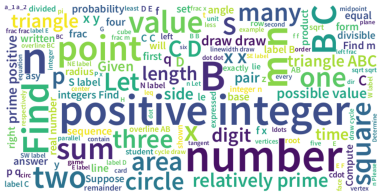
- Cube $ABCDEFGH$, labeled as shown below, has edge length 1 and is cut by a plane passing through vertex D and the midpoints M and N of $\overline{AB}$ and $\overline{CG}$ respectively. The plane divides the cube into two solids. Find the volume of the larger of the two solids. [asy] import cse5; unitsize(8mm); pathpen=black; pair A = (0,0), B = (3.8,0), C = (5.876,1.564), D = (2.076,1.564), E = (0,3.8), F = (3.8,3.8), G = (5.876,5.364), H = (2.076,5.364), M = (1.9,0), N = (5.876,3.465); pair[] dotted = A,B,C,D,E,F,G,H,M,N; D(A–B–C–G–H–E–A); D(E–F–B); D(F–G); pathpen=dashed; D(A–D–H); D(D–C); dot(dotted); label("A",A,SW); label("B",B,S); label("C",C,SE); label("D",D,NW); label("E",E,W); label("F",F,SE); label("G",G,NE); label("H",H,NW); label("M",M,S); label("N",N,NE); [/asy]The answer is in the form
x0cracmn, where gcd(m, n) = 1. Please provide the value of m + n.
- The number $a = \frac{p}{q}$, where p and q are relatively prime positive integers, has the property that the sum of all real numbers x satisfying
$$\lfloor x \rfloor \cdot \{x\} = a \cdot x^2$$
is 420, where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x and $\{x\} = x - \lfloor x \rfloor$ denotes the fractional part of x. What is $p + q$?
- Let a, b be positive integers satisfying
$$\sqrt{\frac{ab}{2b^2 - a}} = \frac{a + 2b}{4b}$$
. Find $|10(a - 5)(b - 15)| + 8$.
- What is the greatest common divisor of $121^2 + 233^2 + 345^2$ and $120^2 + 232^2 + 346^2$?

Example of HIGH-PPL Problems

- At the Lexington High School, each student is given a unique five-character ID consisting of uppercase letters. Compute the number of possible IDs that contain the string "LMT".
- 设3 阶实对称矩阵A 的三个特征值分别为$-1, -1, 2$，且$(1, 1, -1)^T$ 是特征值2 所对应的特征向量. 记A 中所有元素的平方和为I，则$[I]$=_____. (English: Let the eigenvalues of A real symmetric matrix of order 3 be $-1, -1, 2$, respectively, and $(1, 1, -1)^T$be the eigenvector corresponding to eigenvalue 2. For the sum of squares of all elements in the A I, is $[I]$ = _____.)
- 求六个元素的置换群S_6 中6 阶元素的个数。 (English: Find the number of elements of order 6 in the permutation group S_6 of six elements.)
- Three bells begin to ring simultaneously. The intervals between strikes for these bells are, respectively, $\frac{4}{3}$ seconds, $\frac{5}{3}$ seconds, and 2 seconds. Impacts that coincide in time are perceived as one. How many beats will be heard in 1 minute? (Include first and last.)

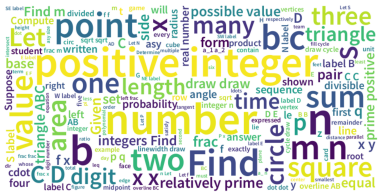
E More Training Dynamics of LOW-PPL and HIGH-PPL Prompts

For Qwen2.5-7B, Qwen2.5-Math-1.5B, and Llama3.1-8B, we found similar trends across training dynamics of LOW-PPL and HIGH-PPL groups, as shown in Figure 6, 7, and 8.



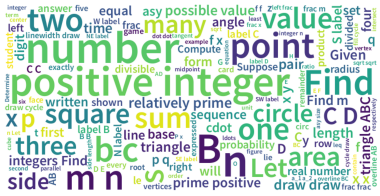
(a) LOW-PPL Prompts (Qwen2.5-7B)

(b) HIGH-PPL Prompts (Qwen2.5-7B)



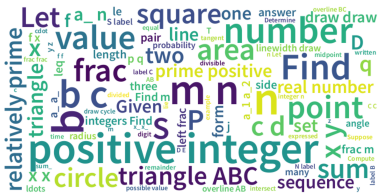
(c) LOW-PPL Prompts (Qwen2.5-Math-1.5B)

(d) HIGH-PPL Prompts (Qwen2.5-Math-1.5B)



(e) LOW-PPL Prompts (Qwen2.5-Math-7B)

(f) HIGH-PPL Prompts (Qwen2.5-Math-7B)



(g) LOW-PPL Prompts (Llama3.1-8B)

(h) HIGH-PPL Prompts (Llama3.1-8B)

Figure 5: Wordcloud of the Most Frequent Words in Low-/HIGH-PPL Prompts

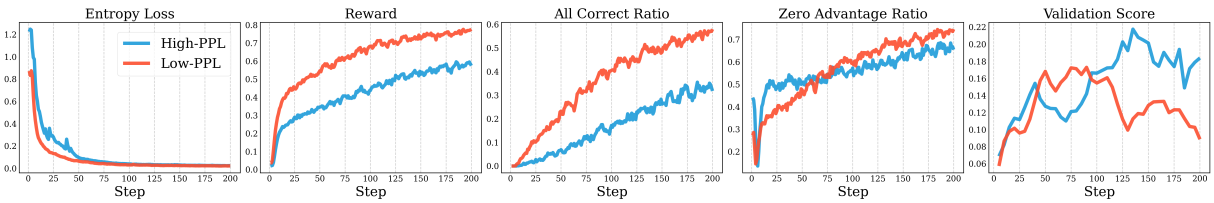


Figure 6: Training Dynamics of LOW-PPL vs. HIGH-PPL Prompts on Qwen2.5-7B.

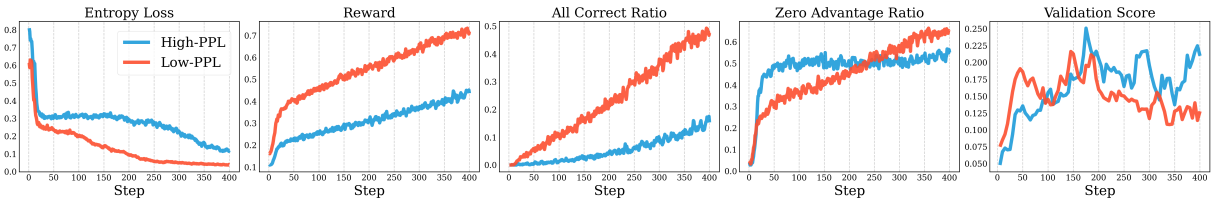


Figure 7: Training Dynamics of LOW-PPL vs. HIGH-PPL Prompts on Qwen2.5-Math-1.5B.

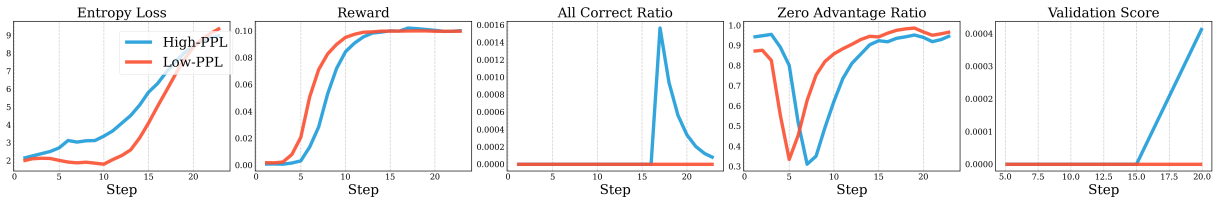


Figure 8: Training Dynamics of LOW-PPL vs. HIGH-PPL Prompts on Llama3.1-8B.

F Description of Baseline Methods

We compare PREPO against the following three baseline strategies.

- **Dynamic Sampling (DS).** Dynamic Sampling, as introduced in DAPO (Yu et al., 2025), dynamically filters out prompt groups whose generated responses all produce identical rewards (i.e., zero variance). In each training batch, DS resamples such uninformative prompt groups, thereby ensuring that the batch maintains a sufficient proportion of prompts that give meaningful gradient signals. However, it can still incur high rollout costs because many sampled prompts may remain uninformative until they are filtered.

- **Random.** The Random baseline uniformly selects prompts and associated rollouts without regard to historical feedback or variance in reward. All prompts are treated equally, so there is no mechanism to avoid rollouts on uninformative or zero-variance prompts. This method serves as a lower bound in terms of data selection sophistication.

- **GRESO.** GRESO (GRPO with Efficient Selective Rollout) (Zheng et al., 2025) is a lightweight, online, pre-rollout filtering approach. It uses statistics of reward dynamics over previous epochs to predict which prompts are likely to be uninformative (e.g., zero variance among responses) and skips them before performing rollouts.

G Code Implementation

At each global step, PPL-schedule computes perplexity on the current batch of prompts and selects the subset that matches the current progress ratio. This removes the need for tuning and avoids extra computational cost.

```
def fit(self):
    ......
    # compute perplexity for each prompt
    batch = self.actor_rollout_wg.get_ppl_batch(batch)
    # compute training progress
    rho = self.global_steps / self.total_training_steps
    # determine the prompts to select
    start_idx = math.floor((N - K) * rho)
    idx = torch.argsort(batch['ppl'])[start_idx : start_idx + K]
    # select prompts based on the indices
    batch = batch.select_via_index(idx)
    ......
```

Similarly, the relative-entropy component uses only a few lines of code, requires no manual tuning, and adds no overhead beyond the entropy computation that is already part of the existing pipeline.

```
def update_policy(self, data: DataProto):
    ......
    # all return: (bsz, response_length)
    entropy, log_prob = self._forward_micro_batch(micro_batch=data, temperature=temperature)
    sample_entropy = verl_F.masked_mean(entropy, response_mask, axis=1).detach()
    batch_entropy = verl_F.masked_mean(entropy, response_mask).detach()
    weight = sample_entropy/batch_entropy
    advantages *= weight[:, None]
    ......
```

H Details of Evaluation Framework

We use the `_validate()`[4] function to evaluate the model. In addition, the basic verifier is `math-verify`, and we add a LaTeX normalization package (`latex2sympy2_extended`) for the AIME and Olympiad bench.

I Additional Experimental Results

In this section, we show the full training-dynamics comparison between random and PREPO in Figures 9 and 10, the full per-benchmark ablation results in Tables 5 and 6, the performance comparison across different evaluation metrics in Table 7, and the sensitivity analysis of the selection ratio in Table 8.

[4]We adopt from the public Github repository: line 569 in `ray_trainer.py`

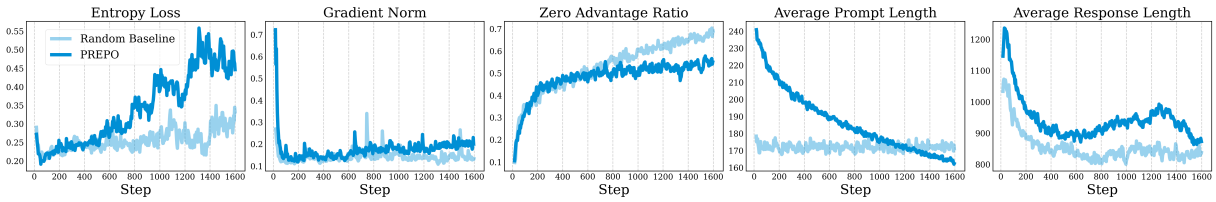


Figure 9: Full comparison between PREPO and random selection on Qwen2.5-Math-7B.

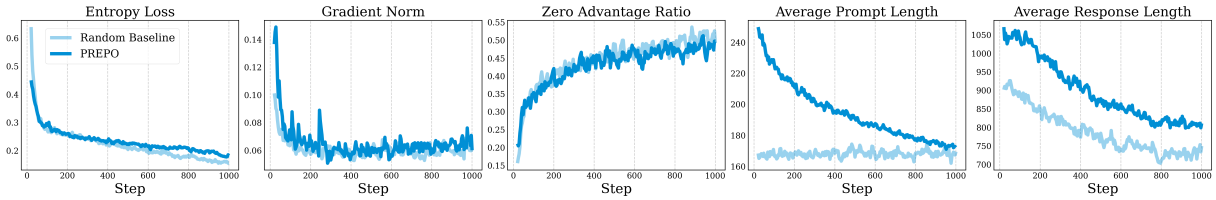


Figure 10: Full comparison between PREPO and random selection on Qwen2.5-Math-1.5B.

Table 5: Ablation study on Qwen. Performance comparison (%) between PREPO and (1) linear PPL-schedule, (2) random filtering with relative-entropy, (3) linear PPL-schedule with entropy loss, (4) non-linear PPL-schedule with relative-entropy. *Best results are highlighted in bold or underlined.*

Model	Method	AIME25	AIME24	MATH	Olympiad Bench	Avg ↑
Qwen2.5-Math-7B	PREPO	12.81	26.15	77.80	41.58	**39.59**
	w/o relative entropy	10.00	23.33	74.60	39.21	36.79
	w/o PPL-schedule	11.87	25.73	76.48	34.43	37.13
	w/ entropy loss	10.73	23.54	75.25	38.81	37.08
	w/ non-linear schedule	12.71	26.15	76.40	40.12	38.85
Qwen2.5-7B	PREPO	10.20	16.09	76.30	39.85	**35.61**
	w/o relative entropy	6.98	16.41	75.70	38.47	34.39
	w/o PPL-schedule	8.89	16.04	79.41	22.54	31.72
	w/ entropy loss	6.25	16.35	77.36	21.04	30.25
	w/ non-linear schedule	9.58	16.25	76.31	39.60	35.44
Qwen2.5-Math-1.5B	PREPO	20.00	16.67	76.25	32.00	**36.23**
	w/o relative entropy	10.21	15.68	72.10	30.50	32.12
	w/o PPL-schedule	11.81	13.12	70.21	30.02	31.29
	w/ entropy loss	6.46	16.56	73.77	30.99	31.95
	w/ non-linear schedule	9.15	15.83	75.85	30.26	32.77
Qwen3-4B	PREPO	66.67	80.00	96.60	60.67	**75.99**
	w/o relative entropy	64.77	72.70	90.54	59.06	71.77
	w/o PPL-schedule	64.99	77.73	95.03	57.86	73.90
	w/ entropy loss	61.10	75.17	88.39	60.54	71.28
	w/ non-linear schedule	61.45	74.47	92.37	60.67	72.24

J Properties of Relative-Entropy Weights

Effective batch size. Relative-entropy weighting does not inflate the effective batch size $\frac{1}{B}\sum_i w_i$; it only redistributes gradient mass across sequences. Since the weights are normalized by $\bar{H}$,

$$\frac{1}{B}\sum_{i=1}^{B} w_i \cdot |o_i| = \frac{1}{B\bar{H}}\sum_{i=1}^{B}|o_i|\bar{H}_i = \frac{1}{B\bar{H}}\sum_{i=1}^{B}\sum_{t=1}^{|o_i|}H_{i,t} = \frac{\sum_{i=1}^{B}|o_i|}{B}. \tag{13}$$

The token-weighted average weight equals the average sequence length; if all sequences have equal length, then $\frac{1}{B}\sum_i w_i = 1$. As shown in Figure 11, the effective batch size for PREPO on Qwen2.5-Math-1.5B stays close to B throughout training. At early steps, LOW-PPL prompts yield low-entropy responses, so higher-entropy rollouts receive more weight, pushing the average above B. As training advances and higher-PPL prompts enter the schedule, the overall entropy rises, and normalization shifts the average slightly below B.

Table 6: Ablation study on Llama. Performance comparison (%) between PREPO and (1) linear PPL-schedule, (2) random filtering with relative-entropy, (3) linear PPL-schedule with entropy loss, (4) non-linear PPL-schedule with relative-entropy. *Best results are highlighted in bold or underlined.*

Model	Method	GSM8K	MATH	Avg ↑
Llama3.1-8B	PREPO	<u>51.10</u>	<u>21.81</u>	**36.55**
	w/o relative entropy	46.85	18.25	32.55
	w/o PPL-schedule	34.55	16.29	25.42
	w/ entropy loss	4.45	8.30	6.33
	w/ non-linear schedule	46.85	20.56	33.71

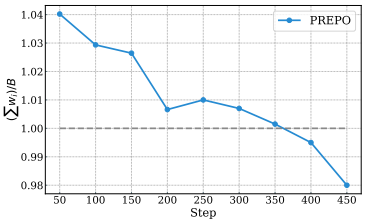


Figure 11: Trend of effective batch size during training (Qwen2.5-Math-1.5B).

K Additional Evidence for PPL as a Policy-Intrinsic Signal

This section provides additional analyses that corroborate the use of perplexity as a meaningful policy-intrinsic signal, independent of superficial confounds such as prompt length or surface-level formatting.

K.1 Controlling for Prompt Length

A natural concern is whether PPL merely reflects prompt length rather than genuine policy–input adaptation. To address this, we partition prompts into fixed-length buckets and recompute the Spearman correlation between PPL and passrate@16 within each bucket. Table 9 reports the results on Qwen2.5-Math-7B.

The correlation remains negative and statistically significant in the medium-length bucket across all models, and persists in the long bucket for Qwen3-4B. This confirms that the PPL signal is not reducible to prompt length alone.

K.2 Language × Length Controlled Analysis

To further test whether cross-language lexical distribution explains the PPL–performance relationship, we compute the Spearman correlation between PPL and passrate@16 within each language group and within each prompt-length bucket. These results show that (1) the PPL–passrate relationship is not limited to a particular language group; (2) the strong negative correlation persists in the overwhelmingly dominant English-only subset; and (3) the correlation remains after controlling for prompt length within each language group. Tables 10, 11, and 12 report the per-bucket results for the English-only, English-mixed, and CJK subsets, respectively.

Across all three language groups, the negative correlation between PPL and passrate@16 persists within each length bucket, with statistical significance in nearly every cell. The only exception is the smallest CJK bucket (N=275, long prompts), where the correlation remains negative but is not significant. This pattern indicates that the PPL signal reflects policy–input adaptation rather than a confound from cross-language lexical distribution or prompt length.

K.3 Structural Complexity Analysis

To further examine what PPL captures beyond surface statistics, we define a quantitative structural complexity score for each prompt based on features including word count, number of equation markers, sentence count, parentheses, LaTeX commands, numeric values, constraints, operators, and logical markers. We then compute the correlation between PPL and this complexity score.

The analysis reveals that PPL correlates positively with structural complexity even after controlling for word count. Prompts with higher PPL tend to contain more mathematical operators, nested structures, and logical constraints, indicating that the policy's adaptation difficulty is driven by genuine compositional complexity rather than length.

K.4 Gradient Contribution by PPL Decile

We measure the average gradient norm contributed by prompts in each PPL decile during training. Table 13 reports the per-decile gradient norms on Qwen2.5-Math-7B.

The gradient norm decreases monotonically from the 20–30% decile onward, with the highest-PPL prompts contributing roughly half the gradient magnitude of lower-PPL prompts. This provides direct evidence that policy–input adaptation level (as measured by PPL) correlates with the magnitude of useful learning signal, supporting the theoretical motivation for our scheduling mechanism.

Table 7: Details of performance comparison (%) on Qwen models. *Avg (avg@16/32/64) means the average of AIME25 (avg@16/32/64), AIME24 (avg16/32/64), MATH, and Olympiad. Best results are highlighted in bold or underlined.*

Method	AIME25 (avg@16)	AIME25 (avg@32)	AIME25 (avg@64)	AIME24 (avg@16)	AIME24 (avg@32)	AIME24 (avg@64)	Avg ↑ (avg@16)	Avg ↑ (avg@32)	Avg ↑ (avg@64)
Qwen2.5-7B	1.25	2.60	1.17	4.17	4.90	4.95	27.69	28.21	27.87
+ DS	7.92	7.08	7.19	17.55	15.00	14.90	34.97	34.76	34.13
+ Random	6.98	6.35	6.51	16.41	15.63	15.89	34.39	34.04	34.14
+ GRESO	9.22	7.29	7.50	10.83	13.65	13.39	34.39	34.92	34.90
+ PREPO (Ours)	10.21	10.62	10.00	16.09	15.52	16.35	**35.61**	**35.72**	**35.63**
Qwen2.5-Math-1.5B	3.54	3.75	4.48	10.21	8.33	8.96	24.23	23.81	24.15
+ DS	10.83	13.76	14.54	25.83	21.72	23.04	34.10	33.80	34.33
+ Random	20.00	17.45	18.81	16.67	15.42	16.50	35.86	34.91	35.52
+ GRESO	15.38	11.40	15.39	20.00	19.20	17.97	34.16	32.86	33.56
+ PREPO (Ours)	20.00	19.69	19.38	16.67	16.98	17.03	**36.23**	**36.23**	**36.17**
Qwen2.5-Math-7B	8.95	9.17	8.80	17.50	20.80	19.22	35.45	35.45	34.96
+ DS	13.33	14.27	14.84	33.33	35.63	34.69	34.10	40.36	40.26
+ Random	10.00	12.50	12.50	26.67	29.34	29.87	35.86	40.73	40.86
+ GRESO	18.33	18.58	18.58	25.83	24.38	25.00	37.46	36.90	37.05
+ PREPO (Ours)	12.81	14.41	15.93	26.15	29.17	29.25	**39.59**	**40.75**	**41.15**
Qwen3-4B	30.00	47.30	44.89	53.33	61.60	28.70	57.53	63.92	55.09
+ DS	63.33	53.90	54.80	66.67	63.50	65.30	70.78	67.63	68.30
+ Random	60.00	58.85	60.16	70.00	65.73	64.69	71.33	69.98	70.05
+ GRESO	56.67	58.33	57.24	69.17	71.77	69.79	69.89	70.96	70.19
+ PREPO (Ours)	66.67	62.19	63.39	80.00	74.28	77.45	**75.99**	**73.44**	**74.53**

Table 8: Performance Comparison (%) of PREPO with Different Selction Ratio ($K/\mathcal{B}$, $\mathcal{B}$ fixed) on Qwen2.5-Math-1.5B. *Best results are highlighted in bold or underlined.*

Selection Ratio	AIME25	AIME24	MATH	Olympiad	Avg ↑	# Rollouts ↓
30%	20.00	20.83	75.85	25.00	35.42	1.4M
25%	13.33	20.00	73.55	24.00	32.72	1.5M
20%	20.00	16.67	76.25	32.00	**36.23**	**1.1M**
15%	13.33	20.00	75.75	22.17	32.81	1.4M
10%	15.83	19.17	70.25	21.33	31.65	1.7M
5%	19.17	16.67	69.85	20.17	31.46	1.8M

Table 9: Spearman correlation between PPL and passrate@16 within fixed prompt-length buckets on Qwen2.5-Math-7B (*** $p < 0.001$, ** $p < 0.05$). The correlation persists within each bucket, confirming that PPL captures information beyond prompt length.

Model	Short	Medium	Long
Qwen2.5-7B	-0.13^{**}	-0.14^{**}	-0.05
Qwen2.5-Math-7B	-0.07	-0.14^{**}	-0.05
Qwen3-4B	-0.10	-0.14^{**}	-0.22
Qwen3-4B HL	-0.06	-0.13^{**}	-0.19

Table 10: Spearman correlation between PPL and passrate@16 within prompt-length buckets for English-only prompts (pure ASCII; dominant group).

Length bucket	N	Spearman ρ	p-value
0K–0.2K	4,711	-0.3392	0.0
0.2K–0.3K	5,782	-0.2659	0.0
0.3K–3.2K	6,555	-0.2119	0.0

Table 11: Spearman correlation between PPL and passrate@16 within prompt-length buckets for English-mixed prompts (minor non-ASCII characters).

Length bucket	N	Spearman ρ	p-value
0.2K–0.3K	51	-0.4420	0.0012
0.3K–3.2K	167	-0.3853	0.0

Table 12: Spearman correlation between PPL and passrate@16 within prompt-length buckets for CJK prompts.

Length bucket	N	Spearman ρ	p-value
0K–0.2K	2,288	-0.1158	0.0
0.2K–0.3K	1,165	-0.0962	0.001
0.3K–3.2K	275	-0.0579	0.3392

Table 13: Average gradient norm by PPL decile (Qwen2.5-Math-7B). Lower-PPL prompts contribute larger gradient signals in early training, consistent with their role as high-adaptation samples that provide reliable learning signal.

PPL Decile	Avg. Gradient Norm
0–10% (lowest PPL)	0.01794
10–20%	0.02015
20–30%	0.01474
30–40%	0.01620
40–50%	0.01526
50–60%	0.01479
60–70%	0.01458
70–80%	0.01395
80–90%	0.01204
90–100% (highest PPL)	0.00867

L Discussion

L.1 Do the Results Generalize to Other Training Datasets?

As DAPO-Math-17K contains both English and Chinese questions, prompt perplexity could correlate with language. To exclude the cofounder of language effects, we trained Qwen2.5-Math-7B on a purely English dataset selected from OpenR1-Math-220k[5], and compared PREPO with several baselines under the same training budget. The results are shown in Table 14 and the Figure 12 below. PREPO again achieves lower rollout usage while achieving a higher average performance. This indicates that the gains of PREPO are not driven by language imbalance.

L.2 What Does the PPL-schedule Contribute During Training?

We compared three configurations on Qwen2.5-Math-7B training exclusively with HIGH-PPL prompts, exclusively with LOW-PPL prompts, and the PPL-schedule, which gradually transitions from Low- to HIGH-PPL prompts. The training dynamics are shown in Figure 13.

In terms of entropy loss, the PPL-schedule achieves a balance between the two extremes: entropy decreases steadily but not excessively, thereby mitigating the risk of collapse. With respect to the zero-advantage ratio, the PPL-schedule consistently maintains a lower value, ensuring that a greater proportion of rollouts remain informative throughout training.

L.3 What Does Relative Entropy Bring to Training?

As shown in Figure 14, PREPO with relative entropy further reduces the zero-advantage ratio, thereby improving sample efficiency.

Case Analysis. In Figure 15, we display examples of responses with their token-level entropy from the same mini-batch, where darker colors mean higher entropy, and each sequence is weighted by its relative entropy.

L.4 Does Prompt Perplexity Change During Training?

As shown in Figure 16, we computed the PPL range at each training epoch and observed that it remained relatively stable throughout training. In addition, the PPL of the prompts used for training exhibited minimal variation.

L.5 What Is the Time Cost of PREPO?

As shown in Figure 17, the time consumption of calculating prompt PPL is barely minimal compared to the rollout generation duration.

L.6 How Does PREPO Affect Problem Diversity?

The analysis indicates that PREPO selects a more diverse set of problems than random sampling. Specifically, PREPO achieves broader coverage across Mathematics Subject Classification[6] (MSC) categories during training, as illustrated in Figure 18.

L.7 How Does PREPO Compare to Training Without Filtering?

For Qwen2.5-Math-7B, we observe that PREPO attains performance comparable to training on the full dataset without any filtering, i.e., using 5 times rollouts per step, as shown in Figure 19. The "Less

[5] https://huggingface.co/datasets/open-r1/OpenR1-Math-220k

[6] MSC : https://zbmath.org/classification/

Table 14: Comparison of PREPO and baselines trained on all-English data (Model: Qwen2.5-Math-7B)

Method	AIME25	AIME24	MATH	Olympiad	Avg ↑	#Rollout ↓
DS	11.98	30.00	72.80	42.02	39.20	3256K
Random	10.79	28.25	80.21	36.72	38.99	2457K
GRESO	6.67	30.00	79.20	45.63	40.38	1331K
PREPO	12.92	33.33	78.05	44.10	**42.10**	**860K**

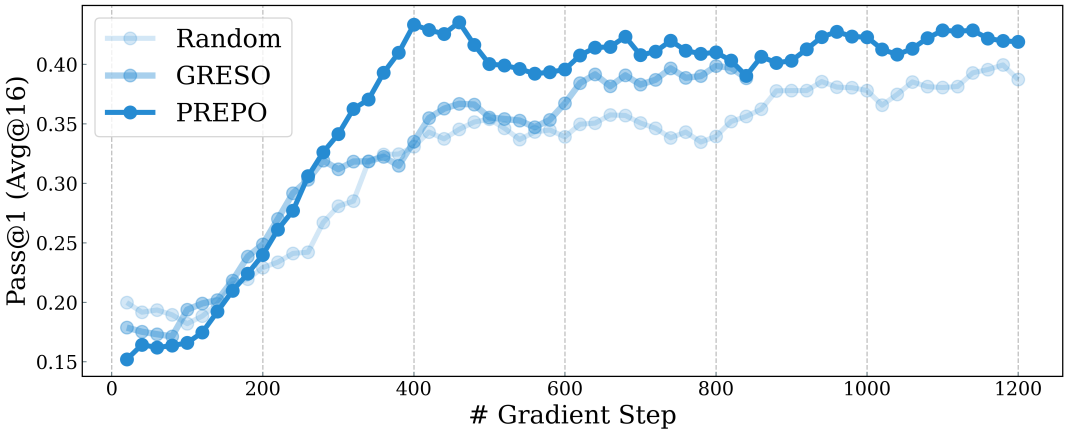
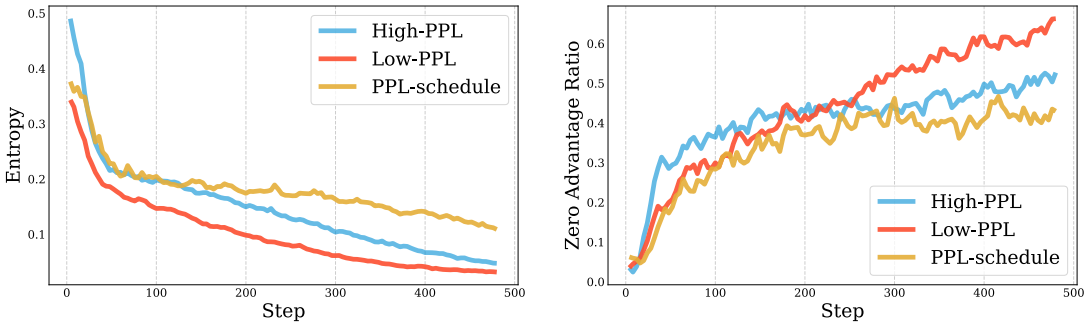
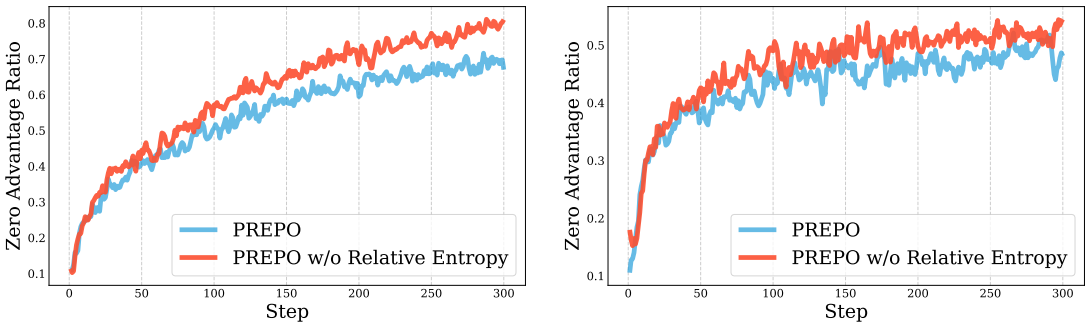


Figure 12: Comparison of performances across PREPO, GRESO, and Random (x-axis: training step, y-axis: validation score, model: Qwen2.5-Math-7B, training data: OpenR1-Math)



(a) Entropy (b) Zero Advantage Ratio

Figure 13: Comparison of Training Dynamics between PPL-schedule and Static PPL Selection (Low- and HIGH-PPL groups)



(a) Qwen2.5-Math-7B (b) Qwen2.5-Math-1.5B

Figure 14: Comparison of zero advantage ratio between PPL-schedule and PREPO.

is More" pattern implies that efficient selection can reduce the amount of data needed for RLVR training.

L.8 Does PREPO Lead to Memorization of Training Data?

Following Wu et al. (2025), we trimmed 40% of each prompt to construct *partial problems*. We then evaluate models on these partial prompts and compute the average pass rate of 16 generations. As shown in Figure 20, the vast majority of partial problems have near-zero pass rate, with only a small fraction achieving non-trivial success. These distributions suggest that the model does not merely memorize the training data but instead requires the full problem context to solve tasks.

L.9 How Sensitive Is PREPO to Extreme Entropy Values?

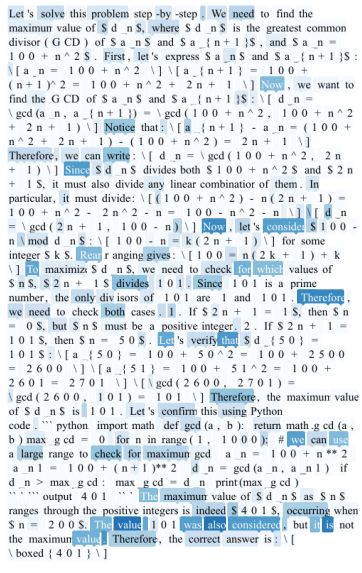
Since $w_i = \bar{H}_i/\bar{H}$ is normalized by the batch mean, a rollout with $\bar{H}_j \gg \bar{H}$ has a large weight $w_j \gg 1$, while simultaneously reducing the weights of the others ($w_i \ll 1$ for $i \neq j$). The derivative

$$\frac{\partial w_i}{\partial \bar{H}_j} = \begin{cases} \dfrac{1}{\bar{H}} - \dfrac{\bar{H}_j}{\bar{H}^2}\dfrac{|o_j|}{\sum_k |o_k|}, & i = j, \\ -\dfrac{\bar{H}_i}{\bar{H}^2}\dfrac{|o_j|}{\sum_k |o_k|}, & i \neq j, \end{cases}$$

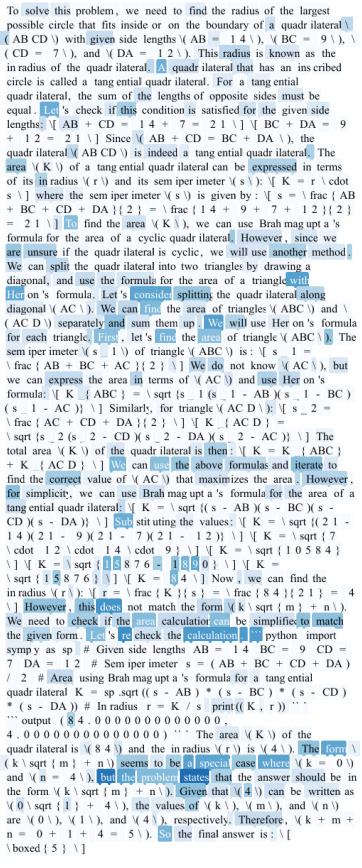
clarifies this behavior: (1) for $i = j$, the two terms partially cancel, so although w_j is already large, its growth rate is moderated as $\bar{H}_j$ increases further; (2) for $i \neq j$, the derivative is negative, confirming that a large $\bar{H}_j$ suppresses the weights of all other rollouts. Empirically, such outliers are rare. As shown in Figure 21, the distribution of relative entropy weights remains stable mainly across training steps 100, 200, 300, and 400 for Qwen2.5-Math-1.5B, where most rollouts clustered near one, with negligible mass on extreme values.

(a) Correct answer
x1.65

(b) Incorrect answer
x1.33

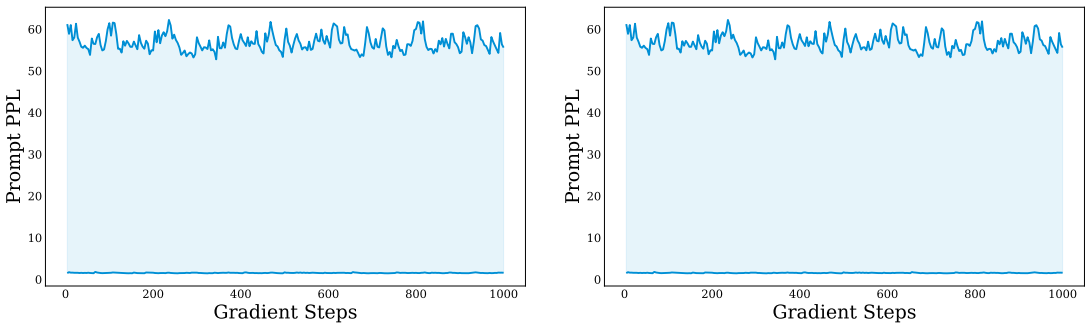


(c) Correct answer
x0.73



(d) Incorrect answer
x0.74

Figure 15: Token-Level Entropy of Sequences within A Mini-batch (with Relative-Entropy in Red).



(a) Qwen2.5-Math-7B

(b) Qwen2.5-Math-1.5B

Figure 16: Range of prompt PPL during training.

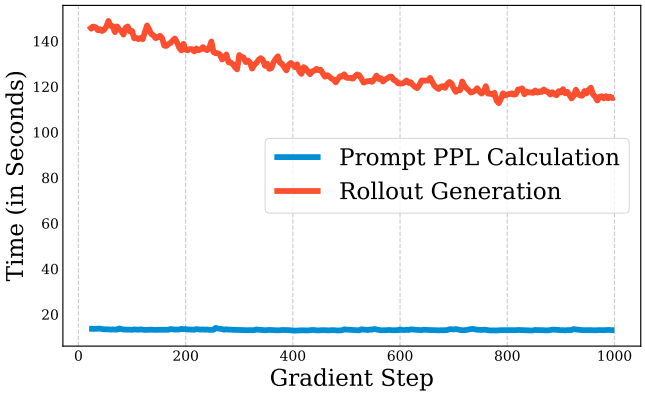
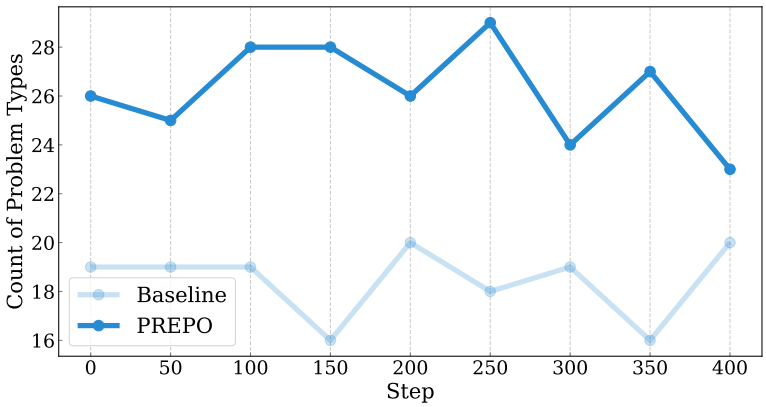
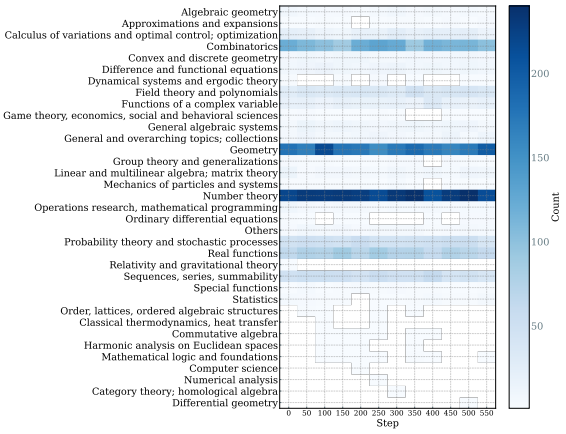


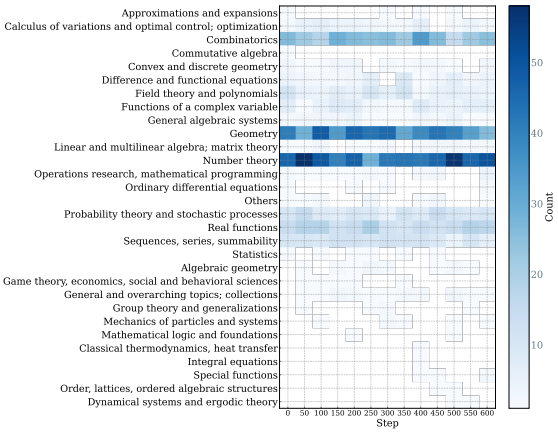
Figure 17: Comparison of calculating prompt PPL and rollout generation



(a) Unique number of MSC tags



(b) MSC frequency of PREPO



(c) MSC frequency of baseline (random)

Figure 18: Comparison of problem diversity between PREPO and baseline on Qwen2.5-Math-7B.

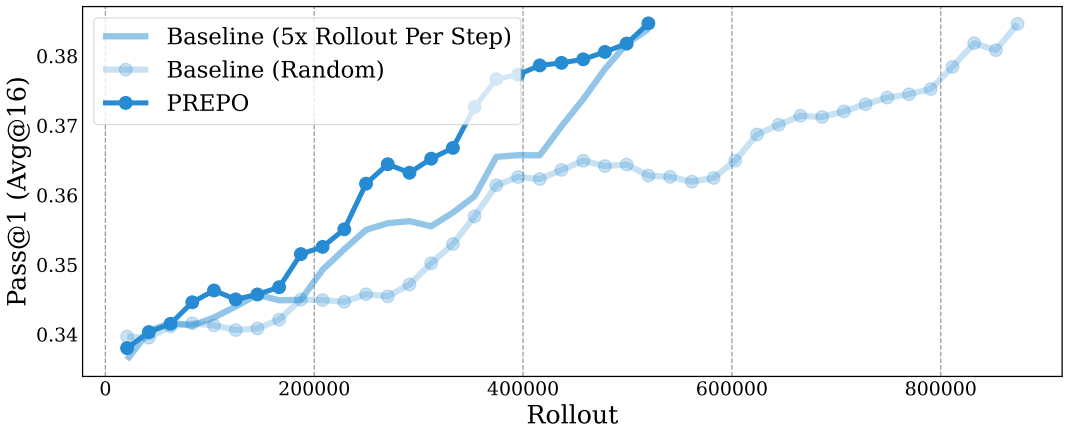
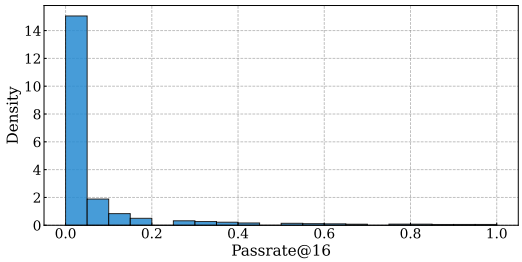
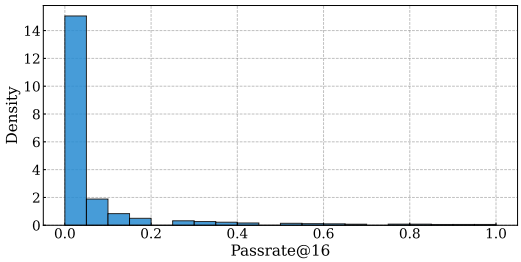


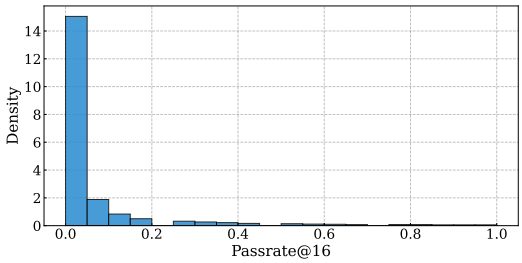
Figure 19: Comparison of PREPO and baselines (a) training w/o filtering (b) random selection.



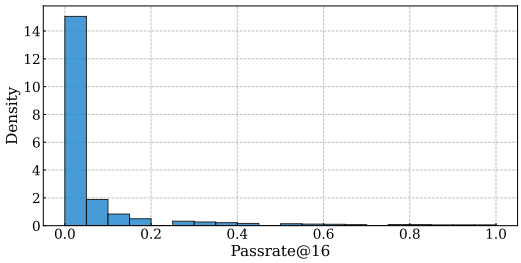
(a) Qwen2.5-Math-1.5B



(b) Qwen2.5-Math-7B

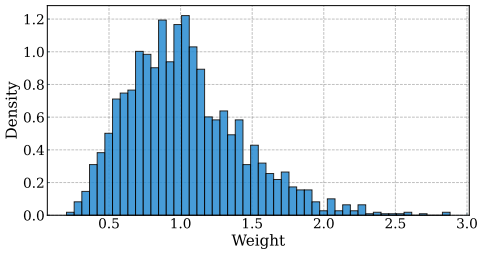


(c) Qwen3-4B

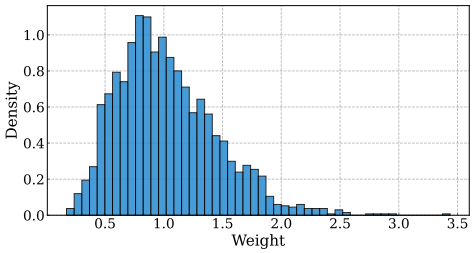


(d) Llama3.1-8B

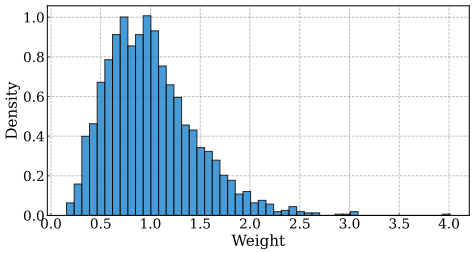
Figure 20: Distribution of passrate@16 over all partial prompts



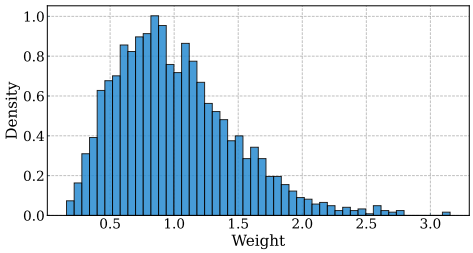
(a) Step 100



(b) Step 200



(c) Step 300



(d) Step 400

Figure 21: Weight Distribution During Training (Qwen2.5-Math-1.5B).