# 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) | $9.98 \times 10^{-7}$ to 0 | $9.98 \times 10^{-7}$ 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. | $1 \times 10^{-2}$ | $1 \times 10^{-2}$ |
| KL Coef. | $4 \times 10^{-2}$ | 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

**Algorithm 1** Two-Stage Entropy-Guided GRPO

**Require:** switch step $\tau_{\text{switch}}$, coefficients $\lambda_{\max}$, $\lambda_{\min}$, total training steps $E$, model $\pi_\theta$ with parameters $\theta$.
**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 $\theta$ with AdamW on $\nabla_\theta \mathcal{L}_{\text{total}}$
**end for**
**return** trained model $\pi_\theta$

**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.

## 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.