

# 000 BEYOND THE EXPLORATION-EXPLOITATION TRADE- 001 OFF: A HIDDEN STATE APPROACH FOR LLM REASON- 002 ING IN RLVR 003

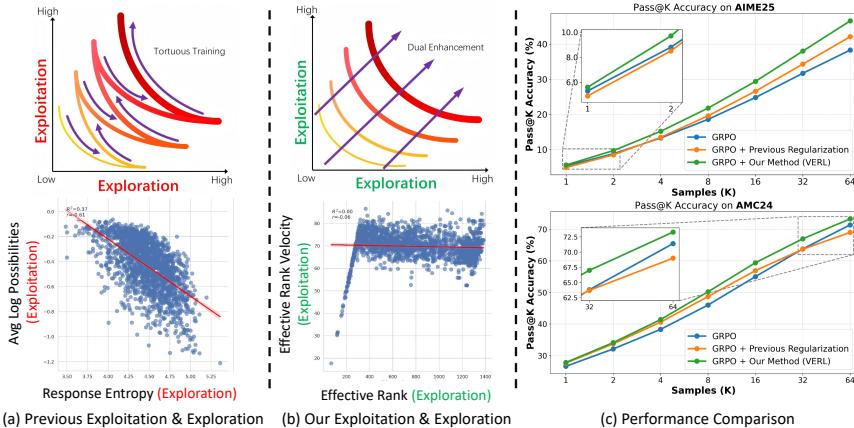
004 **Anonymous authors**  
005  
006

007 Paper under double-blind review  
008  
009

## 010 ABSTRACT 011

012 A prevailing view in Reinforcement Learning with Verifiable Rewards (RLVR)  
013 interprets recent progress through the lens of an exploration-exploitation trade-  
014 off, a perspective largely shaped by token-level metrics. We re-examine this  
015 perspective, proposing that this perceived trade-off may not be a fundamental  
016 constraint but rather an artifact of the measurement level. To investigate this,  
017 we shift the analysis to the semantically rich hidden-state space, adopting Effect-  
018ive Rank (ER) to quantify exploration and proposing its novel first- and second-  
019 order derivatives, named ER Velocity and ER Acceleration, to capture exploita-  
020 tion dynamics. Our analysis reveals that in the semantic space, exploration and  
021 exploitation could be *decoupled* (Sec. 4). This finding reveals an opportunity  
022 to enhance both capacities simultaneously. This insight motivates our method,  
023 Velocity-Exploiting Rank-Learning (VERL), the *first* to operationalize the prin-  
024 ciple of synergistic exploration-exploitation enhancement by directly shaping the  
025 RL advantage function. The key innovation is leveraging the theoretically sta-  
026 ble ERA as a predictive meta-controller to create a synergistic, dual-channel in-  
027 centive structure. Instead of forcing a trade-off, VERL prospectively amplifies  
028 rewards for exploration to preempt overconfidence and reinforces exploitative  
029 gains to consolidate reasoning. Experiments across diverse LLMs and reason-  
030 ing benchmarks show consistent gains, including up to 21.4% absolute accuracy  
031 improvement on the challenging Gaokao 2024 dataset. The code is available at  
032 <https://anonymous.4open.science/r/coding>.  
033

## 034 1 INTRODUCTION



035  
036  
037  
038  
039  
040  
041  
042  
043  
044  
045  
046  
047  
048  
049  
050  
051  
052  
053  
Figure 1: Comparative analysis with the responses of DeepSeek-R1-Distill-Qwen-7B in simpleRL  
test dataset (Zeng et al., 2025). (a) Traditional metrics for exploitation & exploration constrained by  
negative coupling, leading to meandering progress for both capabilities. (b) Our metrics are mutually  
independent. (c) Training regularization with our metrics demonstrates stronger performance in both  
exploitation (small K) and exploration (large K).

054 Recent advancements in Reinforcement Learning with Verifiable Rewards (RLVR) have signifi-  
 055 cantly improved the reasoning abilities of Large Language Models (LLMs). A dominant narrative  
 056 emerging from these recent works (Chen et al., 2025b; Yue et al., 2025; Deng et al., 2025a; Agarwal  
 057 et al., 2025) interprets this progress through the lens of balancing *exploration* (searching for diverse  
 058 reasoning paths) and *exploitation* (refining the most promising known strategies). However, this  
 059 paradigm is almost exclusively rooted in a token-level analysis in action space, where exploration is  
 060 captured by high-entropy token distributions and exploitation by high-confidence, low-entropy ones.  
 061 This has inevitably led to the widespread assumption of an inherent trade-off between the two, as  
 062 the token-level output distribution, which only reflects the model’s hesitation about the next-token  
 063 prediction, is seen as unable to be simultaneously uniform and sharp.

064 While convenient, this token-centric viewpoint introduces significant limitations. Equating explo-  
 065 ration with mere token-level entropy faces an intrinsic dilemma (Fu et al., 2025; Qiao et al., 2025;  
 066 Agarwal et al., 2025): excessively high entropy risks generating incoherent noise, while overly low  
 067 entropy stifles exploration it aims to encourage. Similarly, defining exploitation via hand-crafted  
 068 heuristic rewards (Chen et al., 2025a; Li et al., 2025a; Bensal et al., 2025) produces brittle models  
 069 with poor generalizability as they simply learn to chase surface-level proxies. More fundamentally,  
 070 these token-level proxies are misaligned with how reasoning actually happens (Wei et al., 2022; Yao  
 071 et al., 2023): solutions emerge over multi-token *semantic chunks* (concepts, subgoals), not isolated  
 072 tokens, and a single token cannot correspond to a meaningful greedy decision about a reasoning  
 073 strategy. More related works are discussed in Sec. D. While many works (Cheng et al., 2025a; Deng  
 074 et al., 2025b) are aware to consider both exploration and exploitation as in Fig. 1a, their continued  
 075 reliance on token-level metrics invariably traps them in a cycle of “balancing” the trade-off, instead  
 076 of doubting its existence. This raises a critical question: *Is the exploration-exploitation trade-off*  
 077 *intrinsic to reasoning, or merely an artifact of token-level measurement?*

078 To answer this, we move beyond token-level statistics and study exploration and exploitation di-  
 079 rectly in the semantically rich hidden-state space of response-level trajectories, where prior work has  
 080 shown that transformer representations encode meaningful linguistic and reasoning structure (Jing  
 081 et al., 2025; Sajjad et al., 2022; Valeriani et al., 2023; Matthews et al., 2024; Zhang et al., 2025).  
 082 At this level, we define exploration and exploitation : Effective Rank (ER) measures how broadly  
 083 a hidden-state trajectory spreads across semantic directions, corresponding to representation-level  
 084 exploration, while its temporal derivative, Effective Rank Velocity (ERV), measures how the same  
 085 trajectory refines semantic content along its path, corresponding to representation-level exploitation.  
 086 Concretely, we are the first to apply ER in an RL context and use it to quantify exploration by mea-  
 087 suring the semantic diversity of hidden-state representations: high ER indicates that the model is  
 088 activating diverse semantic directions and widening its search over possible solutions. To capture  
 089 exploitation more precisely, which we define as the efficiency with which a trajectory converges  
 090 toward a solution in representation space, we further introduce Effective Rank Acceleration (ERA),  
 091 the second-order temporal change of ER, which captures the trend of the velocity, indicating whether  
 092 reasoning is accelerating toward a solution or saturating in a stagnant regime. Equipped with these  
 093 semantic-trajectory tools, we uncover a striking result: in the semantic space, exploration and ex-  
 094 ploitation exhibit near-zero correlation (Fig. 1b, bottom). This contrast provides strong evidence that  
 095 the trade-off is not an inherent property of RLVR for reasoning but an artifact of biased token-level  
 096 measurements. It further reveals that these two capacities are not fundamentally antagonistic but  
 097 can, in fact, *be decoupled and enhanced simultaneously* (Fig. 1c).

098 Building on this core insight, we propose Velocity-Exploiting Rank-Learning (VERL), a method  
 099 that moves beyond the trade-off between the two capacities by directly shaping the RL advantage  
 100 function using ER and ERV. Instead of acting as a switch between the two capacities in lower  
 101 dimension, VERL functions as a *tuner* synergistically enhances both capacities in higer dimension.  
 102 Its key innovation is leveraging ERA as a meta-control variable, a choice justified by our theoretical  
 103 proof of its remarkable  $\mathcal{O}(1)$  growth stability (Sec. 3). Specifically, VERL uses ERA as a dynamic  
 104 signal to enhance the training incentives; Specifically, VERL uses ERA to create a synergistic, dual-  
 105 channel incentive structure. Instead of switching between modes, it prospectively shapes the reward  
 106 to simultaneously encourage exploration (via ER) to preempt overconfidence, while also reinforcing  
 107 exploitative gains (via ERV) to consolidate the reasoning path. This unique stability makes ERA  
 108 a robust signal to guide training, allowing VERL to simultaneously encourage exploration from  
 109 productive-potential states while preventing overfitting to local optima. As a result, VERL delivers  
 110 significant performance gains across diverse models and tasks, achieving up to a **21.4%** absolute  
 111 accuracy improvement on the challenging Gaokao 2024 benchmark.

108 **Contributions.** (i) We are the first to probe the exploration-exploitation relationship in the semantically  
 109 rich hidden-state space. By adopting “ER” to quantify exploration and proposing novel metrics  
 110 “ERV” and “ERA” for exploitation, we empirically demonstrate that these two capacities are decoupled,  
 111 moving beyond the conventional token-level trade-off. (ii) We present VERL, a method that  
 112 leverages ERA to manage exploration and exploitation in a unified framework, enabling the simultaneous  
 113 enhancement of both capabilities. (iii) Our extensive experiments demonstrate the efficiency,  
 114 generality, and versatility of VERL across different RL architectures.

## 115 2 PRELIMINARIES

### 117 2.1 PROBLEM FORMULATION AND NOTATIONS

119 We adopt a reinforcement learning perspective on training LLMs for reasoning tasks. The LLM is  
 120 modeled as a policy  $\pi_\theta(\cdot|x)$ , parameterized by  $\theta$ , which generates a reasoning trajectory for a given  
 121 prompt  $x$  sampled from a distribution  $\mathcal{P}_x$ . The model’s output is a sequence of reasoning steps  
 122  $y_{0:T} = (y_0, y_1, \dots, y_T)$ , constrained to a maximum length  $L_{\max}$  (*i.e.*,  $T < L_{\max}$ ). The quality of  
 123 this trajectory is evaluated by a scalar reward function  $r(x, y)$ . The objective is to find the optimal  
 124 policy  $\pi_\phi$  that maximizes the expected reward:

$$125 \phi = \operatorname{argmax}_\theta \mathbb{E}_{x \sim \mathcal{P}_x} \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} [r(x, y)] \text{ s.t. } |y| \leq L_{\max}. \quad (1)$$

127 Conventionally, optimizing this objective at the token level is framed as a fundamental exploration-  
 128 exploitation trade-off. The policy must explore diverse and potentially novel reasoning pathways  
 129 to discover high-reward solutions. Concurrently, it must exploit known strategies by reinforcing  
 130 correct and reliable reasoning patterns that consistently yield high rewards.

### 131 2.2 REINFORCEMENT LEARNING BASELINE FRAMEWORKS

133 **Proximal Policy Optimization (PPO)** from Schulman et al. (2017) is a standard RL algorithm that  
 134 seeks to maximize a clipped surrogate objective function. This objective prevents excessively large  
 135 changes that would destabilize training, which is defined as:

$$136 \mathcal{L}_{\text{PPO}}(\theta) := \mathbb{E}_{x \sim \mathcal{P}_x, y \sim \pi_{\theta_{\text{old}}}(y|x)} \left\{ \sum_{t=1}^{|y|} \min [\rho_t(\theta) A_t, \text{clip}(\rho_t(\theta), 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}}) A_t] \right\}, \quad (2)$$

140 where  $\rho_t(\theta) := \frac{\pi_\theta(y_t|x, y_{<t})}{\pi_{\theta_{\text{old}}}(y_t|x, y_{<t})}$  is the probability ratio between the current and old policies, and  $A_t$   
 141 is the estimated advantage, often calculated using Generalized Advantage Estimation (GAE) from  
 142 Schulman et al. (2015), with clipping (hyperparameter  $\epsilon$ ) to mitigate excessive deviation.

143 **Group Relative Policy Optimization (GRPO)** from Shao et al. (2024) computes a baseline directly  
 144 from the rewards of multiple trajectories. For a given prompt, it samples a group of  $G$  responses,  
 145 obtains their corresponding rewards  $\{r_1, \dots, r_G\}$ , and normalizes these rewards to compute the ad-  
 146 vantage for each response:

$$147 A_{i,t} := \frac{r_i - \text{mean}(\{r_j\}_{j=1}^G)}{\text{std}(\{r_j\}_{j=1}^G)}. \quad (3)$$

149 GRPO would assign a single rule-based reward to the entire output sequence, and the resulting  
 150 group-relative advantage is uniformly propagated to all tokens, then updated as in Eq. 2.

### 152 2.3 CHARACTERIZING HIDDEN STATE REPRESENTATIONS

153 **Response Hidden States.** LLM would generate responses token by token in an autoregressive  
 154 manner. The token  $y_t$  output at step  $t$  represents the current explicit state, while the corresponding  
 155 output in the intermediate layers is referred to as the hidden state  $z_t$ . As the sequence of explicit  
 156 states forms the final response, simultaneously, the hidden states  $\{z_t\}_{t=1}^T$ , ordered by their output  
 157 sequence, collectively form the hidden states matrix  $\mathbf{Z} \in \mathbb{R}^{T \times D}$ , where  $T$  is the output length and  
 158  $D$  is the feature dimension, representing the semantic trajectory. To align with the semantic space,  
 159 we focus on the hidden state of the final layer in this paper.

160 **Dataset Hidden States.** Following the definition in Skean et al. (2025), for a dataset containing  $N$   
 161 prompts, after obtaining a single vector representation for the  $i$ -th response by averaging its token  
 hidden states  $\bar{z}_i = \frac{1}{T_i} \sum_{t=1}^{T_i} z_{i,t}$ , we stack these  $N$  mean embeddings in the dataset hidden states

162 matrix  $\bar{\mathbf{Z}} \in \mathbb{R}^{N \times D}$ , to represent the overall semantic distribution of the entire dataset.  
 163

### 164 3 A HIDDEN-STATE PERSPECTIVE ON REPRESENTATIONAL DYNAMICS

#### 165 3.1 STATIC METRIC: EFFECTIVE RANK (ER)

166 According to Roy & Vetterli (2007), the Effective Rank (ER), which is denoted by  $\text{erank}(\mathbf{Z})$  for a  
 167 response, is computed based on the normalized singular values of its non-padding hidden states  $\mathbf{Z}$ .  
 168 Let  $\sigma_j$  be the  $j$ -th singular value of  $\mathbf{Z}$ , and  $p_j = \frac{\sigma_j}{\sum_k \sigma_k}$  be the normalized singular values. The ER is  
 169 then given by:  
 170

$$171 \text{ER} := \text{erank}(\mathbf{Z}) = \exp \left[ - \sum_j p_j \log(p_j) \right]. \quad (4)$$

172 To quantify a model’s reasoning breadth, we treat exploration as a measurable semantic property.  
 173 Our primary metric for this is ER, which measures the **effective dimensionality** of the hidden-state  
 174 space a model occupies during a response. A high ER signals that the model is leveraging a rich and  
 175 diverse set of internal features, which is a direct signature of exploratory behavior. A low ER, in  
 176 contrast, points to a collapsed, simpler representation, indicating the model is not exploring widely.  
 177 By capturing this dispersion of embeddings, ER provides a more nuanced view of exploration than  
 178 conventional rank, which merely counts dimensions without considering their diversity.  
 179

180 **Theorem 3.1.** *Suppose we have a matrix of embeddings  $\mathbf{Z} \in \mathbb{R}^{T \times D}$ . Then the ER of  $\mathbf{Z}$  is a lower  
 181 bound of conventional rank of  $\mathbf{Z}$ :*

$$182 1 \leq \text{erank}(\mathbf{Z}) \leq \text{rank}(\mathbf{Z}) \leq \min\{T, D\}. \quad (5)$$

183 **Remark 3.2.** Conventional rank offers a discrete count of available dimensions but fails to capture  
 184 the geometric complexity essential for true exploration. In contrast, ER provides a nuanced, continuous  
 185 measure of “effective dimensionality”. In reasoning, this distinction is critical: conventional  
 186 rank may count many potential paths, but ER reveals how uniformly the model is actually exploring  
 187 them. A high ER reflects a more uniform distribution, signaling a broader and more effective  
 188 exploration of the solution space.  
 189

#### 190 3.2 DYNAMIC METRICS: EFFECTIVE RANK VELOCITY (ERV) AND ACCELERATION (ERA)

191 In this section, we develop temporal higher-order metrics, termed ERV and ERA, to characterize  
 192 the dynamics of a policy’s information gain together. Corresponding to the first and second-order  
 193 temporal differences of a representational metric, these tools measure how the quality of hidden  
 194 states evolves, revealing whether the reasoning process is expanding, stabilizing, or saturating.  
 195

196 **Definition 3.3. (First-Order Temporal Difference: ERV)** *To quantify the rate of change for a  
 197 given metric  $M$ , such as the ER or the conventional rank of the hidden states matrix, we define the  
 198 first-order temporal difference, denoted  $\Delta_M^{(1)}$ . This metric captures the “velocity” of information  
 199 gain by measuring how the metric’s value at a given step deviates from its historical average. Let  
 200  $m_t$  be the value of metric  $M$  computed on the token sub-sequence from the start to position  $t$ .  
 201 For a sequence of length  $T$  and a difference stride  $s$ , let the set of evaluation time steps be  $\mathcal{T} =$   
 202  $\{s, 2s, \dots, Ks\}$ , where  $K = \lfloor (T-1)/s \rfloor$ . The overall first-order difference is defined as:*

$$203 \Delta_M^{(1)} := \frac{1}{K-1} \sum_{j=2}^K \delta_{j \cdot s}, \quad \text{where} \quad \delta_{j \cdot s} := m_{j \cdot s} - \frac{1}{j-1} \sum_{k=1}^{j-1} m_{k \cdot s}. \quad (6)$$

204 Equivalently, writing the consecutive-step increments as  $\Delta m_{r \cdot s} := m_{r \cdot s} - m_{(r-1) \cdot s}$ , a simple algebraic  
 205 rearrangement yields

$$206 \delta_{j \cdot s} = \frac{1}{j-1} \sum_{r=2}^j (r-1) \Delta m_{r \cdot s}, \quad j \geq 2, \quad (7)$$

207 showing that each  $\delta_{j \cdot s}$  is in fact a time-weighted average of local consecutive differences, with larger  
 208 weights assigned to more recent steps. Our primary metric for exploitation is ERV, which is designed  
 209 to capture the rate of information gain. It is the average of instantaneous differences ( $\delta_{j \cdot s}$ ), where

216 each difference contrasts the complexity of the current token chunk with the cumulative average of  
 217 all preceding ones. This formulation directly operationalizes our definition of exploitation: a large  
 218 ERV demonstrates that the model is successfully enriching its representation at a rate that outpaces  
 219 its historical trend, signifying a deepening and productive line of inquiry. Conversely, a small ERV  
 220 signals that exploitation of the current path is becoming less effective.

221 **Definition 3.4. (Second-Order Temporal Difference: ERA)** *To measure the rate of change of the  
 222 velocity, we define the second-order temporal difference  $\Delta_M^{(2)}$ , which represents the “acceleration”  
 223 of the metric’s evolution. It reveals whether the process of representation formation is speeding up  
 224 or stabilizing. It is computed as the average change between consecutive instantaneous differences:*  
 225

$$\Delta_M^{(2)} := \frac{1}{K-2} \sum_{j=3}^K [\delta_{j \cdot s} - \delta_{(j-1) \cdot s}]. \quad (8)$$

226 A positive  $\Delta_M^{(2)}$  signifies an accelerating growth rate, indicating that the diversification of the rep-  
 227 resentation is speeding up. A negative value suggests this growth is decelerating, implying that the  
 228 representation’s quality is approaching stability or saturation.

### 229 3.3 SCALING PROPERTIES OF REPRESENTATIONAL DYNAMICS

230 In the preceding sections (Sec. 3.1 and 3.2), we introduced metrics for analyzing the hidden states of  
 231 individual responses. We now analyze the scaling properties of these dynamics at two distinct levels  
 232 of granularity: across an entire dataset as a function of its size ( $N$ ), and within a single reasoning  
 233 trajectory as a function of its length ( $T$ ). The following proposition provides a unified theoretical  
 234 model for both scenarios.

235 **Proposition 3.5.** *Assume a hidden-state matrix is composed of  $k$  approximately orthogonal row  
 236 vectors. The Effective Rank (ER) and its first-order difference (ERV) scale linearly with  $k$ , such that  
 237  $ER = \mathcal{O}(k)$  and  $\Delta_{ER}^{(1)} = \mathcal{O}(k)$ . The second-order difference (ERA) is independent of  $k$ , with a  
 238 scaling order of  $\Delta_{ER}^{(2)} = \mathcal{O}(1)$ .*

239 **Remark 3.6.** This proposition offers a dual interpretation of how our metrics scale under ideal con-  
 240 ditions: **At the dataset level**,  $k$  represents the number of questions  $N$ . The proposition implies  
 241 that as a dataset grows with semantically distinct responses (approaching orthogonality), its over-  
 242 all representational diversity (ER) should increase proportionally. The constant acceleration (ERA)  
 243 suggests a stable, predictable growth pattern for the dataset’s semantic volume. **At the response  
 244 level**,  $k$  represents the sequence length  $T$ . The proposition suggests that for an ideal reasoning  
 245 process where each step contributes novel information (making token embeddings approach orthog-  
 246 onality), the trajectory’s semantic complexity (ER) and information-gain velocity (ERV) should also  
 247 grow linearly with its length. In this context, a constant ERA becomes a signature of a robust and  
 248 non-saturating reasoning process.

## 249 4 DECOUPLING EXPLORATION AND EXPLOITATION IN REASONING

250 In this section, we first investigate the changing trends of the hidden states matrix rank (both ER  
 251 and conventional rank) during regular RL training. Specifically, we utilized the Qwen (Hui et al.,  
 252 2024) and Llama (Dubey et al., 2024) models for our experiments, employing GRPO (Shao et al.,  
 253 2024) reinforcement learning paradigm. The training dataset followed the configuration in Zeng  
 254 et al. (2025), which comprises 8k hard-level 3 to 5 mathematical problems from MATH datasets,  
 255 each accompanied by a verifiable reference answer.

### 256 4.1 ANALYSIS OF RESPONSE-LEVEL METRICS

257 During each training step, we quantitatively analyzed the representational dynamics of hidden states  
 258 within that batch as depicted in Fig. 2, and provided more and diverse details in App. H.1.

259 **Semantic space of hidden states move beyond the exploration-exploitation trade-off towards  
 260 stable enhancements.** While RL consistently improves performance, it interacts differently with  
 261 distinct base models, evidenced in the divergent trends of the ER (first column in Fig. 2), which  
 262 measures the total information within a response. For instance, the Qwen model exhibits an in-  
 263 creasing ER, suggesting more exploratory reasoning, whereas the one of Llama model decreases,

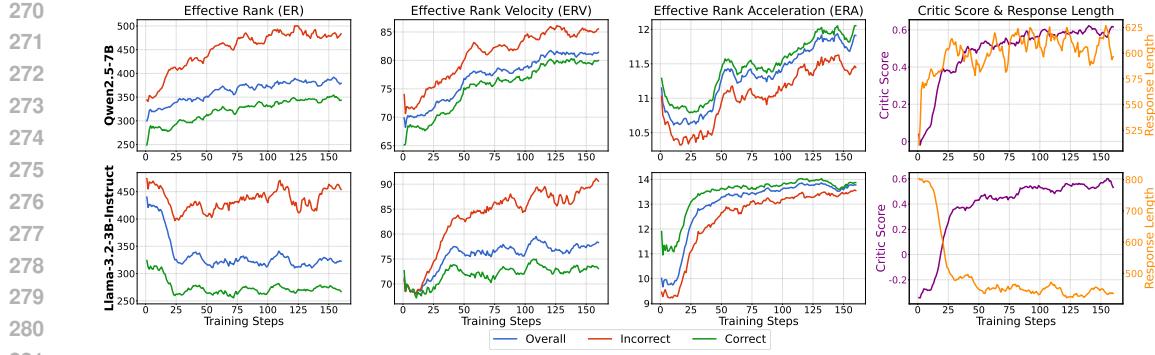


Figure 2: Response-level metrics during GRPO post-training, smoothed with a 10-step rolling window. Metrics are shown for the **Overall** batch, as well as for subsets of **Correct** and **Incorrect** samples. The rightmost column displays the average **Critic Score** (reward) and **Response Length** per batch.

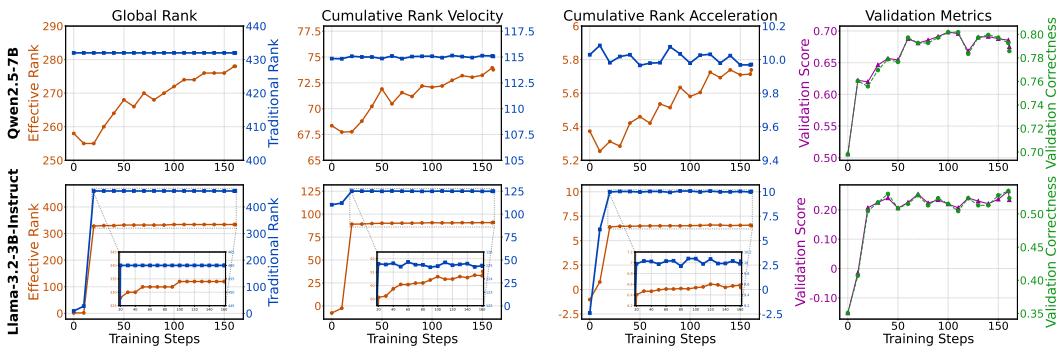


Figure 3: Visualization of dataset-level metrics during GRPO post-training. The figure compares **Traditional** metrics with our **proposed** metrics. Also shown are the **Validation Score** and sample **Correctness**, both averaged over the validation dataset.

indicating more concise, exploitative reasoning. Despite these differences in exploratory behavior, the ERV (second column) demonstrates a consistent upward scaling trend for all models, which suggests while the models’ intrinsic exploratory tendencies differ, RL fine-tuning universally enhances their exploitation capabilities by consistently accelerating the rate of information gain.

**ERA distinguishes correct reasoning.** For both the zero-order metric (ER) and the first-order metric (ERV), incorrect reasoning paths consistently score higher than correct ones. This suggests that excessive exploration (high ER) with new excessive information (high ERV) will potentially derail the reasoning process and lead to incorrect answers. Conversely, for the second-order metric (ERA), correct reasoning trajectories consistently exhibit higher values, which implies the acceleration of information gain—the ability to increasingly speed up the representational evolution—is the key to guide the policy towards a correct solution, distinguishing robust reasoning from flawed exploration.

#### 4.2 ANALYSIS OF DATASET-LEVEL METRICS

Following the framework established in Sec. 2, we extend our analysis from the response level to the entire validation dataset. By computing the dataset hidden states matrix  $\bar{Z}$ , we examine its zero-, first-, and second-order rank dynamics to understand how the policy’s overall representational space evolves. The trends are visualized in Fig. 3. While key experiments are shown here, we refer the reader to App. H.2 for a more diverse range of studies.

**Policy optimization correlates with expanding dataset-level diversity.** Across the training process, we observe a strong positive correlation between performance metrics (accuracy and critic score on the validation set) and the dynamics of dataset-level ER. As the model improves, the zero-order erank( $\bar{Z}$ ) and its first- and second-order differences consistently scale up. This indicates that as the policy is updated, it develops a more diverse and complex repertoire of reasoning strategies for the same set of problems. The increasing ERV and ERA suggest the model becomes progressively more efficient at navigating and expanding this richer semantic space to discover correct solutions.

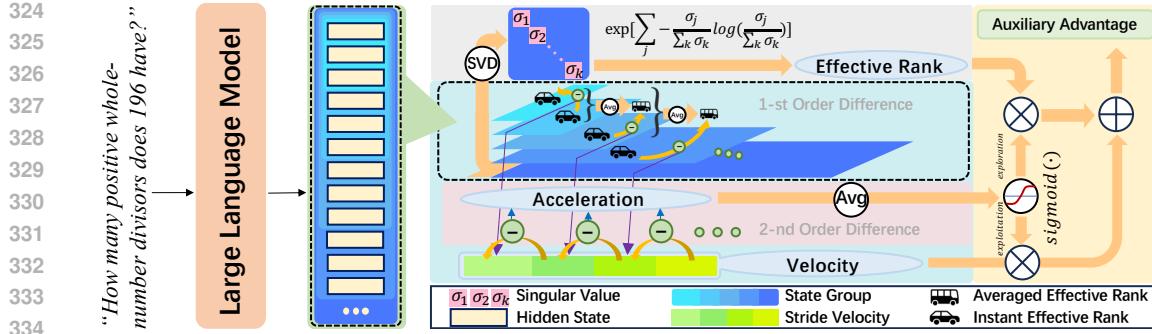


Figure 4: Overview of VERL. Exploration is quantified by computing the ER of the rolling-done hidden states via SVD, while **exploitation** is captured through EMA-smoothed first-order difference (**ERV**) on per-step rolling hidden state and extended to second-order difference (**ERA**). Finally, exploration and exploitation are adaptively integrated to derive the **auxiliary advantage**.

**ER reveals refinement beyond the limits of conventional rank.** During late-stage training, a plateauing conventional rank suggests the model has settled on a fixed number of linearly independent reasoning “directions”. Yet, a simultaneously rising ER points to a more subtle optimization. This trend reveals that the model is improving the quality of its existing solution space by making the “magnitudes” of these directions more uniform. In essence, instead of finding new pathways, the model learns to utilize its established ones more equitably, reducing representational redundancy and fostering a more sophisticated and distributed reasoning capability.

## 5 VELOCITY-EXPLOITING RANK-LEARNING (VERL)

Building upon the insights from Sec. 3 and the empirical observations in Sec. 4, we find that conventional RL objectives overlook the intrinsic hidden-state dynamics that more faithfully characterize exploration and exploitation. This oversight may lead to inefficient training, where policies either wander in unproductive exploration or collapse prematurely to suboptimal reasoning paths. To address this, we propose a novel method named Velocity-Exploiting Rank-Learning (VERL), which refines advantage by incorporating the nuanced dynamics of hidden states, enabling simultaneous enhancement of exploration and exploitation capacities.

### 5.1 STABLE REPRESENTATION DEVIATION INDICATOR

Concretely, we first formalize the representational metrics, letting  $\mathcal{M} = \{M_0, M_1, M_2\}$  denote the set of metrics derived from the hidden states, where  $M_0$  is the 0-order ER,  $M_1 := \Delta_M^{(1)}$  is its first-order temporal difference (ERV), and  $M_2 := \Delta_M^{(2)}$  is its second-order temporal difference (ERA). To create a stable guidance signal, having already computed scalar values  $\{m_0, m_1, m_2\}$  for each trajectory, we normalize these values against their historical trends by maintaining an Exponential Moving Average (EMA)  $\bar{\mu}_k$  for each metric  $M_k$ . The relative deviation for each metric is:

$$d_k := \frac{m_k - \bar{\mu}_k}{|\bar{\mu}_k| + \epsilon}, \quad k \in \{0, 1, 2\}, \quad (9)$$

where  $\epsilon$  is a small constant for numerical stability. This deviation  $d_k$  quantifies how the current trajectory’s representational structure diverges from the policy’s recent average behavior.

### 5.2 BEYOND TRADE-OFF FOR EFFICIENT TRAINING VIA ERA

Our analysis in Sec. 3.1 and 3.2 reveals that ER ( $M_0$ ) and ERV ( $M_1$ ) are effective proxies for exploration and exploitation, respectively. Crucially, these two metrics are also almost independent of each other, as shown in Fig. 1c bottom. This decoupling is key, as it allows us to combine them into a single objective to enhance both capabilities simultaneously.  $M_1$  measures the confidence of the current step as analysis above, so the subtraction of  $M_1$ , namely ERA ( $M_2$ ), would predict the evolution of confidence in subsequent steps. Meanwhile, theoretical analysis (Prop. 3.5) and empirical observations (Fig. 3) indicate that  $M_2$  remains approximately constant across trajectories. Thus,  $M_2$  can serve as a meta-level signal to guide training.

378 An increasing ERV indicates that the model is acquiring progressively more informative evidence,  
 379 reflecting its growing confidence. However, our preliminary experiments reveal that trajectories  
 380 exhibiting excessively high confidence often achieve high in-domain performance (as evidenced  
 381 by the results on the MATH dataset in Tab. 3, where performance suffers without ERA’s dynamic  
 382 unification or when using a simple 50/50 sum “ $\beta = 0.5$ ”) while compromising out-of-domain generalization  
 383 (across most datasets), suggesting severe overfitting. This implies that overconfident  
 384 trajectories reduce the opportunity to learn from less confident yet potentially informative samples.  
 385 To mitigate this, we employ  $M_2$  as the predictive signal to combine exploration and exploitation,  
 386 strategically encouraging exploring lower-confidence samples when trajectories exhibit excessively  
 387 high future confidence, thereby enhancing training efficiency and robustness.

388 Specifically, we define two orthogonal unit vectors of the weights first, an exploration-focused vector  
 389  $\mathbf{w}_{\text{explore}} = [1, 0]$ , which targets  $M_0$ , and an exploitation-focused one  $\mathbf{w}_{\text{exploit}} = [0, 1]$  of  $M_1$ . The  
 390 dynamic weight  $\mathbf{w}_{\text{dyn}}$  is interpolated by the relative deviation of the second-order metric  $d_2$ :

$$391 \quad \mathbf{w}_{\text{dyn}} := \beta \cdot \mathbf{w}_{\text{explore}} + (1 - \beta) \cdot \mathbf{w}_{\text{exploit}}, \quad \text{where } \beta := \text{sigmoid}(d_2). \quad (10)$$

393 The interpolation coefficient  $\beta$  is adaptively determined by the second-order metric  $d_2$  through a  
 394 sigmoid mapping. A high  $M_2$  ( $d_2 \gg 0$ ) means overconfidence in the future, risking overfitting  
 395 to in-domain patterns; thus, VERL increases  $\beta$  to favor the exploration profile  $M_0$ . In contrast,  
 396 when  $M_2$  is low ( $d_2 \leq 0$ ), namely limited confidence and reasoning saturation, VERL decreases  
 397  $\beta$  to emphasize the exploitation profile  $M_1$ . As  $M_2$  typically fluctuates around zero, VERL jointly  
 398 enhances exploration and exploitation. The final auxiliary advantage  $\Phi$  is defined as:

$$399 \quad \Phi := w_{\text{dyn},0} \cdot \tanh(d_0) + w_{\text{dyn},1} \cdot \tanh(d_1), \quad (11)$$

401 where  $w_{\text{dyn},0}$  and  $w_{\text{dyn},1}$  are the first and second entries of the dynamic weight vector  $\mathbf{w}_{\text{dyn}}$ , respec-  
 402 tively. The  $\tanh$  function bounds the magnitude of  $\Phi$  while preserving its sign, thereby stabilizing  
 403 training. This formulation rewards trajectories that exceed the historical average and penalizes those  
 404 that fall short, guiding the policy with adaptive reasoning dynamics while mitigating risks of stag-  
 405 nation and overconfidence.

### 406 5.3 ADVANTAGE SHAPING VIA REPRESENTATIONAL DYNAMICS

408 We refine the policy learning signal by shaping the advantage through a representational auxiliary  
 409 term. Let  $A^{(0)}$  denote the original advantage from GRPO or PPO with GAE, and let  $\Phi_i$  be the  
 410 sequence-level auxiliary signal defined in Sec. 5.2. The shaped objective replaces the original ad-  
 411 vantage with a refined estimate  $\hat{A}_t$ , defined directly within the surrogate loss:

$$412 \quad \mathcal{L}^{\text{shaped}}(\theta) = \mathbb{E}_{x \sim \mathcal{P}_x, y \sim \pi_{\theta_{\text{old}}}} \left[ \sum_{t=1}^{|y|} \min \left( \rho_t(\theta) \hat{A}_t, \text{clip}(\rho_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right], \quad (12)$$

$$416 \quad \text{where } \hat{A}_t = A_t^{(0)} + \min \left( \max(0, \Phi_i), \frac{|A_t^{(0)}|}{\kappa} \right), \quad \rho_t(\theta) = \frac{\pi_{\theta}(y_t \mid x, y_{<t})}{\pi_{\theta_{\text{old}}}(y_t \mid x, y_{<t})}.$$

419 This formulation adds a strictly positive, clipped bonus to trajectories exhibiting desirable repres-  
 420 entational dynamics, while preserving the sign and stability of the original advantage. The mechanism  
 421 applies consistently across advantage structures: for GRPO the shaping is applied once per trajec-  
 422 tory, and for PPO+GAE it applies to every token within the trajectory.

## 423 6 EXPERIMENTS

### 425 6.1 EXPERIMENT SETTINGS

427 *(i) Dataset.* The same datasets as in Sec. 4 are used. *(ii) Reward.* Our rule-based reward function  
 428 assesses mathematical correctness and \boxed{} formatting. Correct answers receive a +1.0 re-  
 429 ward if formatted, and +0.5 if not. Incorrect answers are penalized with -0.5 if formatted and -1.0  
 430 otherwise. *(iii) Training.* Based on verl (Sheng et al., 2025) and vLLM (Kwon et al., 2023) frame-  
 431 work, we set batch size as 48, generating 4 rollouts per prompt for GRPO and 1 rollout for PPO, and  
 set the maximum length  $L_{\text{max}}$  to 1536. More details of the experiments are provided in App. G.

432 Table 1: Performance comparison of models on mathematical reasoning benchmarks (Pass@1).  
 433 “+ GRPO” and “+ PPO” denote RL fine-tuning by GRPO and PPO framework respectively. “w/  
 434 VERL.” indicates incorporating our VERL with original RL type.  $\Delta$  represents the performance  
 435 contrast between original RL method and its VERL variant. See App. H.3 for full details.

Model	AIME24	AIME25	AMC23	AMC24	ASDiv	Carp.En	CMATH	Gaokao 2024.I	Gaokao 2024.Mix	Gaokao MathClose	GSMSK	MAWPS	Olympiad Bench	SVAMP	TabMWP	Avg.
<b>Llama-3.2-3B-Instruct</b>	0.0	0.0	25.0	11.1	74.6	26.5	10.2	14.3	6.8	66.6	86.9	12.7	74.1	41.4	31.0	
+ GRPO	3.3	0.0	27.5	8.9	88.8	45.0	21.4	20.9	23.7	80.7	96.0	16.7	87.7	71.7	41.4	
+ GRPO w/ VERL.	13.3	6.7	25.0	11.1	89.3	45.4	46.2	14.3	22.0	22.9	81.7	96.0	17.6	87.8	72.3	43.4
$\Delta_{\text{GRPO}}$	+10.0	+6.7	-2.5	+2.2	+0.5	+0.4	+17.9	-7.1	+1.1	+1.0	+0.0	+0.9	+0.1	+0.6	+2.0	
+ PPO	10.0	3.3	22.5	13.3	87.9	46.4	21.2	7.1	16.5	20.3	81.4	95.5	17.8	86.8	71.0	40.1
+ PPO w/ VERL.	10.0	3.3	25.0	11.1	88.7	46.0	30.7	14.3	19.8	27.1	82.9	95.7	17.3	85.8	71.3	41.9
$\Delta_{\text{PPO}}$	+0.0	+0.0	+2.5	-2.2	+0.8	-0.4	+9.5	+7.2	+3.3	+6.8	+1.5	+0.2	-0.5	-1.0	+0.3	+1.9
<b>Qwen2.5-7B</b>	6.7	0.0	45.0	15.6	91.4	55.8	86.7	42.9	33.0	49.2	85.8	95.4	25.8	88.5	82.8	53.6
+ GRPO	10.0	6.7	55.0	26.7	94.8	60.2	91.7	14.3	34.1	64.4	90.2	97.6	36.1	92.8	91.3	57.7
+ GRPO w/ VERL.	13.3	10.0	50.0	28.9	95.0	60.8	90.7	35.7	35.2	69.5	89.2	97.7	35.4	92.9	91.9	59.8
$\Delta_{\text{GRPO}}$	+3.3	+3.3	-5.0	+2.2	+0.2	+0.6	-1.0	+21.4	+1.1	+5.1	-1.0	+0.1	-0.7	+0.1	+0.6	+2.1
+ PPO	6.7	3.3	50.0	33.3	94.9	59.6	89.8	28.6	31.9	63.6	89.1	97.3	36.1	92.8	90.8	57.9
+ PPO w/ VERL.	10.0	6.7	52.5	33.3	94.8	60.0	90.3	28.6	34.1	66.9	90.2	97.8	36.1	92.5	90.6	59.0
$\Delta_{\text{PPO}}$	+3.3	+3.3	+2.5	+0.0	-0.1	+0.4	+0.5	+0.0	+2.2	+3.3	+1.1	+0.5	+0.0	-0.3	-0.2	+1.1

444 Table 2: Performance comparison of instruction-tuned models under diverse decoding settings  
 445 (Pass@ $k$ ). For full details, please refer to App. H.4.

Model	MATH500 (Pass@16)	AMC23 (Pass@128)	AMC24 (Pass@128)	AIME24 (Pass@256)	AIME25 (Pass@256)	Avg.
<b>Llama-3.2-3B-Instruct</b>	79.8	93.5	51.5	40.0	30.0	58.96
+ GRPO	80.2	95.4	60.6	40.0	30.0	61.24
+ GRPO w/ VERL.	80.6	95.7	59.0	50.0	36.7	64.40
$\Delta_{\text{GRPO}}$	+0.4	+0.3	-1.6	+10.0	+6.7	+3.16
+ PPO	82.2	94.5	57.0	46.7	36.7	63.42
+ PPO w/ VERL.	82.4	94.7	57.8	46.7	40.0	64.32
$\Delta_{\text{PPO}}$	+0.2	+0.2	+0.8	+0.0	+3.3	+0.90
<b>Qwen2.5-7B</b>	90.6	98.4	73.7	60.0	60.0	76.54
+ GRPO	90.8	97.8	78.3	56.7	50.0	74.72
+ GRPO w/ VERL.	91.4	98.3	79.0	63.3	60.0	78.40
$\Delta_{\text{GRPO}}$	+0.6	+0.5	+0.7	+6.6	+10.0	+3.68
+ PPO	91.2	98.6	74.3	53.3	56.7	74.82
+ PPO w/ VERL.	91.4	98.0	74.4	56.7	66.7	77.44
$\Delta_{\text{PPO}}$	+0.2	-0.6	+0.1	+3.4	+10.0	+2.62

## 456 6.2 MAIN RESULTS

457 **VERL Generalizes across Multiple Benchmarks of Varying Difficulty.** As summarized in Tab. 1  
 458 (Full details in App. H.3), VERL leads to consistent performance gains across benchmarks of dif-  
 459 ferent difficulty levels, ranging from elementary school problems (e.g., ASDiv) to Olympiad-level  
 460 challenges (e.g., OlympiadBench). The improvements are particularly pronounced on benchmarks  
 461 that demand multi-step reasoning rather than simple arithmetic. VERL achieves up to 21.4% and  
 462 10.0% absolute accuracy improvements on Gaokao 2024.I and AIME24 (in Tab. 1), respectively.

463 **VERL Generalizes across RL Algorithms and Base Models.** VERL is a plug-and-play method  
 464 that can be integrated with different RL algorithms to enhance their performance. As shown in  
 465 Tab. 1, applying VERL to GRPO and PPO improves the average results on 15 benchmarks for both  
 466 the Llama and Qwen series, demonstrating its strong generalization ability.

467 **Gains in Both Exploration and Exploitation.** As shown in Tab. 2 (Full details in App. H.4), VERL  
 468 yields larger improvements on Pass@ $k$  (a measure of exploration) than on Pass@1 (a measure of  
 469 exploitation), particularly on more challenging benchmarks. Since Pass@1 reflects exploitation and  
 470 Pass@ $k$  reflects exploration, the combined results of Tab. 1 and Tab. 2 demonstrate that VERL  
 471 effectively enhances both abilities. For a detailed case study, see Sec. I.

472 **Performance Degradation on Some Datasets** As shown in Tab. 1, the minor drop on CMATH  
 473 (-1.0) with Qwen occurs at a high-performance saturation level (91%), likely reflecting statistical  
 474 variance rather than capability degradation. Meanwhile, the drop on Gaokao (-7.1) with Llama is at-  
 475 tributable to the optimization dynamics specific to GRPO, as VERL achieves a substantial +7.2 gain  
 476 on the exact same benchmark under the PPO setting (Row 6). Crucially, these isolated fluctuations  
 477 are outweighed by the consistent improvements in Average accuracy across all models (e.g., +2.0%  
 478 for Llama) and the significant breakthroughs on challenging OOD reasoning tasks (e.g., AIME24  
 479 +10.0%). This confirms that VERL’s benefits in promoting robust reasoning significantly exceed the  
 480 cost of minor local variance.

## 481 6.3 ABLATION ANALYSES

482 We conduct ablation studies on the key hyperparameters or components of VERL: the effectiveness  
 483 of ERA, the stride length ( $s$ ) for temporal dynamics, the advantage clipping factor ( $\kappa$ ), and the  
 484 composition of the auxiliary shaping signal.

486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539

Table 3: Pass@1 performance with variant  $\beta$ . “Adapted  $\beta$ ” denotes  $\beta := \text{sigmoid}(d_2)$ . In this paper, all results in the table are reported in percentage (%), with **Bold** indicating the best performance.

Training Strategy	Score Avg	In Domain			Out of Domain			Hard Problems					
		MATH	MATH500	Avg	Gaokao	CN Middle School	CMATH	Avg	AIME24	AIME25	AMC23	AMC24	Avg
GRPO	0.36	<b>51.4</b>	46.2	48.80	<b>23.7</b>	28.7	28.3	26.90	3.3	0.0	<b>27.5</b>	8.9	9.93
GRPO+VERL ( $\beta = 0.5$ )	<b>0.38</b>	51.2	47.2	49.20	21.2	36.6	38.7	32.17	10.0	0.0	<b>27.5</b>	8.9	11.60
GRPO+VERL (Adapted $\beta$ )	<b>0.38</b>	50.9	<b>51.05</b>	22.9	<b>38.6</b>	<b>46.2</b>	<b>35.90</b>	13.3	<b>6.7</b>	20.0	11.1	<b>12.78</b>	

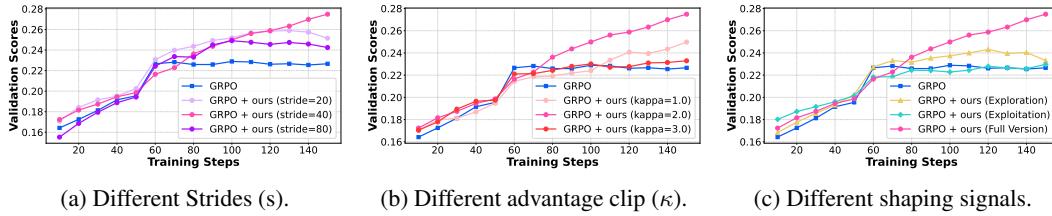


Figure 5: Comparison of various hyperparameters with Llama-3.2-3B-Instruct. It shows that the model performs best with a stride of 40 in (a) and with  $\kappa = 2$  in (b). We adopt these settings for all subsequent experiments. Moreover, (c) indicates that using only one signal, either exploration or exploitation, leads to suboptimal performance, demonstrating the effectiveness of our method.

**Analysis of the Effectiveness of ERA.** As shown in Tab. 3, the comparison between GRPO without/with  $\beta = 0.5$  shows that incorporating hiddenstate-level rewards provides consistent gains over the token-level baseline, demonstrating the advantage of leveraging richer internal representations during optimization. The gap between the fixed  $\beta$  and the Adapted  $\beta$  variant indicates that dynamic adjustment of ERA provides a more reliable estimate of when to emphasize exploration versus exploitation.

**Analysis of Stride ( $s$ ).** The stride  $s$  determines the granularity for calculating temporal difference metrics. As depicted in Fig. 5 (a), VERL’s performance improvement is robust across various stride values, indicating the underlying signal is not overly sensitive to sampling frequency. We find that  $s = 40$  yields optimal validation rewards, striking an effective balance between capturing significant temporal shifts and avoiding noise from minor token-level fluctuations.

**Analysis of Advantage Clip ( $\kappa$ ).** The advantage clipping factor  $\kappa$  stabilizes training by ensuring our auxiliary term acts as a refinement rather than a dominant signal. It constrains the shaping bonus to a fraction of the original advantage, preventing it from overpowering the primary task reward. The results in Fig. 5 (b) show that VERL consistently enhances performance for all tested values of  $\kappa$  underscoring its stability. Optimal performance is achieved at  $\kappa = 2$ , which provides a sufficiently strong and well-proportioned signal to guide the policy without destabilizing the learning process.

**Analysis of Shaping Signals ( $\Phi$ ).** As shown in Fig. 5 (c), compared to the full formulation in Eq. 11, using only the exploration-related term prevents the model from exploiting high-reward trajectories, leading to earlier bottlenecks and inferior final performance. In contrast, using only the exploitation-related term yields higher initial returns but quickly plateaus due to insufficient exploration. When combining both terms, the model achieves more stable training and superior final performance.

## 7 CONCLUSION

We challenge the conventional exploration-exploitation capacities trade-off in LLM reasoning blamed on token-level analysis and shift focus to their decoupled relation in hidden-state representations (measured as semantic diversity and information gain velocity, respectively). We introduce ER, ERV and ERA to quantify the dynamics of semantic complexity, with ERA as a stable indicator distinguishing correct from flawed reasoning. We further propose Velocity-Exploiting Rank-Learning (VERL) method, which uses ERA as a meta-controller to adaptively shape the advantage function, moving beyond the trade-off for simultaneous enhancement. Extensive experiments validate VERL’s superior out-of-domain generalization and performance on complex reasoning tasks.

540  
541  
**ETHICS STATEMENT**

542 The authors of this paper have read and agree to adhere to the ICLR Code of Ethics. Our research  
 543 is focused on the fundamental analysis of internal representations in Large Language Models and  
 544 the development of a novel reinforcement learning algorithm. This work does not involve human  
 545 subjects, and we did not collect any new datasets containing personally identifiable or sensitive in-  
 546 formation. The experiments were conducted using publicly available and widely-used mathematical  
 547 reasoning benchmarks (such as MATH and GSM8K), which are standard in the field and do not  
 548 raise immediate privacy or data bias concerns in the context of this study. While we acknowledge  
 549 that any advancement in LLM reasoning capabilities could be applied in various ways, our work is  
 550 foundational and aimed at improving the robustness and efficiency of AI systems. We do not foresee  
 551 any direct negative societal impacts stemming from this research.  
 552

553  
**REPRODUCIBILITY STATEMENT**

554 To ensure the reproducibility of our findings, we have provided comprehensive details throughout  
 555 the paper and its appendices. The core methodology of our proposed Velocity-Exploiting Rank-  
 556 Learning (VERL) is described in Sec. 5, with a concrete implementation outlined in Algorithm 1.  
 557 All experimental settings, including the base models used (Llama and Qwen series), datasets, reward  
 558 function design, and key hyperparameters for both GRPO and PPO training, are detailed in Sec. 6.1  
 559 and further expanded upon in App. G. The theoretical foundations for our proposed metrics (ER,  
 560 ERV, and ERA) are established in Sec. 3, with complete mathematical proofs for our claims provided  
 561 in App. H. As stated in the abstract, the source code to replicate our experiments will be made  
 562 publicly available upon publication of this work.  
 563

564  
**REFERENCES**

565 Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effec-  
 566 tiveness of entropy minimization in llm reasoning. *arXiv preprint arXiv:2505.15134*, 2025.

567 Shelly Bensal, Umar Jamil, Christopher Bryant, Melisa Russak, Kiran Kamble, Dmytro Mo-  
 568 zolevskyi, Muayad Ali, and Waseem AlShikh. Reflect, retry, reward: Self-improving llms via  
 569 reinforcement learning. *arXiv preprint arXiv:2505.24726*, 2025.

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

573 Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi.  
 574 Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models.  
 575 *arXiv preprint arXiv:2508.10751*, 2025b.

576 Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and  
 577 Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*,  
 578 2025a.

579 Zhengxiang Cheng, Dongping Chen, Mingyang Fu, and Tianyi Zhou. Optimizing length compres-  
 580 sion in large reasoning models. *arXiv preprint arXiv:2506.14755*, 2025b.

581 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,  
 582 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to  
 583 solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

587 Mehl Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob  
 588 Andreas. Beyond binary rewards: Training lms to reason about their uncertainty. *arXiv preprint*  
 589 *arXiv:2507.16806*, 2025.

594 Jia Deng, Jie Chen, Zhipeng Chen, Daixuan Cheng, Fei Bai, Beichen Zhang, Yinqian Min,  
 595 Yanzipeng Gao, Wayne Xin Zhao, and Ji-Rong Wen. From trial-and-error to improvement: A  
 596 systematic analysis of llm exploration mechanisms in rlvr. *arXiv preprint arXiv:2508.07534*,  
 597 2025a.

598 Wenlong Deng, Yi Ren, Yushu Li, Boying Gong, Danica J Sutherland, Xiaoxiao Li, and Christos  
 599 Thrampoulidis. Token hidden reward: Steering exploration-exploitation in group relative deep  
 600 reinforcement learning. *arXiv preprint arXiv:2510.03669*, 2025b.

602 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha  
 603 Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models.  
 604 *arXiv e-prints*, pp. arXiv–2407, 2024.

606 Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. *arXiv*  
 607 *preprint arXiv:2508.15260*, 2025.

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

612 Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han,  
 613 Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting  
 614 agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd*  
 615 *Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp.  
 616 3828–3850, 2024.

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

621 Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang,  
 622 Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*,  
 623 2024.

625 Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bam-  
 626 ford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al.  
 627 Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

629 Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and  
 630 Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement  
 631 learning. *arXiv preprint arXiv:2503.09516*, 2025.

632 Yi Jing, Zijun Yao, Hongzhu Guo, Lingxu Ran, Xiaozhi Wang, Lei Hou, and Juanzi Li. Lingualens:  
 633 Towards interpreting linguistic mechanisms of large language models via sparse auto-encoder. In  
 634 *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp.  
 635 28220–28239, 2025.

637 Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi.  
 638 Mawps: A math word problem repository. In *Proceedings of the 2016 conference of the north*  
 639 *american chapter of the association for computational linguistics: human language technologies*,  
 640 pp. 1152–1157, 2016.

641 Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization  
 642 inhibits data-efficient deep reinforcement learning. *ArXiv*, abs/2010.14498, 2020. URL <https://api.semanticscholar.org/CorpusID:225075792>.

645 Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph  
 646 Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model  
 647 serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

648 Timothée Lesort, Natalia Díaz Rodríguez, Jean-François Goudou, and David Filliat. State represen-  
 649 tation learning for control: An overview. *Neural networks : the official journal of the International*  
 650 *Neural Network Society*, 108:379–392, 2018. URL <https://api.semanticscholar.org/CorpusID:3638188>.  
 651

652 Junyan Li, Wenshuo Zhao, Yang Zhang, and Chuang Gan. Steering llm thinking with budget guid-  
 653 ance. *arXiv preprint arXiv:2506.13752*, 2025a.

654 Pengyi Li, Matvey Skripkin, Alexander Zubrey, Andrey Kuznetsov, and Ivan Oseledets. Confidence  
 655 is all you need: Few-shot rl fine-tuning of language models. *arXiv preprint arXiv:2506.06395*,  
 656 2025b.

657 Xiao Liang, Zhong-Zhi Li, Yeyun Gong, Yang Wang, Hengyuan Zhang, Yelong Shen, Ying Nian  
 658 Wu, and Weizhu Chen. Sws: Self-aware weakness-driven problem synthesis in reinforcement  
 659 learning for llm reasoning. *arXiv preprint arXiv:2506.08989*, 2025a.

660 Xiao Liang, Zhongzhi Li, Yeyun Gong, Yelong Shen, Ying Nian Wu, Zhijiang Guo, and Weizhu  
 661 Chen. Beyond pass@ 1: Self-play with variational problem synthesis sustains rlvr. *arXiv preprint*  
 662 *arXiv:2508.14029*, 2025b.

663 Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter  
 664 Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured  
 665 mathematical reasoning. In *ICLR*, 2023.

666 Clare Lyle, Mark Rowland, and Will Dabney. Understanding and preventing capacity  
 667 loss in reinforcement learning. *ArXiv*, abs/2204.09560, 2022. URL <https://api.semanticscholar.org/CorpusID:248266388>.  
 668

669 Jacob A Matthews, John R Starr, and Marten van Schijndel. Semantics or spelling? probing context-  
 670 ual word embeddings with orthographic noise. *arXiv preprint arXiv:2408.04162*, 2024.

671 Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing  
 672 english math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association*  
 673 *for Computational Linguistics*, pp. 975–984, 2020.

674 Arkil Patel, Satwik Bhattacharya, and Navin Goyal. Are nlp models really able to solve simple  
 675 math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of*  
 676 *the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094,  
 677 2021.

678 Ziqing Qiao, Yongheng Deng, Jiali Zeng, Dong Wang, Lai Wei, Fandong Meng, Jie Zhou, Ju Ren,  
 679 and Yaoxue Zhang. Concise: Confidence-guided compression in step-by-step efficient reasoning.  
 680 *arXiv preprint arXiv:2505.04881*, 2025.

681 Corrado Rainone, Tim Bakker, and Roland Memisevic. Replacing thinking with tool usage enables  
 682 reasoning in small language models. *arXiv preprint arXiv:2507.05065*, 2025.

683 Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007*  
 684 *15th European signal processing conference*, pp. 606–610. IEEE, 2007.

685 Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Firoj Alam, Abdul Khan, and Jia Xu. Analyzing en-  
 686 coded concepts in transformer language models. In *Proceedings of the 2022 Conference of the*  
 687 *North American chapter of the Association for Computational Linguistics: Human language tech-*  
 688 *nologies*, pp. 3082–3101, 2022.

689 John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-  
 690 dimensional continuous control using generalized advantage estimation. *arXiv preprint*  
 691 *arXiv:1506.02438*, 2015.

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

702 Giovanni Servedio, Alessandro De Bellis, Dario Di Palma, Vito Walter Anelli, and Tommaso  
 703 Di Noia. Are the hidden states hiding something? testing the limits of factuality-encoding ca-  
 704 pabilities in llms. *arXiv preprint arXiv:2505.16520*, 2025.

705 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li,  
 706 Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open  
 707 language models, 2024. URL <https://arxiv.org/abs/2402.03300>.

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

711 Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid  
 712 Schwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *arXiv  
 713 preprint arXiv:2502.02013*, 2025.

714 Lucrezia Valeriani, Diego Doimo, Francesca Cuturrello, Alessandro Laio, Alessio Ansuini, and Al-  
 715 berto Cazzaniga. The geometry of hidden representations of large transformer models. *Advances  
 716 in Neural Information Processing Systems*, 36:51234–51252, 2023.

717 Han Wang, Erfan Miah, Martha White, Marlos C. Machado, Zaheer Abbas, Raksha Kumaraswamy,  
 718 Vincent Liu, and Adam White. Investigating the properties of neural network repres-  
 719 entations in reinforcement learning. *ArXiv*, abs/2203.15955, 2022. URL <https://api.semanticscholar.org/CorpusID:247793771>.

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

723 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny  
 724 Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in  
 725 neural information processing systems*, 35:24824–24837, 2022.

726 Tianwen Wei, Jian Luan, Wei Liu, Shuang Dong, and Bin Wang. Cmath: Can your language model  
 727 pass chinese elementary school math test? *CoRR*, 2023.

728 Hang Yan, Fangzhi Xu, Rongman Xu, Yifei Li, Jian Zhang, Haoran Luo, Xiaobao Wu, Luu Anh  
 729 Tuan, Haiteng Zhao, Qika Lin, et al. Mur: Momentum uncertainty guided reasoning for large  
 730 language models. *arXiv preprint arXiv:2507.14958*, 2025.

731 An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li,  
 732 Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint  
 733 arXiv:2407.10671*, 2024.

734 Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik  
 735 Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Ad-  
 736 vances in neural information processing systems*, 36:11809–11822, 2023.

737 Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong  
 738 Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at  
 739 scale. *CoRR*, 2025.

740 Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does re-  
 741 inforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv  
 742 preprint arXiv:2504.13837*, 2025.

743 Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-  
 744 zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025.  
 745 URL <https://arxiv.org/abs/2503.18892>.

746 Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reason-  
 747 ing models know when they’re right: Probing hidden states for self-verification. *arXiv preprint  
 748 arXiv:2504.05419*, 2025.

756 Beichen Zhang, Kun Zhou, Xilin Wei, Xin Zhao, Jing Sha, Shijin Wang, and Ji-Rong Wen. Evalu-  
757 ating and improving tool-augmented computation-intensive math reasoning. *Advances in Neural*  
758 *Information Processing Systems*, 36:23570–23589, 2023.

759

760 Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen  
761 Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint*  
762 *arXiv:2504.16084*, 2025.

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810	<b>CONTENTS OF APPENDIX</b>	
811		
812		
813	<b>A Notations</b>	<b>17</b>
814		
815	<b>B The Use of Large Language Models (LLMs)</b>	<b>18</b>
816		
817	<b>C Algorithm</b>	<b>18</b>
818		
819	<b>D Related Work</b>	<b>19</b>
820	D.1 Reinforcement Learning with Verifiable Rewards . . . . .	19
821	D.2 Exploration and Exploitation in RLVR of LLM . . . . .	19
822	D.3 Representation Dynamics in Deep Reinforcement Learning . . . . .	19
823		
824	<b>E Details of Theorems</b>	<b>20</b>
825	E.1 Proof of Theorem 3.1 . . . . .	20
826	E.2 Proof of Proposition 3.5 . . . . .	20
827		
828	<b>F Additional Theoretical Support for Exploration and Exploitation Metrics</b>	<b>22</b>
829	F.1 Previous vs. our Exploration–Exploitation Metrics . . . . .	22
830	F.2 Effective Rank as Semantic Exploration . . . . .	25
831	F.3 Effective Rank Velocity as Semantic Exploitation . . . . .	26
832		
833		
834	<b>G Implementation Details</b>	<b>28</b>
835	G.1 Training and Evaluation Details . . . . .	28
836	G.2 Efficient Incremental Computation of Higher-Order Metrics . . . . .	28
837	G.3 Time Overhead of VERL Training . . . . .	29
838		
839		
840	<b>H More Experiments</b>	<b>29</b>
841	H.1 Analysis of Response-Level Hidden States . . . . .	29
842	H.2 Analysis of Dataset-Level Hidden States . . . . .	30
843	H.3 Detailed Analysis of Pass@1 Performance . . . . .	34
844	H.4 Detailed Analysis of Pass@ $k$ Performance . . . . .	35
845	H.5 Ablation on the Choice of Hidden Layer . . . . .	35
846		
847		
848	<b>I Case Study</b>	<b>37</b>
849	I.1 Case Study For Pass@1 Setting . . . . .	37
850	I.2 Case Study For Pass@16 Setting . . . . .	38
851		
852		
853		
854		
855		
856		
857		
858		
859		
860		
861		
862		
863		

## A NOTATIONS

Symbol	Description
$\pi$	Large language model policy
$A_{i,t}$	Group-relative advantage for the $t$ -th token in the $i$ -th response in group
$A^{(0)}$	Original advantage estimation
$\hat{A}$	Reshaped advantage value
$z_t$	Hidden state corresponding to the $t$ -th step of the output token
$\bar{z}_i$	Single vector representation for the $i$ -th response by averaging its token hidden states
$\mathbf{Z}_c$	Mean-centered hidden state matrix
$\bar{\mathbf{Z}}_{1:n}$	Dataset-level hidden states matrix formed by the first $n$ prompts
$\mathbf{Z}$	Response-level hidden states matrix
$\Delta_M^{(i)}$	The $i$ -order temporal difference for metric $M$
$\mathcal{M}$	Set of metrics derived from the hidden states
$M_i$	$Ti$ -order temporal difference of ER, exactly the different metrics
$m_t$	Value of metric $M$ computed on the token sub-sequence from the start to position $t$
$\epsilon_{\text{high/low}}$	Hyperparameter for the upper/lower bound used for clipping
$\epsilon$	Small constant for numerical stability
$r_j$	Reward of the $j$ -th response
$\mathbf{w}_{\text{explore/exploit/dyn}}$	Exploration-focused profile/Exploitation-focused profile/Dynamic-weighted profile
$w_{dyn,i}$	The $i$ -th scalar of $\mathbf{w}_{\text{dynamic}}$
$\text{rank}(\cdot)/\text{erank}(\cdot)$	Conventional rank/Effective rank function
$\text{SVD}(\cdot)$	a function to calculate the singular values
$\delta_n^{(i)}$	Instantaneous $i$ -Order Difference for step $n$
$s$	The stride for effective rank velocity calculation
$y_t$	The $t$ -th step (token) of the model's response
$y_{i:j}$	Sequence of reasoning steps from $i$ to $j$
$\bar{\mu}_k$	Exponential Moving Average for metric $M_k$
$\lambda_j(\cdot)$	The $j$ -th eigenvalues of the given matrix
$\beta$	Interpolation coefficient for VERL training
$G$	Size of sampled group in GRPO
$\theta$	Large language model policy's parameter
$\phi$	The parameter corresponding to the optimal policy
$\Phi$	Auxiliary advantage
$t$	Time step
$T$	Output length
$L_{\text{max}}$	The maximum length of model's output
$x$	Prompt
$S$	Sample times per prompt
$\mathcal{P}_x$	Distribution of prompts
$\rho_t$	Probability ratio between the current and old policies for $t$ -step of the output
$p_j$	The $j$ -th normalized singular values
$\mathcal{L}_{\text{PPO}}(\cdot)$	The optimization objective for PPO applied to policy
$D$	Feature dimension of hidden states
$N$	The size of the dataset
$\sigma_j$	The $j$ -th singular value of matrix
$\mathbf{P}$	Singular value distribution
$\mathcal{T}$	Set of time steps
$d_k$	Deviation for metric $M_k$
$\kappa$	Advantage clipping factor
$\Delta$	Performance difference between the baseline RL method and its VERL variant
$H(\cdot)$	Shannon entropy function
$\mathbf{q}_i$	The $i$ -th row of the dataset matrix
$\overline{\mathbf{K}}$	Gram matrix of dataset matrix
$U_t$	Uncentered Gram matrix

918 B THE USE OF LARGE LANGUAGE MODELS (LLMs)  
919920 We used large language models for text polishing.  
921922 C ALGORITHM  
923924 **Algorithm 1** VERL: Training  
925

---

926

927

928 1: **Input:**  $\mathcal{D} = \{x^i\}_{i=1}^N$ , prompt  $x^i$ , policy model  $\pi_\theta$ , hidden-state dimension  $D$ , sample times per  
929 prompt  $S$ .  
930 2: **Parameters:** EMA factor  $\gamma$ , relative deviation stabilizing factor  $\epsilon \ll 1$ , RL fine-tuning stabiliz-  
931 ing factor  $\kappa$ .  
932 3: **Initialize:** Randomly initialize policy parameters  $\pi_\theta$ , historical averages of metrics  $\bar{\mu}_{\text{ER}} =$   
933  $\bar{\mu}_{\text{ERV}} = \bar{\mu}_{\text{ERA}} = 0$ , exploration capacity profile  $\mathbf{w}_{\text{explore}} = [1, 0]$ , exploitation capacity profile  
934  $\mathbf{w}_{\text{exploit}} = [0, 1]$ .  
935 4: **Output:** A well-trained policy model  $\pi_\theta$ .  
936 5: **repeat**  
937 6:   **for**  $x^i \in \mathcal{D}$  **do:** // Pick a sample from dataset  
938 7:     **for** 1 to  $S$  **do:** // Rolling  $S$  times for one sample  
939 8:        $y_0^i \leftarrow x^i$ ,  $\mathbf{Z}_0^i \leftarrow \emptyset$ ,  $t \leftarrow 1$   
940 9:       **repeat** // Generation process  
941 10:        $y_t^i, \mathbf{z}_t^i \sim \pi_\theta(\cdot | y_{t-1}^i)$   
942 11:        $y_t^i \leftarrow [y_{t-1}^i; y_t^i]$  // Concatenate token sequence  
943 12:        $\mathbf{Z}_t^i \leftarrow [\mathbf{Z}_{t-1}^i; \mathbf{z}_t^i] \in \mathbb{R}^{t \times D}$   
944 13:        $\sigma_t^i \leftarrow \text{SVD}(\mathbf{Z}_t^i)$   
945 14:        $j \leftarrow |\sigma_t^i|$   
946 15:        $p_{j,t}^i \leftarrow \sigma_{j,t}^i / \sum_j \sigma_{j,t}^i$   
947 16:        $\text{erank}_t^i \leftarrow \exp\left(-\sum_j p_{j,t}^i \log p_{j,t}^i\right)$   
948 17:       **If**  $t > 1$  **then:**  $\delta_{\text{ERV},t} \leftarrow \text{erank}_t^i - \frac{1}{t-1} \sum_{k=1}^{t-1} \text{erank}_k^i$   
949 18:       **If**  $t > 2$  **then:**  $\delta_{\text{ERA},t} \leftarrow \delta_{\text{ERV},t} - \delta_{\text{ERV},t-1}$   
950 19:        $t \leftarrow t + 1$   
951 20:     **until** rolling done the sentence; //  $t - 1$  is the final timestep while rolling done  
952 21:      $A_{\text{origin}}^i \leftarrow \text{base RL evaluating on } y_{t-1}^i$  // Calculating ER metric  
953 22:      $m_{\text{ER}}^i \leftarrow \text{erank}_{t-1}^i$  // Calculating ERV metric  
954 23:      $m_{\text{ERV}}^i \leftarrow \frac{1}{t-2} \sum_{t=2}^{t-1} \delta_{\text{ERV}}^t$  // Calculating ERA metric  
955 24:      $m_{\text{ERA}}^i \leftarrow \frac{1}{t-3} \sum_{t=3}^{t-1} \delta_{\text{ERA}}^t$   
956 25:      $\bar{\mu}_k \leftarrow \gamma \bar{\mu}_k + (1 - \gamma) m_k^i, k \in \{\text{ER, ERV, ERA}\}$   
957 26:      $d_k^i \leftarrow \frac{m_k^i - \bar{\mu}_k}{|\bar{\mu}_k| + \epsilon}, k \in \{\text{ER, ERV, ERA}\}$   
958 27:      $\beta^i \leftarrow \text{sigmoid}(d_{\text{ERA}}^i)$   
959 28:      $\mathbf{w}_{\text{dyn}}^i \leftarrow \beta^i \mathbf{w}_{\text{explore}} + (1 - \beta^i) \mathbf{w}_{\text{exploit}}$   
960 29:      $w_{\text{dyn,ER}}^i \leftarrow \text{the first scalar value of } \mathbf{w}_{\text{dyn}}^i$   
961 30:      $w_{\text{dyn,ERV}}^i \leftarrow \text{the second scalar value of } \mathbf{w}_{\text{dyn}}^i$   
962 31:      $\Phi^i \leftarrow w_{\text{dyn,ER}}^i \tanh(d_{\text{ER}}^i) + w_{\text{dyn,ERV}}^i \tanh(d_{\text{ERV}}^i)$   
963 32:      $\hat{A}^i \leftarrow A_{\text{origin}}^i + \min\left(\max(0, \Phi^i), \frac{|A_{\text{origin}}^i|}{\kappa}\right)$   
964 33:     **end for**  
965 34:   **end for**  
966 35:   Update  $\theta$  via base RL objective with  $\hat{A}^i$   
967 36:   **until**  $\theta$  converges;

---

971

972 **D RELATED WORK**  
973974 **D.1 REINFORCEMENT LEARNING WITH VERIFIABLE REWARDS**  
975976 To unlock the full reasoning potential of Large Language Models (LLMs), Reinforcement Learning  
977 with Verifiable Rewards (RLVR) has become a prominent training paradigm. This approach was  
978 notably employed by DeepSeek-R1-Zero (Guo et al., 2025) and , which executes complex reasoning  
979 processes through actions such as reflection and validation. Following the success of DeepSeek-R1,  
980 a significant body of research has investigated the efficacy of RLVR on popular open-source LLMs,  
981 including Qwen (Yang et al., 2024), Mistral (Jiang et al., 2024), and LLaMA (Dubey et al., 2024).  
982983 This has fostered an optimistic view that RLVR can not only enhance existing model capabilities  
984 but also enable the acquisition of novel reasoning knowledge, facilitating a path toward continuous  
985 self-improvement (Zeng et al., 2025; Yu et al., 2025). RLVR training has been shown to grant LLMs  
986 controllable output length for efficient inference (Yan et al., 2025; Cheng et al., 2025b), deepen their  
987 reasoning pathways (Bensal et al., 2025), mitigate their weaknesses (Liang et al., 2025a;b), enable  
988 the use of external tools (Rainone et al., 2025; Jin et al., 2025), and even facilitate unsupervised  
989 reasoning (Zuo et al., 2025). However, Some studies (Yue et al., 2025) argue that while RLVR  
990 significantly improves the confidence and reliability of model reasoning, it may inadvertently con-  
991 strain the model’s exploratory capacity. The core of this issue lies in RLVR’s optimization objective:  
992 maximizing expected rewards. This objective function inherently biases the policy gradient toward  
993 reinforcing known trajectories that lead to high rewards (i.e., “exploitation”), while suppressing the  
994 exploration of unknown paths that may offer potentially higher returns but also carry greater risk  
995 (i.e., “exploration”). Consequently, the outputs of RLVR-optimized models often remain confined  
996 within the sampling distribution of the base model, suggesting the paradigm excels at refining ex-  
997 isting knowledge rather than generating new knowledge. This trade-off between exploration and  
998 exploitation constitutes a central challenge in the contemporary RLVR landscape.  
9991000 **D.2 EXPLORATION AND EXPLOITATION IN RLVR OF LLM**  
10011002 Recent perspectives (Wang et al., 2025; Cui et al., 2025) on the exploration-exploitation dilemma  
1003 have predominantly been shaped by analyses at the token level, focusing on the prediction dis-  
1004 tribution over the vocabulary. From this viewpoint, higher entropy in the token-level predic-  
1005 tion—indicating a more uniform distribution over the next token—is interpreted as a sign of greater  
1006 exploratory behavior, as it suggests a capacity for more diverse responses. This has led to the  
1007 adoption of techniques (Deng et al., 2025a; Cheng et al., 2025a) such as entropy regularization to  
1008 explicitly encourage the policy to explore novel reasoning paths. Conversely, lower entropy in the  
1009 token-level prediction is taken to signify higher model confidence in its reasoning chain, thus re-  
1010 presenting strong exploitation. Subsequent work (Fu et al., 2025) has also utilized metrics derived  
1011 from the top-k probabilities of the token prediction to quantify confidence. For instance, some ap-  
1012 proaches (Damani et al., 2025; Qiao et al., 2025) leverage the model’s internal “confidence” signals  
1013 to dynamically evaluate and filter the quality of reasoning steps, while others have employed high  
1014 confidence as a feedback signal to enable unsupervised reinforcement learning (Li et al., 2025b).  
1015 Ultimately, however, these confidence-based metrics are not fundamentally different from entropy.  
1016 This token-level standard of measurement introduces an endogenous contradiction: classifying be-  
1017 havior as either exploratory or exploitative requires the introduction of a prior assumptions, a prac-  
1018 tice that is disadvantageous for LLM research.  
10191020 In this paper, we depart from this paradigm. We shift the analysis from the token level to the se-  
1021 mantic space at the response level. This approach allows us to decouple the intertwined elements  
1022 of exploration and exploitation, aiming to achieve a simultaneous enhancement of both during rein-  
1023 forcement learning.  
10241025 **D.3 REPRESENTATION DYNAMICS IN DEEP REINFORCEMENT LEARNING**  
10261027 Beyond RLVR for language models, there is a line of classical deep RL work that explicitly studies  
1028 how neural representations evolve during training and how this affects exploration and sample effi-  
1029 ciency. State representation learning (SRL) for control aims to construct low-dimensional, action-  
1030 dependent embeddings that preserve task-relevant dynamics while discarding nuisance variation.  
1031 This work Lesort et al. (2018) provide a comprehensive overview of SRL methods for robotics and  
1032 control, emphasizing how compact latent states can improve both data efficiency and stability of  
1033 downstream RL algorithms. More recently, several works have directly analyzed the feature dynam-  
1034

ics of deep RL agents. The work Kumar et al. (2020) identify an “implicit under-parameterization” phenomenon in value-based deep RL: repeated bootstrapping updates lead to a collapse in the effective rank of value-network features, which in turn correlates with degraded performance in both online and offline settings. The work Lyle et al. (2022) further study capacity loss, showing that networks trained on non-stationary targets can lose their ability to fit new value functions over time, and proposing Initial Feature Regularization (InFeR) to stabilize the feature subspace and improve performance on sparse-reward Atari tasks. Complementary to these analyses, Wang et al. (2022) systematically measure multiple representational properties (e.g., dynamics-awareness, orthogonality) across thousands of agent–task combinations, and relate them to transfer performance, highlighting that not all good control policies arise from equally useful representation geometries.

## E DETAILS OF THEOREMS

### E.1 PROOF OF THEOREM 3.1

Suppose we have a matrix of embeddings  $\mathbf{Z} \in \mathbb{R}^{T \times D}$ . Then the effective rank of  $\mathbf{Z}$  is a lower bound of  $\text{rank}(\mathbf{Z})$ :

$$1 \leq \text{erank}(\mathbf{Z}) \leq \text{rank}(\mathbf{Z}) \leq \min\{T, D\} \quad (13)$$

*Proof.* Let the singular value distribution of the matrix  $\mathbf{Z}$  be  $\mathbf{p} = (p_1, p_2, \dots, p_{\min\{T, D\}})$ . The Shannon entropy of this distribution  $H(\mathbf{p})$  is bounded. Its minimum is 0, which occurs when only one element of  $\mathbf{p}$  is 1 and all others are 0. Its maximum is  $\log k$ , where  $k$  is the number of non-zero singular values, and this occurs when the distribution is uniform ( $p_j = 1/k$  for all non-zero values). The lower bound is established from the minimum entropy value:

$$\text{erank}(\mathbf{Z}) = \exp(H(\mathbf{p})) \geq \exp(0) = 1 \quad (14)$$

Equality holds if and only if the singular value distribution is  $(1, 0, \dots, 0)$ , meaning  $\mathbf{Z}$  has only one non-zero singular value. For the upper bound, let  $k = \text{rank}(\mathbf{Z})$  be the number of non-zero singular values of  $\mathbf{Z}$ . The entropy of the distribution  $\mathbf{p}$  is calculated only over these  $k$  values and is maximized when they are uniform. Therefore

$$H(\mathbf{p}) \leq \log k \quad (15)$$

Applying the exponential function to this inequality gives:

$$\text{erank}(\mathbf{Z}) = \exp(H(\mathbf{p})) \leq \exp(\log k) = k = \text{rank}(\mathbf{Z}) \quad (16)$$

This establishes that the effective rank is upper-bounded by the conventional rank. The final inequality,  $\text{rank}(\mathbf{Z}) \leq \min\{T, D\}$ , is a standard property of matrix rank. Equality for  $\text{erank}(\mathbf{Z}) = \text{rank}(\mathbf{Z})$  holds if and only if the non-zero singular values are all equal, corresponding to a uniform singular value distribution over its support.

### E.2 PROOF OF PROPOSITION 3.5

The zero-order metric and first-order difference of the effective rank scales linearly with the number of responses,  $\Delta_M^{(0)} = \mathcal{O}(N)$ ,  $\Delta_M^{(1)} = \mathcal{O}(N)$ . The second-order difference of the effective rank is constant and does not depend on  $N$ , yielding a scaling order of  $\Delta_M^{(2)} = \mathcal{O}(1)$ .

*Proof.* Without loss of generality, we take the effective rank for example. We adopt the provided definition of effective rank for a representation matrix  $\mathbf{Z}$  with singular values  $\{\sigma_i\}$ :

$$\text{erank}(\mathbf{Z}) = \exp\left(-\sum_j p_j \log(p_j)\right), \quad \text{where } p_j = \frac{\sigma_j}{\sum_k \sigma_k} \quad (17)$$

Our analysis focuses on the dataset matrix  $\bar{\mathbf{Z}} \in \mathbb{R}^{N \times D}$ , whose rows  $\{\mathbf{q}_i\}_{i=1}^N$  are the mean token embeddings of  $N$  responses. The singular values  $\sigma_i(\bar{\mathbf{Z}})$  of  $\bar{\mathbf{Z}}$  are the square roots of the eigenvalues of the Gram matrix  $\bar{\mathbf{K}} = \bar{\mathbf{Z}}\bar{\mathbf{Z}}^\top$ ; i.e.,  $\sigma_j(\bar{\mathbf{Z}}) = \sqrt{\lambda_j(\bar{\mathbf{K}})}$ . Given that the rows of  $\bar{\mathbf{Z}}$  are nearly

1080 orthogonal, the Gram matrix  $\bar{\mathbf{K}}$  is strongly diagonal-dominant. Its eigenvalues can be approximated  
 1081 by its diagonal entries:  
 1082

$$1083 \quad 1084 \quad \lambda_j(\bar{\mathbf{K}}) \approx \bar{\mathbf{K}}_{jj} = \|\mathbf{q}_j\|^2 = \frac{1}{T} \quad \text{for } j = 1, \dots, N \quad (18)$$

1085 The matrix has  $N$  significant eigenvalues, each approximately equal to  $1/T$ . The singular values of  
 1086  $\bar{\mathbf{Z}}$  are the square roots of the eigenvalues of  $\bar{\mathbf{K}}$ :  
 1087

$$1088 \quad 1089 \quad \sigma_j(\bar{\mathbf{Z}}) = \sqrt{\lambda_j(\bar{\mathbf{K}})} \approx \sqrt{\frac{1}{T}} = \frac{1}{\sqrt{T}} \quad \text{for } j = 1, \dots, N \quad (19)$$

1091 To calculate the effective rank, we first normalize these singular values to form a probability distribution  
 1092  $\{p_j\}$ . The sum of singular values is:  
 1093

$$1094 \quad 1095 \quad \sum_{k=1}^N \sigma_k(\bar{\mathbf{Z}}) \approx \sum_{k=1}^N \frac{1}{\sqrt{T}} = \frac{N}{\sqrt{T}} \quad (20)$$

1097 The individual probabilities are therefore:  
 1098

$$1099 \quad 1100 \quad p_j = \frac{\sigma_j}{\sum_k \sigma_k} \approx \frac{1/\sqrt{T}}{N/\sqrt{T}} = \frac{1}{N} \quad (21)$$

1102 The distribution  $\mathbf{p} = \{p_1, p_2, \dots, p_N\}$  is a uniform distribution over  $N$  states. The Shannon entropy  
 1103 of this distribution is maximal:

$$1104 \quad 1105 \quad H(\mathbf{p}) = - \sum_{j=1}^N p_j \log(p_j) = - \sum_{j=1}^N \frac{1}{N} \log\left(\frac{1}{N}\right) = -N \left( \frac{-\log(N)}{N} \right) = \log(N) \quad (22)$$

1108 The effective rank is the exponential of this entropy:  $\text{erank}(\bar{\mathbf{Z}}) = \exp(H(\mathbf{p})) = \exp(\log(N)) = N$ .  
 1109 In the maximal prompt entropy regime, the effective rank of the dataset matrix scales as  $\mathcal{O}(N)$ .

1110 We adapt them to our context by defining the metric's value at "time"  $n$  as the Effective Rank  
 1111 computed on the dataset matrix formed by the first  $n$  prompts, denoted  $\bar{\mathbf{Z}}_{1:n}$ . Let  $m_n = \text{erank}(\bar{\mathbf{Z}}_{1:n})$ .  
 1112 From our previous analysis, we established a crucial result that forms the basis of this derivation:  
 1113 for maximal cases, the effective rank of a dataset with  $n$  prompts scales linearly with  $n$ .  
 1114

$$1115 \quad m_n = \text{erank}(\bar{\mathbf{Z}}_{1:n}) \approx n \quad (23)$$

1116 We will use this linear approximation to derive the scaling orders of the difference metrics, assuming  
 1117 a stride of  $s = 1$  for simplicity. The first-order difference quantifies the average "velocity" of change  
 1118 in the metric relative to its historical mean. Instantaneous First-Order Difference ( $\delta_n^{(1)}$ ) is the value  
 1119 at step  $n$  minus the average of all preceding values.  
 1120

$$1121 \quad 1122 \quad \delta_n^{(1)} = m_n - \left( \frac{1}{n-1} \sum_{k=1}^{n-1} m_k \right) \quad (24)$$

1124 Substituting our approximation  $m_k \approx k$ :

$$1126 \quad 1127 \quad \delta_n^{(1)} \approx n - \left( \frac{1}{n-1} \sum_{k=1}^{n-1} k \right) \quad (25)$$

$$1130 \quad 1131 \quad \delta_n^{(1)} \approx n - \left( \frac{1}{n-1} \cdot \frac{(n-1)n}{2} \right) = n - \frac{n}{2} = \frac{n}{2} \quad (26)$$

1132 The instantaneous difference grows linearly with  $n$ . Overall First-Order Difference  $\Delta_{\text{erank}}^{(1)}$ : This is  
 1133 the average of the instantaneous differences over the entire dataset of size  $N$ .

$$\Delta_{\text{erank}}^{(1)} = \frac{1}{N-1} \sum_{n=2}^N \delta_n^{(1)} \approx \frac{1}{N-1} \sum_{n=2}^N \frac{n}{2} \quad (27)$$

$$\Delta_{\text{erank}}^{(1)} \approx \frac{1}{2(N-1)} \left( \left( \sum_{n=1}^N n \right) - 1 \right) = \frac{1}{2(N-1)} \left( \frac{N(N+1)}{2} - 1 \right) \quad (28)$$

For large  $N$ , the expression is dominated by the highest power of  $N$ :

$$\Delta_{\text{erank}}^{(1)} \sim \frac{N^2/4}{N} = \frac{N}{4} \quad (29)$$

The first-order difference of the effective rank scales linearly with the number of prompts,  $\Delta_{\text{erank}}^{(1)} = \mathcal{O}(N)$ . As for second-order difference, we compute the change in Instantaneous Differences between consecutive values of  $\delta_n^{(1)}$ .

$$\delta_n^{(1)} - \delta_{n-1}^{(1)} \approx \frac{n}{2} - \frac{n-1}{2} = \frac{1}{2} \quad (30)$$

This change is a constant, indicating a linear increase in the first-order difference. Overall Second-Order Difference  $\Delta_{\text{erank}}^{(2)}$ :

$$\Delta_{\text{erank}}^{(2)} = \frac{1}{N-2} \sum_{n=3}^N \left( d_n^{(1)} - d_{n-1}^{(1)} \right) \approx \frac{1}{N-2} \sum_{n=3}^N \frac{1}{2} \quad (31)$$

$$\Delta_{\text{erank}}^{(2)} \approx \frac{1}{N-2} \cdot (N-2) \cdot \frac{1}{2} = \frac{1}{2} \quad (32)$$

The second-order difference of the effective rank is constant and does not depend on  $N$ , yielding a scaling order of  $\Delta_{\text{erank}}^{(2)} = \mathcal{O}(1)$ .

## F ADDITIONAL THEORETICAL SUPPORT FOR EXPLORATION AND EXPLOITATION METRICS

In this section we formalize the relationship between our proposed hidden-state metrics (*Effective Rank* and *Effective Rank Velocity*) and the classical notions of exploration and exploitation in reinforcement learning. We first show that the *old* token-level metrics (average log probability and response entropy) are algebraically coupled, whereas our *new* hidden-state metrics are not. We then provide a representation-level justification for interpreting Effective Rank as a measure of semantic exploration, and Effective Rank Velocity as a measure of representation-level exploitation that is strongly correlated with greedy value improvement under the PPO-style architecture used in RLVR. Throughout, we consider a conditional language model  $p_\theta(y | x)$  and a Transformer backbone that produces hidden states  $z_t \in \mathbb{R}^D$  at each time step  $t$  for a given prompt  $x$  and generated response  $y_{1:T}$ .

## F.1 PREVIOUS VS. OUR EXPLORATION-EXPLOITATION METRICS

In this subsection, we formalize the difference between the *previous* token-level metrics used in prior RLHF/RLVR work and the *our* hidden-state metrics proposed in this paper. For a given prompt  $x$  and generated response  $y_{1:T}$ , let  $\pi_\theta(\cdot \mid x, y_{<t})$  denote the model's token-level policy distribution at step  $t$ , i.e. the softmax over the vocabulary induced by the logits at that position.

**Previous metrics (token-level action space, log-probability, and entropy).** We define the *average log probability* of an response and the *response entropy* as

$$\text{AvgLogProb}(x, y_{1:T}) := \frac{1}{T} \sum_{t=1}^T \log \pi_\theta(y_t \mid x, y_{<t}), \quad (33)$$

$$\text{RespEnt}(x, y_{1:T}) := \frac{1}{T} \sum_{t=1}^T H(\pi_\theta(\cdot \mid x, y_{<t})), \quad H(p) := - \sum_v p(v) \log p(v). \quad (34)$$

Thus RespEnt is the *token-level* entropy averaged over the response: at each step we compute the Shannon entropy of the vocabulary distribution and then average over time. At the response level semantic space, we consider  $x$  drawn from a prompt distribution  $p(x)$  and, for the purpose of analysis, responses  $y_{1:T}$  drawn *on-policy* from the model  $p_\theta(\cdot | x)$ :

$$\mathcal{L}_{\text{avg-log}}(\theta) := \mathbb{E}_{x \sim p(x)} \mathbb{E}_{y_{1:T} \sim p_\theta(\cdot | x)} [\text{AvgLogProb}(x, y_{1:T})], \quad (35)$$

$$\mathcal{H}_{\text{resp}}(\theta) := \mathbb{E}_{x \sim p(x)} \mathbb{E}_{y_{1:T} \sim p_\theta(\cdot | x)} [\text{RespEnt}(x, y_{1:T})]. \quad (36)$$

**Proposition F.1** (Token-level exploitation and exploration are tightly coupled). *Under on-policy sampling  $y_{1:T} \sim p_\theta(\cdot | x)$ , the corpus-level average log probability  $\mathcal{L}_{\text{avg-log}}(\theta)$  and response entropy  $\mathcal{H}_{\text{resp}}(\theta)$  satisfy*

$$\mathcal{L}_{\text{avg-log}}(\theta) = -\mathcal{H}_{\text{resp}}(\theta). \quad (37)$$

*In particular, under the same sampling distribution, any change of the model that increases token-level exploitation in action space as measured by  $\mathcal{L}_{\text{avg-log}}$  necessarily decreases  $\mathcal{H}_{\text{resp}}$  by the same amount (and vice versa).*

*Proof.* Fix a prompt  $x$  and a time step  $t$ . Conditioned on  $x$  and the history  $y_{<t}$ , the next token  $y_t$  is drawn from  $\pi_\theta(\cdot | x, y_{<t})$ . Taking the expectation of  $\log \pi_\theta(y_t | x, y_{<t})$  under this distribution yields:

$$\mathbb{E}_{y_t \sim \pi_\theta(\cdot | x, y_{<t})} [\log \pi_\theta(y_t | x, y_{<t})] = \sum_v \pi_\theta(v | x, y_{<t}) \log \pi_\theta(v | x, y_{<t}) = -H(\pi_\theta(\cdot | x, y_{<t})). \quad (38)$$

Now consider a full response  $y_{1:T} \sim p_\theta(\cdot | x)$ . By the law of iterated expectations,

$$\mathbb{E}_{y_{1:T} \sim p_\theta(\cdot | x)} [\log \pi_\theta(y_t | x, y_{<t})] = \mathbb{E}_{y_{<t} \sim p_\theta(\cdot | x)} \left[ \mathbb{E}_{y_t \sim \pi_\theta(\cdot | x, y_{<t})} [\log \pi_\theta(y_t | x, y_{<t})] \right] \quad (39)$$

$$= -\mathbb{E}_{y_{<t} \sim p_\theta(\cdot | x)} \left[ H(\pi_\theta(\cdot | x, y_{<t})) \right], \quad (40)$$

where we used equation 38 in the last step. Averaging over  $t = 1, \dots, T$  and dividing by  $T$  gives

$$\mathbb{E}_{y_{1:T} \sim p_\theta(\cdot | x)} [\text{AvgLogProb}(x, y_{1:T})] = \mathbb{E}_{y_{1:T} \sim p_\theta(\cdot | x)} \left[ \frac{1}{T} \sum_{t=1}^T \log \pi_\theta(y_t | x, y_{<t}) \right] \quad (41)$$

$$= -\mathbb{E}_{y_{1:T} \sim p_\theta(\cdot | x)} \left[ \frac{1}{T} \sum_{t=1}^T H(\pi_\theta(\cdot | x, y_{<t})) \right] \quad (42)$$

$$= -\mathbb{E}_{y_{1:T} \sim p_\theta(\cdot | x)} [\text{RespEnt}(x, y_{1:T})]. \quad (43)$$

Finally, taking expectation over prompts  $x \sim p(x)$  on both sides of equation 43 yields

$$\mathcal{L}_{\text{avg-log}}(\theta) = -\mathcal{H}_{\text{resp}}(\theta), \quad (44)$$

which is exactly Eq. equation 37. This shows that under on-policy sampling, the two token-level metrics are related by a fixed negative sign and thus cannot be decoupled in the action space.  $\square$

**Our metrics (hidden-state Effective Rank and velocity).** The next proposition shows that these two hidden-state metrics in Sec. 3.1 and 3.3 are *structurally decoupled* at the level of trajectories: knowing the final Effective Rank alone does not determine ERV, and conversely.

**Proposition F.2** (Hidden-state metrics are structurally decoupled). *Fix  $K \geq 3$  evaluation steps. Consider the map that associates to each Effective Rank trajectory  $m = (m_1, \dots, m_K) \in \mathbb{R}^K$  its final value*

$$\text{ER}_{\text{final}}(m) := m_K \quad (45)$$

*and its Effective Rank velocity*

$$\text{ERV}(m) := \frac{1}{K-1} \sum_{j=2}^K \left( m_j - \frac{1}{j-1} \sum_{k=1}^{j-1} m_k \right). \quad (46)$$

*Then:*

1242 1. *There is no function  $f : \mathbb{R} \rightarrow \mathbb{R}$  such that  $\text{ERV}(m) = f(\text{ER}_{\text{final}}(m))$  for all trajectories*  
 1243  *$m \in \mathbb{R}^K$ .*  
 1244 2. *There is no function  $g : \mathbb{R} \rightarrow \mathbb{R}$  such that  $\text{ER}_{\text{final}}(m) = g(\text{ERV}(m))$  for all trajectories*  
 1245  *$m \in \mathbb{R}^K$ .*

1247 *Equivalently,  $\text{ER}_{\text{final}}$  and  $\text{ERV}$  are not functionally dependent: they capture genuinely different*  
 1248 *aspects of the Effective Rank sequence.*

1250 *Proof.* We view  $\text{ER}_{\text{final}}$  and  $\text{ERV}$  as real-valued functions on  $\mathbb{R}^K$ . The proof is purely algebraic and  
 1251 does not rely on any monotonicity of  $m_j$ .

1252 Step 1:  $\text{ERV}$  is a non-trivial linear functional. Introduce the shorthand

$$1253 \quad \Delta m_j := m_j - m_{j-1}, \quad j \geq 2. \quad (47)$$

1255 A direct calculation shows that each increment  $\delta_j$  can be written as

$$1257 \quad \delta_j = \frac{1}{j-1} \sum_{r=2}^j (r-1) \Delta m_r, \quad j \geq 2, \quad (48)$$

1259 so that  $\text{ERV}$  is a linear functional of  $m$ :

$$1261 \quad \text{ERV}(m) = \sum_{j=1}^K \alpha_j m_j, \quad (49)$$

1264 for some fixed coefficients  $\alpha_1, \dots, \alpha_K$  that depend only on  $K$  (and  $s$ ) and satisfy  $\sum_{j=1}^K \alpha_j = 0$  and  
 1265  $\alpha_j \neq 0$  for at least two indices  $j$  (e.g.  $\alpha_1 \neq 0$  and  $\alpha_K \neq 0$ ). In particular,  $\text{ERV}$  is *not* proportional  
 1266 to the projection onto any single coordinate  $m_j$ .

1267 Step 2: No functional dependence of  $\text{ERV}$  on the final ER. Suppose, for contradiction, that there  
 1268 exists a function  $f : \mathbb{R} \rightarrow \mathbb{R}$  such that

$$1270 \quad \text{ERV}(m) = f(\text{ER}_{\text{final}}(m)) = f(m_K) \quad \text{for all } m \in \mathbb{R}^K. \quad (50)$$

1271 Fix any constant  $c \in \mathbb{R}$ . Consider the affine subspace

$$1273 \quad \mathcal{A}_c := \{m \in \mathbb{R}^K : m_K = c\}. \quad (51)$$

1274 On this subspace,  $\text{ER}_{\text{final}}(m) \equiv c$  is constant, so by assumption  $\text{ERV}(m) \equiv f(c)$  must also be  
 1275 constant. However,  $\text{ERV}$  is a non-trivial linear functional that depends on at least one coordinate  
 1276  $m_j$  with  $j < K$ . Therefore, restricted to  $\mathcal{A}_c$ , the map  $m \mapsto \text{ERV}(m)$  varies with those coordinates  
 1277 and cannot be constant. This yields a contradiction. Hence no such  $f$  exists.

1278 Step 3: No functional dependence of final ER on  $\text{ERV}$ . The argument is symmetric. Suppose there  
 1279 exists  $g : \mathbb{R} \rightarrow \mathbb{R}$  such that

$$1281 \quad m_K = \text{ER}_{\text{final}}(m) = g(\text{ERV}(m)) \quad \text{for all } m \in \mathbb{R}^K. \quad (52)$$

1282 Fix any constant  $c \in \mathbb{R}$  and consider the affine subspace

$$1284 \quad \mathcal{B}_c := \{m \in \mathbb{R}^K : \text{ERV}(m) = c\}. \quad (53)$$

1285 Since  $\text{ERV}$  is a non-trivial linear functional,  $\mathcal{B}_c$  is an affine hyperplane of codimension 1, and  $m_K$   
 1286 can vary freely among its points. Yet the assumed relation  $m_K = g(\text{ERV}(m)) = g(c)$  would force  
 1287  $m_K$  to be constant on  $\mathcal{B}_c$ , which is impossible. Thus no such  $g$  exists.

1288 Combining the two steps, we conclude that  $\text{ER}_{\text{final}}$  and  $\text{ERV}$  are not functionally dependent on each  
 1289 other.  $\square$

1291 **Summary.** Proposition F.1 shows that the classical token-level metrics—average log probability  
 1292 and response entropy—are *algebraically coupled* under on-policy sampling and cannot be varied  
 1293 independently. In contrast, Proposition F.2 demonstrates that the proposed hidden-state metrics—  
 1294 terminal Effective Rank and Effective Rank velocity—are structurally decoupled: they are distinct  
 1295 functionals of the Effective Rank trajectory and capture complementary aspects of exploration (se-  
 1296 mantic diversity level) and exploitation (semantic diversity gain speed).

1296  
1297

## F.2 EFFECTIVE RANK AS SEMANTIC EXPLORATION

1298  
1299

We now formalize the interpretation of Effective Rank as a measure of semantic diversity and uncertainty in the hidden-state space, and hence as a representation-level proxy for exploration in LLM reasoning. We assume that the hidden states are semantic representations in the sense that downstream semantic properties can be approximately recovered as linear functionals of the hidden vectors. This is standard in representation learning and probing work on large language models.

1300  
1301  
1302  
1303  
1304  
1305

**Assumption F.3** (Semantic linear decodability). There exists a collection of  $K$  semantic features  $s^{(1)}, \dots, s^{(K)}$  (e.g., semantic roles, entity identities, factual attributes, intermediate reasoning states) such that for each time step  $t$  and feature index  $k$  we have

1306  
1307

$$s_t^{(k)} \approx w_k^\top h_t, \quad w_k \in \mathbb{R}^D. \quad (54)$$

1308

That is, semantic features are approximately linearly decodable from hidden states.

1309  
1310  
1311  
1312  
1313  
1314

**Assumption F.4** (Bounded energy and finite support). For a given trajectory  $Z_{1:T}$ , there exists an orthonormal basis of semantic directions  $\{e_1, \dots, e_D\}$  such that each hidden state admits a decomposition

1315

$$h_t = \sum_{i=1}^D a_{t,i} e_i, \quad (55)$$

1316  
1317

with  $\sum_{t=1}^T a_{t,i}^2 < \infty$  for all  $i$ , and only finitely many coordinates  $a_{t,i}$  carry task-relevant semantic variation.

1318  
1319  
1320

**Proposition F.5** (Effective Rank as semantic diversity and uncertainty). *Let  $Z_{1:T}$  be a hidden-state trajectory satisfying Assumptions F.3–F.4. Let  $Z_{1:T} = U\Sigma R^\top$  be its SVD, and  $\text{ER}(Z_{1:T})$  its Effective Rank. Then:*

1321  
1322  
1323  
1324  
1325  
1326  
1327

1. *If the trajectory uses exactly  $k$  orthogonal semantic directions with equal energy and no others, i.e. the singular values satisfy  $\sigma_1 = \dots = \sigma_k > 0$  and  $\sigma_{k+1} = \dots = \sigma_r = 0$ , then  $\text{ER}(Z_{1:T}) = k$ .*
2. *More generally, if the singular value spectrum of  $Z_{1:T}$  becomes more spread out over more directions in the sense of majorization (i.e. the normalized singular value vector becomes more uniform over a larger support), then  $\text{ER}(Z_{1:T})$  increases.*

1328  
1329  
1330

Consequently,  $\text{ER}(Z_{1:T})$  is a basis-invariant, strictly increasing measure of the number of independent semantic directions that are effectively used by the hidden states, and thus a natural representation-level proxy for semantic exploration and uncertainty.

1331

*Proof.* We proceed in two parts.

1332  
1333  
1334  
1335

**(1) Equal-energy  $k$ -dimensional semantic subspace.** Suppose  $Z_{1:T}$  uses exactly  $k$  orthogonal semantic directions with equal energy. Then, up to permutation, the non-zero singular values satisfy

1336

$$\sigma_1 = \dots = \sigma_k = c > 0, \quad \sigma_{k+1} = \dots = \sigma_r = 0. \quad (56)$$

1337

The normalized singular values are thus

1338  
1339  
1340  
1341

$$q_i = \begin{cases} 1/k, & i = 1, \dots, k, \\ 0, & i > k, \end{cases} \quad (57)$$

1342

and the entropy of  $q$  is

1343  
1344  
1345

$$H(q) = - \sum_{i=1}^k \frac{1}{k} \log \frac{1}{k} = \log k. \quad (58)$$

1346

Therefore

1347

$$\text{ER}(Z_{1:T}) = \exp(H(q)) = \exp(\log k) = k. \quad (59)$$

1348

1349

This shows that, in the idealized case of exactly  $k$  equi-energetic semantic directions, Effective Rank matches the true semantic dimensionality  $k$ .

1350 (2) **Monotonicity under majorization.** Consider two hidden-state trajectories  $Z$  and  $\tilde{Z}$  with normalized singular value spectra  $q$  and  $\tilde{q}$ , respectively. Suppose that  $q$  is *majorized* by  $\tilde{q}$  (denoted  $q < \tilde{q}$ ), meaning intuitively that  $\tilde{q}$  is “more spread out” and therefore more uniform across a larger support.

1354 It is a standard result in information theory that the Shannon entropy  $H(\cdot)$  is Schur-concave: if  
1355  $q < \tilde{q}$ , then  $H(q) \leq H(\tilde{q})$  with strict inequality whenever  $q \neq \tilde{q}$ . Therefore

$$1356 \quad \text{ER}(Z) = \exp(H(q)) \leq \exp(H(\tilde{q})) = \text{ER}(\tilde{Z}), \quad (60)$$

1358 with strict inequality when the majorization is strict. In words, whenever the singular value spectrum  
1359 becomes more spread out across more directions, the Effective Rank strictly increases.

1360 Combining the two parts, we see that Effective Rank equals the number of equi-energetic semantic  
1361 directions in the idealized case and increases whenever the representation distributes energy over  
1362 more orthogonal directions. Since, by Assumption F.3, semantic features are linearly decodable  
1363 along such directions,  $\text{ER}(Z_{1:T})$  provides a basis-invariant measure of how many independent semantic  
1364 dimensions are explored by the hidden states and how evenly they are used. This justifies its  
1365 interpretation as a representation-level exploration and uncertainty metric.  $\square$

### 1366 F.3 EFFECTIVE RANK VELOCITY AS SEMANTIC EXPLOITATION

1368 Building on Sec. 3.1, where Effective Rank (ER) is shown to measure the number and uniform use  
1369 of semantic directions in the hidden-state space, we now give a representation-only justification for  
1370 interpreting Effective Rank Velocity (ERV) as *semantic exploitation*.

1371 Throughout this subsection we fix a single trajectory  $Z_{1:T}$  and a stride  $s$ . Let the evaluation positions  
1372 be  $t_j = js$  for  $j = 1, \dots, K$  with  $K = \lfloor (T-1)/s \rfloor$ , and write

$$1373 \quad m_j := \text{ER}(Z_{1:t_j}), \quad j = 1, \dots, K. \quad (61)$$

1375 Thus,  $\{m_j\}_{j=1}^K$  is the ER trajectory of the growing prefixes of the same response.

1377 **ERV as a recency-weighted sum of ER increments.** For convenience we recall the notation of  
1378 Def. 3.3 with  $M = \text{ER}$ . Define the local ER increments

$$1380 \quad \Delta m_r := m_r - m_{r-1}, \quad r \geq 2. \quad (62)$$

1381 Def. 3.3 introduces the “instantaneous difference”

$$1383 \quad \delta_j := m_j - \frac{1}{j-1} \sum_{k=1}^{j-1} m_k, \quad j \geq 2, \quad (63)$$

1386 and the first-order temporal difference (ERV) as

$$1388 \quad \Delta_{\text{ER}}^{(1)} := \frac{1}{K-1} \sum_{j=2}^K \delta_j. \quad (64)$$

1391 The following lemma makes explicit that ERV is a recency-weighted average of the consecutive ER  
1392 increments.

1393 **Lemma F.6** (Recency-weighted velocity of ER). *For any sequence  $(m_j)_{j=1}^K$ , the instantaneous  
1394 differences admit the representation*

$$1396 \quad \delta_j = \frac{1}{j-1} \sum_{r=2}^j (r-1) \Delta m_r, \quad j \geq 2, \quad (65)$$

1398 and hence ERV can be written as

$$1400 \quad \Delta_{\text{ER}}^{(1)} = \sum_{r=2}^K w_r \Delta m_r, \quad w_r := \frac{r-1}{K-1} \sum_{j=r}^K \frac{1}{j-1} > 0. \quad (66)$$

1403 In particular, ERV is a positive linear combination of the local ER increments  $\Delta m_r$ , assigning larger  
weights to more recent steps.

1404 *Proof.* Eq. 65 is exactly in Def. 3.3 with  $M = \text{ER}$ , obtained by expressing  $\delta_j$  in terms of the  
 1405 increments  $\Delta m_r$  via telescoping. Plugging Eq. 65 into the definition of  $\Delta_{\text{ER}}^{(1)}$  and exchanging the  
 1406 order of summation yields  
 1407

$$\Delta_{\text{ER}}^{(1)} = \frac{1}{K-1} \sum_{j=2}^K \frac{1}{j-1} \sum_{r=2}^j (r-1) \Delta m_r = \sum_{r=2}^K \left[ \frac{r-1}{K-1} \sum_{j=r}^K \frac{1}{j-1} \right] \Delta m_r, \quad (67)$$

1411 which gives Eq. 66 with  $w_r$  as stated. Since  $r-1 > 0$  and the harmonic tail  $\sum_{j=r}^K (j-1)^{-1}$  is  
 1412 positive, we have  $w_r > 0$  for all  $r$ .  $\square$   
 1413

1414 **Semantic exploitation as positive ER drift in a fixed semantic subspace.** App. F.1 states that, up  
 1415 to an orthonormal change of basis, the hidden states can be written as  $h_t = \sum_i a_{t,i} e_i$  with bounded  
 1416 energy along each semantic direction  $e_i$ , and that ER is a strictly increasing, basis-invariant measure  
 1417 of how many semantic directions are effectively used and how evenly energy is distributed among  
 1418 them. In particular, if the set of active directions (support of the singular value spectrum) is kept fixed  
 1419 and the spectrum becomes more uniform (in the sense of majorization), then ER strictly increases.  
 1420 Motivated by this, we isolate an idealized *semantic exploitation* regime in which the trajectory has  
 1421 already selected a semantic subspace and is refining it.

1422 **Definition F.7** (Semantic exploitation regime). Let  $(m_j)_{j=1}^K$  be the ER trajectory of a response, and  
 1423 let  $q^{(j)}$  denote the normalized singular value vector of  $Z_{1:t_j}$ . We say that steps  $j = 2, \dots, K$  form  
 1424 a *semantic exploitation regime with rate  $\mu > 0$*  if:

- 1425 1. (Fixed semantic support) The support of  $q^{(j)}$  is independent of  $j$ , i.e., the set of active  
 1426 semantic directions is fixed.
- 1427 2. (Uniformization within the support) For every  $j \geq 2$ ,  $q^{(j)}$  is more uniform than  $q^{(j-1)}$  on  
 1428 this fixed support, in the sense of majorization, so that by Prop. F.5 we have  $m_j - m_{j-1} =$   
 1429  $\Delta m_j \geq \mu$  for some  $\mu > 0$ .

1430 Intuitively, condition (i) says that the model has committed to a particular semantic subspace (a line  
 1431 of reasoning), and condition (ii) says that it keeps redistributing energy within this subspace to make  
 1432 use of all its semantic directions more evenly. This is precisely the notion of “refining a promising  
 1433 strategy” in representation space.

1434 **ERV lower-bounds the semantic exploitation rate.** Under Def. F.7, ER experiences a persistent  
 1435 positive drift along the trajectory. The next proposition shows that ERV is a quantitative lower bound  
 1436 on this drift, and thus a natural measure of semantic exploitation strength.

1437 **Proposition F.8** (ERV as a lower bound on semantic exploitation rate). *Assume the hidden states  
 1438 satisfy Assumptions F.3 and F.4 and that steps  $j = 2, \dots, K$  form a semantic exploitation regime  
 1439 with rate  $\mu > 0$  in the sense of Def. F.7, so that  $\Delta m_j \geq \mu$  for all  $j \geq 2$ . Then*

$$\Delta_{\text{ER}}^{(1)} \geq \frac{\mu K}{4}. \quad (68)$$

1441 *In particular, ERV is strictly positive and grows linearly with the length  $K$  of the exploitation seg-  
 1442 ment.*

1443 *Proof.* By Eq. 65 and the assumption  $\Delta m_r \geq \mu$  we obtain, for each  $j \geq 2$ ,

$$\delta_j = \frac{1}{j-1} \sum_{r=2}^j (r-1) \Delta m_r \geq \frac{1}{j-1} \sum_{r=2}^j (r-1) \mu = \frac{\mu}{j-1} \sum_{r=2}^j (r-1) = \frac{\mu j}{2}. \quad (69)$$

1450 Averaging over  $j$  then yields

$$\Delta_{\text{ER}}^{(1)} = \frac{1}{K-1} \sum_{j=2}^K \delta_j \geq \frac{1}{K-1} \sum_{j=2}^K \frac{\mu j}{2} = \frac{\mu}{2(K-1)} \sum_{j=2}^K j. \quad (70)$$

1454 For  $K \geq 2$  we have  $\sum_{j=2}^K j \geq \frac{K(K-1)}{2}$ , so

$$\Delta_{\text{ER}}^{(1)} \geq \frac{\mu}{2(K-1)} \cdot \frac{K(K-1)}{2} = \frac{\mu K}{4}, \quad (71)$$

1455 which proves Eq. 68.  $\square$   
 1456

1458 Thus, in an idealized regime where the model has already discovered a useful semantic subspace  
 1459 and is consistently enriching it, ERV provides a strictly positive, linearly growing lower bound on  
 1460 the rate at which semantic complexity within that subspace is being exploited.  
 1461

## 1462 G IMPLEMENTATION DETAILS

### 1463 G.1 TRAINING AND EVALUATION DETAILS

1464 We typically use the same set of hyperparameters to train and evaluate all models in the SimpleRL-  
 1465 Zoo series (Zeng et al., 2025) in the default main experiment setting.  
 1466

1467 **Training.** We conduct all experiments with 4 A800-PCIE-80GB GPUs. For GRPO and PPO, we  
 1468 use a prompt batch size of 48 with a maximum rollout length of 1536 tokens. Training is performed  
 1469 using a mini-batch size of 24. For GRPO, we generate 4 rollouts per prompt. For PPO, we use  
 1470 DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025) as the value model and generate 1 rollout per  
 1471 prompt. The default sampling temperature is set to 1.0, and the clip ratio is 0.2. For all actor models  
 1472 ranging from 3B to 8B parameters, we use a learning rate of 1e-6 and a KL loss coefficient of 1e-4.  
 1473 For critic models in PPO, we use a learning rate of 1e-5. For our training datasets, we follow the  
 1474 same setup as in Zeng et al. (2025), where the data is filtered from GSM8K (Cobbe et al., 2021)  
 1475 and MATH (Hendrycks et al., 2021) configured with different difficulty levels for models of varying  
 1476 capabilities. We tested using the checkpoint model trained up to 120 steps.  
 1477

1478 **Evaluation.** We build our evaluation script based on that of Zeng et al. (2025), using a temper-  
 1479 ature of 0.6 and a maximum generation length of 2048 tokens. To ensure consistency, we adopt  
 1480 the same prompt template used during training. For most benchmarks, we report Pass@1 results.  
 1481 However, for benchmarks like AIME 2024, which contains fewer problems, we report both Pass@1  
 1482 and average accuracy (Pass@256), computed over 256 generated samples per problem.  
 1483

1484 **Base Models.** To demonstrate the universality of our insights and methods, we conduct zero  
 1485 RL training experiments on Llama-3.2 (3B), Llama-3.1 (8B) (Dubey et al., 2024), Mistral-v0.3-  
 1486 7B (Jiang et al., 2024), and Qwen-2.5 (1.5B, 3B, 7B) (Hui et al., 2024). For value model in PPO,  
 1487 we use DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025) for all experiments.  
 1488

1489 **Benchmark.** We evaluate on a diverse suite of mathematical reasoning benchmarks. These include  
 1490 standard benchmarks such as GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), AS-  
 1491 Div (Miao et al., 2020), Carp (English Version) (Zhang et al., 2023), MAWPS (Koncel-Kedziorski  
 1492 et al., 2016), SVAMP (Patel et al., 2021), TabMWP (Lu et al., 2023), and OlympiadBench (He  
 1493 et al., 2024); Chinese mathematics collections like CMATH (Wei et al., 2023) and Gaokao 2024;  
 1494 and benchmarks from mathematics competitions, including the 2024/2025 AIME and the 2023/2024  
 1495 AMC.  
 1496

### 1497 G.2 EFFICIENT INCREMENTAL COMPUTATION OF HIGHER-ORDER METRICS

1498 A naive computation of the temporal difference metrics would be computationally prohibitive. Our  
 1499 method’s feasibility hinges on an efficient, incremental algorithm that computes the required metrics  
 1500 without redundant operations on the growing hidden state matrix  $\mathbf{Z} \in \mathbb{R}^{T \times D}$ .  
 1501

1502 The effective rank is derived from the singular values of the mean-centered hidden state matrix  $\mathbf{Z}_c$ .  
 1503 These are equivalent to the square roots of the eigenvalues of the centered Gram matrix  $G = \mathbf{Z}_c \mathbf{Z}_c^\top$ .  
 1504 Instead of recomputing  $G_t$  from scratch at each time step  $t$ , our algorithm incrementally updates  
 1505 two sufficient statistics: the uncentered Gram matrix  $U_t = Z_{1:t} Z_{1:t}^\top$  and the sum of hidden state  
 1506 vectors  $s_t = \sum_{i=1}^t z_i$ . When extending the analysis window, the new uncentered Gram matrix  $U_t$  is  
 1507 constructed from the prior matrix  $U_{t-s}$  and the new chunk of hidden states  $\Delta Z_t = Z_{t-s+1:t}$ . This  
 1508 update follows a recursive block matrix structure:  
 1509

$$U_t = \begin{pmatrix} U_{t-s} & Z_{1:t-s}(\Delta Z_t)^\top \\ (\Delta Z_t)Z_{1:t-s}^\top & (\Delta Z_t)(\Delta Z_t)^\top \end{pmatrix} \quad (72)$$

1510 From the efficiently updated  $U_t$  and  $s_t$ , we can directly construct the centered Gram matrix  $G_t$ .  
 1511 Letting  $\mu_t = s_t/t$  be the mean vector and  $\mathbf{1}_t$  be a column vector of ones,  $G_t$  can be expressed as:  
 1512

$$G_t = U_t - (Z_{1:t} \mu_t) \mathbf{1}_t^\top - \mathbf{1}_t (Z_{1:t} \mu_t)^\top + (\mu_t^\top \mu_t) \cdot (\mathbf{1}_t \mathbf{1}_t^\top) \quad (73)$$

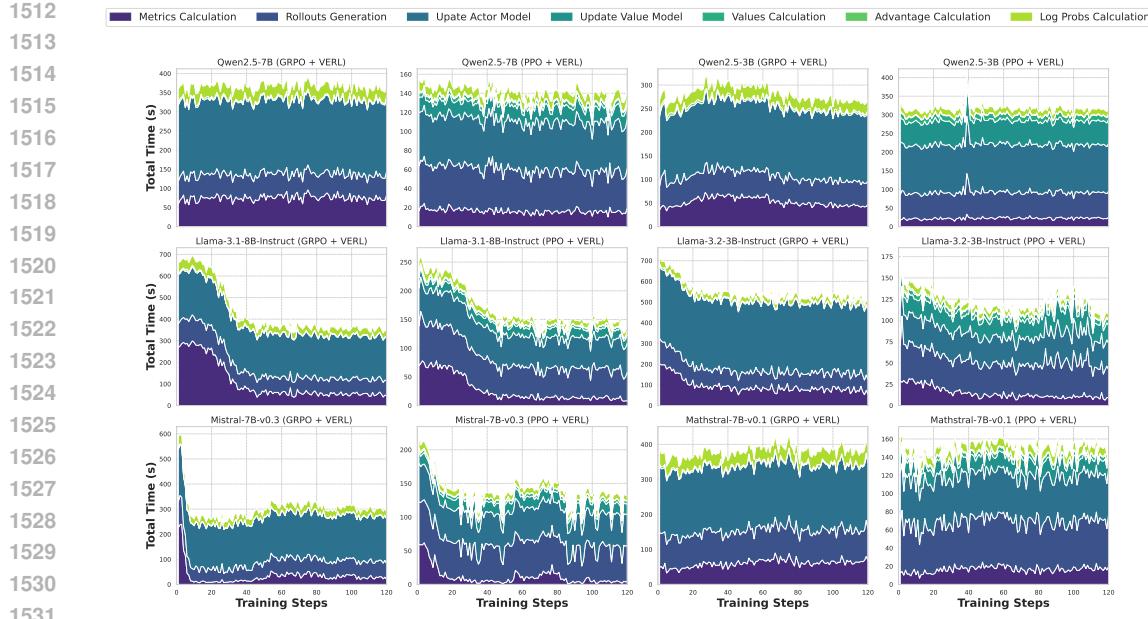


Figure 6: Time overhead of the main computation of RL Training.

This allows for the calculation of  $G_t$  without re-accessing the full history of hidden states. The eigenvalues  $\{\lambda_j\}$  of  $G_t$  are then used to derive the effective rank. First, the singular values of the centered matrix are obtained,  $\sigma_j = \sqrt{\lambda_j}$ . These are normalized to form a probability distribution,  $p_j = \sigma_j / \sum_k \sigma_k$ . The effective rank is then the exponential of the Shannon entropy of this distribution:  $\text{erank}(Z_{c,t}) = \exp\left(-\sum_j p_j \log p_j\right)$ . This pipeline efficiently yields a sequence of effective rank values,  $m_{j,s} = \text{erank}(Z_{c,j,s})$ , at each stride  $s$ . From this sequence, we compute the instantaneous first-order difference,  $\delta$ , which compares the current value to the running average of all preceding values. This is defined recursively as:  $\delta_{j,s} = m_{j,s} - \frac{1}{j-1} \sum_{k=1}^{j-1} m_{k,s}$ .

The computational advantage of this incremental approach is substantial. While the total cost for the series of eigenvalue decompositions  $\mathcal{O}(T^4/s)$ , is common to both methods, the cost of matrix construction differs significantly. The naive method's recalculation totals  $\mathcal{O}(DT^3/s)$ , whereas our incremental update method reduces this to  $\mathcal{O}(DT^2)$ . This reduction of the polynomial dependency on sequence length  $T$  from cubic to quadratic is critical, as this term is multiplied by the large hidden dimension  $D$ , making it the dominant factor in practical performance and rendering the dense calculation of temporal dynamics feasible. In the worst-case regime where the sequence length  $T$  exceeds the hidden dimension  $D$ , and both  $D$  and the stride  $s$  can be treated as constants. The naïve approach that reconstructs matrices independently at each stride has a matrix-construction cost scaling as  $\mathcal{O}(T^2)$ , VERL's incremental Gram/covariance updates scale as  $\mathcal{O}(T)$ . So asymptotically, our implementation has a strictly better dependency on  $T$  than a naïve SVD-based design.

### G.3 TIME OVERHEAD OF VERL TRAINING

We conducted post-training with Zero RL on several base models. The Fig. 6 illustrates the time associated with each computational stage. The 'metrics calculation' component, which represents the cost of computing metrics for hidden states, accounts for an insignificant portion of the total processing time. This demonstrates that our method does not introduce substantial time overhead. To further stress-test the worst-case scenario, we deliberately compute ER, ERV, and ERA on the CPU rather than the GPU, and still observe that the additional time overhead remains negligible.

## H MORE EXPERIMENTS

### H.1 ANALYSIS OF RESPONSE-LEVEL HIDDEN STATES

As shown in Figs. 7 and 8, our analysis of response-level hidden states across additional LLMs confirms that the insights presented in Sec. 4.1 hold true for various base models and RL paradigms.

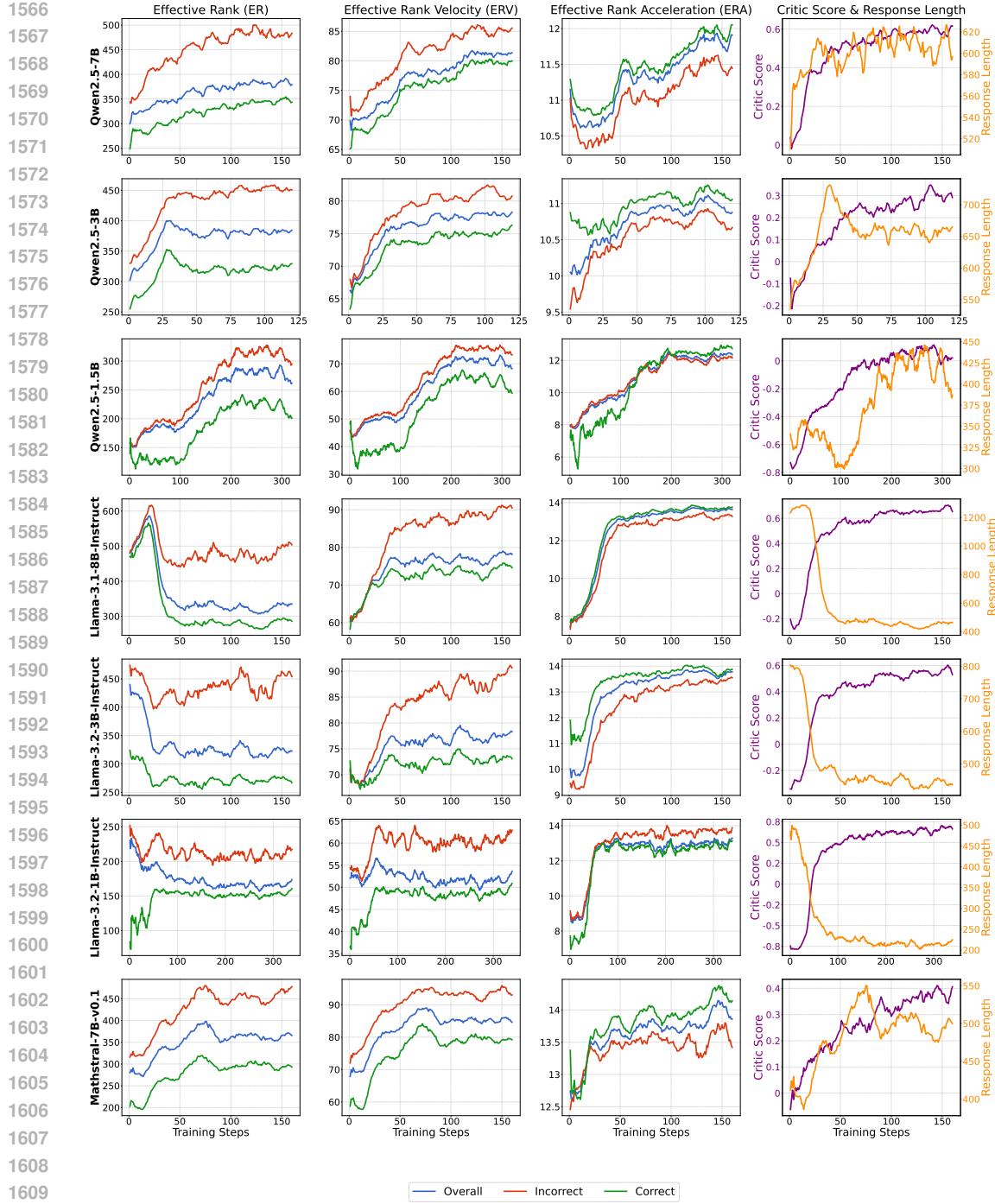


Figure 7: Visualization of response-level metrics for GRPO post-training. 'Overall' (blue) represents the metric across the entire data batch, while 'Correct' (green) and 'Incorrect' (red) show the metrics specifically for correctly and incorrectly classified samples, respectively. Data is smoothed using a rolling window of 10 steps to highlight underlying trends.

## H.2 ANALYSIS OF DATASET-LEVEL HIDDEN STATES

As shown in Figs. 9 and 10, our analysis of dataset-level hidden states across additional LLMs confirms that the insights presented in Sec. 4.2 hold for various base models and RL paradigms.

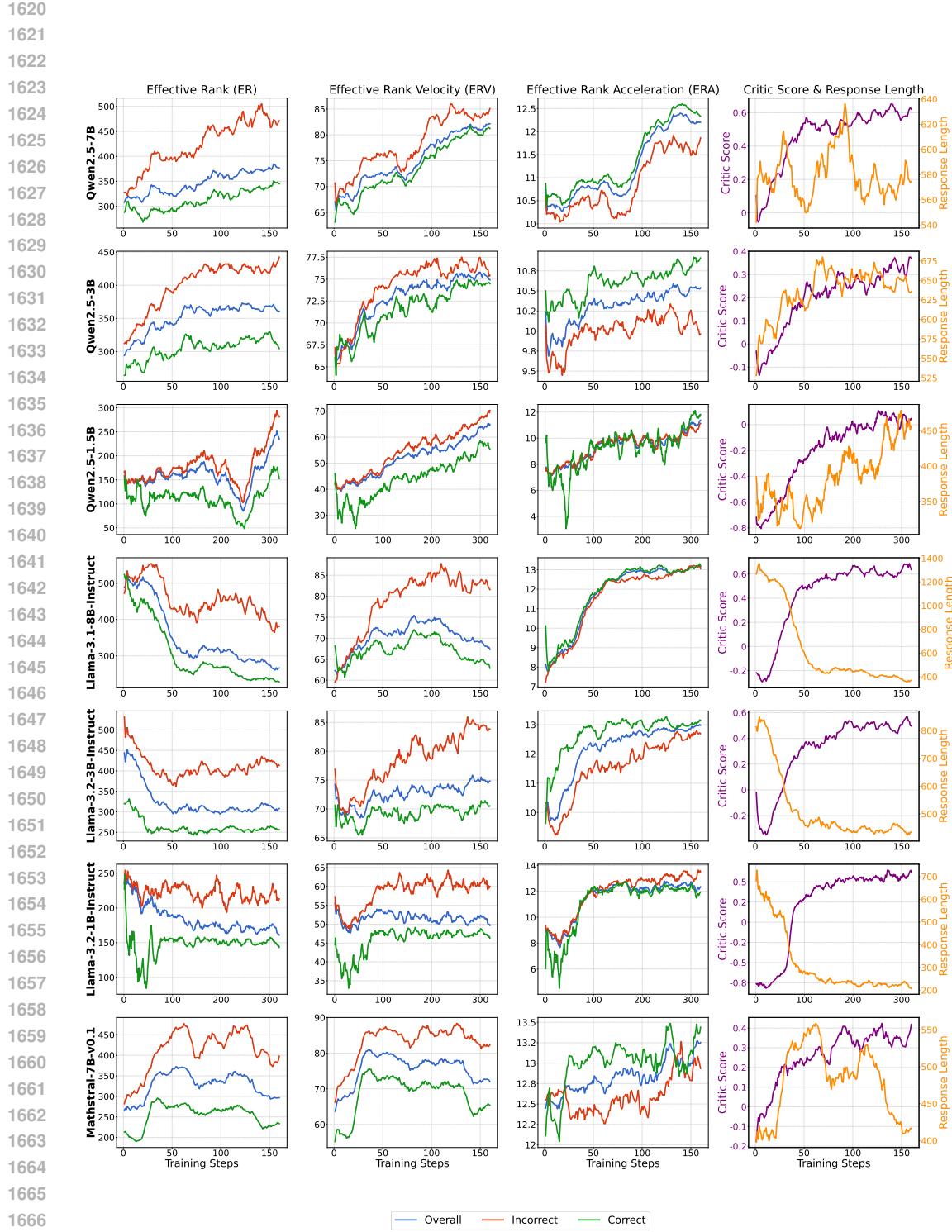


Figure 8: Visualization of response-level metrics for PPO post-training. 'Overall' (blue) represents the metric across the entire data batch, while 'Correct' (green) and 'Incorrect' (red) show the metrics specifically for correctly and incorrectly classified samples, respectively. Data is smoothed using a rolling window of 10 steps to highlight underlying trends.

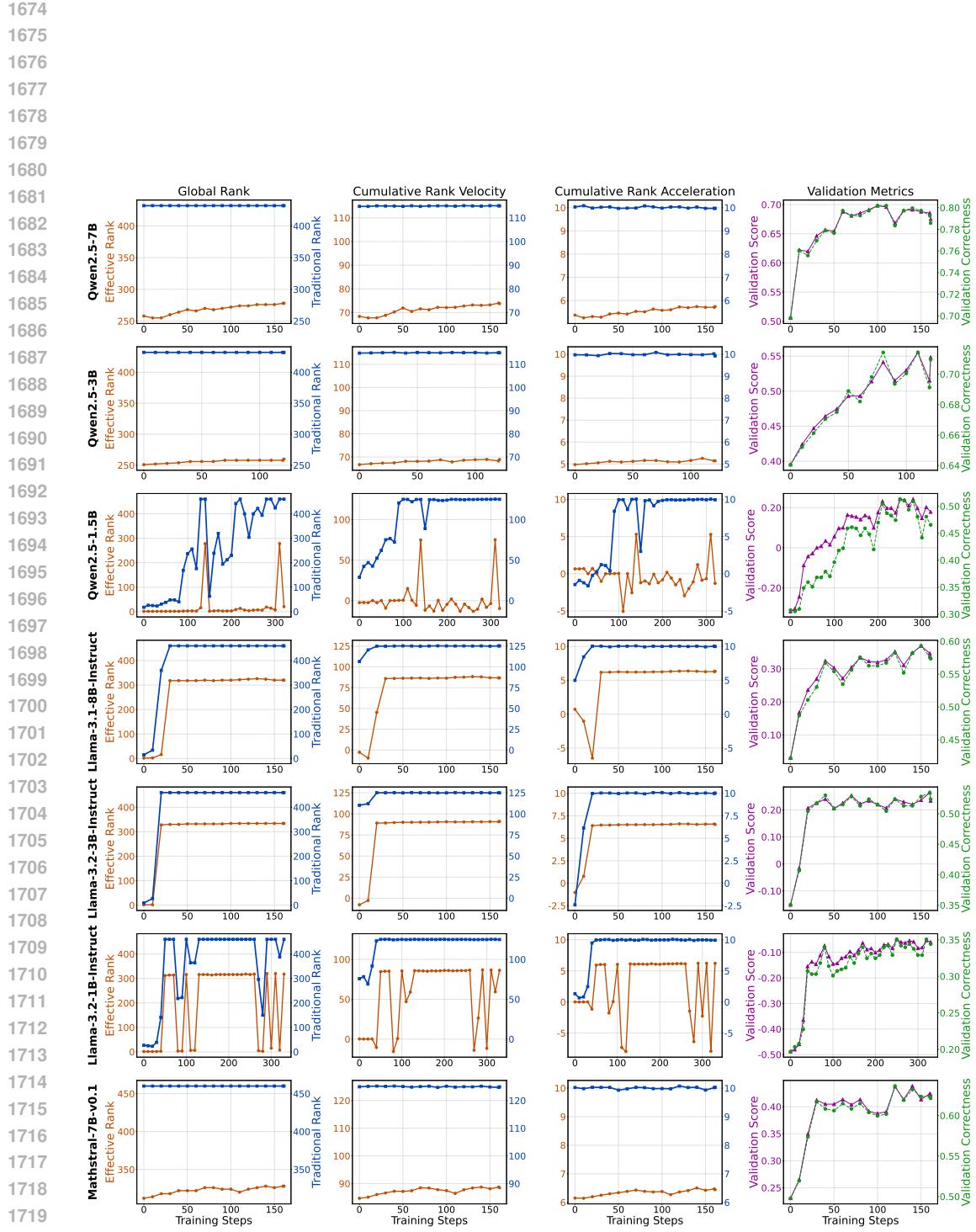


Figure 9: Visualization of dataset-level metrics for GRPO post-training

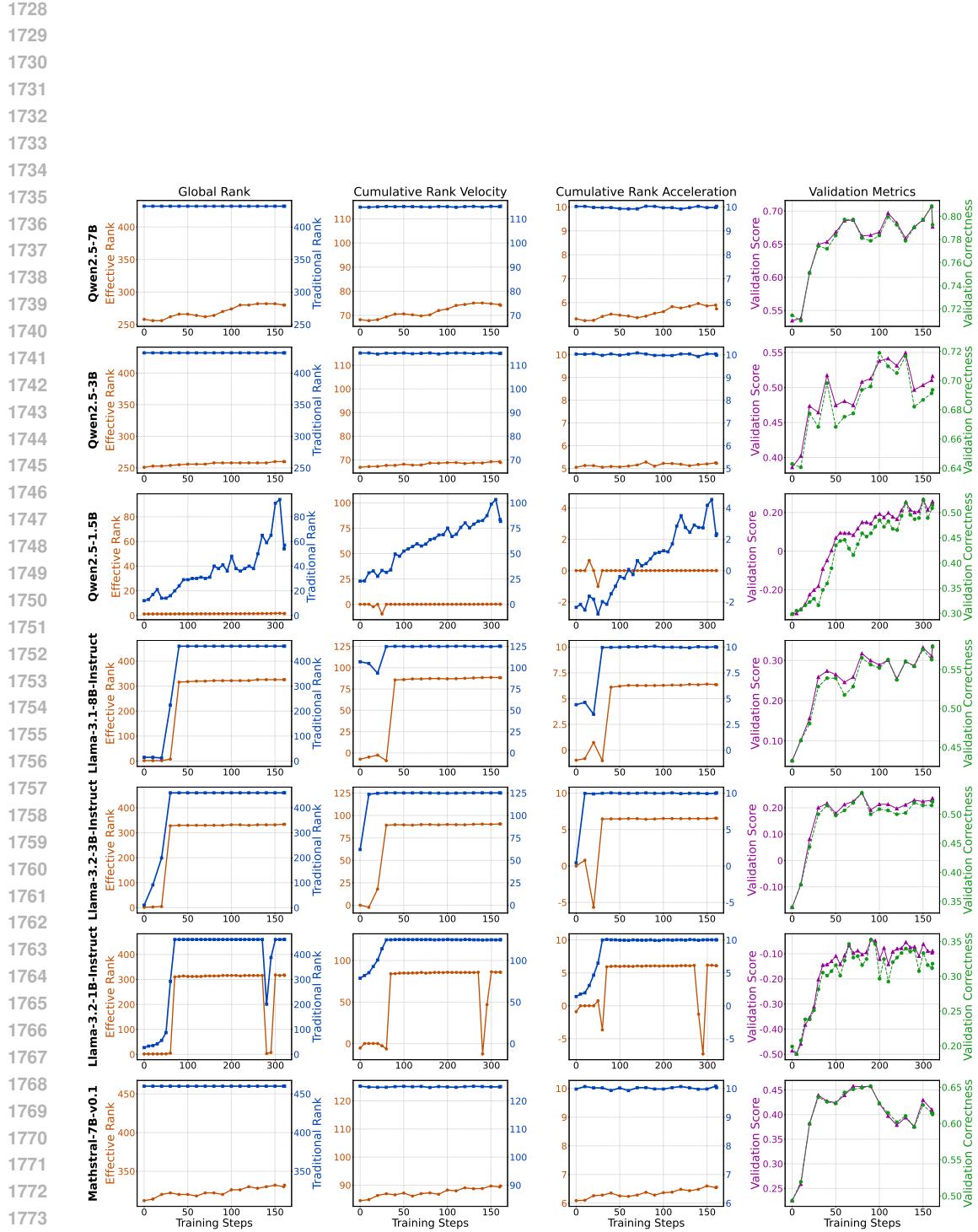


Figure 10: Visualization of dataset-level metrics for PPO post-training

1782 **H.3 DETAILED ANALYSIS OF PASS@1 PERFORMANCE**  
1783

1784 As shown in Tab. 4, Pass@1 measures the model’s ability to generate a correct answer in a single  
1785 attempt, which directly reflects its exploitation ability. We fine-tune the base model by integrating  
1786 our VERL-based Advantage method into two reinforcement learning paradigms, GRPO and PPO.

1787 Table 4: Performance comparison of instruction-tuned models on mathematical reasoning benchmarks  
1788 (Pass@1). “+ GRPO” and “+ PPO” denote reinforcement learning fine-tuning from the base  
1789 model using GRPO and PPO, respectively. “w/ VERL.” indicates the application of our VERL-based  
1790 advantage to the corresponding RL algorithm.  $\Delta$  represents the performance difference between the  
1791 baseline RL method and its VERL-advanced variant. All results are reported in percentage (%).

Model	AIME24	AIME24	AMC2	AMC2	ASDiv	Carp.Em	CMATH	Gaokao 2024.1	Gaokao 2024.Mix	Gaokao MathCloze	GSM8	MAWPS	Olympiad Bench	SVAMP	TabMWP	Avg.
Llama-3-2-3B-Instruct	0.0	0.0	25.0	11.1	74.6	26.5	10.2	14.3	14.3	6.8	66.6	86.9	12.7	74.1	41.4	31.0
+ GRPO	3.3	0.0	27.5	8.9	88.8	45.0	28.3	21.4	20.9	23.7	80.7	96.0	16.7	87.7	71.7	41.4
+ GRPO w/ VERL.	13.3	6.7	25.0	11.1	89.3	45.4	46.2	14.3	22.0	22.9	81.7	96.0	17.6	87.8	72.3	43.4
$\Delta_{\text{GRPO}}$	+10.0	+6.7	-2.5	+2.2	+0.5	+0.4	+17.9	-7.1	+1.1	-0.8	+1.0	+0.0	+0.9	+0.1	+0.6	+2.0
+ PPO	10.0	3.3	22.5	13.3	87.9	46.4	21.2	7.1	16.5	20.3	81.4	95.5	17.8	86.8	71.0	40.1
+ PPO w/ VERL.	10.0	3.3	25.0	11.1	88.7	46.0	30.7	14.3	19.8	27.1	82.9	95.7	17.3	85.8	71.3	41.9
$\Delta_{\text{PPO}}$	+0.0	+0.0	+2.5	-2.2	+0.8	-0.4	+9.5	+7.2	+3.3	+6.8	+1.5	+0.2	-0.5	-1.0	+0.3	+1.9
Llama-3-1-8B-Instruct	0.0	3.3	17.5	8.9	48.0	34.1	18.5	0.0	15.4	16.9	47.4	43.5	10.4	48.5	34.3	23.1
+ GRPO	6.7	0.0	25.5	15.6	90.3	42.4	60.7	7.1	14.3	32.2	88.4	96.4	19.7	88.5	82.7	44.5
+ GRPO w/ VERL.	10.0	3.3	32.5	15.6	90.7	45.0	72.7	14.3	14.3	30.5	88.6	96.9	21.3	88.4	83.1	47.2
$\Delta_{\text{GRPO}}$	+3.3	+3.3	+10.0	+0.0	+0.4	+2.6	+12.0	+7.2	+0.0	-1.7	+0.2	+0.5	+1.6	-0.1	+0.4	+2.7
+ PPO	6.7	0.0	30.0	17.8	89.8	42.0	60.0	0.0	14.3	25.4	86.4	95.7	18.2	88.6	82.3	43.8
+ PPO w/ VERL.	10.0	0.0	35.0	13.3	90.7	42.6	62.0	14.3	22.0	28.8	87.3	96.6	19.1	88.1	83.0	46.2
$\Delta_{\text{PPO}}$	+3.3	+0.0	+5.0	-4.5	+0.9	+0.6	+2.0	+14.3	+7.7	+3.4	+0.9	+0.9	-0.5	-0.5	+0.7	+2.4
Qwen-2.5-3B	6.7	0.0	20.0	24.4	90.7	54.7	76.7	0.0	22.0	41.5	89.7	95.1	23.0	84.3	71.3	46.1
+ GRPO	3.3	0.0	40.0	22.2	92.6	56.0	82.7	7.1	27.5	42.4	82.8	96.6	23.6	89.0	81.4	49.8
+ GRPO w/ VERL.	6.7	0.0	30.0	17.8	92.6	56.9	84.8	21.4	33.0	49.2	82.2	96.4	24.4	88.5	81.0	51.0
$\Delta_{\text{GRPO}}$	+3.4	+0.0	-10.0	-4.4	+0.0	+0.9	+2.1	+14.3	+5.5	+6.8	-0.6	-0.2	+0.8	-0.5	-0.4	+1.2
+ PPO	3.3	0.0	32.5	15.6	92.8	56.5	83.2	0.0	28.6	50.0	81.7	96.6	24.4	86.0	80.8	48.8
+ PPO w/ VERL.	6.7	0.0	32.5	17.8	92.6	57.0	84.3	21.4	29.7	47.5	81.8	96.5	24.6	88.3	81.4	50.8
$\Delta_{\text{PPO}}$	+3.4	+0.0	+0.0	+2.2	-0.2	+0.5	+1.1	+21.4	+1.1	-2.5	+0.1	-0.1	+0.2	+2.3	+0.6	+2.0
Qwen2.5-7B	6.7	0.0	45.0	15.6	91.4	55.8	86.7	42.9	33.0	49.2	85.8	95.4	25.8	88.5	82.8	53.6
+ GRPO	10.0	6.7	55.0	26.7	94.8	60.2	91.7	14.3	34.1	64.4	90.2	97.6	36.1	92.8	91.3	57.7
+ GRPO w/ VERL.	13.3	10.0	50.0	28.9	95.0	60.8	90.7	35.7	69.5	89.2	97.7	35.4	92.9	91.9	59.8	
$\Delta_{\text{GRPO}}$	+3.3	+3.3	-5.0	+2.2	+0.2	+0.6	-1.0	+21.4	+1.1	+5.1	-1.0	+0.1	-0.7	+0.1	+0.6	+2.1
+ PPO	6.7	3.3	50.0	33.3	94.9	59.6	89.8	28.6	31.9	63.6	89.1	97.3	36.1	92.8	90.8	57.9
+ PPO w/ VERL.	10.0	6.7	52.5	33.3	94.8	60.0	90.3	28.6	34.1	66.9	90.2	97.8	36.1	92.5	90.6	59.0
$\Delta_{\text{PPO}}$	+3.3	+3.3	+2.5	+0.0	-0.1	+0.4	+0.5	+0.0	+2.2	+3.3	+1.1	+0.5	+0.0	-0.3	-0.2	+1.1
Mathtral-7B-v0.1	0.0	0.0	12.5	8.9	87.1	51.1	74.2	28.6	33.0	31.4	81.6	93.8	17.9	87.7	54.7	44.2
+ GRPO	0.0	0.0	47.5	17.8	92.9	55.9	81.3	35.7	44.0	49.2	88.1	97.6	25.6	93.0	81.5	54.0
+ GRPO w/ VERL.	6.7	0.0	45.0	20.0	93.3	55.5	81.5	50.0	40.7	46.6	89.5	97.2	29.3	90.7	83.5	55.3
$\Delta_{\text{GRPO}}$	+6.7	+0.0	-2.5	+2.2	+0.4	-0.4	+0.2	+14.3	-3.3	-2.6	+1.4	-0.4	+3.7	-2.3	+2.0	+1.3
+ PPO	6.7	3.3	32.5	20.0	90.9	51.8	78.3	42.9	37.4	49.2	87.0	96.0	28.4	89.9	70.7	52.3
+ PPO w/ VERL.	10.0	0.0	27.5	22.2	93.0	53.8	78.2	42.9	51.6	48.3	87.4	96.7	26.1	89.6	84.1	54.1
$\Delta_{\text{PPO}}$	+3.3	-3.3	-5.0	+2.2	+2.1	+2.0	-0.1	+0.0	+14.2	-0.9	+0.4	+0.7	-2.3	-0.3	+13.4	+1.8
Mistral-7B-v0.3	0.0	0.0	10.0	0.0	40.5	12.4	21.8	14.3	13.2	3.4	24.0	50.8	1.6	39.1	30.6	17.4
+ GRPO	0.0	0.0	2.5	4.4	58.2	11.1	42.3	0.0	15.4	5.1	52.4	79.2	3.0	47.6	37.7	23.9
+ GRPO w/ VERL.	0.0	0.0	7.5	2.2	59.1	15.0	43.0	0.0	6.6	4.2	40.3	69.5	2.8	57.5	53.0	24.0
$\Delta_{\text{GRPO}}$	+0.0	+0.0	+5.0	-2.2	+0.9	+3.9	+0.7	+0.0	-8.8	-0.9	-12.1	-9.7	-0.2	+9.9	+15.3	+0.1
+ PPO	0.0	0.0	0.0	0.0	8.9	6.6	7.7	7.1	11.0	2.5	3.3	8.6	2.1	6.9	12.0	5.1
+ PPO w/ VERL.	0.0	0.0	2.5	0.0	44.7	10.6	35.7	7.1	16.5	5.1	28.8	70.7	2.4	57.5	35.1	21.1
$\Delta_{\text{PPO}}$	+0.0	+0.0	+2.5	+0.0	+35.8	+4.0	+28.0	+0.0	+5.5	+2.6	+25.5	+62.1	+0.3	+50.6	+23.1	+16.0

1836 **H.4 DETAILED ANALYSIS OF PASS@ $k$  PERFORMANCE**  
1837

1838 As shown in Tab. 5, which provides a comprehensive analysis of the models’ performance on the  
 1839 Pass@ $k$  metric, which is a direct measure of the model’s exploration ability. As a supplement  
 1840 to the main paper’s discussion, it presents the detailed performance of our VERL-based models  
 1841 across a variety of mathematical reasoning benchmarks. These results demonstrate that our method  
 1842 consistently leads to significant improvements, confirming its effectiveness in enhancing the models’  
 1843 exploration capabilities.

1844 Table 5: Performance comparison of instruction-tuned models under diverse decoding settings  
 1845 (Pass@ $k$ ). All results are reported in percentage (%).

1847 <b>Model</b>	1848 <b>MATH500</b> (Pass@16)	1848 <b>AMC23</b> (Pass@128)	1848 <b>AMC24</b> (Pass@128)	1848 <b>AIME24</b> (Pass@256)	1848 <b>AIME25</b> (Pass@256)	1848 <b>Avg.</b>
<b>Llama-3.2-3B-Instruct</b>	79.8	93.5	51.5	40.0	30.0	58.96
+ GRPO	80.2	95.4	60.6	40.0	30.0	61.24
+ GRPO w/ VERL.	80.6	95.7	59.0	50.0	36.7	64.40
$\Delta_{\text{GRPO}}$	+0.4	+0.3	-1.6	+10.0	+6.7	+3.16
+ PPO	82.2	94.5	57.0	46.7	36.7	63.42
+ PPO w/ VERL.	82.4	94.7	57.8	46.7	40.0	64.32
$\Delta_{\text{PPO}}$	+0.2	+0.2	+0.8	+0.0	+3.3	+0.90
<b>Llama-3.1-8B-Instruct</b>	79.8	94.6	57.4	46.7	36.7	63.04
+ GRPO	83.4	94.9	56.9	53.3	36.7	65.04
+ GRPO w/ VERL.	83.4	95.1	63.1	50.0	36.7	65.66
$\Delta_{\text{GRPO}}$	+0.0	+0.2	+6.2	-3.3	+0.0	+0.62
+ PPO	79.2	92.4	59.0	46.7	36.7	62.80
+ PPO w/ VERL.	82.4	91.9	60.0	53.3	36.7	64.86
$\Delta_{\text{PPO}}$	+3.2	-0.5	+1.0	+6.6	+0.0	+2.06
<b>Qwen2.5-3B</b>	86.0	96.7	69.0	56.7	40.0	69.68
+ GRPO	86.6	92.2	68.5	46.7	40.0	66.80
+ GRPO w/ VERL.	87.6	95.9	67.8	53.3	43.3	69.58
$\Delta_{\text{GRPO}}$	+1.0	+3.7	-0.7	+6.6	+3.3	+2.78
+ PPO	87.8	96.5	67.9	43.3	43.3	67.76
+ PPO w/ VERL.	88.2	96.8	67.3	53.3	43.3	69.78
$\Delta_{\text{PPO}}$	+0.4	+0.3	-0.6	+10.0	+0.0	+2.02
<b>Qwen2.5-7B</b>	90.6	98.4	73.7	60.0	60.0	76.54
+ GRPO	90.8	97.8	78.3	56.7	50.0	74.72
+ GRPO w/ VERL.	91.4	98.3	79.0	63.3	60.0	78.40
$\Delta_{\text{GRPO}}$	+0.6	+0.5	+0.7	+6.6	+10.0	+3.68
+ PPO	91.2	98.6	74.3	53.3	56.7	74.82
+ PPO w/ VERL.	91.4	98.0	74.4	56.7	66.7	77.44
$\Delta_{\text{PPO}}$	+0.2	-0.6	+0.1	+3.4	+10.0	+2.62
<b>Mathstral-7B-v0.1</b>	80.4	88.5	60.9	43.3	36.7	61.96
+ GRPO	84.8	87.3	69.2	36.7	40.0	63.60
+ GRPO w/ VERL.	87.0	97.0	76.9	50.0	50.0	72.18
$\Delta_{\text{GRPO}}$	+2.2	+9.7	+7.7	+13.3	+10.0	+8.58
+ PPO	82.4	91.7	70.7	53.3	40.0	67.62
+ PPO w/ VERL.	84.8	93.8	69.9	53.3	46.7	69.70
$\Delta_{\text{PPO}}$	+2.4	+2.1	-0.8	+0.0	+6.7	+2.08
<b>Mistral-7B-v0.3</b>	36.0	73.5	39.6	20.0	16.7	37.16
+ GRPO	33.0	63.2	36.0	10.0	10.0	30.44
+ GRPO w/ VERL.	34.4	64.5	38.0	16.7	13.3	33.38
$\Delta_{\text{GRPO}}$	+1.4	+1.3	+2.0	+6.7	+3.3	+2.94
+ PPO	21.8	46.4	25.1	6.7	6.7	21.34
+ PPO w/ VERL.	19.2	46.5	30.1	3.3	13.3	22.48
$\Delta_{\text{PPO}}$	-2.6	+0.1	+5.0	-3.4	+6.6	+1.14

1886 **H.5 ABLATION ON THE CHOICE OF HIDDEN LAYER**  
1887

1888 We focus on the final hidden layer because our exploration/exploitation metrics are defined in the  
 1889 *semantic* space along the reasoning trajectory, and prior interpretability work (Jing et al., 2025;  
 Sajjad et al., 2022; Valeriani et al., 2023; Matthews et al., 2024; Servedio et al., 2025; Zhang et al.,

1890 Table 6: Comparison of GRPO + VERL using an intermediate layer (layer 14) versus the final layer,  
 1891 evaluated by pass@1 on multiple math benchmarks. Using the last layer yields the strongest average  
 1892 improvement.

Method	aimc24	aimc25	amc23	amc24	asdiv	carp.en	cmath	gao kao24.1	gao kao24.mix	gao kao.math.cloze	gsm8k	mawps	olympiadbench	svamp	tabnlp	Avg.
Llama-3.2-3B-Instruct	0.0	0.0	25.0	11.1	74.6	26.5	10.2	14.3	14.3	6.8	66.6	86.9	12.7	74.1	41.4	30.97
GRPO	3.3	0.0	27.5	8.9	88.8	45.0	28.3	21.4	20.9	23.7	80.7	96.0	16.7	87.7	71.7	41.37
GRPO w/ VERL (layer = 14)	10.0	0.0	27.5	11.1	88.6	43.6	30.7	21.4	16.5	22.0	81.9	95.5	18.1	87.0	71.4	41.69
GRPO w/ VERL (layer = last)	13.3	6.7	25.0	11.1	89.3	45.4	46.2	14.3	22.0	22.9	81.7	96.0	17.6	87.8	72.3	<b>43.44</b>

1897 Table 7: Comparison of GRPO + VERL using an intermediate layer (layer 14) versus the final layer,  
 1898 evaluated by pass@k on several math benchmarks. Again, using the last layer yields the best average  
 1899 improvement.

Method	math500@16	amc23@128	amc24@128	aimc24@256	aimc25@256	Avg.
Llama-3.2-3B-Instruct	79.8	93.5	51.5	40.0	30.0	58.96
GRPO	80.2	95.4	60.6	40.0	30.0	61.24
GRPO w/ VERL (layer = 14)	81.0	92.9	57.0	40.0	36.7	61.52
GRPO w/ VERL (layer = last)	80.6	95.7	59.0	50.0	36.7	<b>64.40</b>

1905 2025) suggests that the last layers are most aligned with semantic meaning and model predictions.  
 1906 Empirically, using the final layer gives consistently better performance than using an intermediate  
 1907 layer: for example, GRPO + VERL with the last layer improves the average pass@1 from 41.69%  
 1908 (layer 14) to 43.44%, and the average pass@k from 61.52% (layer 14) to 64.40%. This subsection  
 1909 provides the detailed analysis supporting our design choice to base VERL on the final layer.

1910 Intermediate layers in large language models can encode rich features. However, our notion of  
 1911 exploration and exploitation is explicitly defined in the semantic space of a reasoning trajectory.  
 1912 Existing interpretability studies (Jing et al., 2025; Sajjad et al., 2022; Valeriani et al., 2023; Matthews  
 1913 et al., 2024; Servedio et al., 2025; Zhang et al., 2025) indicate that hidden states in the last layers are  
 1914 most tightly aligned with token-level semantics and the model’s predictive distribution, while mid-  
 1915 layer representations tend to mix morphology, syntax, and other lower-level or task-specific signals.  
 1916 For this reason, we consider the final layer more suitable for semantic diversity metrics such as ER  
 1917 and ERV.

1918 **Intermediate-layer vs. final-layer VERL (pass@1).** We implemented VERL on an intermediate  
 1919 layer (layer 14) and on the last layer, keeping everything else fixed. The pass@1 results are reported  
 1920 in Table 6. Both VERL variants improve over GRPO, but the last-layer version clearly gives the  
 1921 strongest overall gains in average pass@1.

1922 **Intermediate-layer vs. final-layer VERL (pass@k).** A similar pattern holds for pass@k, sum-  
 1923 marized in Table 7. Using the last layer yields the best average improvement, especially on the more  
 1924 challenging AIME-style benchmarks, further supporting the choice of the final hidden layer as the  
 1925 basis for our semantic exploration and exploitation metrics.

1944  
1945

## I CASE STUDY

1946  
1947  
1948  
1949

For all case studies presented in this section, we employ Qwen2.5-7B as the base model. The answers shown in the gray boxes are generated by Qwen2.5-7B trained with vanilla GRPO for 120 steps, while those in the purple boxes are produced by Qwen2.5-7B trained with GRPO augmented with the auxiliary shaping term—our proposed method VERL—for 120 steps.

1950  
1951

### I.1 CASE STUDY FOR PASS@1 SETTING

1952  
1953  
1954  
1955  
1956  
1957  
1958

**Case Study I.** For the case shown in Fig. 11, the model trained with vanilla GRPO incorrectly judges  $-1 < -13$ , whereas the model trained with GRPO augmented with the auxiliary shaping term gives the correct answer. This example highlights the limitation of vanilla GRPO in exploitation: it fails to consistently apply basic numerical commonsense—namely, that larger absolute values correspond to smaller magnitudes for negative numbers. By contrast, introducing the Auxiliary Shaping Term strengthens exploitation capacity, enabling the model to make such comparisons more reliably.

1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966**Question**

What is the smallest possible integer value of  $x$  if  $12 \div (x + 1)$  is an integer?

**Ground Truth**

-13

**Vanilla GRPO**1968  
1969  
1970  
1971  
1972  
1973

To find the smallest possible integer value of  $x$  such that  $\frac{12}{x+1}$  is an integer, we need to consider the factors of 12. The expression  $\frac{12}{x+1}$  will be an integer if  $x + 1$  is a divisor of 12. Let's list the divisors of 12:

1. The divisors of 12 are:  $\pm 1, \pm 2, \pm 3, \pm 4, \pm 6, \pm 12$ .
2. We need to find the smallest  $x$ , so we start with the smallest positive divisor, which is 1:

$$x + 1 = 1 \implies x = 1 - 1 \implies x = 0$$

3. To confirm, we substitute  $x = 0$  back into the original expression:

$$\frac{12}{0+1} = \frac{12}{1} = 12$$

Since 12 is an integer,  $x = 0$  is indeed a valid solution.

4. We should also consider the negative divisors to ensure we have the smallest possible integer value. The next smallest divisor is -1:

$$x + 1 = -1 \implies x = -1 - 1 \implies x = -2$$

5. Substituting  $x = -2$  back into the original expression:

$$\frac{12}{-2+1} = \frac{12}{-1} = -12$$

Since -12 is an integer,  $x = -2$  is also a valid solution, but it is not smaller than  $x = 0$ .

Therefore, the smallest possible integer value of  $x$  is

0

1984

1985

1986

1987

**GRPO with Auxiliary Shaping Term**

To find the smallest possible integer value of  $x$  such that  $\frac{12}{x+1}$  is an integer, we need to determine the divisors of 12. The divisors of 12 are  $\pm 1, \pm 2, \pm 3, \pm 4, \pm 6, \pm 12$ . For  $\frac{12}{x+1}$  to be an integer,  $x + 1$  must be one of these divisors. We want the smallest possible  $x$ , so we should consider the smallest positive divisor, which is 1.

If  $x + 1 = 1$ , then  $x = 0$ .

We should also consider the negative divisors to ensure we find the smallest possible integer value of  $x$ . The smallest negative divisor is -12. If  $x + 1 = -12$ , then  $x = -13$ .

Comparing  $x = 0$  and  $x = -13$ , the smallest possible integer value of  $x$  is -13.

Therefore, the final answer is

-13

1995  
1996  
1997

Figure 11: Case study: Vanilla GRPO fails to exploit numerical commonsense when comparing negative values (-1 vs. -13), while our method gives the correct answer.

1998 **Case Study II.** For the case in Fig. 12, the crucial distinction lies in how the models interpret the  
 1999 condition “No house faces two different streets.” The vanilla GRPO-trained model incorrectly infers  
 2000 that each street contains only 10 houses, overlooking the structural implication of the condition. In  
 2001 contrast, the model trained with GRPO augmented by the auxiliary shaping term exploits this com-  
 2002 monsense constraint more effectively: it decomposes the problem step by step, correctly deduces  
 2003 that each street has 20 houses, and thereby arrives at the correct total.

2004 **Question**

2005 In my neighborhood, there are six streets. There are 10 houses on each side of each street. No house faces two different streets.  
 2006 How many houses are in my neighborhood?

2008 **Ground Truth**

2009 120

2010 **Vanilla GRPO**

2011 To determine the total number of houses in the neighborhood, we can follow these steps:

1. **Identify the number of streets and houses per street:** - There are 6 streets. - Each street has 10 houses on each side.
2. **Calculate the number of houses on one street:** Since there are 10 houses on each side of a street, and each house faces only one street, the total number of houses on one street is 10 (since they are all on the same street).
3. **Calculate the total number of houses in the neighborhood:** There are 6 streets, and each street has 10 houses. Therefore, the total number of houses is  $6 \times 10 = 60$ .

2012 Putting the final answer within  we get:

2013 60

2014 **GRPO with Auxiliary Shaping Term**

2015 To find the total number of houses in the neighborhood, let's break down the problem step by step.

1. There are 6 streets in the neighborhood.
2. Each street has houses on both sides.
3. There are 10 houses on each side of every street.
4. No house faces two different streets, meaning each house is counted only once.

2016 First, calculate the number of houses on one side of a single street:

2017 10 houses/side

2018 Since there are houses on both sides of each street, multiply by 2:

$$2019 \quad 10 \text{ houses/side} \times 2 \text{ sides/street} = 20 \text{ houses/street}$$

2020 Now, since there are 6 streets, multiply the number of houses per street by the number of streets:

$$2021 \quad 20 \text{ houses/street} \times 6 \text{ streets} = 120 \text{ houses}$$

2022 So, the total number of houses in the neighborhood is

2023 120

2024 Figure 12: Case study: Vanilla GRPO overlooks the constraint and infers 10 houses per street, while  
 2025 our method exploits it correctly and infers 20.

2026 **Case Study III.** As shown in Fig. 13, the task concerns solving a differential equation under the  
 2027 constraint of finding a purely exponential solution rather than a general one. The model trained  
 2028 with vanilla GRPO overlooks this constraint and defaults to the generic procedure for obtaining the  
 2029 full solution. In contrast, the model trained with GRPO augmented by the auxiliary shaping term  
 2030 respects the constraint and produces the correct purely exponential answer. This case illustrates  
 2031 how the auxiliary shaping term enhances exploitation by enforcing adherence to problem-specific  
 2032 constraints.

2033 **I.2 CASE STUDY FOR PASS@16 SETTING**

2034 **Case Study I.** As shown in Fig. 14, the question is a simple elementary-level math problem re-  
 2035 quiring the integer solution  $x \geq 1, x \neq 2$ . While the vanilla GRPO model consistently outputs 3,  
 2036 which is incorrect, the model trained with the auxiliary shaping term produces 1 in 5 out of 16 roll-  
 2037 outs, the correct answer. This case highlights that the auxiliary shaping term enhances the model's  
 2038 exploration ability, enabling it to consider a wider range of possibly valid solutions.

2052  
2053  
2054  
2055  
2056**Question**2057  
2058 Find a purely exponential solution of  $\frac{d^4 x}{dt^4} - x = e^{-2t}$ .  
2059

2060

2061  
2062  
2063**Ground Truth** $\frac{e^{-2t}}{15}$ 2064  
2065**Vanilla GRPO**2066 To find a purely exponential solution of the differential equation ( $\frac{d^4 x}{dt^4} - x = e^{-2t}$ ), we will follow these steps:  
2067**1. Find the complementary solution by solving the homogeneous equation:**

2068  
2069 
$$\frac{d^4 x}{dt^4} - x = 0$$

2070 The characteristic equation is:  
2071

$$r^4 - 1 = 0$$

2072 Solving for  $r$ , we get:  
2073

$$r^4 = 1 \implies r = \pm 1, \pm i$$

2074 Therefore, the complementary solution is:  
2075

$$x_c(t) = C_1 e^t + C_2 e^{-t} + C_3 \cos(t) + C_4 \sin(t)$$

**2. Find a particular solution using the method of undetermined coefficients:**2076 Since the right-hand side of the non-homogeneous equation is  $e^{-2t}$ , we assume a particular solution of the form:  
2077

2078  
2079 
$$x_p(t) = A e^{-2t}$$

2080 Taking the fourth derivative of  $x_p(t)$ :  
2081

$$\frac{d^4}{dt^4}(A e^{-2t}) = A(-2)^4 e^{-2t} = 16 A e^{-2t}$$

2083 Substituting  $x_p(t)$  and its fourth derivative into the original differential equation:  
2084

$$16 A e^{-2t} - A e^{-2t} = e^{-2t}$$

2086 Simplifying, we get:  
2087

$$(16A - A) e^{-2t} = e^{-2t} \implies 15A e^{-2t} = e^{-2t}$$

2088 Therefore:  
2089

$$15A = 1 \implies A = \frac{1}{15}$$

2090 So, the particular solution is:  
2091

$$x_p(t) = \frac{1}{15} e^{-2t}$$

**3. Combine the complementary and particular solutions:**2092 The general solution to the differential equation is:  
2093

2094  
2095 
$$x(t) = x_c(t) + x_p(t) = C_1 e^t + C_2 e^{-t} + C_3 \cos(t) + C_4 \sin(t) + \frac{1}{15} e^{-2t}$$

2096 Thus, the final answer is:  
2097

$$C_1 e^t + C_2 e^{-t} + C_3 \cos(t) + C_4 \sin(t) + \frac{1}{15} e^{-2t}$$

2099  
2100  
2101  
2102  
2103  
2104  
2105

Figure 13: Case study: Vanilla GRPO ignores the constraint of a purely exponential solution, while our method enforces it and yields the correct result.

2106

2107

2108

**GRPO with Auxiliary Shaping Term**

2109

To find a purely exponential solution to the differential equation

2110

2111

$$\frac{d^4 x}{dt^4} - x = e^{-2t}$$

2112

, we will follow these steps:

2113

1. **Find the complementary solution by solving the homogeneous equation:**

2114

2115

$$\frac{d^4 x}{dt^4} - x = 0$$

2116

The characteristic equation is:

2117

Solving for  $r$ , we get:

2118

$$r^4 - 1 = 0$$

2119

Therefore, the complementary solution is:

2120

2121

$$x_c(t) = C_1 e^t + C_2 e^{-t} + C_3 \cos(t) + C_4 \sin(t)$$

2122

2123

2. **Find a particular solution using the method of undetermined coefficients:** Since the right-hand side of the non-homogeneous equation is  $e^{-2t}$ , we assume a particular solution of the form:

2124

2125

$$x_p(t) = A e^{-2t}$$

2126

Taking the fourth derivative of  $x_p(t)$ :

2127

$$\frac{d^4 x_p}{dt^4} = 16 A e^{-2t}$$

2128

Substituting  $x_p(t)$  and its fourth derivative into the original differential equation:

2129

2130

$$16 A e^{-2t} - A e^{-2t} = e^{-2t}$$

2131

Simplifying, we get:

2132

$$15 A e^{-2t} = e^{-2t}$$

2133

Solving for  $A$ :

2134

$$15 A = 1 \implies A = \frac{1}{15}$$

2135

Therefore, the particular solution is:

2136

$$x_p(t) = \frac{1}{15} e^{-2t}$$

2137

3. **Combine the complementary and particular solutions:** The general solution to the differential equation is:

2138

2139

$$x(t) = x_c(t) + x_p(t) = C_1 e^t + C_2 e^{-t} + C_3 \cos(t) + C_4 \sin(t) + \frac{1}{15} e^{-2t}$$

2140

Thus, the purely exponential solution is:

2141

2142

$$\boxed{\frac{1}{15} e^{-2t}}$$

2143

2144  
2145  
Figure 13: Case study: Vanilla GRPO ignores the constraint of a purely exponential solution, while  
our method enforces it and yields the correct result (continued).

2146

2147

2148

2149

**Question**

2150

Given  $f(x) = \frac{\sqrt{x-1}}{x-2}$ , what is the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value?

2151

2152

**Ground Truth**

2153

2154

1

2155

2156

2157  
2158  
Figure 14: Case study demonstrating improved exploration: the auxiliary shaping term enables the  
model to explore and reach the correct solution.

2159

2160

2161

2162

2163

2164

2165

2166

2167

2168

2169

2170

2171

2172

2173

2174

2175

2176

2177

2178

2179

2180

2181

2182

2183

2184

2185

2186

2187

2188

2189

2190

2191

2192

2193

2194

2195

2196

2197

2198

2199

2200

2201

2202

2203

2204

2205

2206

2207

2208

2209

2210

2211

2212

2213

**Vallina GRPO#1-#8**

(Details omitted as they are not crucial for this case study.)

- Numerator Condition:** The numerator is  $\sqrt{x-1}$ . For the square root to be defined and result in a real number, the expression inside the square root must be non-negative.

$$x - 1 \geq 0 \implies x \geq 1$$

- Denominator Condition:** The denominator is  $x - 2$ . For the function to be defined, the denominator must not be zero.

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these two conditions, we get:

$$x \geq 1 \text{ and } x \neq 2$$

The smallest integer that satisfies  $x \geq 1$  and  $x \neq 2$  is  $x = 3$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is  $\boxed{3}$ .

To determine the smallest possible integer value for  $x$  such that  $f(x) = \frac{\sqrt{x-1}}{x-2}$  has a real number value, we need to ensure that both the numerator and the denominator satisfy their respective conditions for real numbers.

- Numerator Condition:** The expression inside the square root must be non-negative.

$$x - 1 \geq 0 \implies x \geq 1$$

- Denominator Condition:** The denominator must not be zero to avoid division by zero.

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these conditions, we need  $x \geq 1$  and  $x \neq 2$ . The smallest integer that satisfies these conditions is  $x = 3$ . Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is  $\boxed{3}$ .

To determine the smallest possible integer value for  $x$  such that  $f(x) = \frac{\sqrt{x-1}}{x-2}$  has a real number value, we need to ensure that both the numerator and the denominator satisfy certain conditions.

- Numerator Condition:** The expression inside the square root must be non-negative because the square root of a negative number is not a real number.

$$x - 1 \geq 0 \implies x \geq 1$$

- Denominator Condition:** The denominator must not be zero because division by zero is undefined.

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these conditions, we get:

$$x \geq 1 \text{ and } x \neq 2$$

The smallest integer that satisfies  $x \geq 1$  and  $x \neq 2$  is  $x = 3$ .

(Details omitted as they are not crucial for this case study.)

- Square Root Condition:** The expression inside the square root,  $x - 1$ , must be greater than or equal to zero for the square root to be a real number.

$$x - 1 \geq 0 \implies x \geq 1$$

- Denominator Condition:** The denominator,  $x - 2$ , must not be zero to avoid division by zero.

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these two conditions, we get:

$$x \geq 1 \text{ and } x \neq 2$$

The smallest integer that satisfies  $x \geq 1$  and  $x \neq 2$  is  $x = 3$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is  $\boxed{3}$ .

(Details omitted as they are not crucial for this case study.)

- For the square root to be defined, the expression inside it must be greater than or equal to zero:

$$x - 1 \geq 0$$

Solving this inequality:

$$x \geq 1$$

- For the denominator to be non-zero:

$$x - 2 \neq 0$$

Solving this equation:

$$x \neq 2$$

(Details omitted as they are not crucial for this case study.)

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is:

$$\boxed{3}$$

(Details omitted as they are not crucial for this case study.)

- For the square root to be defined, the expression inside it must be greater than or equal to zero:

$$x - 1 \geq 0$$

Solving this inequality:

$$x \geq 1$$

- For the function to be defined, the denominator must not be zero:

$$x - 2 \neq 0$$

Solving this equation:

$$x \neq 2$$

(Details omitted as they are not crucial for this case study.)

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is  $\boxed{3}$ .

(Details omitted as they are not crucial for this case study.)

- For the square root to be defined, we need:

$$x - 1 \geq 0$$

Solving this inequality:

$$x \geq 1$$

- For the denominator to be non-zero, we need:

$$x - 2 \neq 0$$

Solving this equation:

$$x \neq 2$$

Combining these two conditions,  $x$  must be greater than or equal to 1 but not equal to 2. The smallest integer that satisfies this condition is  $x = 3$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is  $\boxed{3}$ .

Figure 14: Case study demonstrating improved exploration: the auxiliary shaping term enables the model to explore and reach the correct solution (continued).

2214

2215

2216

2217

2218

2219

2220

2221

2222

2223

2224

2225

2226

2227

2228

2229

2230

2231

2232

2233

2234

2235

2236

2237

2238

2239

2240

2241

2242

2243

2244

2245

2246

2247

2248

2249

2250

2251

2252

2253

2254

2255

2256

2257

2258

2259

2260

2261

2262

2263

2264

2265

2266

2267

**Vanilla GRPO#9-#16***(Details omitted as they are not crucial for this case study.)*

**1. Numerator Condition:** The expression inside the square root,  $x - 1$ , must be non-negative because the square root of a negative number is not a real number. Therefore, we need:

$$x - 1 \geq 0 \implies x \geq 1$$

**2. Denominator Condition:** The denominator,  $x - 2$ , must not be zero because division by zero is undefined. Therefore, we need:

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these two conditions, we get:

$$x \geq 1 \text{ and } x \neq 2$$

The smallest integer that satisfies  $x \geq 1$  and  $x \neq 2$  is  $x = 3$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is  $\boxed{3}$ .

*(Details omitted as they are not crucial for this case study.)*

**1. Numerator Condition:** The expression inside the square root,  $x - 1$ , must be non-negative because the square root of a negative number is not a real number. Therefore, we need:

$$x - 1 \geq 0 \implies x \geq 1$$

**2. Denominator Condition:** The denominator,  $x - 2$ , must not be zero because division by zero is undefined. Therefore, we need:

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these two conditions, we get:

$$x \geq 1 \text{ and } x \neq 2$$

The smallest integer that satisfies  $x \geq 1$  and  $x \neq 2$  is  $x = 3$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is  $\boxed{3}$ .

*(Details omitted as they are not crucial for this case study.)*

1. For the square root to be defined, we need:

$$x - 1 \geq 0$$

$$x \geq 1$$

2. For the denominator to be non-zero, we need:

$$x - 2 \neq 0$$

$$x \neq 2$$

Combining these two conditions, we get:

$$x \geq 1$$

$$x \neq 2$$

The smallest integer that satisfies  $x \geq 1$  and  $x \neq 2$  is  $x = 3$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is  $\boxed{3}$ .

*(Details omitted as they are not crucial for this case study.)*

**1. Numerator Condition:** The expression under the square root,  $x - 1$ , must be non-negative because the square root of a negative number is not a real number. Therefore, we need:

$$x - 1 \geq 0 \implies x \geq 1$$

**2. Denominator Condition:** The denominator,  $x - 2$ , must not be zero because division by zero is undefined. Therefore, we need:

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these two conditions, we get:

$$x \geq 1 \text{ and } x \neq 2$$

The smallest integer that satisfies  $x \geq 1$  and  $x \neq 2$  is  $x = 3$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is  $\boxed{3}$ .

*(Details omitted as they are not crucial for this case study.)*

**1. Numerator Condition:** The numerator is  $\sqrt{x - 1}$ . For the square root to be defined and result in a real number, the expression inside the square root must be non-negative.

$$x - 1 \geq 0 \implies x \geq 1$$

**2. Denominator Condition:** The denominator is  $x - 2$ . For the function to be defined, the denominator must not be zero.

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these two conditions, we need  $x$  to be greater than or equal to 1, but not equal to 2. Therefore, the smallest possible integer value for  $x$  that satisfies both conditions is 3.

$$\boxed{3}$$

*(Details omitted as they are not crucial for this case study.)*

**1. Square Root Condition:** The expression inside the square root,  $x - 1$ , must be greater than or equal to zero for the square root to be a real number.

$$x - 1 \geq 0 \implies x \geq 1$$

**2. Denominator Condition:** The denominator,  $x - 2$ , must not be zero to avoid division by zero.

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these two conditions, we get:

$$x \geq 1 \text{ and } x \neq 2$$

The smallest integer that satisfies  $x \geq 1$  and  $x \neq 2$  is  $x = 3$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is  $\boxed{3}$ .

To determine the smallest possible integer value for  $x$  such that  $f(x) = \frac{\sqrt{x-1}}{x-2}$  has a real number value, we need to ensure that both the numerator and the denominator satisfy their respective conditions for real numbers.

**1. Numerator Condition:** The expression inside the square root must be non-negative.

$$x - 1 \geq 0 \implies x \geq 1$$

**2. Denominator Condition:** The denominator must not be zero.

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these conditions, we need  $x$  to be

greater than or equal to 1 but not equal to 2. Therefore, the smallest possible integer value for  $x$  that satisfies both conditions is 3.

$$\boxed{3}$$

*(Details omitted as they are not crucial for this case study.)*

**1. Numerator Condition:** The numerator is  $\sqrt{x - 1}$ . For the square root to be a real number, the expression inside the square root must be non-negative.

$$x - 1 \geq 0 \implies x \geq 1$$

**2. Denominator Condition:** The denominator is  $x - 2$ . For the function to be defined, the denominator must not be zero.

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these two conditions, we get:

$$x \geq 1 \text{ and } x \neq 2$$

The smallest integer that satisfies  $x \geq 1$  and  $x \neq 2$  is  $x = 3$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is  $\boxed{3}$ .

Figure 14: Case study demonstrating improved exploration: the auxiliary shaping term enables the model to explore and reach the correct solution (continued).

2268

2269 **GRPO with Auxiliary Shaping Term#1-#8**

2270

2271

2272

2273 *(Details omitted as they are not crucial for this case study.)*

2274 1. **Numerator Condition:** The numerator is  $\sqrt{x-1}$ . For the square root to be defined and real, the expression inside the square root must be non-negative.

2275  $x-1 \geq 0 \implies x \geq 1$

2276

2277 2. **Denominator Condition:** The denominator is  $x-2$ . For the function to be defined, the denominator must not be zero.

2278  $x-2 \neq 0 \implies x \neq 2$

2279

2280 Combining these conditions, we need  $x$  to be greater than or equal to 1, but not equal to 2. The smallest integer that satisfies these conditions is  $x = 3$ .

2281 Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is 3.

2282

2283

2284

2285

2286

2287

2288

2289

2290

2291

2292

2293

2294

2295

2296 *(Details omitted as they are not crucial for this case study.)*

2297 1. **Numerator Condition:** The expression under the square root,  $x-1$ , must be non-negative because the square root of a negative number is not a real number. Therefore, we need:

2298  $x-1 \geq 0 \implies x \geq 1$

2299

2300 2. **Denominator Condition:** The denominator,  $x-2$ , must not be zero because division by zero is undefined. Therefore, we need:

2301  $x-2 \neq 0 \implies x \neq 2$

2302

2303 Combining these two conditions, we get:

2304  $x \geq 1 \text{ and } x \neq 2$

2305

2306 The smallest integer that satisfies  $x \geq 1$  but not  $x = 2$  is  $x = 1$ .

2307 Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is 1.

2308

2309

2310

2311

2312

2313

2314

2315

2316

2317

2318

2319

2320

2321

(*Details omitted as they are not crucial for this case study.*)

1. For the square root to be defined in the real numbers, the expression inside it must be greater than or equal to zero:

$$x-1 \geq 0$$

Solving this inequality:

$$x \geq 1$$

2. For the function to be defined, the denominator must not be zero:

$$x-2 \neq 0$$

Solving this equation:

$$x \neq 2$$

(*Details omitted as they are not crucial for this case study.*)

Thus, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is 3.

(*Details omitted as they are not crucial for this case study.*)

1. The expression inside the square root must be non-negative. 2. The denominator must not be zero.

(*Details omitted as they are not crucial for this case study.*)

### Step 3: Combine the conditions From Step 1, we know  $x \geq 1$ . From Step 2, we know  $x \neq 2$ . Therefore, the smallest integer value for  $x$  that satisfies both conditions is the smallest integer greater than 1, which is 3.

### Final Answer The smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is 3.

(*Details omitted as they are not crucial for this case study.*)

1. For  $\sqrt{x-1}$  to be defined, we need:

$$x-1 \geq 0 \implies x \geq 1$$

2. For the denominator  $x-2$  to be non-zero, we need:

$$x-2 \neq 0 \implies x \neq 2$$

Combining these two conditions, we get:

$$x \geq 1 \text{ and } x \neq 2$$

The smallest integer that satisfies  $x \geq 1$  and  $x \neq 2$  is  $x = 1$ .

(*Details omitted as they are not crucial for this case study.*)

Since  $f(1) = 0$  is a real number,  $x = 1$  is indeed a valid solution.

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is 1.

(*Details omitted as they are not crucial for this case study.*)

1. For  $\sqrt{x-1}$  to be defined in the real numbers, we need:

$$x-1 \geq 0 \implies x \geq 1$$

2. For the denominator  $x-2$  to not be zero, we need:

$$x-2 \neq 0 \implies x \neq 2$$

Combining these two conditions,  $x$  must be greater than or equal to 1 but not equal to 2. The smallest integer that satisfies this condition is  $x = 3$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is 3.

Figure 14: Case study demonstrating improved exploration: the auxiliary shaping term enables the model to explore and reach the correct solution (continued).

2322

2323

2324

2325

2326

2327

2328

2329

2330

2331

2332

2333

2334

2335

2336

2337

2338

2339

2340

2341

2342

2343

2344

2345

2346

2347

2348

2349

2350

2351

2352

2353

2354

2355

2356

2357

2358

2359

2360

2361

2362

2363

2364

2365

2366

2367

2368

2369

2370

2371

2372

2373

2374

2375

### GRPO with Auxiliary Shaping Term#9-#16

(Details omitted as they are not crucial for this case study.)

1. **Numerator Condition:** The expression inside the square root,  $x - 1$ , must be non-negative because the square root of a negative number is not a real number. Therefore, we need:

$$x - 1 \geq 0 \implies x \geq 1$$

2. **Denominator Condition:** The denominator,  $x - 2$ , must not be zero because division by zero is undefined. Therefore, we need:

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these two conditions, we get:

$$x \geq 1 \text{ and } x \neq 2$$

The smallest integer that satisfies  $x \geq 1$  and  $x \neq 2$  is  $x = 1$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is 1.

(Details omitted as they are not crucial for this case study.)

1. **Numerator Condition:** The expression under the square root must be non-negative.

$$x - 1 \geq 0 \implies x \geq 1$$

2. **Denominator Condition:** The denominator must not be zero.

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these two conditions, we get:

$$x \geq 1 \text{ and } x \neq 2$$

The smallest integer that satisfies  $x \geq 1$  and  $x \neq 2$  is  $x = 3$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is 3.

(Details omitted as they are not crucial for this case study.)

1. **Numerator Condition:** The expression inside the square root must be non-negative.

$$x - 1 \geq 0 \implies x \geq 1$$

This ensures that  $\sqrt{x - 1}$  is a real number.

2. **Denominator Condition:** The denominator must not be zero.

$$x - 2 \neq 0 \implies x \neq 2$$

This ensures that the function is defined.

Combining these two conditions, we get:

$$x \geq 1 \text{ and } x \neq 2$$

The smallest integer that satisfies  $x \geq 1$  and  $x \neq 2$  is  $x = 3$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is 3.

(Details omitted as they are not crucial for this case study.)

1. For the square root to be defined, we need:

$$x - 1 \geq 0$$

Solving this inequality:

$$x \geq 1$$

2. For the denominator to be non-zero, we need:

$$x - 2 \neq 0$$

Solving this equation:

$$x \neq 2$$

Combining these two conditions,  $x$  must be greater than or equal to 1 but not equal to 2. The smallest integer that satisfies this condition is  $x = 3$ . Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is 3.

(Details omitted as they are not crucial for this case study.)

1. The expression inside the square root must be non-negative. 2. The denominator must not be zero.

Let's address these conditions one by one:

1. For the square root to be defined, the expression inside it must be greater than or equal to zero:

$$x - 1 \geq 0 \implies x \geq 1$$

2. For the denominator to be non-zero:

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these conditions, we need  $x$  to be greater than or equal to 1 but not equal to 2. The smallest integer that satisfies this is  $x = 3$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is 3.

(Details omitted as they are not crucial for this case study.)

1. **Numerator Condition:** The expression inside the square root must be non-negative because the square root of a negative number is not a real number.

$$x - 1 \geq 0 \implies x \geq 1$$

2. **Denominator Condition:** The denominator must not be zero because division by zero is undefined.

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these two conditions, we get:

$$x \geq 1 \text{ and } x \neq 2$$

The smallest integer that satisfies  $x \geq 1$  and  $x \neq 2$  is  $x = 3$ .

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is 3.

(Details omitted as they are not crucial for this case study.)

1. **Numerator Condition:** The expression under the square root,  $x - 1$ , must be non-negative because the square root of a negative number is not a real number.

$$x - 1 \geq 0 \implies x \geq 1$$

2. **Denominator Condition:** The denominator,  $x - 2$ , must not be zero because division by zero is undefined.

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these conditions, we need  $x \geq 1$  and  $x \neq 2$ . The smallest integer that satisfies these conditions is  $x = 1$ .

(Details omitted as they are not crucial for this case study.)

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is 1.

To determine the smallest possible integer value for  $x$  such that  $f(x) = \frac{\sqrt{x-1}}{x-2}$  has a real number value, we need to ensure that both the numerator and the denominator satisfy their respective conditions for real numbers.

1. **Numerator Condition:** The expression inside the square root must be non-negative.

$$x - 1 \geq 0 \implies x \geq 1$$

2. **Denominator Condition:** The denominator,  $x - 2$ , must not be zero because division by zero is undefined.

$$x - 2 \neq 0 \implies x \neq 2$$

Combining these conditions, we need  $x \geq 1$  and  $x \neq 2$ . The smallest integer that satisfies these conditions is  $x = 3$ .

(Details omitted as they are not crucial for this case study.)

Therefore, the smallest possible integer value for  $x$  such that  $f(x)$  has a real number value is 3.

Figure 14: Case study demonstrating improved exploration: the auxiliary shaping term enables the model to explore and reach the correct solution (continued).

2376     **Case Study II.** As shown in Fig. 15, this high-school level geometry problem illustrates the con-  
2377     trast between vanilla GRPO and GRPO with the auxiliary shaping term. The vanilla GRPO model  
2378     fails to produce the correct answer in any rollout. In contrast, the model trained with the auxiliary  
2379     shaping term succeeds in 5 out of 16 rollouts. Moreover, its reasoning traces display higher ex-  
2380     ploratory diversity: it applies the law of sines 9 times, the law of cosines 4 times, and other general  
2381     geometric properties 3 times. By comparison, the vanilla GRPO model relies more narrowly on  
2382     the law of sines (11 times), with only limited use of other strategies. This case highlights that the  
2383     auxiliary shaping term enhances the model’s exploration capacity, enabling it to leverage a broader  
2384     range of problem-solving strategies rather than over-committing to a single path.

2385

2386

2387

2388

2389

2390

2391

2392

2393

2394

2395

2396

2397

2398

2399

2400

2401

2402

2403

2404

2405

2406

2407

2408

2409

2410

2411

2412

2413

2414

2415

2416

2417

2418

2419

2420

2421

2422

2423

2424

2425

2426

2427

2428

2429

2430

2431

2432

2433

2434

2435

2436

2437

2438

2439

2440

2441

2442

2443

2444

2445

2446

2447

2448

2449

2450

2451

2452

2453

2454

2455

2456

2457

2458

2459

2460

2461

2462

2463

2464

2465

2466

2467

2468

2469

2470

2471

2472

2473

2474

2475

2476

2477

2478

2479

2480

2481

2482

2483

**Question**

Abdul and Chiang are standing 48 feet apart in a field. Bharat is standing in the same field as far from Abdul as possible so that the angle formed by his lines of sight to Abdul and Chiang measures  $60^\circ$ . What is the square of the distance (in feet) between Abdul and Bharat?

**Ground Truth**

3072

**Vanilla GRPO#1-#8**

(Details omitted as they are not crucial for this case study.)

1. **Identify the triangle and its properties:**  
(Details omitted as they are not crucial for this case study.)

2. **Use the Law of Sines:**  
(Details omitted as they are not crucial for this case study.)

3. **Apply the Law of Sines:**  
(Details omitted as they are not crucial for this case study.)

4. **Find the square of the distance:**  
(Details omitted as they are not crucial for this case study.)

Therefore, the square of the distance between Abdul and Bharat is 2304.

To solve this problem, we can use the Law of Sines in trigonometry. Let's denote the distance between Abdul and Bharat as  $x$  and the distance between Chiang and Bharat as  $y$ . We know that the angle between Abdul and Chiang as seen from Bharat is  $60^\circ$ , and the distance between Abdul and Chiang is 48 feet. Given:  $\angle ABC = 60^\circ$  -  $AB = 48$  feet - We need to find  $x^2$ . First, we apply the Law of Sines in triangle  $ABC$ :  
(Details omitted as they are not crucial for this case study.) Thus, the square of the distance between Abdul and Bharat is 9216.

To solve this problem, we can use the Law of Sines in trigonometry. Let's denote the distance between Abdul and Bharat as  $AB$ , and the distance between Chiang and Bharat as  $BC$ . Since Bharat is as far from Abdul as possible to form a  $60^\circ$  angle, triangle  $ABC$  is a right triangle with the right angle at Chiang. This means  $\angle ACB = 90^\circ$  and  $\angle BAC = 60^\circ$ .  
(Details omitted as they are not crucial for this case study.) Thus, the square of the distance between Abdul and Bharat is 1728. To verify the accuracy of this result, we can use Python to perform the calculations.  
(Details omitted as they are not crucial for this case study.)

To solve this problem, we can use the Law of Sines and some geometric properties. Let's break it down step by step.

1. **Identify the triangle and its properties:**  
(Details omitted as they are not crucial for this case study.)

2. **Apply the Law of Sines:**  
(Details omitted as they are not crucial for this case study.)

3. **Calculate the distances:**  
(Details omitted as they are not crucial for this case study.)

4. **Find the square of the distance:**  
(Details omitted as they are not crucial for this case study.)

So, the square of the distance between Abdul and Bharat is 768.

To solve this problem, we need to use some geometric principles, specifically involving the Law of Sines and properties of triangles.

1. **Identify the triangle and the given information:**  
(Details omitted as they are not crucial for this case study.)

2. **Use the Law of Sines to find the distance  $BC$  (the distance from Bharat to Chiang):**  
(Details omitted as they are not crucial for this case study.)

3. **Find the distance  $AC$  (the distance from Abdul to Bharat):**  
(Details omitted as they are not crucial for this case study.)

4. **Apply the Pythagorean theorem:**  
(Details omitted as they are not crucial for this case study.)

Thus, the square of the distance between Abdul and Bharat is 11520.

Figure 15: Case study: Geometry problem. GRPO with auxiliary shaping term not only achieves higher accuracy but also explores more diverse reasoning strategies than vanilla GRPO.

2484  
2485  
2486  
2487  
2488  
2489  
2490  
2491  
2492  
2493  
2494  
2495  
2496  
2497  
2498  
2499  
2500  
2501  
2502  
2503  
2504  
2505  
2506  
2507  
2508  
2509  
2510  
2511  
2512  
2513  
2514  
2515  
2516  
2517  
2518  
2519  
2520  
2521  
2522  
2523  
2524  
2525  
2526  
2527  
2528  
2529  
2530  
2531  
2532  
2533  
2534  
2535  
2536  
2537

### Vanilla GRPO#9-#16

To solve this problem, we can use the Law of Sines and some geometric properties. Let's break it down step by step.

- 1. Identify the triangle and the given information:**  
(Details omitted as they are not crucial for this case study.)
- 2. Determine the largest possible angle at A:**  
(Details omitted as they are not crucial for this case study.)
- 3. Use the Pythagorean theorem:**  
(Details omitted as they are not crucial for this case study.)
- 4. Find the square of the distance between Abdul and Bharat:**  
(Details omitted as they are not crucial for this case study.)

So, the square of the distance between Abdul and Bharat is 9216.

To solve this problem, we can use some geometric properties and trigonometry. Let's break it down step by step:

- 1. Identify the triangle formed:**  
(Details omitted as they are not crucial for this case study.)
- 2. Use the Law of Sines:**  
(Details omitted as they are not crucial for this case study.)
- 3. Maximize the distance:**  
(Details omitted as they are not crucial for this case study.)
- 4. Square the distance:**  
(Details omitted as they are not crucial for this case study.)

Let's calculate it:

$$48^2 = 2304$$

So, the square of the distance between Abdul and Bharat is 2304.

To solve the problem, we need to determine the distance between Abdul and Bharat, given that the angle between their lines of sight to Chiang is  $60^\circ$  and that Bharat is positioned to be as far from Abdul as possible.

(Details omitted as they are not crucial for this case study.)

Using the Law of Cosines in triangle  $ABC$ :

(Details omitted as they are not crucial for this case study.)

$$BC^2 = d^2 + 2304 - 48d$$

(Details omitted as they are not crucial for this case study.)

The square of the distance between Abdul and Bharat is:

$$d^2 = 96^2 = 9216$$

Thus, the square of the distance between Abdul and Bharat is 9216.

(Details omitted as they are not crucial for this case study.)

#### 1. Identify the known values:

(Details omitted as they are not crucial for this case study.)

#### 2. Set up the triangle:

(Details omitted as they are not crucial for this case study.)

#### 3. Apply the Law of Sines:

(Details omitted as they are not crucial for this case study.)

#### 4. Determine the angles:

(Details omitted as they are not crucial for this case study.)

#### 5. Simplify the Law of Sines equation:

(Details omitted as they are not crucial for this case study.)

Therefore, the square of the distance between Abdul and Bharat is 2304.

To solve this problem, we need to use some geometric properties and trigonometry. Let's break it down step by step.

- 1. Understanding the Problem:**  
(Details omitted as they are not crucial for this case study.)
- 2. Geometric Insight:**  
(Details omitted as they are not crucial for this case study.)
- 3. Using Trigonometry:**  
(Details omitted as they are not crucial for this case study.)
- 4. Calculating the Distance:** - Let the distance AB be  $x$ . Since  $\angle ABC = 60^\circ$  and we need to maximize AB, we use the Law of Cosines in triangle ABC:  
(Details omitted as they are not crucial for this case study.)

Therefore, the square of the distance between Abdul and Bharat is 432.

(Details omitted as they are not crucial for this case study.)

- 1. Identify the given information:**  
(Details omitted as they are not crucial for this case study.)
- 2. Visualize the problem:**  
(Details omitted as they are not crucial for this case study.)
- 3. Use the Law of Sines:**  
(Details omitted as they are not crucial for this case study.)
- 4. Properties of an equilateral triangle:**  
(Details omitted as they are not crucial for this case study.)
- 5. Calculate the square of the distance:**  
(Details omitted as they are not crucial for this case study.)

Thus, the square of the distance between Abdul and Bharat is 2304.  
Provide the final answer within  .

To solve this problem, we need to use some geometric properties and trigonometry. Let's break it down step by step.

#### 1. Understand the Problem:

(Details omitted as they are not crucial for this case study.)

#### 2. Visualize the Problem:

(Details omitted as they are not crucial for this case study.)

#### 3. Use Geometric Properties:

(Details omitted as they are not crucial for this case study.)

#### 4. Apply Trigonometry:

(Details omitted as they are not crucial for this case study.)

#### 5. Calculate the Distance:

$- AB = 2 * AC = 2 * 48 = 96$  feet. - The square of the distance AB is  $96^2$ .

#### 6. Final Calculation:

$- 96^2 = 9216$ .  
So, the square of the distance between Abdul and Bharat is 9216.

To solve this problem, we can use the Law of Sines in trigonometry. Let's break it down step by step:

#### 1. Identify the given information:

(Details omitted as they are not crucial for this case study.)

#### 2. Set up the triangle:

(Details omitted as they are not crucial for this case study.)

#### 3. Use the Law of Sines:

(Details omitted as they are not crucial for this case study.)

#### 4. Find the square of the distance:

The distance between Abdul and Bharat is 24 feet. The square of this distance is:

$$24^2 = 576$$

Therefore, the square of the distance between Abdul and Bharat is 576.

Figure 15: Case study: Geometry problem. GRPO with auxiliary shaping term not only achieves higher accuracy but also explores more diverse reasoning strategies than vanilla GRPO (continued).

2538  
2539  
2540

## GRPO with Auxiliary Shaping Term#1-#8

2541  
2542  
2543  
2544  
2545  
2546  
2547  
2548  
2549  
2550  
2551  
2552  
2553  
2554  
2555  
2556  
2557  
2558  
2559  
2560  
2561  
2562  
2563  
2564  
2565  
2566  
2567  
2568  
2569  
2570  
2571  
2572  
2573  
2574  
2575  
2576  
2577  
2578  
2579  
2580  
2581  
2582  
2583  
2584  
2585  
2586  
2587  
2588  
2589  
2590  
2591*(Details omitted as they are not crucial for this case study.)*Given that  $\angle ABC = 60^\circ$  and  $AC = 48$  feet, we can use the Law of Sines to find  $AB$ :*(Details omitted as they are not crucial for this case study.)*The square of the distance  $AB$  is:

$$(32\sqrt{3})^2 = 32^2 \times 3 = 1024 \times$$

Thus, the square of the distance between Abdul and Bharat is 3072.

To solve this problem, we can use the Law of Sines in trigonometry.

*(Details omitted as they are not crucial for this case study.)*Given that  $AC = 48$  feet, we can use the Law of Sines to find  $AB$ :

$$\frac{AB}{\sin(60^\circ)} = \frac{AC}{\sin(30^\circ)}$$

*(Details omitted as they are not crucial for this case study.)*Solving for  $AB$ :

$$AB = 96 \times \frac{\sqrt{3}}{2} = 48\sqrt{3}$$

*(Details omitted as they are not crucial for this case study.)*Therefore, the square of the distance between Abdul and Bharat is 6912.*(Details omitted as they are not crucial for this case study.)*

## 1. Identify the given information:

*(Details omitted as they are not crucial for this case study.)*

## 2. Visualize the problem:

*(Details omitted as they are not crucial for this case study.)*

## 3. Determine the type of triangle:

- Since angle  $ABC$  and angle  $ACB$  are both  $60^\circ$ , triangle  $ABC$  is an equilateral triangle. In an equilateral triangle, all sides are equal.

## 4. Calculate the side length of the equilateral triangle:

- Since  $AC = 48$  feet and triangle  $ABC$  is equilateral,  $AB$  (the distance from Abdul to Bharat) is also 48 feet.*(Details omitted as they are not crucial for this case study.)*So, the square of the distance between Abdul and Bharat is 2304.To solve this problem, we can use the Law of Sines and some geometric properties. Let's denote the positions of Abdul, Chiang, and Bharat as points  $A$ ,  $C$ , and  $B$  respectively. The distance between  $A$  and  $C$  is given as 48 feet. The angle  $\angle ACB$  is  $60^\circ$ . We need to find the square of the distance  $AB$ , which is the farthest distance Bharat can be from Abdul.*(Details omitted as they are not crucial for this case study.)*Therefore, the square of the distance between Abdul and Bharat is 6912.*(Details omitted as they are not crucial for this case study.)*Step 3: Recognize that if triangle  $ABC$  is equilateral, then all sides are equal. Therefore,  $AB = BC = AC = 48$  feet.Step 4: Calculate the square of the distance  $AB$ . Since  $AB = 48$  feet, we square this value to get  $AB^2$ .

$$AB^2 = 48^2 = 2304$$

So, the square of the distance between Abdul and Bharat is 2304.To find the square of the distance between Abdul and Bharat, we can use the Law of Sines in trigonometry. Let's denote the distance between Abdul and Bharat as  $AB$ , and the distance between Chiang and Bharat as  $BC$ . Since Bharat is as far from Abdul as possible, triangle  $ABC$  is a right triangle with  $\angle ACB = 90^\circ$ .Given: -  $\angle ABC = 60^\circ$   
-  $AC = 48$  feetWe want to find  $AB^2$ .First, we apply the Law of Sines in triangle  $ABC$ :*(Details omitted as they are not crucial for this case study.)*Thus, the square of the distance between Abdul and Bharat is 3072.*(Details omitted as they are not crucial for this case study.)*In a right triangle, if one of the angles is  $60^\circ$ , the triangle is a 30-60-90 triangle. In a 30-60-90 triangle, the sides are in the ratio  $1 : \sqrt{3} : 2$ . Here,  $AC$  is the hypotenuse, so we can set up the relationship as follows:

$$AC = 2 \times AB$$

$$48 = 2 \times AB$$

$$AB = \frac{48}{2} = 24 \text{ feet}$$

We need to find the square of the distance  $AB$ :

$$AB^2 = 24^2 = 576$$

Thus, the square of the distance between Abdul and Bharat is 576.*(Details omitted as they are not crucial for this case study.)*Using the properties of a 30-60-90 triangle, we know that the side opposite the  $60^\circ$  angle is  $\sqrt{3}$  times the side opposite the  $30^\circ$  angle. Let's denote the side opposite the  $30^\circ$  angle (which is half the distance between Abdul and Chiang) as  $x$ . Therefore,  $AB = x\sqrt{3}$  and  $BC = 2x$ .*(Details omitted as they are not crucial for this case study.)*

$$AB^2 = 2304(21 - 12\sqrt{3})$$

Since we are asked to find the square of the distance, and we have already calculated it, we can box the final answer:

2304

Figure 15: Case study: Geometry problem. GRPO with auxiliary shaping term not only achieves higher accuracy but also explores more diverse reasoning strategies than vanilla GRPO (continued).

2592

2593

2594

2595

2596

2597

2598

2599

2600

2601

2602

2603

2604

2605

2606

2607

2608

2609

2610

2611

2612

2613

2614

2615

2616

2617

2618

2619

2620

2621

2622

2623

2624

2625

2626

2627

2628

2629

2630

2631

2632

2633

2634

2635

2636

2637

2638

2639

2640

2641

2642

2643

**GRPO with Auxiliary Shaping Term#9-#16**

To solve this problem, we can use the Law of Sines in trigonometry.

(Details omitted as they are not crucial for this case study.)

Using the Law of Sines:

$$\frac{AB}{\sin(120^\circ)} = \frac{AC}{\sin(60^\circ)}$$

(Details omitted as they are not crucial for this case study.)

Therefore, the square of the distance between Abdul and Bharat is 3072.

To solve this problem, we need to use some geometric properties and trigonometric relationships. Let's break it down step by step.

**1. Identify the given information:**

(Details omitted as they are not crucial for this case study.)

**2. Visualize the problem:**

(Details omitted as they are not crucial for this case study.)

**3. Use the Law of Cosines:**

(Details omitted as they are not crucial for this case study.)

**4. Apply the Law of Cosines:**

(Details omitted as they are not crucial for this case study.)

**5. Find the square of the distance:** - The square of the distance between Abdul and Bharat is  $d^2 = 2304$ .

Therefore, the square of the distance between Abdul and Bharat is 2304.

To solve this problem, we can use the Law of Sines and some geometric properties.

(Details omitted as they are not crucial for this case study.)

Given that the angle  $ABC$  is  $60^\circ$  and  $AB$  is maximized, the triangle  $ABC$  is a right triangle with angle  $ACB$  being  $90^\circ$  (since the angle at Bharat is  $60^\circ$  and the sum of angles in a triangle is  $180^\circ$ , the remaining angle at  $A$  must be  $30^\circ$ , making  $AB$  the hypotenuse of a 30-60-90 triangle).

(Details omitted as they are not crucial for this case study.)

Therefore, the square of the distance between Abdul and Bharat is 3072.

To solve this problem, we need to use some geometric principles. Let's denote the positions of Abdul, Chiang, and Bharat as points  $A$ ,  $C$ , and  $B$  respectively. The distance between  $A$  and  $C$  is 48 feet, and we need to find the distance  $AB$  when the angle  $\angle ABC$  is  $60^\circ$  and  $B$  is as far from  $A$  as possible.

**1. Identify the configuration:**

(Details omitted as they are not crucial for this case study.)

**2. Use the Law of Cosines:**

(Details omitted as they are not crucial for this case study.)

**3. Substitute into the Law of Cosines:**

(Details omitted as they are not crucial for this case study.)

So, the square of the distance between Abdul and Bharat is 3840 square feet.

To solve this problem, we need to use some geometric principles, specifically involving the Law of Sines and properties of triangles. Let's break it down step by step:

**1. Identify the given information:**

(Details omitted as they are not crucial for this case study.)

**2. Visualize the problem:**

(Details omitted as they are not crucial for this case study.)

**3. Use the Law of Sines:**

(Details omitted as they are not crucial for this case study.)

**4. Determine the angles:**

(Details omitted as they are not crucial for this case study.)

**5. Calculate the distance AB:**

(Details omitted as they are not crucial for this case study.)

Therefore, the square of the distance between Abdul and Bharat is 3072.

To find the square of the distance between Abdul and Bharat, let's denote the distance between Abdul and Bharat as  $d$ , and the distance between Chiang and Bharat as  $d'$ .

Since the angle formed by the lines of sight to Abdul and Chiang is  $60^\circ$ , we can use the Law of Cosines in the triangle formed by Abdul, Chiang, and Bharat. The Law of Cosines states that for any triangle with sides  $a$ ,  $b$ , and  $c$ , and the angle  $\theta$  opposite side  $c$ :

$$c^2 = a^2 + b^2 - 2ab \cos(\theta)$$

(Details omitted as they are not crucial for this case study.)

Therefore, the square of the distance between Abdul and Bharat is 2304.

To solve this problem, we'll use some geometric principles, particularly the Law of Sines and properties of triangles. Let's break down the problem step-by-step:

**1. Identify the given information:**

(Details omitted as they are not crucial for this case study.)

**2. Visualize the problem:**

(Details omitted as they are not crucial for this case study.)

**3. Use the Law of Sines:**

(Details omitted as they are not crucial for this case study.)

**4. Apply the Law of Sines:**

(Details omitted as they are not crucial for this case study.)

**5. Use trigonometry in the right triangle:**

(Details omitted as they are not crucial for this case study.)

Therefore, the square of the distance between Abdul and Bharat is 1728.

To solve this problem, we need to use some geometric properties and trigonometry. Let's denote the positions of Abdul, Chiang, and Bharat as points  $A$ ,  $C$ , and  $B$  respectively in the field. We know that  $AC = 48$  feet and the angle  $\angle ACB = 60^\circ$ . We want to find the distance  $AB$ , which we'll denote as  $d$ , and then find  $d^2$ .

**1. Identify the type of triangle:**

(Details omitted as they are not crucial for this case study.)

**2. Apply the Law of Cosines:**

(Details omitted as they are not crucial for this case study.)

**3. Maximize the distance  $d$ :**

(Details omitted as they are not crucial for this case study.)

Therefore, the square of the distance between Abdul and Bharat is 1728.

Figure 15: Case study: Geometry problem. GRPO with auxiliary shaping term not only achieves higher accuracy but also explores more diverse reasoning strategies than vanilla GRPO (continued).