From Exploration to Exploitation: A Two-Stage Entropy RLVR Approach for Noise-Tolerant MLLM Training

Donglai Xu[1,*], **Hongzheng Yang**[2,*], **Yuzhi Zhao**[3,†], **Pingping Zhang**[3], **Jinpeng Chen**[3], **Wenao Ma**[2], **Zhijian Hou**[3], **Mengyang Wu**[2], **Xiaolei Li**[4], **Senkang Hu**[3], **Ziyi Guan**[5], **Jason Chun Lok Li**[5], **Lai-Man Po**[3]

[1]Independent Researcher [2]The Chinese University of Hong Kong [3]City University of Hong Kong
[4]Hong Kong University of Science and Technology [5]University of Hong Kong
donglaixu99@gmail.com; hzyang22@cse.cuhk.edu.hk; yzzhao2-c@my.cityu.edu.hk

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) for Multimodal Large Language Models (MLLMs) is highly dependent on high-quality labeled data, which is often scarce and prone to substantial annotation noise in real-world scenarios. Existing RLVR methods under noisy supervision can overfit to incorrect labels and suppress the response diversity essential for the reward ranking signal in Group Relative Policy Optimization (GRPO). To address these challenges and enhance noise tolerance, we propose a two-stage token-level entropy optimization method for RLVR. This approach dynamically guides the model from exploration to exploitation during training. In the initial exploration phase, token-level entropy maximization promotes diverse outputs, serving as a regularizer that mitigates premature convergence to noisy labels and ensures sufficient intra-group variation, enabling more reliable advantage estimation in GRPO. As training progresses, the method transitions into the exploitation phase, where token-level entropy minimization encourages the model to produce confident outputs, thereby consolidating acquired knowledge and refining prediction accuracy. Empirically, across diverse MLLM backbones, various noise settings, and multiple tasks, our phased entropy schedule delivers superior overall robustness and outperforms representative external-signal, internal-signal, and entropy-based baselines.

1. Introduction

Recently, Reinforcement Learning with Verifiable Rewards (RLVR) has gained recognition for its effectiveness, as evidenced by its superior generalization compared to supervised fine-tuning (SFT) [5], its ability to elicit reasoning, and its ease of implementation. A notable example is Group Relative Policy Optimization (GRPO) [12], applied by DeepSeek-R1 [12], which exemplifies these strengths. RLVR has demonstrated success across a wide range of domains, including mathematical reasoning [28, 32, 47], formal verification [30, 45], and code generation [43]. Moreover, RLVR has been extended to multimodal tasks, enhancing the reasoning capabilities of Multimodal Large Language Models (MLLMs). These applications span image classification and object grounding [2, 15, 25, 33], image segmentation [24], medical reasoning [17], video understanding [8, 39], and graphical user interface (GUI) reasoning [26, 27]. Despite these advancements, a critical challenge remains: RLVR methods typically rely on high-quality labeled data to compute verifiable rewards. In real-world scenarios, datasets are frequently accompanied by annotation noise, posing a significant barrier to effective RLVR implementation.

To address the challenge of applying RLVR to datasets with annotation noise, recent methodologies can be grouped into three primary categories:

1. **External-Signal-Based Methods**: These approaches utilize external verifiable signals to guide RLVR training, such as compilers for code generation [21], Large Language Models (LLMs) as evaluators (e.g., LLM-as-a-Judge) [11], and Test-time Reinforcement Learning (TTRL) [42, 51]. These methods exhibit inconsistent performance due to variations in LLM capabilities across domains, and tools like compilers are often task-specific, limiting their applicability.

2. **Internal-Signal-Based Methods**: These methods derive rewards directly from model outputs, such as random rewards or format rewards. These internal reward signals do not rely on labeled data or external tools [31]. Although these approaches offer flexibility, their effectiveness is constrained, as the internal reward signals are often not closely

[*]Equal contribution.
[†]Corresponding author.

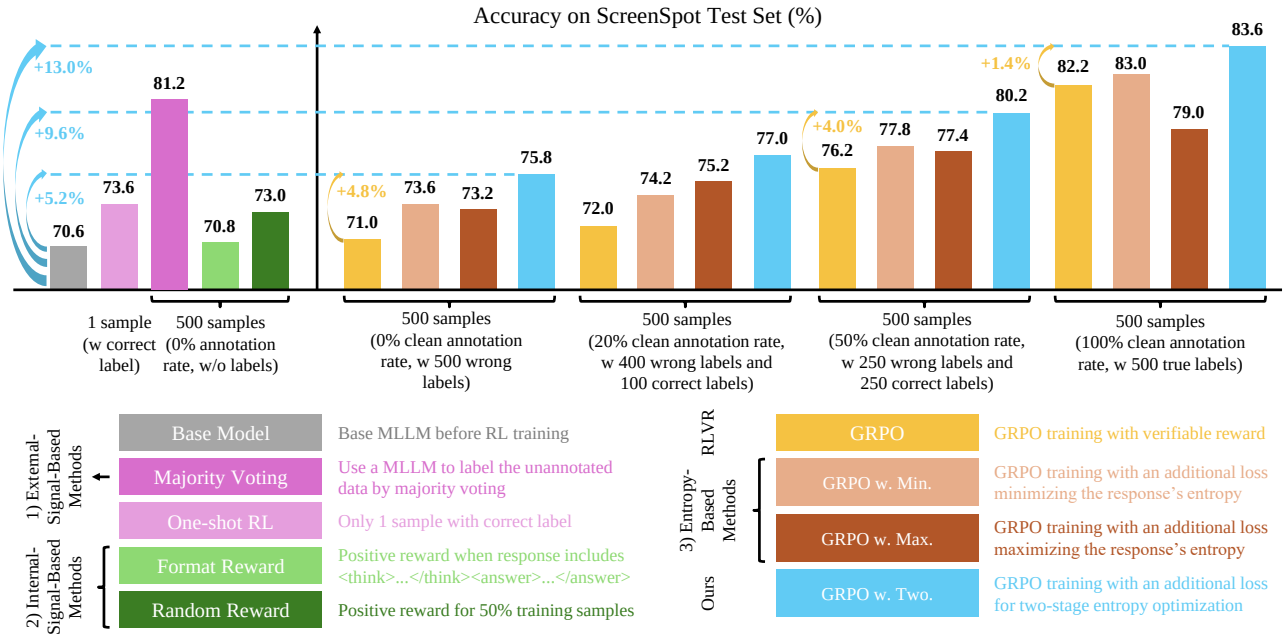

Figure 1. ScreenSpot accuracy after 1000 steps of different training strategies on Qwen2.5-VL-3B model. The horizontal axis includes different training data configurations. The proposed two-stage entropy-guided RLVR training method (GRPO w. Two.) performs better than one-shot RL [9], RLVR with "spurious rewards" (including format reward and random reward) [31], and RLVR with pure entropy minimization or maximization [49] on diverse noisy label settings, with clean annotation rates ranging from (0%, 20%, 50%, 100%).

aligned with task-specific objectives.

3. **Entropy-Based Methods**: These methods leverage entropy to guide training. For example, Wang et al. [41] proposed a one-shot RL scheme that achieves significant improvements in mathematical reasoning using the entropy loss. Similarly, Zhao et al. [50] employed self-certainty signals, while EMPO [49] minimized predictive entropy directly. However, these approaches often overemphasize entropy reduction, potentially overlooking the dynamic role of entropy across different training stages.

To investigate the robustness of RLVR under noisy data conditions, we evaluate the performance of MLLMs trained with different RL methods on two visual tasks: GUI grounding and fine-grained classification. We systematically vary the proportion of mislabeled data while maintaining a fixed training set size. The results for the GUI grounding task are presented in Figure 1. Our key observations regarding the three methodological categories are as follows:

1. As the proportion of mislabeled data decreases, the model accuracy after training generally increases. External-signal-based methods, such as TTRL [51], rely heavily on MLLMs for pre-labeling. The capability of the MLLM used for pre-labeling directly affects the ratio of noisy data introduced into the training set, which consequently imposes an upper bound on the final model performance.

2. With a small proportion of correctly labeled data, standard GRPO training outperforms internal-signal-based methods, such as those relying on spurious rewards [31].

3. Augmenting GRPO with entropy-based losses [49] consistently yields superior performance compared to using GRPO alone. Similar trends are observed across other vision tasks.

Based on these observations, we find that standard GRPO alone can already match or exceed the performance of internal-signal-based methods under moderate noise conditions (i.e., excluding purely random noise). Furthermore, augmenting standard GRPO with entropy-based methods can yield additional improvements against noisy annotations. However, if the optimization objective is naively reduced to either only entropy maximization or only entropy minimization, the learning dynamics can become problematic. Pure entropy maximization makes convergence difficult, while exclusive entropy minimization may trap the model in sub-optimal deterministic behaviors, especially facing the label noise. Furthermore, pure entropy minimization suppresses the response diversity, which is necessary for the informative advantage estimation required by GRPO. We argue that entropy optimization should be scheduled and switched between the two regimes, which could offer a controlled trade-off between exploration and

exploitation without sacrificing convergence stability.

Specifically, we propose a two-stage entropy-guided RLVR training method. During the early phase of training, we maximize token-level entropy to encourage more diverse outputs. This promotes exploration and mitigates overfitting to noisy labels. As training progresses, the model has captured most of the information from the datasets. We then proceed to the second stage, where entropy minimization is applied to encourage more confident and deterministic output. By explicitly guiding the model from exploration to exploitation, this two-stage method enhances the model's ability to learn from noisy datasets. For instance, by applying the two-stage entropy optimization to Qwen2.5-VL-3B [1] with 50% noisy labels, the method further boosts performance from 76.2% to 80.2% on ScreenSpot dataset [4], with similar gains observed across other levels of label noise (e.g., from 71% to 75.8% for 100% noisy labels, and from 82.2% to 83.6% for 0% noisy labels), as shown in Figure 1. It also outperforms pure entropy maximization or minimization in most noise settings, yielding the strongest overall robustness. Our contributions can be summarized as follows:

- We conduct comprehensive experiments across multiple dimensions: 1) varying noisy annotation rates, 2) diverse model architectures and scales (Qwen2-VL-2B, Qwen2.5-VL-3B, Qwen2-VL-7B [36], InternVL-3.5-2B [38]), and 3) multiple tasks (GUI grounding, fine-grained classification, and open-vocabulary object detection), to evaluate the impact of noisy labels on RLVR.
- We identify the limitations of existing RLVR methods under noisy supervision and introduce a two-stage entropy-guided optimization method. By transitioning from exploration to exploitation, our approach mitigates overfitting to noisy labels while consolidating knowledge.
- Our phased entropy strategy outperforms standard GRPO and existing entropy-based methods across different models, task types and noise conditions.

2. Related Works

2.1. Reinforcement Learning with Verifiable Rewards

RLVR leverages verifiable signals to compute rewards, particularly for tasks with well-defined correctness criteria, such as mathematical reasoning and code generation [14, 18, 32, 34]. Unlike traditional reinforcement learning approaches that rely on learned reward models, RLVR employs rule-based verification functions, such as exact answer matching, to mitigate the complexities and potential biases associated with learned rewards. This characteristic has enabled RLVR to achieve state-of-the-art reasoning capabilities in LLMs, as exemplified by DeepSeek-R1 [12]. The GRPO algorithm and its variants [32] have further extended RLVR to multimodal scenarios, including image classification [25], geometry reasoning [15], GUI grounding [27], and multi-step reasoning tasks such as search [16]. Despite these successes, RLVR's effectiveness is limited to domains with reliable verifiable signals and high-quality annotations, posing challenges in scenarios with noisy data.

2.2. Reinforcement Learning with Noisy Annotations

In scenarios where accurate and clean data is unavailable, existing methods often have to utilize noisy annotations to guide RLVR training. LLM-as-a-Judge [46, 48] is a well-known method, which utilizes the LLM itself as a noisy reward signal when accurate human feedback is not available. Recently, TTRL [51] employs majority voting across diverse model outputs to generate noisy pseudo labels, which serve as verifiable rewards to enhance mathematical reasoning through RL training. Additionally, research on spurious rewards [31] has explored format rewards and random rewards. These internal reward signals do not rely on any labeled data, either with clean or noisy annotations. The majority of these studies have focused on math reasoning and code generation tasks with either pure unlabeled data or partially labeled data with clean annotations. In this work, we systematically evaluate the impact of these reward signals on multimodal tasks under noisy supervision.

2.3. Entropy in Reinforcement Learning

Recently, entropy minimization [10] has been adapted to RLVR [41, 50]. For instance, Zhao et al. [50] only utilized self-certainty as a reward signal in RL training, achieving superior out-of-domain performance and matching standard GRPO training on mathematical reasoning benchmarks. Similarly, the EMPO framework [49] minimizes the entropy of output sequences, leveraging internal model consistency as an effective reward signal. Additionally, Seed-GRPO [3] employs entropy to modulate the magnitude of policy updates, enhancing training stability.

However, existing approaches primarily utilize pure entropy as a standalone reward signal, or focus on improving training stability under partially labeled datasets with strictly clean annotations. In contrast, our work investigates the dynamic role of entropy-based mechanisms for multimodal tasks under noisy supervision.

3. Preliminary

3.1. Group Relative Policy Optimization (GRPO)

RLVR leverages binary rewards for policy optimization. Unlike traditional reinforcement learning approaches that rely on human feedback or learned preference models, RLVR employs rule-based verification labels, such as exact answer matching, compiler feedback, or mathematical correctness checks to determine reward assignment.

GRPO serves as the primary algorithm for RLVR training. The GRPO training process begins by sampling K responses $\{y_1, y_2, \ldots, y_K\}$ from the current policy $\pi_\theta(\cdot|x)$ for each input x. Each response y_i is evaluated using a verifiable reward function $\mathcal{R}(y_i, y^*)$ that returns a binary signal based on correctness verification. The key innovation of GRPO is its group-wise advantage estimation that normalizes rewards within each group to reduce variance. For a given group of K responses with rewards $\{r_1, r_2, \ldots, r_K\}$, GRPO computes the advantage for each response as:

$$A_i = \frac{r_i - \text{mean}(r(y_{1:K}))}{\text{std}(r(y_{1:K}))}, \tag{1}$$

where $\text{mean}(r(y_{1:K}))$ and $\text{std}(r(y_{1:K}))$ are the mean and standard deviation of rewards within the group, respectively.

The policy gradient becomes:

$$\nabla_\theta \mathcal{L}_{\text{GRPO}} = -\mathbb{E}_{x\sim\mathcal{D}}\left[\sum_{i=1}^{K}\sum_{t=1}^{T_i} A_i \nabla_\theta \log \pi_\theta(y_{i,t}|y_{i,<t}, x)\right]. \tag{2}$$

In practice, we use a clipped surrogate objective to constrain the update relative to the old policy and avoid overly aggressive parameter changes.

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}}\left[\sum_{i=1}^{K}\sum_{t=1}^{T_i} \min\left(\frac{\pi_\theta(y_{i,t}|y_{i,<t}, x)}{\pi_{\theta_{\text{old}}}(y_{i,t}|y_{i,<t}, x)} A_i, \right.\right.$$
$$\left.\left. \text{clip}\left(\frac{\pi_\theta(y_{i,t}|y_{i,<t}, x)}{\pi_{\theta_{\text{old}}}(y_{i,t}|y_{i,<t}, x)}, 1-\epsilon, 1+\epsilon\right) A_i\right)\right], \tag{3}$$

where $T_i = |y_i|$ is the response length.

4. Methodology

4.1. Token-Level Entropy

The foundation of our approach lies in leveraging token-level entropy as a granular measure of uncertainty in text generation. Unlike sequence-level entropy, which captures the overall uncertainty of an output, token-level entropy quantifies the predictability of each token at every generation step. Formally, for an input sequence x and partially generated tokens $y_{<t}$, the model produces a conditional probability distribution $\pi_\theta(v \mid x, y_{<t})$ over vocabulary V. The per-token entropy is computed as:

$$\mathcal{H}_t(x, y) = -\sum_{v\in\mathcal{V}} \pi_\theta(v \mid x, y_{<t}) \log \pi_\theta(v \mid x, y_{<t}). \tag{4}$$

The token-level entropy for the entire sequence is then computed by averaging over all T tokens in the trajectory:

$$\mathcal{H}_{\text{token}}(x, y) = \frac{1}{T}\sum_{t=1}^{T} \mathcal{H}_t(x, y). \tag{5}$$

where $T = |y|$ is the response length. The corresponding entropy loss is then defined as:

$$\mathcal{L}_{\text{entropy}} = -\mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{K}\sum_{i=1}^{K} \mathcal{H}_{\text{token}}(x, y_i)\right]. \tag{6}$$

where K is the number of responses sampled per input x.

4.2. Two-Stage Entropy-Guided GRPO

The role of entropy in learning has been studied from the complementary perspectives of exploration and exploitation. Early work in semi-supervised classification [10, 19] argues that optimizing the predictive distribution towards low entropy transforms unlabeled inputs into effective constraints on the classification decision boundary. Deep reinforcement learning [13] literature, by contrast, argues for maximizing policy entropy to support exploration until the optimal behavior is reliably discovered. Existing RLVR studies inherit one of these viewpoints in isolation. EMPO [49] and one-shot RL [41] minimize predictive entropy to exploit the base model prior knowledge, while CLIP-Cov [7] prevents the collapse of the entropy, thus promoting exploration.

Both choices may break down under noisy supervision. Let $\mathcal{L}_{\text{entropy}}$ be the token-level entropy loss defined in Eq. (6) and λ be a positive constant. GRPO using $-\lambda\mathcal{L}_{\text{entropy}}$ as a regularization term in the total loss may drive the model to place overly high confidence on potentially incorrect labels. Furthermore, this minimization simultaneously suppresses the response diversity that GRPO's group-wise normalization requires for informative advantage estimation. In contrast, regularizing with $+\lambda\mathcal{L}_{\text{entropy}}$ alleviates overconfidence and preserves the alternative candidates necessary for GRPO response diversity. However, under consistent entropy maximization, the policy struggles to converge because the probability mass is never encouraged to concentrate. Therefore, we argue that the direction of entropy optimization should not remain static. Rather, it should be dynamically scheduled throughout the training process. As illustrated in Figure 3, token-level entropy should be maximized early in training. This initial exploration resists overfitting to noisy labels and provides the response diversity necessary for informative advantage estimation. Later in training, the entropy should be minimized to transition the model from exploration to exploitation, allowing it to consolidate learned knowledge.

Based on the above intuition, we propose a two-stage token-level entropy optimization framework for RLVR training, thereby realizing the exploration-to-exploitation trajectory. Let $\mathcal{L}_{\text{GRPO}}$ denote the standard GRPO loss derived from the group-wise advantage formulation, and let $\lambda(\tau)$ be a scalar coefficient that varies with the training step τ. The unified objective function is defined as:

Table 1. Accuracy (%) of Qwen2.5-VL-3B across different annotation noise levels on GUI grounding (ScreenSpot) and fine-grained classification (Pets37, 4-shot) tasks.

	GUI Grounding								Fine-grained Classification							
Method	**Base**	**100%**	**80%**	**60%**	**50%**	**40%**	**20%**	**0%**	**Base**	**100%**	**80%**	**60%**	**50%**	**40%**	**20%**	**0%**
Base Model	70.6	–	–	–	–	–	–		59.2	–	–	–	–	–	–	–
GRPO	–	71.0	72.0	75.8	76.2	79.8	81.8	82.2	–	54.7	64.7	67.3	68.5	68.8	68.8	**70.7**
GRPO w. Min.	–	73.2	75.2	77.4	77.4	77.6	79.0	79.0	–	**59.3**	64.6	66.9	**68.6**	68.7	69.5	70.4
GRPO w. Max.	–	73.6	74.2	76.6	77.8	**81.0**	**82.6**	83.0	–	51.0	64.5	67.5	67.8	68.5	68.9	69.8
GRPO w. Two.	–	**75.8**	**77.0**	**79.4**	**80.2**	80.6	82.4	**83.6**	–	54.3	**65.5**	**67.5**	68.4	**69.0**	**69.7**	70.0

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{GRPO}} + \lambda(\tau)\,\mathcal{L}_{\text{entropy}}. \tag{7}$$

We define the schedule for $\lambda(\tau)$ as a simple piecewise function:

$$\lambda(\tau) = \begin{cases} \lambda_{\max}, & \text{if } \tau \le \tau_{\text{switch}} \quad \text{(Stage 1: exploration)}, \\ -\lambda_{\min}, & \text{otherwise} \quad \text{(Stage 2: exploitation)}, \end{cases} \tag{8}$$

with hyper-parameters $\lambda_{\max}, \lambda_{\min} > 0$. During Stage 1, the positive coefficient instantiates an entropy maximization variant of GRPO, which encourages diverse sampling. The switching point is triggered at the τ_{switch} training step. We studied τ_{switch} in Section 5.4. Subsequently, Stage 2 flips the coefficient $-\lambda_{\min}$ to minimize entropy. This shifts the optimization objective, directing the model to produce confident outputs and consolidate the knowledge acquired during the exploration phase. The adaptive scheduling ensures that the model benefits from both regimes. The pseudo code is given in Algorithm 1.

5. Experiments

5.1. Experimental Setup

Datasets and Training. We use GRPO [23] to train base models. For Qwen-VL backbones, we adopt UI-R1 framework [27] for GUI grounding and Visual-RFT framework [25] for fine-grained classification. For Intern-VL backbones, we adopt verl-internvl framework [37, 38]. For GUI grounding task, we randomly select 500 samples from ScreenSpot [4] as a training set, with an equal distribution between mobile, web, and desktop. For fine-grained classification task, we utilize Pets37 [29] with 4-shot setting.

Evaluation. For evaluation of the GUI grounding task, we select 500 samples from ScreenSpot as a test set, which is different from the training samples but with the same platform distribution. For the fine-grained classification task, we use the official test split of the Pets37 dataset for evaluation. We adopt grounding and prediction accuracy as our evaluation metrics, which is calculated by matching the bounding box for the GUI grounding task and matching the label text for fine-grained classification. We compare five configurations: (1) Base pretrained model without RL (Base Model), (2) Standard GRPO (GRPO), (3) GRPO training with an additional entropy-minimization regularization term (GRPO w. Min.), (4) GRPO with an additional entropy-maximization regularization term (GRPO w. Max.), and (5) Our proposed two-stage entropy-guided method (GRPO w. Two.).

Noisy Supervision Simulation. For the GUI grounding task, we simulate noisy labels by randomly generating a new bounding box in the image with the same size as the original ground truth bounding box, ensuring no overlap between them, and using it as the new noisy target. We reward the response if the grounding point is within the target noisy bounding box. For the fine-grained classification task, to create noisy annotations, we randomly replace the correct label with an incorrect one drawn from the remaining set of labels. We reward the response if the prediction matches the noisy label. Across both tasks, we generate datasets with noise levels $\{100\%, 80\%, 60\%, 50\%, 40\%, 20\%, 0\%\}$.

5.2. Main Results

Quantitative Analysis. Table 1 reports Qwen2.5-VL-3B's results on two different tasks. For GUI grounding, the proposed two-stage entropy-guided optimization method maintains 80.2% accuracy at 50% noise, just 2% below GRPO trained on clean labels, demonstrating remarkable noise tolerance. Our method consistently outperforms the standard GRPO baseline across all noise levels, with a particularly strong improvement of 4.8% absolute gain at high noise (100%). Absolute gains of 5.2% to 13.0% were achieved over the base Qwen2.5-VL-3B model under different noise conditions. These results validate our core hypothesis that strategic entropy modulation enhances model performance under noisy data settings. For fine-grained classification, the results share similar trends with GUI grounding. In particular, entropy minimization performs best at 100% noise (59.3%), while maximization excels at 0% noise (69.8%). Our method balances these regimes, delivering robust performance across noise levels. This confirms the task-agnostic benefits of our method.

Table 2 further reports results for three Qwen-VL backbones and Internvl-3.5 on GUI grounding. Our two-stage method delivers gains across different model sizes, model

Table 2. Effect of Various Base Models on the ScreenSpot Dataset. Accuracy (%) of 4 backbones, Qwen2-VL-2B, Qwen2-VL-7B, Qwen2.5-VL-3B, InternVL-3.5-2B, trained with standard GRPO versus the proposed two-stage entropy-guided method (i.e., **w. Two.**).

	Qwen2-VL-2B			Qwen2-VL-7B			Qwen2.5-VL-3B			InternVL-3.5-2B		
Noise Level	**Base**	**GRPO**	**w. Two.**	**Base**	**GRPO**	**w. Two.**	**Base**	**GRPO**	**w. Two.**	**Base**	**GRPO**	**w. Two.**
-	11.2	–	–	37.2	–	–	70.6	–	–	48.6	–	–
100%	–	**17.0**	14.4	–	**37.4**	34.8	–	71.0	**75.8**	–	49.2	**50.2**
50%	–	**32.8**	25.2	–	61.2	**69.8**	–	76.2	**80.2**	–	56.8	**59.2**
20%	–	**50.0**	44.4	–	74.0	**76.6**	–	81.8	**82.4**	–	63.2	**69.8**
0%	–	55.2	**55.6**	–	75.4	**78.0**	–	82.2	**83.6**	–	66.8	**69.8**

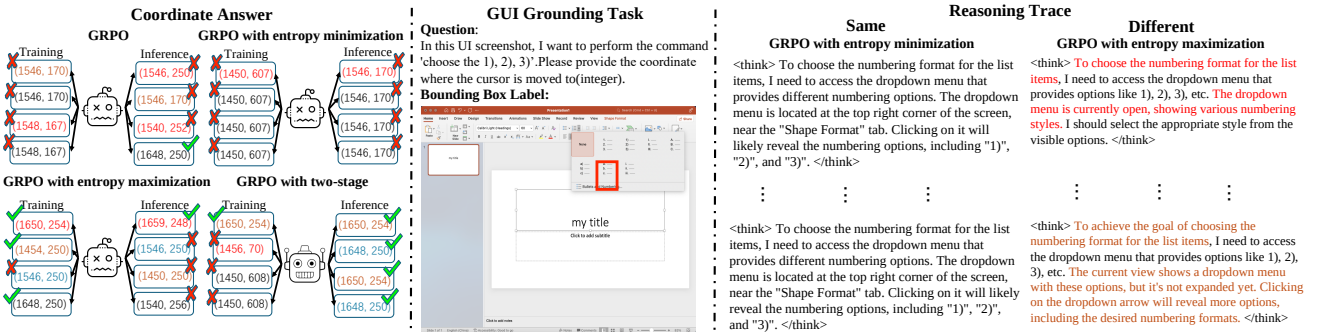

Figure 2. Qualitative effect of entropy scheduling on the GUI grounding task. We visualise the reasoning trace ($\langle$think$\rangle \ldots \langle$/think$\rangle$) and predicted coordinate produced by: GRPO, GRPO with entropy minimization, GRPO with entropy maximization, and GRPO with two-stage entropy-guided optimization. The ground-truth bounding box is outlined in red on the image.

families, and noise levels. Interestingly, we find that larger backbones benefit more from the two–stage schedule, showing the potential scalability of our approach. Qwen2-VL-7B records a significantly larger 8.6% improvement at 50% noise, while Qwen2-VL-2B has an opposite phenomenon. We also find our proposed approach has more significant gains on later models, e.g., Qwen2.5-VL-3B gains 4.8% at 100% noise and 4.2% at 50% noise over the GRPO baseline. Beyond Qwen Model Family, we also train InternVL-3.5-2B on ScreenSpot, where the two-stage method consistently outperforms the standard GRPO baseline across all noise levels. This confirms that the benefits of phased entropy optimization are not limited to the Qwen model family. We include the full results of InternVL-3.5-2B in Appendix D.

Qualitative Analysis. Figure 2 provides an illustrative comparison of how the three entropy regimes shape both the sampled reasoning traces and the final predictions. For GRPO with entropy minimization, the policy collapses almost immediately onto a single confident decoding path. All rollouts verbalize an almost identical chain of thought, so noisy rewards are propagated unchecked, and the model converges to the same incorrect coordinate at inference. In contrast, pure entropy maximization generates various reasoning paths that include at least one trajectory consistent with the true label, thus reducing the susceptibility to misleading reward signals. However, the lack of consolidation leaves its accuracy short of the best. For our two-stage method, the reasoning traces remain diverse enough to resist noise but also coherent enough to pinpoint the correct GUI region.

5.3. Discussions

Generalizable Findings. To further verify the general applicability of our method, we extend the study to the open-vocabulary object detection task (OVOD). Specifically, we randomly sampled 975 annotations from the COCO dataset [22], which includes 65 categories with 15 images per category. Similarly to GUI grounding task, we simulate label noise by generating bounding boxes that do not intersect with the original ground-truth boxes. Evaluation is performed on the remaining 15 categories that are unseen during training, using mean Average Precision (mAP) as the metric. We adopt the same GRPO method as in GUI grounding, with rewards computed based on box-overlap verification at an Intersection over Union (IoU) threshold of 0.5. Table 3 shows that the proposed two-stage entropy schedule significantly enhances the GRPO baseline across all noise conditions. Notably, at 50% label noise, the two-stage approach improves the mAP of Qwen2-VL-2B by 3.53 from 15.94 (standard GRPO) to 19.47, matching the best score among all configurations.

Noisy Data Scaling. To further investigate the impact of

Table 3. Performance comparison of Qwen2-VL-2B across different annotation noise levels on the OVOD task (mAP @ 0.5 IoU).

Method	OVOD			
	Base	100%	50%	0%
Base Model	9.56	–	–	–
GRPO	–	10.79	15.94	16.00
GRPO w. Max.	–	14.60	19.47	17.20
GRPO w. Min.	–	**15.94**	18.91	**18.79**
GRPO w. Two.	–	15.54	**19.47**	18.44

Table 4. Performance comparison of Qwen2.5-VL-3B for the scaling effect of adding noisy training data to 500 clean GUI-grounding samples.

Method	ScreenSpot					
	Base	+50	+100	+150	+200	+250
Base Model	70.6	–	–	–	–	–
GRPO	–	79.4	80.8	78.0	77.6	78.0
GRPO w. Min.	–	79.8	79.4	78.6	79.6	79.0
GRPO w. Max.	–	**82.2**	81.4	82.0	80.4	**80.4**
GRPO w. Two.	–	81.4	**81.8**	**82.8**	**81.8**	80.0

noisy data on GRPO training, we fixed 500 correctly labeled samples and incrementally added 50 noisy samples at a time to train models. As shown in Table 4, the standard GRPO baseline achieves its peak performance when 100 noisy samples are added to the 500 clean samples, after which accuracy begins to decline. In contrast, our two-stage method maintains a highly robust 80.0%-82.8% accuracy across all noise scaling levels, demonstrating superior stability. The noise effect is most pronounced for entropy maximization at +50 samples (82.2%), but it degrades with additional noise. The consistent performance of our method confirms that the phased entropy optimization effectively exploits noisy data benefits while mitigating its risks.

Out-of-distribution Generalization. To assess the out-of-distribution (OOD) generalization ability, we evaluate on the ScreenSpot-Pro [20], OS-World-G [44], and MMBench-GUI L2 [40] benchmarks, which differ significantly from the training distribution (ScreenSpot) in both visual complexity and domain coverage. For ScreenSpot-Pro, we randomly sample 150 samples from each category (Development, Creative, CAD, Scientific, Office, OS) to ensure equal amount for each category. For MMBench-GUI L2, we randomly sample 500 samples while ensuring the data distribution is uniformed across six platforms (Windows, macOS, linux, iOS, Android, and Web). For OS-World-G, we use the whole dataset.

As shown in Table 5, the two-stage method achieves the best OOD performance (20.7%) with 500 clean samples +150 mislabeled samples configuration (i.e., +150 configuration). This 2.7-5.4% improvement over alternatives indicates that the two-stage entropy optimization method improves knowledge transfer. In particular, entropy maximization alone achieves competitive OOD performance at

Table 5. OOD evaluation accuracy (%) of Qwen2.5-VL-3B trained on ScreenSpot, evaluated on ScreenSpot-Pro and OS-World-G across adding noisy training data configurations.

Method	ScreenSpot-Pro					OS-World-G				
	+50	+100	+150	+200	+250	+50	+100	+150	+200	+250
GRPO	16.7	16.7	18.0	17.3	**19.3**	38.1	39.8	36.5	37.7	**41.4**
GRPO w. Min.	16.0	16.7	15.3	16.0	16.7	39.8	42.0	40.2	38.9	38.7
GRPO w. Max.	**20.7**	16.7	18.0	17.3	12.7	**42.6**	39.8	**42.3**	40.2	41.3
GRPO w. Two.	16.7	**19.3**	**20.7**	**18.0**	18.0	42.1	**42.1**	41.2	**42.4**	40.0

Table 6. OOD evaluation accuracy (%) of Qwen2.5-VL-3B trained on ScreenSpot, evaluated on ScreenSpot-Pro and MMBench-GUI L2 across different annotation noise levels.

Method	ScreenSpot-Pro				MMBench-GUI L2			
	Base	0%	50%	100%	Base	0%	50%	100%
Base Model	6.4	–	–	–	45.0	–	–	–
GRPO	–	16.7	13.3	8.7	–	57.0	53.8	46.4
GRPO w. Min.	–	18.7	11.3	**8.7**	–	58.0	55.6	**51.0**
GRPO w. Max.	–	16.7	14.0	7.3	–	58.2	54.6	48.8
GRPO w. Two.	–	**21.3**	**18.0**	8.0	–	**60.6**	**57.4**	49.0

+50 samples (20.7%) but degrades with additional noise, while our method maintains robust generalization. Across all noise levels on OS-World-G, our two-stage method (GRPO w. Two.) consistently delivers robust OOD performance, maintaining accuracy between 40.0% and 42.4%, outperforming standard GRPO (37.7%–41.4%) and single-stage variants (e.g., GRPO w. Max.: 38.7%–42.6%; GRPO w. Min.: 38.7%–40.2%). We further evaluate our two-stage method for OOD generalization across ScreenSpot-Pro and MMBench-GUI L2, as shown in Table 6. For instance, our two-stage method achieves the best overall OOD performance. It achieves 60.6% accuracy on clean data and 57.4% at 50% noise, outperforming standard GRPO and single-stage entropy methods on MMBench-GUI L2. We include additional results on the MMBench-GUI L2 in Appendix D.

GRPO tolerance to Data Noise. Fig. 1 and Table 1 reveal that standard GRPO already exhibits moderate robustness to noisy labels. With 50% noisy GUI-grounding labels, Qwen2.5-VL-3B trained with standard GRPO attains 76.2% accuracy, only 6% below the clean-data ceiling. We hypothesize that this robustness arises partly from GRPO's group-wise advantage normalization. When evaluating a mislabeled sample, if the model's prior ability leads all K rollouts to consistently predict the actual correct answer, every response in the group receives the same zero reward. Consequently, the normalized advantages become zero, which yields no learning signal for that group. This self-gating effect establishes a robust baseline on top of which entropy scheduling can operate. To test whether the noise tolerance observed in GRPO is an inherent algorithmic property rather than an artifact of data formatting or preprocessing, we conduct additional ablation studies in Appendix D. Results show GRPO remains robust to noisy labels across preprocessing variations.

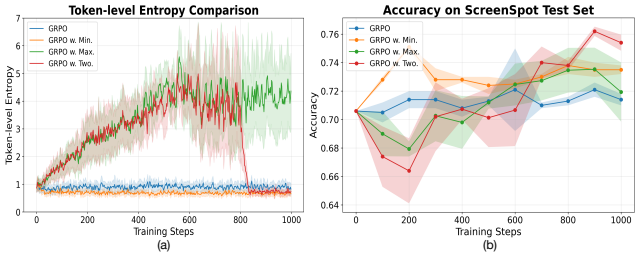

Figure 3. (a) Comparison of token-level entropy dynamics during training with 100% noise; (b) Comparison of ScreenSpot test set accuracy at each training step under 100% noise level. We compare 4 strategies: standard GRPO, GRPO with entropy maximization, GRPO with entropy minimization, and GRPO with two-stage entropy-guided optimization.

Table 7. Influence of the exploration-to-exploitation switch point for Qwen2.5-VL-3B on the GUI grounding task.

	Transition Point			
Noise Level	**Step 500**	**Step 700**	**Step 800**	**Step 900**
100%	73.6	75.0	**75.8**	73.6
50%	79.6	79.8	**80.2**	79.0
0%	80.4	81.8	**83.6**	82.0

5.4. Ablation Study

Training Dynamics of Entropy. For our proposed two-stage entropy-guided optimization method, Figure 3 illustrates the training dynamics of token-level entropy. During Phase 1 (steps 0-800), the entropy increases steadily to ∼400% of the initial value, confirming effective exploration. The transition to Phase 2 (steps 800-1000) triggers a rapid reduction in entropy, stabilizing at ∼20% of the peak value after 900 steps. These dynamics validate our core design: extended exploration prevents premature convergence, while subsequent exploitation consolidates knowledge into confident predictions. This smooth phase transition is crucial for maintaining stability under noisy supervision.

Switching Points Analysis. We examine the effect of switching point on the GUI grounding task (ScreenSpot as the training set), by varying $\tau_{\text{switch}} \in \{500, 700, 800, 900\}$. As shown in Table 7, the performance is generally robust to the fixed switching point. Switching at step 800, corresponding to 80% of the total training steps, achieves the best balance between sufficient exploration and late-stage consolidation on GUI grounding.

Stage-wise entropy scheduling ("Max. then Min.") outperforms subset-wise entropy assignment. Table 8 compares four possible ways of combining entropy maximization and minimization under 100%, 50% and 0% noise level (ScreenSpot as the training set). Across all noise levels, "Max. then Min." outperforms "Min. then Max." by 5.6% at 100% noise level, 3.4% at 50% noise level and 3.8% on clean data. Beginning with entropy minimization prematurely encourages the policy to converge. This causes the model to overfit to the initial noisy reward signals, reducing the response diversity required for GRPO to discover better trajectories later in training. Conversely, starting with entropy maximization supports the exploration and the diversity needed for effective group-wise advantage estimation in GRPO. The subsequent minimization phase then consolidates the high reward behaviors discovered during the exploration phase into a confident policy.

Table 8. Performance Comparison Across Two-stage Methods for Qwen2.5-VL-3B on the GUI grounding task. LT. refers to training samples with correct labels. LF. refers to training samples with incorrect labels. LT. Max. LF. Min. refers to maximizing entropy on the correctly-labeled subset and minimize it on the noisy subset.

	Noise Level		
Methods	**100%**	**50%**	**0%**
LT. Max. LF. Min.	73.2	76.8	83.0
LF. Max. LT. Min.	73.6	78.0	79.0
Min. then Max.	70.2	76.8	79.8
Max. then Min.	**75.8**	**80.2**	**83.6**

When maximizing entropy is restricted to the noisy subset only (i.e.,"LF. Max. LT. Min."), its performance is better than "Min. then Max." but still inferior to "Max. then Min." schedule. Applying entropy maximization exclusively to the noisy subset restricts the model's overall ability to explore. Even for correctly labeled data, initial exploration is beneficial for discovering potentially better reasoning paths before committing to a final policy. Furthermore, in practical settings, the clean or noisy status of a label is unknown, making a unified schedule much more applicable. Symmetrically, "LT. Max. LF. Min." works well with 50% and 0% noise levels, because half or all data are reliable, but it suffers under 100% noise level when there are no clean labels to guide the exploitation.

6. Conclusion

We explore the effectiveness of RLVR under noisy supervision for multimodal tasks. To augment RLVR methods like GRPO, we propose a Two-Stage Entropy-Guided GRPO that first maximizes and then minimizes the token-level entropy during training. This strategy encourages early exploration and later exploitation, leading to improved robustness against label noise. Through extensive experiments with Qwen and InternVL models and on different tasks, we demonstrate that our method maintains high performance even under substantial annotation noise. In particular, the two-stage method contributes to more stable convergence and better generalization. Our findings highlight the potential of entropy-aware policy optimization as a powerful tool for learning from imperfect data in multimodal scenarios.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 3

[2] Liang Chen, Lei Li, Haozhe Zhao, Yifan Song, and Vinci. R1-v: Reinforcing super generalization ability in vision-language models with less than $3. https://github.com/Deep-Agent/R1-V, 2025. Accessed: 2025-02-02. 1

[3] Minghan Chen, Guikun Chen, Wenguan Wang, and Yi Yang. Seed-grpo: Semantic entropy enhanced grpo for uncertainty-aware policy optimization. *arXiv preprint arXiv:2505.12346*, 2025. 3

[4] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*, 2024. 3, 5

[5] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. In *Forty-second International Conference on Machine Learning*, 2025. 1

[6] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 2

[7] Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025. 4

[8] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025. 1

[9] Zitian Gao, Lynx Chen, Joey Zhou, and Bryan Dai. One-shot entropy minimization. *arXiv preprint arXiv:2505.20282*, 2025. 2

[10] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004. 3, 4

[11] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024. 1

[12] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 1, 3

[13] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018. 4

[14] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025. 3

[15] Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025. 1, 3

[16] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025. 3

[17] Yuxiang Lai, Jike Zhong, Ming Li, Shitian Zhao, and Xiaofeng Yang. Med-r1: Reinforcement learning for generalizable medical reasoning in vision-language models. *arXiv preprint arXiv:2503.13939*, 2025. 1

[18] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024. 3

[19] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, page 896. Atlanta, 2013. 4

[20] Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use, 2025. 7

[21] Shangzhan Li, Zefan Wang, Ye He, Yuxuan Li, Qi Shi, Jianling Li, Yonggang Hu, Wanxiang Che, Xu Han, Zhiyuan Liu, et al. Autotriton: Automatic triton programming with reinforcement learning in llms. *arXiv preprint arXiv:2507.05687*, 2025. 1

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014*, pages 740–755, Zurich, Switzerland, 2014. Springer. 6

[23] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 5

[24] Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025. 1

[25] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-

rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*, 2025. 1, 3, 5

[26] Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Guanjing Xiong, and Hongsheng Li. Ui-r1: Enhancing efficient action prediction of gui agents by reinforcement learning. *arXiv preprint arXiv:2503.21620*, 2025. 1

[27] Run Luo, Lu Wang, Wanwei He, and Xiaobo Xia. Gui-r1: A generalist r1-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458*, 2025. 1, 3, 5

[28] Hieu Trung Nguyen, Bao Nguyen, Wenao Ma, Yuzhi Zhao, Ruifeng She, and Viet Anh Nguyen. Adaptive rollout allocation for online reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2602.01601*, 2026. 1

[29] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 5

[30] ZZ Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, et al. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2504.21801*, 2025. 1

[31] Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, et al. Spurious rewards: Rethinking training signals in rlvr. *arXiv preprint arXiv:2506.10947*, 2025. 1, 2, 3

[32] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 1, 3

[33] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025. 1

[34] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025. 3

[35] Qwen Team. Qwen2.5: A party of foundation models, 2024. 2

[36] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 3

[37] Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, and Jifeng Dai. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *arXiv preprint arXiv:2411.10442*, 2024. 5

[38] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. 3, 5

[39] Xiaodong Wang and Peixi Peng. Open-r1-video. https://github.com/Wang-Xiaodong1899/Open-R1-Video, 2025. Accessed: 21-July-2025. 1

[40] Xuehui Wang, Zhenyu Wu, JingJing Xie, Zichen Ding, Bowen Yang, Zehao Li, Zhaoyang Liu, Qingyun Li, Xuan Dong, Zhe Chen, et al. Mmbench-gui: Hierarchical multi-platform evaluation framework for gui agents. *arXiv preprint arXiv:2507.19478*, 2025. 7

[41] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025. 2, 3, 4

[42] Lai Wei, Yuting Li, Chen Wang, Yue Wang, Linghe Kong, Weiran Huang, and Lichao Sun. Unsupervised post-training for multi-modal llm reasoning via grpo. *arXiv preprint arXiv:2505.22453*, 2025. 1

[43] Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I Wang. Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution. *arXiv preprint arXiv:2502.18449*, 2025. 1

[44] Tianbao Xie, Jiaqi Deng, Xiaochuan Li, Junlin Yang, Haoyuan Wu, Jixuan Chen, Wenjing Hu, Xinyuan Wang, Yuhui Xu, Zekun Wang, Yiheng Xu, Junli Wang, Doyen Sahoo, Tao Yu, and Caiming Xiong. Scaling computer-use grounding via user interface decomposition and synthesis, 2025. 7

[45] Huajian Xin, ZZ Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, et al. Deepseek-prover-v1.5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search. *arXiv preprint arXiv:2408.08152*, 2024. 1

[46] Wei Xiong, Hanning Zhang, Chenlu Ye, Lichang Chen, Nan Jiang, and Tong Zhang. Self-rewarding correction for mathematical reasoning. *arXiv preprint arXiv:2502.19613*, 2025. 3

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

[48] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 3, 2024. 3

[49] Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025. 2, 3, 4

[50] Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey

Levine, and Dawn Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025. 2, 3

[51] Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025. 1, 2, 3

From Exploration to Exploitation: A Two-Stage Entropy RLVR Approach for Noise-Tolerant MLLM Training

Supplementary Material

A. Limitations.

A limitation of the Two-Stage Entropy-Guided GRPO approach is that it works best when the base model has a reasonable prior ability on the target task. If the zero-shot ability of the base model in the target task is weak, early maximization of entropy can amplify incorrect modes before the model samples a correct trajectory. This likely explains the weaker gains for Qwen2-VL-2B in Table 2 and the limited benefit under fully noisy supervision on fine-grained classification in Table 1.

B. Implementation Details

Training Details. We provide a brief summary of the training settings in Table B.1. For both the GUI grounding and fine-grained classification tasks, the base model is trained using 8 NVIDIA L20 GPUs, requiring approximately 8 hours and 1 hour, respectively. OVOD tasks share the same setting as fined-grained classification tasks. Code website: https://github.com/xudonglai0426/RLVR-from-Exploration-to-Exploitation.

Table B.1. Hyperparameter settings used in the experiments.

Hyperparameter	GUI Ground.	Fine. Class.
Learning rate (lr)	9.98×10^{-7} to 0	9.98×10^{-7} to 0
Max pixels	12,845,056	401,408
Number of generations	8	8
Number of training epochs	4	24
Max prompt length	1024	1024
Per-device train batch size	1	1
Gradient accumulation steps	2	2
Entropy Coef.	1×10^{-2}	1×10^{-2}
KL Coef.	4×10^{-2}	0

Evaluation Details. For the MMBench-GUI L2 benchmark, we randomly sample 500 samples for the training set and 500 samples for the test set. Both sets share the same data composition, with an equal distribution across the six platforms (Windows, macOS, Linux, iOS, Android, and Web) and the two instruction types (basic and advanced). For OS-World-G benchmark, we use the whole dataset with refined instruction for evaluation.

C. Entropy Optimization Schedule

Why Training Starts with Entropy Maximization. Our two-stage schedule begins with token-level entropy maximization because diversity is the currency that GRPO relies on to compute meaningful advantage signals. Maximization enlarges the variance of responses within each group, sharpening the relative ranking and, consequently, the gradient. At the same time, it regularizes the policy against premature converge to spurious labels. When the correct supervision is missing or wrong, a more diverse distribution prevents the policy from overfitting to the noisy target. Empirically, this exploration phase already yields a non-trivial improvement over either entropy minimization or the plain GRPO baseline (e.g. 77.8% vs. 76.2% at 50% noise on ScreenSpot).

Why Training ends with Entropy Minimization. Exploration alone is insufficient. Once the policy has discovered high-reward regions, it must consolidate. After token entropy plateaus, the sign of the entropy coefficient is flipped. Minimizing entropy concentrates probability mass on the best trajectory identified earlier, reduces variance at inference time and sharpens predictions. The switch consistently achieves improvements across all noise levels, confirming that exploitation effectively complements exploration.

Algorithm 1 Two-Stage Entropy-Guided GRPO

Require: switch step τ_{switch}, coefficients $\lambda_{\max}$, $\lambda_{\min}$, total training steps E, model π_θ with parameters θ.
for $\tau = 1$ **to** E **do**
  Sample K responses $\{y_1, \ldots, y_K\}$ from $\pi_\theta(\cdot|x)$
  Compute rewards $r_i = \mathcal{R}(y_i, y^*)$ for each response
  Compute normalized advantages:

$$A_i = \frac{r_i - \text{mean}(r(y_{1:K}))}{\text{std}(r(y_{1:K}))}$$

  if $\tau \leq \tau_{\text{switch}}$ **then**
    $\lambda(\tau) \leftarrow +\lambda_{\max}$
  else
    $\lambda(\tau) \leftarrow -\lambda_{\min}$
  end if
  Compute standard GRPO loss: $\mathcal{L}_{\text{GRPO}}$ (see Eq. (3))
  Compute entropy regularization term: $\mathcal{L}_{\text{entropy}}$ (see Eq. (6))
  Compute total loss: $\mathcal{L}_{\text{total}}$ (see Eq. (7))
  Update θ with AdamW on $\nabla_\theta \mathcal{L}_{\text{total}}$
end for
return trained model π_θ

D. Additional Experiments

Robust Analysis of GRPO. As observed in Fig. 1 and Table 1, the standard GRPO already exhibits moderate robustness to noisy labels. To address potential concerns that this noise tolerance might be an artifact of specific data pre-

Table C.1. Robustness Analysis of GRPO on ScreenSpot at 100% noise level.

Model	GRPO	GRPO with Two.	GRPO w. Abs. Coord.	GRPO w. Two. and Abs. Coord.	GRPO w. Resize	GRPO w. Two. w. Resize
Qwen2-VL-2B	12.4	13.8	14.5	16.0	13.4	16.6
Qwen2.5-VL-3B	69.8	73.8	69.8	73.8	70.6	74.2
InternVL3.5-2B	49.2	50.2	49.2	50.2	46.2	49.8

Table D.1. Accuracy (%) of InternVL-3.5-2B across annotation noise levels on the GUI grounding (ScreenSpot) task.

Method	Base	100%	80%	50%	20%	0%
Base Model	48.6	-	-	-	-	-
GRPO	-	49.2	49.6	56.8	63.2	66.8
GRPO w. Min.	-	48.0	51.0	**59.8**	65.6	69.2
GRPO w. Max.	-	49.8	51.6	57.6	66.0	65.2
GRPO w. Two.	-	**50.2**	**53.0**	59.2	**69.8**	**69.8**

Table D.2. In-domain training Accuracy (%) on MMBench-GUI L2 of Qwen2.5-VL-3B. The model is trained on MMBench-GUI L2 under {100%, 50%, 0%} annotation noise.

Method	Base	100%	50%	0%
Base Model	45.0	-	-	-
GRPO	-	47.0	53.6	55.0
GRPO w. Min.	-	51.0	54.6	57.6
GRPO w. Max.	-	49.6	56.0	58.0
GRPO w. Two.	-	49.4	53.8	55.0

Table D.3. In-domain training Accuracy (%) on GSM8K of Qwen2.5-3B. The model is trained on GSM8K under {100%, 50%, 0%} annotation noise.

Method	Base	100%	50%	0%
Base Model	77.2	-	-	-
GRPO	-	80.4	78.6	81.4
GRPO w. Min.	-	80.4	81.4	83.2
GRPO w. Max.	-	81	81.6	80.6
GRPO w. Two.	-	80.4	80.6	79.6

processing choices, we conducted an ablation study on coordinate formatting and image scaling with 4 rollouts during training. Specifically, we evaluated the GRPO baseline under 100% noise using absolute coordinates (GRPO w. Abs. Coord.) instead of relative ones, and with dynamic image resizing enabled (GRPO w. Resize). As shown in Table C.1, performance remains stable across these preprocessing variations. This confirms that the noise tolerance is an inherent algorithmic property of GRPO. Specifically, the self-gating effect where uniform incorrect predictions within a group yield zero normalized advantage mitigates harmful gradient updates.

Evaluation on MMBench-GUI L2. To resolve potential concerns about training data contamination, we conducted additional experiments using the MMBench-GUI L2 dataset. Since MMBench-GUI L2 was published after the knowledge cutoff of the Qwen2.5-VL-3B base model, it serves as an ideal benchmark for data contamination evaluation. We evaluate our approach under in-domain training settings. For in-domain training on the MMBench-GUI L2, we train and evaluate the model directly on the MMBench-GUI L2 dataset under different annotation noise levels. We set the transition step as 400 and evaluate at training step 500. As shown in Table D.2, our two-stage method achieves 49.4% accuracy, outperforming the base model (45.0%) and standard GRPO (47.0%).

Experiments on Text-based Tasks. To further investigate our two-stage method, we conducted additional experiments using the GSM8K [6] dataset with Qwen2.5-3B [35] in Table D.3. We randomly select 500 samples from the training set and 500 samples from the test set of the full dataset for training and evaluation, respectively.