NoisyRollout: Reinforcing Visual Reasoning with Data Augmentation

Xiangyan Liu*[1] **Jinjie Ni***[1] **Zijian Wu***[1] **Chao Du**[2] **Longxu Dou**[2] **Haonan Wang**[1]
Tianyu Pang†[2] **Michael Qizhe Shieh**[1]

[1]National University of Singapore [2]Sea AI Lab **Code** **Collection**

Abstract

Recent advances in reinforcement learning (RL) have strengthened the reasoning capabilities of vision-language models (VLMs). However, enhancing policy exploration to better scale test-time compute remains largely underexplored. In addition, VLMs continue to struggle with imperfect visual perception, which in turn affects the subsequent reasoning process. We introduce **NoisyRollout**, a simple yet effective data augmentation method that addresses these issues by mixing training trajectories from both clean and moderately distorted images. This approach injects perceptual diversity, encouraging better policy exploration and leading to more robust reasoning. A noise annealing schedule gradually reduces distortion strength, aiding exploration early in training while ensuring later stability. Crucially, our method is easy-to-adopt—**requiring no additional training cost and no modifications to the RL objective**. Extensive experiments on **2** distinct training datasets demonstrate that NoisyRollout achieves state-of-the-art performance among open-source RL-tuned models across **5** out-of-domain reasoning and perception benchmarks. Furthermore, we validate the effectiveness of NoisyRollout across model sizes (7B and 32B), data scales (from 1K to 6K) and image augmentation types (Gaussion noise and rotation), highlighting its generalizability and scalability.

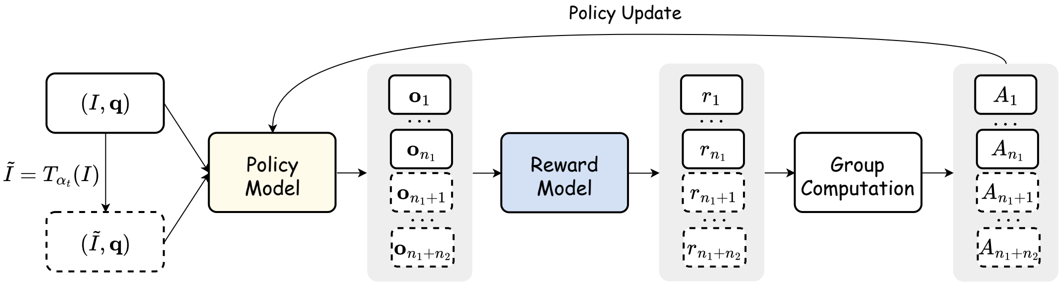


Figure 1: An Illustration of the NoisyRollout workflow. Solid lines depict the generation and use of clean rollouts from the clean (original) input $(I, \mathbf{q})$, while dashed lines depict the generation and use of noisy rollouts from the corresponding noisy input $(\tilde{I}, \mathbf{q})$. The distorted image $\tilde{I}$ is obtained by applying a distortion function $\tilde{I} = T_{\alpha_t}(I)$ with distortion strength α_t. The distortion level α_t is controlled by a noise annealing schedule, which gradually decreases distortion during training. Rollouts from both sources are mixed to form the final trajectories $\{\mathbf{o}_i\}_{i=1}^{n_1+n_2}$, rewards $\{r_i\}_{i=1}^{n_1+n_2}$, and advantages $\{A_i\}_{i=1}^{n_1+n_2}$. Crucially, policy optimization conditions only on clean inputs $(I, \mathbf{q})$; the corresponding noisy inputs $(\tilde{I}, \mathbf{q})$ are used solely to collect diverse rollouts for exploration.

---

*Equal contribution. †Correspondence to Tianyu Pang.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

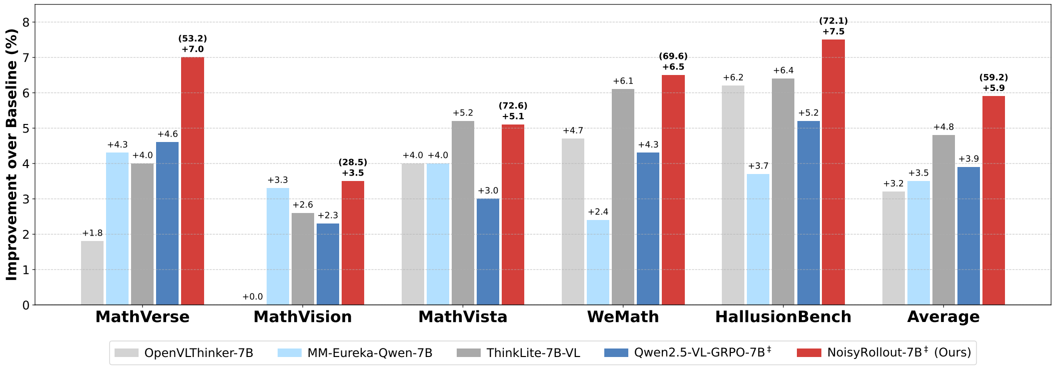


Figure 2: Accuracy improvement over Qwen2.5-VL-7B-Instruct on 5 out-of-domain benchmarks, covering both visual reasoning tasks (from MathVerse to WeMath) and a visual perception task (HallusionBench). Both Qwen2.5-VL-GRPO-7B and NoisyRollout-7B are fine-tuned by ourselves (denoted with ‡) using vanilla GRPO with only **2.1K** training samples from Geometry3K. The exact accuracy of NoisyRollout-7B is annotated above each corresponding bar in parentheses.

1 Introduction

Scaling test-time compute—often referred to as *reasoning*—through reinforcement learning (RL) has emerged as a promising axis for advancing model intelligence [25, 11]. While this idea has been primarily explored in the context of large language models (LLMs) [21, 90], the vision-language model (VLM) community is also actively investigating this direction [46, 56, 53]. Recent endeavours suggests that VLMs can also benefit from RL-driven scaling of test-time compute [23, 44, 72, 49, 86].

However, scaling test-time compute via RL requires more than sheerly generating longer outputs [45], and VLMs face unique challenges in this process. A key challenge is effective policy exploration, enabling policies to discover behaviors that generalize well beyond training data [87, 82]—an area largely underexplored in VLM research. Traditional practices, such as increasing rollout temperature to promote decoding diversity [89], often introduce superficial variability without meaningfully directing policies toward more robust or informative behaviors.

Moreover, VLMs inherently struggle with imperfect visual perception [40, 78], which negatively impacts subsequent reasoning processes [93, 97, 27]. Despite this, recent efforts [46, 53, 13] tend to adapt RL methods directly from the LLM domain. Such approaches often fail to take these perceptual challenges into consideration, thereby hindering the efficient development of visual reasoning capabilities through RL training.

Tackling the challenges of policy exploration and perceptual limitations in VLMs during RL training, we propose **NoisyRollout**, a simple yet powerful data augmentation technique for VLMs that introduces *meaningful rollout diversity*. Specifically, for each training sample consisting of an input image I and a corresponding text query $\mathbf{q}$, the old policy ($\pi_{\theta_{\text{old}}}$) produces two sets of rollouts based on the original clean image and a moderately distorted version of the same image, respectively.

While the current policy (π_θ) is updated solely by conditioning on the clean image and text query pair $(I, \mathbf{q})$, the two sets of rollouts form a group, collectively contributing to computing the reward baseline and normalized advantage in Group Relative Policy Optimization (GRPO) [64]. This *hybrid rollout strategy* enables the policy to achieve more targeted and efficient exploration, ultimately leading to *more robust visual reasoning* via RL through two key mechanisms:

❶ Successful reasoning trajectories from noisy inputs with distorted images reveal alternative, potentially more robust reasoning strategies, **improving reasoning generalization to harder or out-of-domain visual conditions**.

❷ When the same query yields different outcomes for clean and distorted inputs, the resulting reward differences expose *perceptual discrepancies* that affect reasoning. These discrepancies act as implicit contrastive signals, **helping refine the model's visual perception during reasoning** by constraining the negative perceptual exploration space.

While incorporating noisy rollouts can facilitate more effective and efficient exploration, it may also introduce instability in policy gradient estimation. To further enhance scalability and training stability, we employ a *noise annealing schedule* that gradually reduces the strength of image distortions over training. Such a strategy mitigates distributional mismatch between the evolving policy and the noisy trajectories generated from it when conditioned on clean inputs—an issue that often arises in later training stages—while retaining the benefits of noisy signals during the early phases of training.

We conduct extensive experiments to validate the effectiveness of NoisyRollout. Trained with only 2.1K samples from the Geometry3K [47] dataset using Qwen2.5-7B-VL-Instruct [4], Figure 2 shows that NoisyRollout achieves superior performance across 5 out-of-domain visual reasoning and perception benchmarks [48, 19, 93, 74, 57] (MathVerse 53.2%, MathVision 28.5%, and HallusionBench 72.1%). It outperforms both open-source RL-tuned models [53, 77] and those utilizing large-scale supervised fine-tuning (SFT) before RL [83, 92, 13]. Furthermore, it consistently surpasses its direct baseline (vanilla GRPO) on both in-domain and out-of-domain tasks, all within a fixed total rollout budget. Crucially, these out-of-domain improvements generalize across different model sizes (e.g., 7B to 32B) as well as training corpora and data scales (e.g., MMK12 [53] with 1K to 6K samples). These empirical results, combined with its simplicity and lightweight characteristics, establish NoisyRollout as a potentially *scalable* approach.

2 NoisyRollout: A Free-Lunch with Noisy Reinforcement Learning

We introduce NoisyRollout, a data augmentation method that enhances visual reasoning in VLMs during RL training, particularly by improving the rollout diversity for better policy exploration. NoisyRollout achieves this by incorporating a hybrid rollout strategy that leverages reasoning trajectories from both clean and distorted images, and a noise annealing schedule that progressively reduces distortion strength. These designs require no additional training cost and integrate seamlessly with standard GRPO implementations. A simplified overview is provided in Figure 1 and Algorithm 1.

GRPO. Group Relative Policy Optimization (GRPO) [64] was originally developed to improve mathematical reasoning in LLMs but can also be effectively adapted to enhance visual reasoning in VLMs. For a given input pair $(I, \mathbf{q})$ consisting of an image and text query from the training set $p_{\mathcal{D}}$, a rule-based outcome reward function $r(I, \mathbf{q}, \mathbf{o})$ is adopted to avoid reward hacking. This function assigns $r(I, \mathbf{q}, \mathbf{o}) = 1$ if the generated response $\mathbf{o}$ correctly addresses the query (as verified by a parser) with the required format, and $r(I, \mathbf{q}, \mathbf{o}) = 0$ otherwise. For each input, the old policy $\pi_{\theta_{\text{old}}}$ generates n response rollouts. The baseline reward is then calculated as $\text{mean}(\mathbf{r})$, where $\mathbf{r} = \{r_i\}_{i=1}^{n} = \{r(I, \mathbf{q}, \mathbf{o}_i)\}_{i=1}^{n}$ represents the rewards for all rollouts. The normalized advantage for the i-th rollout is defined as $\hat{A}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$. Derived from PPO [62], the GRPO objective function is:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(I,\mathbf{q})\sim p_{\mathcal{D}}, \mathbf{o}\sim\pi_{\theta_{\text{old}}}(\cdot|I,\mathbf{q})} \left[\frac{1}{n}\sum_{i=1}^{n} \min\left(\frac{\pi_\theta(\mathbf{o}_i \mid I, \mathbf{q})}{\pi_{\theta_{\text{old}}}(\mathbf{o}_i \mid I, \mathbf{q})}\hat{A}_i, \text{clip}\left(\frac{\pi_\theta(\mathbf{o}_i \mid I, \mathbf{q})}{\pi_{\theta_{\text{old}}}(\mathbf{o}_i \mid I, \mathbf{q})}, 1-\epsilon, 1+\epsilon\right)\hat{A}_i\right)\right], \quad (1)$$

where π_θ is the current policy, $\epsilon > 0$ sets the clipping range. We omit the KL divergence constraint $\mathbb{D}_{\text{KL}}[\pi_\theta | \pi_{\theta_{\text{ref}}}]$ following recent practices in Meng et al. [53] and Liu et al. [45].

Hybrid rollout strategy. Building upon GRPO, NoisyRollout introduces a hybrid rollout strategy to enhance the rollout diversity. For each input pair $(I, \mathbf{q})$, we generate an augmented version of the image $\tilde{I}$ through a noise transformation function T_α parameterized by a distortion strength α, i.e., $\tilde{I} = T_\alpha(I)$. As illustrated in Figure 1, the old policy $\pi_{\theta_{\text{old}}}$ produces two sets of rollouts: n_1 responses conditioned on the clean input $(I, \mathbf{q})$, and n_2 responses conditioned on the corresponding noisy input $(\tilde{I}, \mathbf{q})$. All rollouts from both clean and distorted images are then combined into a single group for reward calculation, yielding $\mathbf{r} = \{r_i\}_{i=1}^{n_1+n_2} = \{r(I, \mathbf{q}, \mathbf{o}_j)\}_{j=1}^{n_1} \cup \{r(I, \mathbf{q}, \mathbf{o}_k)\}_{k=n_1+1}^{n_1+n_2}$. Crucially, the policy update step remains conditioned solely on the clean image I and query $\mathbf{q}$ for better policy exploration. We defer the discussion of optimizing noisy and clean trajectories on their

Algorithm 1 NoisyRollout: Noisy Reinforcement Fine-Tuning

```
Input: Current policy $\pi_\theta$, old policy $\pi_{\theta_{old}}$, dataset $p_\mathcal{D}$, training steps $t_{\text{max}}$, clean rollout number $n_1$, noisy rollout number $n_2$, clip parameter $\epsilon$, initial noise strength $\alpha_0$, noise scheduler $\eta(\cdot)$, noise transformation function $T(\cdot)$
for $t = 1$ to $t_{\text{max}}$ do
    Sample batch $(I, \mathbf{q}) \sim p_\mathcal{D}$
    Set noise strength $\alpha_t = \eta(\alpha_0, t, t_{\text{max}})$                         ▷ Annealing schedule
    Generate distorted images $\tilde{I} = T_{\alpha_t}(I)$
    Sample $\{\mathbf{o}_j\}_{j=1}^{n_1}$ from $\pi_{\theta_{\text{old}}}(\mathbf{o} \mid I, \mathbf{q})$                         ▷ Clean rollouts
    Sample $\{\mathbf{o}_k\}_{k=n_1+1}^{n_1+n_2}$ from $\pi_{\theta_{\text{old}}}(\mathbf{o} \mid \tilde{I}, \mathbf{q})$                         ▷ Noisy rollouts
    Compute rewards $r_i = r(I, \mathbf{q}, \mathbf{o}_i)$ for all $i \in \{1, \dots, n_1 + n_2\}$
    Compute advantages $\hat{A}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$, where $\mathbf{r} = \{r_i\}_{i=1}^{n_1+n_2}$
    Update policy using:
    $\mathcal{J}(\theta) = \mathbb{E}\Big[\frac{1}{n_1+n_2}\sum_{i=1}^{n_1+n_2} \min\Big(\frac{\pi_\theta(\mathbf{o}_i|I,\mathbf{q})}{\pi_{\theta_{\text{old}}}(\mathbf{o}_i|I,\mathbf{q})}\hat{A}_i, \text{clip}\big(\frac{\pi_\theta(\mathbf{o}_i|I,\mathbf{q})}{\pi_{\theta_{\text{old}}}(\mathbf{o}_i|I,\mathbf{q})}, 1-\epsilon, 1+\epsilon\big)\hat{A}_i\Big)\Big]$
    $\theta \leftarrow \theta - \nabla_\theta \mathcal{J}(\theta)$                         ▷ Update conditioned on clean images only
    $\theta_{\text{old}} \leftarrow \theta$                         ▷ Update old policy parameters
end for
```

corresponding inputs to Appendix B. The NoisyRollout objective function is defined as:

$$\mathcal{J}(\theta) = \mathbb{E}_{(I,\mathbf{q})\sim p_\mathcal{D}, \{\mathbf{o}_j\}_{j=1}^{n_1}\sim\pi_{\theta_{\text{old}}}(\cdot|I,\mathbf{q}), \{\mathbf{o}_k\}_{k=n_1+1}^{n_1+n_2}\sim\pi_{\theta_{\text{old}}}(\cdot|\tilde{I},\mathbf{q})}$$
$$\left[\frac{1}{n_1+n_2}\sum_{i=1}^{n_1+n_2} \min\left(\frac{\pi_\theta(\mathbf{o}_i \mid I,\mathbf{q})}{\pi_{\theta_{\text{old}}}(\mathbf{o}_i \mid I,\mathbf{q})}\hat{A}_i, \text{clip}\left(\frac{\pi_\theta(\mathbf{o}_i \mid I,\mathbf{q})}{\pi_{\theta_{\text{old}}}(\mathbf{o}_i \mid I,\mathbf{q})}, 1-\epsilon, 1+\epsilon\right)\hat{A}_i\right)\right]. \quad (2)$$

Noise annealing schedule. Applying fixed-strength distortions throughout training often leads to training instability, primarily due to a distributional mismatch between noisy rollouts and the evolving policy. To mitigate this, we introduce a noise annealing schedule $\eta(\cdot)$ that gradually reduces the distortion strength over time. Specifically, at training step t, the noise level is defined as $\alpha_t = \eta(\alpha_0, t, t_{\text{max}})$, where α_0 is the initial noise strength and t_{max} denotes the total number of training steps. As shown in Figure 1, the distorted image is then generated as $\tilde{I} = T_{\alpha_t}(I)$.

Consequently, this schedule keeps diverse and informative supervision signals early in training, when the policy is constrained by its perceptual capacity. As training progresses, the noise level α_t is gradually reduced, narrowing the gap between noisy rollouts ($\{\mathbf{o}_k\}_{k=n_1+1}^{n_1+n_2}$) and the trajectories that $\pi_{\theta_{\text{old}}}(\cdot|I,\mathbf{q})$ would typically produce. This decay helps mitigate abrupt distribution shifts after policy updates, which can arise from unstable or high-variance policy gradients. Over time, rollouts generated from $(\tilde{I}, \mathbf{q})$ become progressively more "on-policy" *w.r.t* the clean-input-conditioned policy $\pi_{\theta_{\text{old}}}(\cdot|I,\mathbf{q})$, fostering a smoother transition from exploration to exploitation in later training stages.

Summary. NoisyRollout aims to improve the visual reasoning abilities of VLMs by enhancing rollout diversity to enable more effective policy exploration during RL training. Built on top of GRPO, it introduces a *hybrid rollout strategy* and a *noise annealing schedule*. These additions require no extra training cost and preserve the original RL objective. This design offers several benefits:

- **Robust reasoning:** Positive trajectories[1] from distorted inputs offer alternative, and potentially more robust reasoning paths, improving generalization to challenging or out-of-domain visual conditions.
- **Contrastive perceptual signals:** When clean and distorted inputs yield divergent outcomes for the same text query, the resulting reward differences shape a better perceptual exploration space, serving as implicit contrastive signals that refine the model's perceptual behaviors during reasoning.
- **Stable training dynamics for better exploitation:** The noise annealing schedule enables a smooth transition from early-stage noisy signals to fully on-policy learning, mitigating distributional mismatch and ensuring stable convergence as the model gradually improves its perception and reasoning. This provides a solid foundation for further exploitation in the later stages of RL training.

---

[1] We regard a trajectory as positive if it receives a reward of 1.

Table 1: Performance comparison of VLMs with moderate parameter sizes on a suite of out-of-domain benchmarks. Accuracy scores (%) are reported for all benchmarks for clarity. Models marked with "$\star$" are evaluated using our evaluation suite. For R1-related models, the corresponding `reasoning` templates are used by default, while "$\dagger$" indicates results obtained using the `direct-answer` template. Data sizes used for SFT and RL are annotated in blue and red, respectively. The best value in each column is shown in **bold**, and the second-best is underlined.

Model	**Data Size**	**MathVerse**	**MathVision**	**MathVista**	**WeMath**	**HallusionBench**
Open-source						
InternVL-2.5-8B-Instruct [9]	-	39.5	19.7	64.4	-	67.3†
LLaVA-OneVision-7B [35]	-	26.2	-	63.2	-	48.4†
Kimi-VL-16B [29]	-	44.9	21.4	68.7	-	66.2†
URSA-8B [50]	-	45.7	26.2	59.8	-	-
Mulberry-7B [84]	-	-	-	63.1	-	-
R1-related (reinforcement learning with verifiable reward)						
R1-VL-7B [92]	260K+10K	40.0	24.7	63.5	-	-
Vision-R1-7B [23]	200K+10K	52.4	-	<u>73.5</u>	-	-
R1-OneVision-7B* [83]	155K+10K	46.1	22.5	63.9	62.1	65.6
OpenVLThinker-7B* [13]	35K+15K	48.0	25.0	71.5	67.8	70.8
MM-Eureka-Qwen-7B* [53]	15K	50.5	28.3	71.5	65.5	68.3
ADORA-7B* [20]	2.1K	50.1	27.6	71.1	67.1	53.1
ThinkLite-7B-VL* [77]	11K	50.2	27.6	72.7	69.2	71.0
VLAA-Thinker-7B* [5]	25K	49.9	26.9	68.8	67.9	68.6
Qwen2.5-VL-7B-Instruct* [4]	-	46.2	25.0	67.5	63.1	64.6 (71.2†)
+ Vanilla GRPO* ($n = 12$)	2.1K (Geometry3K)	50.8	27.3	70.5	67.4	69.8
+ NoisyRollout* ($n_1 = 6, n_2 = 6$)	2.1K (Geometry3K)	**53.2**	28.5	72.6	69.6	<u>72.1</u>
+ Vanilla GRPO* ($n = 12$)	6.4K (MMK12)	51.8	<u>29.4</u>	73.2	<u>70.2</u>	70.3
+ NoisyRollout* ($n_1 = 6, n_2 = 6$)	6.4K (MMK12)	<u>53.0</u>	**30.6**	**74.5**	**70.3**	**72.2**

Table 2: Performance comparison of VLMs with large parameter sizes on a suite of out-of-domain benchmarks. The notation and evaluation protocols are consistent with those described in Table 1.

Model	**#Data**	**MathVerse**	**MathVision**	**MathVista**	**WeMath**	**HallusionBench**
Close-source						
GPT-4o [24]	-	50.8	30.4	63.8	69.0	71.4†
Claude-3.5-Sonnet [3]	-	26.5	38.0	67.7	-	71.6†
Kimi1.5 [28]	-	-	38.6	74.9	-	-
Open-source						
InternVL-2.5-78B-Instruct [9]	-	51.7	32.2	72.3	-	72.9†
QVQ-72B-Preview [58]	-	-	35.9	71.4	-	-
Qwen2.5-VL-72B-Instruct [4]	-	-	38.1	74.8	-	71.9†
R1-related (RL-tuned with verifiable reward)						
MM-Eureka-Zero-38B [53]	9.4K	48.9	26.6	64.2	-	-
MM-Eureka-Qwen-32B* [53]	17K	56.5	39.8	76.7	76.7	71.4
Qwen2.5-VL-32B-Instruct* [4]	-	58.5	37.6	76.5	74.0	66.6
+ Vanilla GRPO* ($n = 8$)	2.1K (Geometry3K)	<u>58.9</u>	39.2	77.0	76.1	72.3
+ NoisyRollout* ($n_1 = 4, n_2 = 4$)	2.1K (Geometry3K)	<u>58.9</u>	39.9	**77.8**	<u>77.2</u>	**73.5**
+ Vanilla GRPO* ($n = 8$)	6.4K (MMK12)	<u>58.9</u>	<u>40.0</u>	76.7	76.9	72.1
+ NoisyRollout* ($n_1 = 4, n_2 = 4$)	6.4K (MMK12)	**59.3**	**41.6**	<u>77.4</u>	**77.6**	<u>73.2</u>

3 Experiments

Dataset. We use EasyR1 [96] as our reinforcement learning training framework, which is built on verl [65] and specifically designed for VLMs. Our experiments utilize two datasets: Geometry3K [47], focused on geometric problem solving, and MMK12 [53], covering diverse K-12 math topics. These datasets comprise 2.1K and 6.4K training samples respectively. We processed them by converting all questions from multiple-choice to free-form format to prevent reward hacking and model guessing.

Evaluation. We mainly evaluate model performance along two dimensions. First, we assess out-of-domain generalization across five benchmarks: four visual reasoning benchmarks, including MathVerse [93], MathVision [74], MathVista [48], and WeMath [57], as well as one visual perception benchmark, HallusionBench [19]. Second, we evaluate the in-domain performance of NoisyRollout by comparing it with the vanilla GRPO baseline on the Geometry3K test set.

Moreover, we develop an evaluation suite for consistent assessment of our trained checkpoints and most open-source R1-related checkpoints using vLLM [31] for accelerated inference (marked with * in Tables 1 and 2), while adopting reported results for others.[2] We employ *greedy decoding* for model inference and use Gemini-2.0-Flash-001 [17] as the judge model to parse generated responses.

Implementation details. Following prior work [53, 77], we initialize our policy models with Qwen2.5-VL-7/32B-Instruct, which exhibit strong foundational capabilities well-suited for subsequent RL training. All experiments are conducted using 8 A100 GPUs (40G for 7B model, 80G for 32B model). We keep the vision encoder frozen for training stability and parameter efficiency. For other general RL-related hyperparameters, we adopt the default settings from EasyR1: a global batch size of 128, a rollout batch size of 512, a rollout temperature of 1.0, and a learning rate of 1e$-$6. To prevent token-length bias, we compute the policy loss using the `token-mean` aggregation strategy.[3] For NoisyRollout-specific configurations, we adopt **Gaussian noise** as the default image distortion strategy, and apply a *sigmoid-shaped* annealing schedule:

$$\alpha_t = \eta(\alpha_0, t, t_{\max}) = \alpha_0 \cdot \left(1 - \frac{1}{1 + e^{-\lambda(t-\gamma)/t_{\max}}}\right), \tag{3}$$

where γ determines the midpoint of the annealing curve and λ controls its steepness. Figure 7 illustrates the visual effects of applying different levels of Gaussian noise to a clean image. We defer the discussion of unsuccessful image distortion strategies (e.g., cropping), noise annealing strategies (e.g., power, exponential), and proportions of noisy rollouts in total rollouts to Appendix A. The `reasoning` and `direct-answer` templates used in our experiments are shown in Appendix H. Additional implementation details regarding the number of training steps/epochs and the hyperparameters for image distortion and noise annealing are presented in Appendix J.

3.1 Main Results

Result 1: Out-of-domain generalization. When trained on the Geometry3K dataset using Qwen2.5-VL-7B-Instruct, NoisyRollout not only improves in-domain performance (Figure 3, lower left subplot), but more importantly, demonstrates strong out-of-domain generalization. As shown in Table 1, NoisyRollout achieves superior performance across five visual reasoning and perception benchmarks, consistently outperforming the vanilla GRPO baseline in every case. This advantage is further illustrated in Figure 3, which presents detailed comparisons across benchmarks as training progresses. Specifically, NoisyRollout achieves 53.2% on MathVerse, 28.5% on MathVision, and 69.6% on WeMath, surpassing existing R1-related baselines and even outperforming GPT-4o.

Moreover, while Qwen2.5-7B-VL-Instruct's perception accuracy on HallusionBench drops from 71.2% to 64.6% when switching from `direct-answer` to `reasoning` templates,[4] NoisyRollout achieves 72.1% with the `reasoning` prompt (compared to vanilla GRPO's 69.8%). The final subplot in Figure 3 further confirms that NoisyRollout enhances perception quality during reasoning, achieving a higher Bradley–Terry win rate over vanilla GRPO (See Appendix C for details). These results indicate that our hybrid rollout strategy enhances visual perception by promoting better policy exploration through vision-oriented inductive biases.

Result 2: Sample efficiency. NoisyRollout demonstrates exceptional data efficiency by generalizing with only **2.1K** training samples from Geometry3K, whereas comparable models require significantly more data or even additional SFT as warm-up training. For example, Table 1 indicates that OpenVLThinker-7B needs **35K** SFT samples and **15K** RL samples but reaches only 48.0% on MathVerse and 71.5% on MathVista. This efficiency stems from NoisyRollout's use of noisy training signals that foster targeted exploration during RL, enabling effective generalization from limited samples.

Result 3: Robustness across training datasets and model sizes. NoisyRollout consistently improves upon vanilla GRPO, demonstrating strong robustness across model sizes and training datasets. As shown in Tables 1 and 2, the 7B model trained on MMK12 achieves gains of 1.2%, 1.3%, and

---

[2] While we closely follow system (or format) prompts from relevant codebases or papers, minor result discrepancies may occur due to differences in judge models or inference engines, which we consider acceptable.

[3] The code implementation can be found at verl.

[4] This degradation caused by the `reasoning` template has also been observed in previous studies [6, 27].

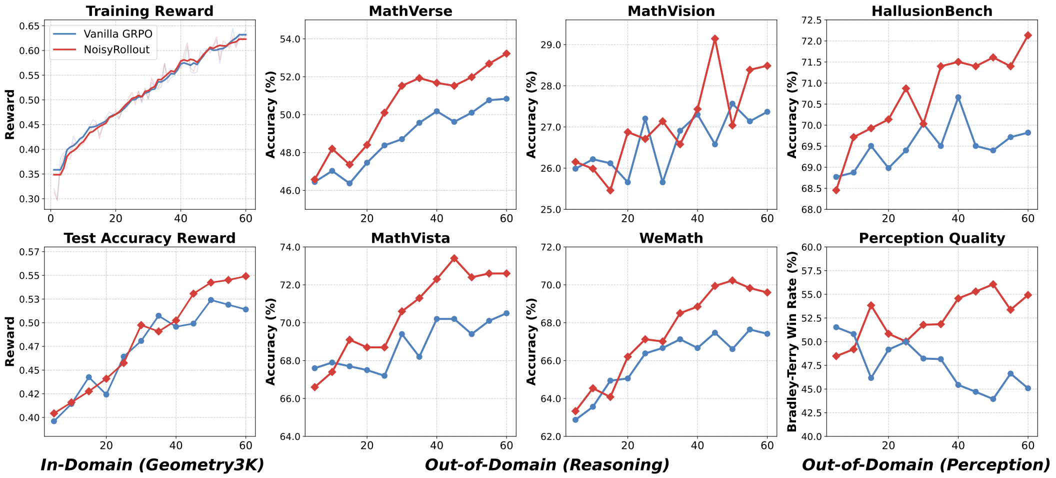


Figure 3: Comparison of NoisyRollout and vanilla GRPO on Qwen2.5-VL-7B-Instruct across in-domain and out-of-domain scenarios with the same total rollout number (12). The X-axis in all subplots represents *RL training steps*. **First column:** Reward comparison on the in-domain dataset during training. **Second and third columns:** Comparison on four out-of-domain visual reasoning benchmarks. **Last column:** Evaluation of visual perception capabilities, where the upper subplot directly compares their perception performance on HallusionBench and the lower subplot presents the model-ranked Bradley–Terry win rates *w.r.t.* the perception qualities of their reasoning traces.

1.9% over GRPO on MathVerse, MathVista, and HallusionBench, respectively. Similarly, the 32B model trained on MMK12 surpasses GRPO by 0.4%, 0.7%, and 1.1% on the same benchmarks.

Notably, in certain benchmarks like MathVerse and MathVista, the performance gains of NoisyRollout over vanilla GRPO are smaller for the 32B model than for the 7B model. This is likely because the 32B model's initial policy was already fine-tuned via RL,[5] whereas the 7B model's was not.

3.2 Ablation Study: More Effective Rollout Diversity with Noisy Trajectories

Setup. Unless otherwise specified, all ablation studies in this and the following subsection use Geometry3K as the training dataset on Qwen2.5-VL-7B-Instruct. In this part, we aim to examine the effectiveness of our NoisyRollout from the perspective of *rollout diversity*, a key factor for effective policy exploration in RL training. Here, we define rollout diversity as the average pairwise cosine distance between trajectory embeddings, where higher values indicate greater diversity. We randomly sample 256 instances from the Geometry3K training set. For each sample, we generate either $n = 12$ trajectories in vanilla GRPO or a combination of $n_1 = 6$ and $n_2 = 6$ trajectories in NoisyRollout, then encode them with an embedding model.[6] We track both diversity and accuracy across training steps (Figure 4) and evaluate final performance on in-domain and out-of-domain benchmarks (Table 3). We use vanilla GRPO with a rollout temperature of 1.0 as *the control group*.

Result. As shown in Figure 4, NoisyRollout enhances rollout diversity in early training stages compared to the control group, similar to increasing rollout temperature in vanilla GRPO from 1.0 to 1.2. This initial diversity boost, though accompanied by lower starting accuracy, ultimately leads to higher final training accuracy. Moreover, both NoisyRollout and higher-temperature vanilla GRPO show diversity decreasing below the control group in later training stages.

Table 3 reveals that NoisyRollout with temperature 1.0 consistently outperforms vanilla GRPO across all temperature settings (0.8 to 1.4), as well as mixed-temperature variants. Moreover, when applying temperature 1.2 to both approaches, NoisyRollout still demonstrates significant improvement over vanilla GRPO. These results indicate that NoisyRollout introduces more targeted and effective diversity than simply adjusting temperature parameters, which increases diversity in a less focused manner.

---

[5] https://qwenlm.github.io/blog/qwen2.5-vl-32b/

[6] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Table 3: Performance comparison under different rollout temperature settings, with the total number of rollouts fixed at 12. **In vanilla GRPO**, "$n(6)$: 1.0, $n(6)$: 1.2" indicates 6 rollouts with temperature 1.0 and another 6 with temperature 1.2. **In NoisyRollout**, "$n_1(6)$: 1.0" denotes 6 rollouts per sample generated from clean input $(I, \mathbf{q})$ with temperature 1.0, while "$n_2(6)$: 1.0" denotes 6 rollouts per sample from noisy input $(\tilde{I}, \mathbf{q})$ with temperature 1.0. "Geo3K" represents the test set of Geometry3K dataset. "Avg." represents average accuracy (%) across six benchmarks.

Method	Rollout Temperature	Geo3K	MathVerse	MathVision	MathVista	WeMath	HallusionBench	Avg.
Vanilla GRPO	$n(12)$: 0.8	50.1	50.5	26.7	69.9	65.8	70.1	55.5
	$n(12)$: 1.0	51.4	50.8	27.3	70.5	67.4	69.8	56.2
	$n(12)$: 1.1	50.4	50.2	27.7	70.4	68.1	69.4	56.0
	$n(12)$: 1.2	53.2	51.2	27.1	69.3	68.3	70.9	56.7
	$n(12)$: 1.4	51.4	50.6	25.8	70.1	69.0	69.6	56.1
	$n(6)$: 1.0, $n(6)$: 1.2	50.8	50.7	26.8	70.1	67.4	68.2	55.7
NoisyRollout	$n_1(6)$: 1.0, $n_2(6)$: 1.0	**54.9**	**53.2**	**28.5**	<u>72.6</u>	<u>69.6</u>	**72.1**	**58.5**
	$n_1(6)$: 1.2, $n_2(6)$: 1.2	<u>53.4</u>	<u>52.6</u>	<u>28.3</u>	**72.9**	**70.9**	<u>70.9</u>	<u>58.2</u>

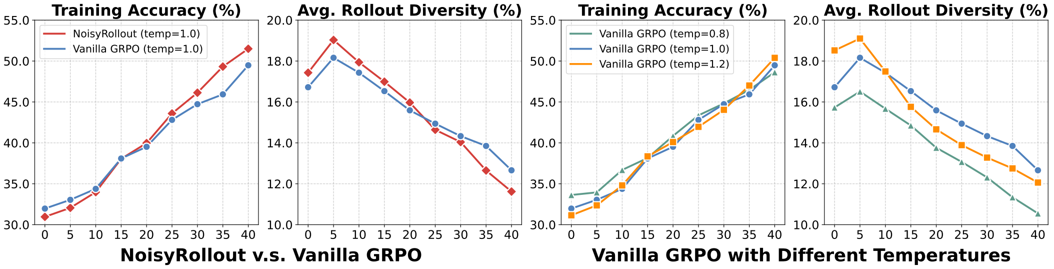


Figure 4: Comparison of accuracy and diversity metrics (%) across RL training steps (0 to 40). The left two subfigures contrast NoisyRollout versus vanilla GRPO (both with temperature 1.0), while the right two demonstrate the effects of different temperature settings (0.8, 1.0, 1.2) on vanilla GRPO.

3.3 Ablation Study: Impact of Hyperparameters and Module Design

Table 4: Ablation study on the noise annealing strategy.

Method	Geometry3K	OOD Avg.
Qwen2.5-VL-7B-Instruct	39.4	53.3
+ Vanilla GRPO	<u>51.4</u>	57.2
+ NoisyRollout w.o. Noise Annealing	43.9	<u>58.0</u>
+ NoisyRollout	**54.9**	**59.2**

Noise annealing. As shown in Figure 5, removing noise annealing causes the in-domain performance of our method to drop sharply around training step 45. This drop is due to divergence caused by a distributional mismatch—an issue discussed in Section 2 and further illustrated by the training dynamics in the same figure. Additionally, Table 4 shows that disabling noise annealing leads to lower performance in both in-domain and out-of-domain settings (43.9% and 58.0%, respectively), compared to our standard setting with noise annealing (54.9% and 59.2%). These results further highlight the effectiveness of noise annealing.

Table 5: Empirical validation with additional augmentation types.

Method	Augmentation	MathVerse	MathVision	MathVista	WeMath	HallusionBench	Average
GRPO	None	50.8	27.3	70.5	67.4	69.8	57.2
NoisyRollout	Gaussian noise	**53.2** (+2.4)	**28.5** (+1.2)	**72.6** (+2.1)	**69.6** (+2.2)	**72.1** (+2.3)	**59.2** (+2.0)
NoisyRollout	Rotation	52.5 (+1.7)	28.1 (+0.8)	71.9 (+1.4)	68.1 (+0.7)	70.2 (+0.4)	58.2 (+1.0)

Image augmentation type. To validate that the benefits of NoisyRollout are not limited to a single type of distortion, we evaluate its performance using rotation as a representative geometric transformation, in addition to our default Gaussian noise. The results, presented in Table 5, show that both augmentation strategies outperform the vanilla GRPO baseline. Specifically, using rotation boosts the average score from 57.2% to 58.2%. While Gaussian noise yields superior results with an average score of 59.2%, the meaningful gains from rotation demonstrate the generalizability of our approach. This suggests that the core mechanism of NoisyRollout —enhancing policy exploration through diverse visual inputs—is robust and effective across different augmentation techniques.

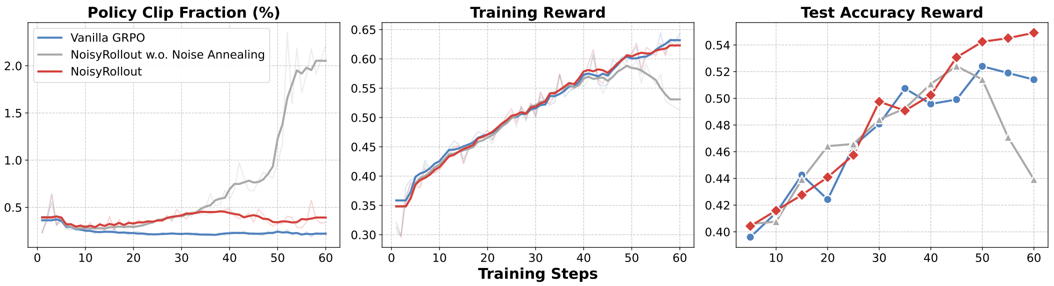


Figure 5: Comparison of NoisyRollout *w.* and *w.o.* noise annealing, and vanilla GRPO in terms of training dynamics (policy clip fraction and training reward) and accuracy on the in-domain test set.

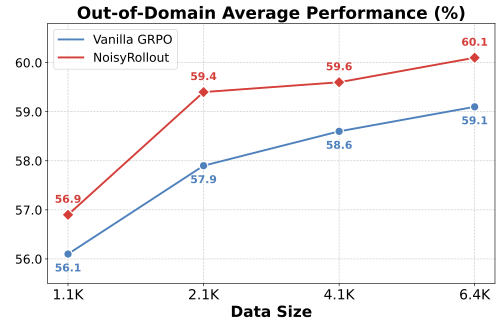


Figure 6: Performance comparison on MMK12 when scaling up the training data size.

Table 6: Ablation study on the impact of initial noise steps (α_0). "OOD Avg." represents average accuracy (%) across five out-of-domain benchmarks.

Noise Step	Geometry3K	OOD Avg.
0	51.4	57.2
100	52.7	57.4
300	53.4	57.7
400	54.6	58.1
500	**54.9**	**59.2**
550	39.6	57.7
600	Diverged	

Data scale. Although Geometry3K is a high-quality training dataset, its limited size (2.1K samples) prevents a thorough investigation of the scaling behavior of NoisyRollout compared to vanilla GRPO. To enable such analysis, we additionally consider MMK12, which contains 6.4K samples after preprocessing. Figure 6 shows NoisyRollout consistently outperforms vanilla GRPO across various data scales, ranging from 1.1K to 6.4K. Notably, the performance gains do not diminish as the dataset size increases, suggesting that NoisyRollout has strong potential for use in large-scale training regimes.

Initial noise step. We evaluate the impact of noise strength by varying the initial Gaussian noise step, as shown in Table 6. Gradually increasing the initial noise step α_0 from 0 to 500 *consistently* improves performance across all evaluation categories, suggesting that moderate noise promotes exploration and enriches the training signal. However, exceeding this threshold leads to performance degradation, as overly distorted images (see Figure 7) yield noisy rollouts with average near-zero rewards. These excessively noisy samples introduce harmful distribution shifts during policy updates, ultimately destabilizing the learning process. Additional ablation results on the MMK12 dataset are deferred to Appendix A.[7]

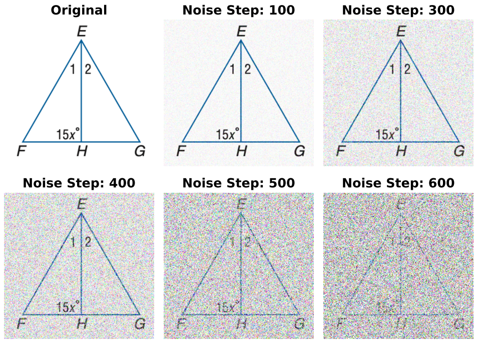


Figure 7: Illustration of visual degradation under increasing Gaussian noise steps.

GRPO variant. Recently, several variants have been proposed to enhance the original GRPO implementation. Specifically, Liu et al. [45] identified a question-level difficulty bias and proposed removing the standard deviation normalization ($\mathrm{std}(\mathbf{r})$) to address this issue. In addition, Yu et al. [87] increased the upper clipping threshold (ϵ_{high}) to mitigate entropy collapse. As shown in Table 7, applying NoisyRollout consistently improves performance not only on the original GRPO implementation but also across these variants. This highlights that NoisyRollout provides complementary benefits alongside optimization-focused modifications, underscoring its broad applicability.

---

[7] We also include additional ablations (e.g., **number of rollouts** and **data seed variations**) in Appendix A.

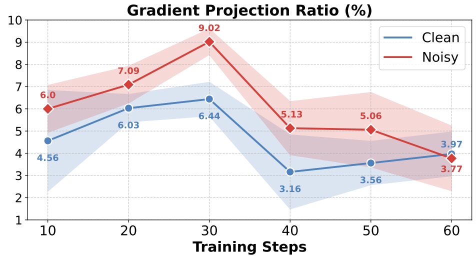


Figure 8: Comparison of gradient projection ratio over training.

Table 7: Performance comparison when using GRPO variants for policy optimization.

Method	Geometry3K	OOD Avg.
Qwen2.5-VL-7B-Instruct	39.4	53.3
+ GRPO (w.o. $\text{std}(\mathbf{r})$)	51.3	57.0
+ NoisyRollout (w.o. $\text{std}(\mathbf{r})$)	**56.1**	58.9
+ GRPO ($\epsilon_{\text{high}} = 0.28$)	52.6	58.2
+ NoisyRollout ($\epsilon_{\text{high}} = 0.28$)	53.9	**59.6**

3.4 Further Analysis: Quantitative Contribution of Noisy Rollouts on RL Optimization

Setup. For each training sample, we partition the collected rollouts into `Clean` and `Noisy` subgroups, containing n_1 and n_2 rollouts, respectively. We measure each subgroup's contribution by projecting its specific effective gradients onto an **anchor gradient** $\mathbf{g}^t = \theta^{t+\Delta t} - \theta^t$, which represents the overall model update over Δt optimization steps, beginning at training step t. The subgroup effective gradients, $\mathbf{g}^t_{\text{clean}}$ and $\mathbf{g}^t_{\text{noisy}}$, are derived from actual optimization steps starting from θ^t, using only rollouts from the respective subgroup (by masking losses from the other subgroup). The projection ratios are then calculated as $r^t_{\text{clean}} = (\mathbf{g}^t_{\text{clean}} \cdot \mathbf{g}^t)/\|\mathbf{g}^t\|^2$ and $r^t_{\text{noisy}} = (\mathbf{g}^t_{\text{noisy}} \cdot \mathbf{g}^t)/\|\mathbf{g}^t\|^2$. These ratios provide a quantitative estimate of each subgroup's contribution to the overall model update $\mathbf{g}^t$. More details are included in Appendix I.

Result. Figure 8 shows that the `Noisy` subgroup consistently contributes more significantly to policy optimization compared to `Clean`, especially during early training phases when distortion strength α_t is high and the policy π_θ still struggles with visual understanding. This trend gradually diminishes towards the final stages of training as the learning is gradually "on-policy". These findings quantitatively confirm that our method effectively leverages noisy rollouts to enhance training signals.

4 Related Work

VLMs have rapidly advanced through integrating vision encoders [59, 91] with large language models [2, 36, 41, 42, 24, 17], with specialized efforts in reasoning tasks [66, 94]. Reinforcement learning training in LLMs and VLMs, initially employed for alignment via human feedback (RLHF) [54, 1, 88], has evolved to incorporate rule-based rewards and advanced optimization methods like GRPO [64], as exemplified by DeepSeek-R1 [21] and Kimi-1.5 [28]. Emerging RL approaches in multimodal domains include LMM-R1 [56], Vision-R1 [23], R1-V [7], OpenVLThinker [13], and MM-Eureka [53], which extend RL to visual reasoning tasks. However, existing studies on training VLMs via RL have not adequately explored techniques that can enhance the explorative capabilities of models. Our method addresses this gap by proposing a data augmentation technique with visual-oriented inductive biases. A detailed discussion of related work is deferred to the Appendix D due to space limit.

5 Conclusion

In this paper, we investigate scaling test-time compute in VLMs via RL. We introduce NoisyRollout, a simple yet effective data augmentation technique that promotes diversity by mixing trajectories from both clean and distorted inputs with vision-oriented inductive biases. This approach enhances policy exploration during RL training without incurring additional training costs. Empirically, NoisyRollout demonstrates improved generalization and robustness, achieving state-of-the-art performance across multiple visual reasoning and perception benchmarks with high sample efficiency.

Acknowledgement

This project was partially supported by the Singapore Ministry of Education Academic Research Fund Tier 1 (Award Number: T1 251RES2514) and TPU Research Cloud. Additional computational support was provided by Sea AI Lab.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736, 2022.

[3] Anthropic. Claude 3.5 sonnet. `https://www.anthropic.com`, 2024.

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[5] Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. Sft or rl? an early investigation into training r1-like reasoning large vision-language models. `https://github.com/UCSC-VLAA/VLAA-Thinking`, 2025.

[6] Jiacheng Chen, Tianhao Liang, Sherman Siu, Zhengqing Wang, Kai Wang, Yubo Wang, Yuansheng Ni, Wang Zhu, Ziyan Jiang, Bohan Lyu, et al. Mega-bench: Scaling multimodal evaluation to over 500 real-world tasks. *arXiv preprint arXiv:2410.10563*, 2024.

[7] 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.

[8] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

[9] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[10] Chris, Yichen Wei, Yi Peng, Xiaokun Wang, Weijie Qiu, Wei Shen, Tianyidan Xie, Jiangbo Pei, Jianhao Zhang, Yunzhuo Hao, Xuchen Song, Yang Liu, and Yahui Zhou. Skywork r1v2: Multimodal hybrid reinforcement learning for reasoning, 2025. URL `https://arxiv.org/abs/2504.16656`.

[11] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards, 2025. URL `https://arxiv.org/abs/2502.01456`.

[12] Huilin Deng, Ding Zou, Rui Ma, Hongchen Luo, Yang Cao, and Yu Kang. Boosting the generalization and reasoning of vision language models with curriculum reinforcement learning. *arXiv preprint arXiv:2503.07065*, 2025.

[13] Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. Openvlthinker: An early exploration to complex vision-language reasoning via iterative self-improvement, 2025. URL `https://arxiv.org/abs/2503.17352`.

[14] Yuhao Dong, Zuyan Liu, Hai-Long Sun, Jingkang Yang, Winston Hu, Yongming Rao, and Ziwei Liu. Insight-v: Exploring long-chain visual reasoning with multimodal large language models. *arXiv preprint arXiv:2411.14432*, 2024.

[15] Difei Gao, Lei Ji, Luowei Zhou, Kevin Qinghong Lin, Joya Chen, Zihan Fan, and Mike Zheng Shou. Assistgpt: A general multi-modal assistant that can plan, execute, inspect, and learn, 2023. URL `https://arxiv.org/abs/2306.08640`.

[16] Yuying Ge, Sijie Zhao, Jinguo Zhu, Yixiao Ge, Kun Yi, Lin Song, Chen Li, Xiaohan Ding, and Ying Shan. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*, 2024.

[17] Gemini Team. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[18] Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

[19] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14375–14385, 2024.

[20] Lujun Gui and Qingnan Ren. Training reasoning model with dynamic advantage estimation on reinforcement learning. https://github.com/ShadeCloak/ADORA, 2025. Notion Blog.

[21] 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.

[22] Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. *arXiv preprint arXiv:2011.13389*, 2020.

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

[24] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[25] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[26] Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. *arXiv preprint arXiv:2505.00703*, 2025.

[27] Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanwei Li, Yu Qi, Xinyan Chen, Liuhui Wang, Jianhan Jin, Claire Guo, Shen Yan, et al. Mme-cot: Benchmarking chain-of-thought in large multimodal models for reasoning quality, robustness, and efficiency. *arXiv preprint arXiv:2502.09621*, 2025.

[28] Kimi Team. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599.

[29] Kimi Team. Kimi-VL technical report, 2025. URL https://arxiv.org/abs/2504.07491.

[30] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

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

[32] 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.

[33] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020.

[34] Bo Li, Kaichen Zhang, Hao Zhang, Dong Guo, Renrui Zhang, Feng Li, Yuanhan Zhang, Ziwei Liu, and Chunyuan Li. Llava-next: Stronger llms supercharge multimodal capabilities in the wild, May 2024. URL https://llava-vl.github.io/blog/2024-05-10-llava-next-stronger-llms/.

[35] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[36] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

[37] Weiqi Li, Xuanyu Zhang, Shijie Zhao, Yabin Zhang, Junlin Li, Li Zhang, and Jian Zhang. Q-insight: Understanding image quality via visual reinforcement learning. *arXiv preprint arXiv:2503.22679*, 2025.

[38] Yunxin Li, Shenyuan Jiang, Baotian Hu, Longyue Wang, Wanqi Zhong, Wenhan Luo, Lin Ma, and Min Zhang. Uni-moe: Scaling unified multimodal llms with mixture of experts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[39] Yuting Li, Lai Wei, Kaipeng Zheng, Jingyuan Huang, Guilin Li, Bo Wang, Linghe Kong, Lichao Sun, and Weiran Huang. Revisiting visual understanding in multimodal reasoning through a lens of image perturbation, 2025. URL https://arxiv.org/abs/2506.09736.

[40] Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*, 2024.

[41] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[42] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024.

[43] Xiangyan Liu, Rongxue Li, Wei Ji, and Tao Lin. Towards robust multi-modal reasoning via model selection, 2024. URL https://arxiv.org/abs/2310.08446.

[44] 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.

[45] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

[46] 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.

[47] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*, 2021.

[48] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

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

[50] Ruilin Luo, Zhuofan Zheng, Yifan Wang, Yiyao Yu, Xinzhe Ni, Zicheng Lin, Jin Zeng, and Yujiu Yang. Ursa: Understanding and verifying chain-of-thought reasoning in multimodal mathematics. *arXiv preprint arXiv:2501.04686*, 2025.

[51] Yan Ma, Steffi Chern, Xuyang Shen, Yiran Zhong, and Pengfei Liu. Rethinking rl scaling for vision language models: A transparent, from-scratch framework and comprehensive evaluation scheme. *arXiv preprint arXiv:2504.02587*, 2025.

[52] Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai yu, Liang Zhao, Yisong Wang, Jiaying Liu, and Chong Ruan. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation, 2024.

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

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

[55] Yi Peng, Chris, Xiaokun Wang, Yichen Wei, Jiangbo Pei, Weijie Qiu, Ai Jian, Yunzhuo Hao, Jiachun Pan, Tianyidan Xie, Li Ge, Rongxian Zhuang, Xuchen Song, Yang Liu, and Yahui Zhou. Skywork r1v: Pioneering multimodal reasoning with chain-of-thought, 2025. URL https://arxiv.org/abs/2504.05599.

[56] Yingzhe Peng, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl. *arXiv preprint arXiv:2503.07536*, 2025.

[57] Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*, 2024.

[58] Qwen. Qvq: To see the world with wisdom. https://qwenlm.github.io/blog/qvq-72b-preview/, 2024.

[59] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pp. 8748–8763. PMLR, 2021.

[60] Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 34:5402–5415, 2021.

[61] Fereshteh Sadeghi and Sergey Levine. CAD^{2}RL: Real single-image flight without a single real image. In *Robotics: Science and Systems*, 2017.

[62] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[63] Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

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

[65] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

[66] Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. *arXiv preprint arXiv:2406.17294*, 2024.

[67] Bharat Singh, Rajesh Kumar, and Vinay Pratap Singh. Reinforcement learning in robotic applications: a comprehensive survey. *Artificial Intelligence Review*, pp. 945–990, 2022.

[68] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 5639–5650. PMLR, 2020.

[69] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

[70] Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, Hisham Cholakkal, Ivan Laptev, Mubarak Shah, Fahad Shahbaz Khan, and Salman Khan. Llamav-o1: Rethinking step-by-step visual reasoning in llms, 2025. URL `https://arxiv.org/abs/2501.06186`.

[71] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

[72] Haonan Wang, Chao Du, and Tianyu Pang. V1: Toward multimodal reasoning by designing auxiliary tasks, 2025. URL `https://v1-videoreasoning.notion.site`.

[73] Junke Wang, Zhi Tian, Xun Wang, Xinyu Zhang, Weilin Huang, Zuxuan Wu, and Yu-Gang Jiang. Simplear: Pushing the frontier of autoregressive visual generation through pretraining, sft, and rl. *arXiv preprint arXiv:2504.11455*, 2025.

[74] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024.

[75] 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, 2025. URL `https://arxiv.org/abs/2411.10442`.

[76] Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, et al. Visualprm: An effective process reward model for multimodal reasoning. *arXiv preprint arXiv:2503.10291*, 2025.

[77] Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement, 2025. URL `https://arxiv.org/abs/2504.07934`.

[78] Haoran Wei, Youyang Yin, Yumeng Li, Jia Wang, Liang Zhao, Jianjian Sun, Zheng Ge, Xiangyu Zhang, and Daxin Jiang. Slow perception: Let's perceive geometric figures step-by-step. *arXiv preprint arXiv:2412.20631*, 2024.

[79] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024.

[80] Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, et al. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024.

[81] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

[82] Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance, 2025. URL `https://arxiv.org/abs/2504.14945`.

[83] Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*, 2025.

[84] Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, et al. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search. *arXiv preprint arXiv:2412.18319*, 2024.

[85] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021.

[86] En Yu, Kangheng Lin, Liang Zhao, Jisheng Yin, Yana Wei, Yuang Peng, Haoran Wei, Jianjian Sun, Chunrui Han, Zheng Ge, Xiangyu Zhang, Daxin Jiang, Jingyu Wang, and Wenbing Tao. Perception-r1: Pioneering perception policy with reinforcement learning, 2025. URL `https://arxiv.org/abs/2504.07954`.

[87] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[88] Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, and Tat-Seng Chua. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback, 2024. URL `https://arxiv.org/abs/2312.00849`.

[89] Weihao Zeng, Yuzhen Huang, Lulu Zhao, Yijun Wang, Zifei Shan, and Junxian He. B-star: Monitoring and balancing exploration and exploitation in self-taught reasoners. *arXiv preprint arXiv:2412.17256*, 2024.

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

[91] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

[92] Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. *arXiv preprint arXiv:2503.12937*, 2025.

[93] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pp. 169–186. Springer, 2024.

[94] Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Yichi Zhang, Ziyu Guo, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, Shanghang Zhang, et al. Mavis: Mathematical visual instruction tuning. *arXiv e-prints*, pp. arXiv–2407, 2024.

[95] Yi-Fan Zhang, Tao Yu, Haochen Tian, Chaoyou Fu, Peiyan Li, Jianshu Zeng, Wulin Xie, Yang Shi, Huanyu Zhang, Junkang Wu, et al. Mm-rlhf: The next step forward in multimodal llm alignment. *arXiv preprint arXiv:2502.10391*, 2025.

[96] Yaowei Zheng, Junting Lu, Shenzhi Wang, Zhangchi Feng, Dongdong Kuang, and Yuwen Xiong. Easyr1: An efficient, scalable, multi-modality rl training framework. `https://github.com/hiyouga/EasyR1`, 2025.

[97] Wenwen Zhuang, Xin Huang, Xiantao Zhang, and Jin Zeng. Math-puma: Progressive upward multimodal alignment to enhance mathematical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 26183–26191, 2025.

[98] Zhuofan Zong, Bingqi Ma, Dazhong Shen, Guanglu Song, Hao Shao, Dongzhi Jiang, Hongsheng Li, and Yu Liu. Mova: Adapting mixture of vision experts to multimodal context. *arXiv preprint arXiv:2404.13046*, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The introduction (Section 1) clearly outlines the motivation, problem setting, proposed method (NoisyRollout), and the key contributions. These are consistent with the experimental results and scope discussed throughout the paper (Section 3).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of NoisyRollout are detailed in Appendix F.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The training details of NoisyRollout are included in Appendix J. The details of used datasets are included in Section 3.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and running instructions are provided in the supplementary materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The details of training and evaluation are included in Section 3 and Appendix J.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: NoisyRollout requires large-scale reinforcement learning (RL) training. Similar to previous related works [53, 13, 23, 83], we do not report error bars due to the high computational cost. However, in certain ablation and analytical experiments—where computational demands are more manageable—we perform multiple independent runs to ensure statistical significance (e.g., Table 10 and Figure 8).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The details of compute resources are included in Table 13.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the Code of Ethics and it conforms with the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The details of broader impacts are included in Appendix E.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our model and data focus on vision-language understanding and reasoning, with minimal risk of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The licenses are mentioned in Appendix K.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code and model are provided in the supplementary materials.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components. LLMs are used solely for editing (e.g., grammar, spelling, word choice) and data processing/filtering.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Additional Ablation Studies

Noise annealing strategy. On the Geometry3K training dataset, we further examine the impact of different noise annealing schedules on NoisyRollout's performance by comparing our default sigmoid strategy with power ($\alpha_t = \alpha_0 \cdot (1 - t/t_{\max})^p$, $p = 3.0$) and exponential ($\alpha_t = \alpha_0 \cdot \gamma^{t/t_{\max}}$, $\gamma = 0.98$) decay functions. As shown in Table 8, all three strategies enable NoisyRollout to outperform the vanilla GRPO baseline on benchmarks. Among them, the sigmoid schedule achieves the highest average score (58.5%), surpassing both power and exponential decays (57.1% and 57.0%). The superior performance of the sigmoid schedule likely results from its characteristic "slow-fast-slow" decay, which balances exploration and stability more effectively by maintaining sufficient early exploration and promoting rapid convergence afterward.

Table 8: Ablation study on the strategies of noise annealing. "Avg." denotes the average accuracy across the six benchmarks. Best value per column is **bold**, second best is underlined.

Method	Geo3K	MathVerse	MathVision	MathVista	WeMath	HalluBench	Avg.
Qwen2.5-VL-7B-Instruct	39.4	46.2	25.0	67.5	63.1	64.6	51.0
+ GRPO	52.0	50.8	27.3	70.5	67.4	69.8	56.3
+ NoisyRollout (Pow.)	52.2	52.1	26.4	72.0	68.7	71.0	57.1
+ NoisyRollout (Exp.)	51.9	52.6	27.7	70.5	**70.1**	69.1	57.0
+ NoisyRollout (Sig.)	**54.9**	**53.2**	**28.5**	**72.6**	69.6	**72.1**	**58.5**

Total rollout number. We analyze the impact of total rollout number by comparing vanilla GRPO and NoisyRollout under varying rollout budgets. As shown in Table 9, increasing the number of rollouts in vanilla GRPO from $n = 8$ to $n = 16$ improves in-domain performance (from 49.6% to 54.7%), but only marginally benefits out-of-domain generalization (from 56.8% to 57.5%). NoisyRollout consistently outperforms vanilla GRPO even when the total number of rollouts is held constant. Notably, NoisyRollout with $n_1 = n_2 = 6$ (total 12) achieves both higher in-domain (54.9%) and out-of-domain (59.2%) accuracy than vanilla GRPO with 16 rollouts.

Table 9: Comparision of NoisyRollout and GRPO on Qwen2.5-VL-7B-Instruct across different rollout configurations. "OOD Avg." denotes the average accuracy across all five out-of-domain benchmarks.

Rollout	Geo3K	MathVerse	MathVision	MathVista	WeMath	HalluBench	OOD Avg.
$n = 8$	49.6	50.2	27.3	68.8	68.1	69.7	56.8
$n = 12$	52.0	50.8	27.3	70.5	67.4	69.8	57.2
$n = 16$	54.7	51.3	27.5	71.4	68.0	69.3	57.5
$n = 20$	53.6	50.8	27.2	70.0	68.5	69.9	57.3
$n_1 = n_2 = 4$	51.7	51.8	27.4	**72.6**	69.5	70.3	58.3
$n_1 = n_2 = 6$	**54.9**	**53.2**	28.5	**72.6**	**69.6**	**72.1**	**59.2**
$n_1 = n_2 = 8$	54.7	52.6	**28.7**	72.0	69.1	72.0	58.9

Unsuccessful image data augmentation. In our experiments, we explored two classic image augmentation techniques: Gaussian noise and rotation with expansion (`expand=True`). Both proved effective as they introduce perceptual diversity while preserving all essential visual information. In contrast, we also investigated two augmentation strategies that were ultimately unsuccessful: cropping and rotation without expansion. The latter refers to rotating an image and then cropping it back to its original dimensions, which cuts off the corners of the rotated content. Both of these unsuccessful methods frequently resulted in **critical information loss**, as key parts of a problem or diagram were removed from the image. This led to rollouts with

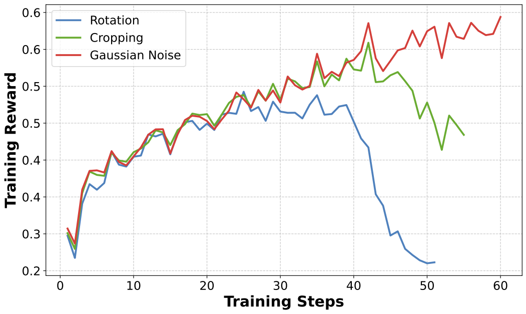


Figure 9: Performance of Rotation without expansion and cropping.

Table 10: Comparision of NoisyRollout and GRPO on Qwen2.5-VL-7B-Instruct across data seeds.

Method	Seed	Geo3K	MathVerse	MathVision	MathVista	WeMath	HalluBench	OOD Avg.
GRPO	1	52.0	50.8	27.3	70.5	67.4	69.8	57.2
NoisyRollout		**54.9**	53.2	**28.5**	72.6	69.6	**72.1**	**59.2**
GRPO	2	51.9	51.5	27.8	70.7	68.1	69.4	57.5
NoisyRollout		54.1	**53.6**	26.6	**72.9**	69.7	71.4	58.8
GRPO	3	51.1	50.8	27.1	69.7	67.8	69.5	57.0
NoisyRollout		53.7	52.4	27.8	72.3	**70.5**	70.9	58.8
GRPO	42	51.9	50.5	27.2	71.3	68.6	68.6	57.2
NoisyRollout		54.1	52.7	27.3	72.2	69.6	70.3	58.4

Table 11: Performance of NoisyRollout on the MMK12 dataset (2.1K) across different noise steps.

Method	Initial Noise Step	MathVerse	MathVision	MathVista	WeMath	HalluBench	OOD Avg.
GRPO	0	50.7	28.5	71.7	68.6	69.8	57.9
NoisyRollout	300	51.4	28.4	72.5	69.5	70.5	58.5
	400	50.3	28.2	71.8	70.3	70.8	58.3
	450	**52.8**	**28.9**	72.9	**71.9**	70.7	**59.4**
	500	50.2	27.4	**73.2**	71.4	**71.3**	58.7

consistently near-zero rewards, providing unreliable policy gradient estimates that caused training instability and eventual divergence. This distinction underscores a key finding: for NoisyRollout to be effective, the augmentation must introduce meaningful perceptual variance without fundamentally compromising the integrity of the input data.

Data seed. To evaluate the robustness of our approach to data sampling variations, we examine the performance consistency of NoisyRollout when using different random seeds for data sampling during training on the Geometry3K dataset. We compare these results against vanilla GRPO trained with the same seeds to verify that the improvements offered by NoisyRollout are consistent across different data orderings and not merely an artifact of a specific seed (Table 10).

Initial noise step (MMK12). Based on our experience, the initial noise step α_0 is a critical hyperparameter. In addition to the Geometry3K dataset, we also evaluate the impact of different initial noise steps on the MMK12 dataset using Qwen2.5-VL-7B-Instruct. Table 11 shows NoisyRollout outperforms standard GRPO on MMK12 when configured with an appropriate initial noise step ($\alpha_0 = 450$).

Proportion of noisy rollouts. Table 12 demonstrates that incorporating noisy rollouts during training significantly enhances model performance across both reasoning and perception benchmarks. A balanced $50/50$ distribution of clean and noisy rollouts achieves optimal results (58.5% average accuracy), outperforming both the no-noise baseline (56.3%) and higher noise proportions. This finding aligns with our earlier observations about the effectiveness of noise as a regularizer, confirming that the ideal approach combines clean rollouts for exploitation of current state with noisy rollouts for exploration, rather than relying exclusively on either strategy.

Table 12: Comparison across different proportions of noisy rollouts $n_2/(n_1 + n_2)$ during rollout collection, with a fixed total of 12 rollouts.

Proportion	Geo3K	MathVerse	MathVision	MathVista	WeMath	HalluBench	Avg.
$0/12$	52.0	50.8	27.3	70.5	67.4	69.8	56.3
$3/12$	51.1	52.8	**28.5**	71.8	69.8	70.6	57.4
$6/12$	**54.9**	**53.2**	**28.5**	**72.6**	69.6	72.1	**58.5**
$9/12$	54.2	52.1	27.3	72.3	**71.1**	**72.3**	57.9
$12/12$	53.1	52.8	**28.5**	72.3	69.1	71.1	57.8

B Unsuccessful Attempts

During the early development of NoisyRollout, we encountered several unsuccessful trials that are worth documenting. Due to limited computational resources, we could only explore a limited set of hyperparameter combinations heuristically. Some of these approaches might prove effective with further hyperparameter optimization and better design.

Optimizing noisy and clean trajectories on corresponding inputs. We explored an alternative design in which policy updates were conditioned on the same input used to generate each rollout—i.e., using clean inputs for clean rollouts and distorted inputs for noisy rollouts. However, this approach did not yield meaningful improvements over the original GRPO baseline. We hypothesize that this design reduces the benefits of group-based advantage estimation. Specifically, by decoupling clean and noisy rollouts during optimization, the method effectively degrades into a form of sample-level data augmentation. This fragmentation weakens the shared reward signal across rollouts, thereby diminishing the informativeness of group-level statistics such as the normalized advantage. As a result, the exploration benefits introduced by noisy rollouts are not fully leveraged during policy updates.

In contrast, our proposed approach treats clean and noisy rollouts as a unified group for advantage calculation, while anchoring all policy optimization to the clean inputs. This design retains the distributional diversity introduced by noise, but preserves a consistent input distribution for policy updates—striking a balance between exploration and stable learning.

Reward penalty on noisy subgroup. We experimented with applying explicit reward penalties (e.g., -0.1) to all noisy rollouts, aiming to encourage the model to better capture contrastive learning signals. However, this approach quickly led to training divergence. Rather than improving its core reasoning and perception abilities, the policy model learned to distinguish between clean and noisy rollouts. As a result, the noisy rollouts rapidly became highly "off-policy", since the model could easily identify. This distributional mismatch destabilized training and undermined the learning.

C Evaluating the Perception Quality of Reasoning Traces

Visual Information Extraction Prompt

Extract all visual perception and information recognition components from the following reasoning trace.
Original question: {question}
Reasoning trace: {reasoning}

Your task is to extract and summarize ONLY the parts that relate to visual perception, information extraction, and understanding of visual elements from the image.

This includes:
1. Any measurements, dimensions, or numerical values extracted from the image
2. Description of visual elements like shapes, objects, positions, or spatial relationships
3. Recognition of text, symbols, diagrams, or graphs from the image
4. Any visual features mentioned or used in the reasoning

Format your response with the tag:
<visual_perception> [Extracted visual information here] </visual_perception>

Include ONLY visual perception elements, not mathematical reasoning that happens after the information is extracted. If there are no clear visual perception elements, respond with "No clear visual perception elements identified."

Visual Perception Comparison Prompt

Compare the quality of visual perception between two models based on the image and the original question.
Original question: {question}
Visual perception from Model A: {visual A}
Visual perception from Model B: {visual B}

Your task is to determine which model better captures and correctly extracts visual information from the image. Compare their visual perception quality based on:

- Accuracy of visual information extraction (measurements, shapes, relationships)
- Complete identification of all relevant visual elements
- Proper recognition of visual information required to solve the problem

Score both models and determine the winner: If Model A demonstrates significantly better perception than Model B, respond: <result>A</result>; if Model B demonstrates significantly better perception than Model A, respond: <result>B</result>; if both models show similar quality of visual perception, respond: <result>tie</result>.

Now:
1. identify what visual information is required to solve this problem.
2. analyze how each model perceives this information.
3. provide your comparative judgment with specific reasons.
4. provide your <result> tag with exactly A, B, or tie.

To further evaluate the perception quality of models trained with NoisyRollout and vanilla GRPO during reasoning, we perform a paired comparison using a strong VLM.[8] We sample 300 reasoning traces from the evaluation logs of the models performing visual reasoning on the MathVerse and MathVista benchmarks, forming paired comparisons between NoisyRollout and vanilla GRPO.

To isolate visual perception, we extract only the visual components from each reasoning trace using a specialized prompt, removing any influence from mathematical reasoning or final answers. To reduce potential position bias in the comparisons, each pair of traces is evaluated twice: once with the

---

[8] Specifically, we use Gemini-Flash-2.0-001 in this experiment.

NoisyRollout trace shown first and the vanilla GRPO trace second, and once with the order reversed. We combine the results using the Bradley-Terry model to compute win rates. This methodology offers a reliable measure focused specifically on visual perception quality during reasoning. The results are presented in Figure 3 (the 8th subfigure). The extraction and evaluation prompts are shown above.

D Detailed Related Work

Large Vision-Language Models. VLMs have rapidly evolved to understand and reason with both visual and textual information [70]. These models combine visual encoders with large language models to enable comprehension and inference across modalities. Early VLMs like Flamingo [2] and BLIP-2 [36] established foundational integration techniques between vision and language components. The LLaVA series [41, 42, 35, 34] introduced effective visual instruction tuning methodologies that significantly advanced multimodal capabilities. For mathematical reasoning, specialized approaches [66, 94] have employed mathematical visual instruction tuning to enhance VLMs' abilities to interpret and solve mathematical problems in multimodal contexts.

Advanced VLMs including GPT-4o [24] and Gemini [17] have demonstrated unprecedented general visual understanding through massive pretraining. Mixture-of-Experts approaches in DeepSeek-VL-2 [80], Uni-MoE [38], and MoVA [98] improved computational efficiency by selectively activating specialized components based on input characteristics. Meanwhile, unified models like SEED-X [16], Chameleon [69], Show-o [81], and Janus series [79, 52, 8] integrated visual understanding and generation capabilities within single architectures. However, most existing VLMs still lack robust visual reasoning capabilities [14], especially for tasks requiring sophisticated analysis of visual information combined with complex reasoning [43, 75].

Reinforcement Learning-Enhanced Visual Reasoning. RL has emerged as a key methodology for enhancing the capabilities of LLMs and VLMs. Early research primarily focused on Reinforcement Learning from Human Feedback (RLHF) [54], which aligned model outputs with human preferences [1]. Recent advancements have further demonstrated that RL-based techniques can significantly enhance reasoning abilities. For instance, DeepSeek-R1 [21] utilizes rule-based rewards combined with Group Relative Policy Optimization (GRPO) [64], whereas Kimi-1.5 [28] employs a variant of online policy mirror descent, both methods showing notable improvements in reasoning performance.

In the multimodal domain, research on leveraging RL to enhance VLMs' reasoning capabilities remains in early stages. Some approaches explore using generative reward models [95, 76] to enhance VLMs' general capability, but these typically require powerful closed-source models for training data generation. Recent work including LMM-R1 [56], Vision-R1 [23], R1-V [7] and OpenVLThinker [13] has applied R1-type RL to VLMs in diverse specific subdomains like geometry problems and object counting tasks [55, 10, 12, 39, 37, 73, 26]. Further more, pilot studies [53, 51] further extend large-scale rule-based RL to broader multimodal mathematical reasoning, demonstrating significant performance gains without relying on in-domain training data.

Data Augmentation in Visual Reinforcement Learning. Data augmentation has played a central role in visual reinforcement learning (RL) from pixels by improving sample efficiency and generalization without architectural changes. RAD (Reinforcement Learning with Augmented Data) first demonstrated that simple image transformations—random crop, translation, color jitter, and cutout—can substantially enhance policy learning, particularly in DMControl and ProcGen benchmarks [33]. CURL combined off-policy RL with contrastive learning on augmented views, improving representation quality and sample efficiency [68]. DrQ regularized Q-value targets and policy updates via random shifts, achieving stable gains with minimal computational overhead [30]; DrQ-v2 further refined training schedules, exploration, and multi-step targets to master high-dimensional control from pixels [85].

Beyond these baselines, several extensions explored stronger invariance and generalization. DrAC introduced actor–critic regularizers that promote consistency under augmentation and automated augmentation search for diverse environments [60]. SVEA mitigated instability under heavy augmentations by mixing augmented and clean views to reduce target variance [22]. Self-Predictive Representations (SPR) encouraged temporal consistency between augmented latent states to improve representation quality and low-data efficiency [63]. In the sim-to-real context, domain randomization perturbs textures, lighting, and geometry to improve transferability from simulation to real-world environments [71, 61].

While these studies have established augmentation as a key ingredient for robust visual control, most methods optimize control policies rather than multimodal reasoning. Current augmentation pipelines are designed to stabilize perception and dynamics modeling but rarely connect augmented visual representations to higher-level symbolic or linguistic reasoning. Bridging this gap will likely require augmentations that preserve both control-relevant and semantic information, enabling joint optimization for perception, reasoning, and alignment in future multimodal agents.

E Broader Impacts

Our proposed method, **NoisyRollout**, introduces a simple yet powerful data augmentation approach designed to improve visual reasoning and perceptual robustness in VLMs. Given its effectiveness, especially noted through strong out-of-domain performance and high sample efficiency, this approach has broad applicability within resource-constrained training scenarios. This is particularly beneficial in domains where acquiring or annotating large-scale datasets is costly or practically challenging, such as medical imaging [32], robotic perception [67], and assistive technologies [15].

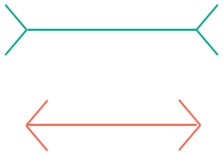
Moreover, by enhancing model robustness to visual conditions, our method can also facilitate safer and more reliable deployment of VLMs in real-world applications, potentially leading to more trustworthy human-AI interactions. Furthermore, as our method involves relatively simple augmentation steps without additional computational overhead or complex training protocols, along with strong performance on scaling experiments as shown in Table 2 and Figure 6, it is suitable for integration into existing large-scale training pipelines, supporting broader adoption in both academia and industry.

F Limitations

Despite easy-to-adopt designs and promising empirical results, our study has several limitations. **First**, due to computational constraints, our experiments are limited in scale: we primarily explore model sizes up to 32B parameters and training dataset scales in the order of a few thousand samples. Future work should validate and extend our findings using significantly larger-scale training scenarios—such as models with 72B parameters or training datasets in the range of hundreds of thousands of sample. **Second**, NoisyRollout is applied during the RL fine-tuning phase of an already pre-trained VLM. A more fundamental, but vastly more complex, direction would be to explore how the principles of NoisyRollout (i.e., learning from noisy signals in RL) could be integrated into the large-scale pre-training phase of the VLM itself. **Finally**, while empirically effective, our study lacks a formal theoretical analysis of how NoisyRollout, with its specific hybrid trajectory mixing and noise annealing, affects the exploration-exploitation trade-off and the convergence properties of the RL algorithm. It's unclear if the introduced noise guarantees broader state-space coverage in a principled way or if certain noise characteristics could inadvertently hinder convergence.

G Case Study

We present two case studies to demonstrate the improved perception and reasoning capabilities of our NoisyRollout compared to vanilla GRPO, as illustrated in Figure 10 and Figure 11, respectively.



Question: Is the green line longer than the red line in the image?

Source: HallusionBench, set_id:27, figure_id:0 question_id:1

Perception Issue

NoisyRollout

<think>
To determine if the green line is longer than the red line, let's analyze the image step by step:

1. The green line consists of two segments that meet at a point, forming a V-shape with the segments extending outward.
2. The red line is a single straight segment with arrows at both ends, indicating it extends in both directions but does not have any additional segments like the green line.

By visually comparing the two lines:
- The green line, when extended, appears to be the same length as the red line when considering the total length of the segments.
- The red line is a single, continuous segment without any additional segments, and it appears to be the same length as the green line when considering the total length.

Since both lines appear to be the same length, the green line is not longer than the red line.
</think>
\boxed{no}

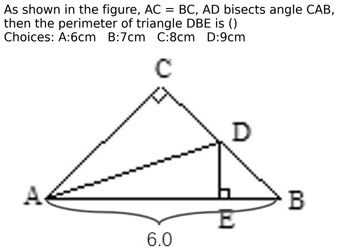
Vanilla GRPO

<think>
To determine if the green line is longer than the red line, I need to compare their lengths visually. The green line appears to be a straight line segment with two branches extending from the middle, forming a Y-shape. The red line is a single straight segment with arrows at both ends, indicating it is a bidirectional arrow but does not change its length.

By comparing the two, the green line extends further out from the central point, suggesting it is longer than the red line, which is a straight segment without any branches extending from it.

Therefore, the green line is longer than the red line.
</think>
\boxed{yes}

Figure 10: Case study showing improved perception capability of NoisyRollout over vanilla GRPO.



As shown in the figure, AC = BC, AD bisects angle CAB, then the perimeter of triangle DBE is ()
Choices: A:6cm B:7cm C:8cm D:9cm

Question: According to the question shown in the image, please first conduct reasoning, and then answer the question and provide the correct option letter, e.g., A, B, C, D, at the end.

Source: MathVerse-VisionOnly, id: 45

Reasoning Issue

NoisyRollout

<think>
Since AC = BC and AD bisects angle CAB, triangle CAD is congruent to triangle BAD by the SAS (Side-Angle-Side) congruence criterion. This means that CD = BD and AD = AD.

Given that AD bisects angle CAB, and since AC = BC, triangle ACD is isosceles with AC = BC. The perpendicular from D to AB at E implies that DE is the altitude and also the median in triangle ADB, making DE = CE.

The perimeter of triangle DBE is the sum of DB, BE, and DE. **Since CD = BD and DE = CE, we can express the perimeter as:**

$$DB + BE + DE = BD + BE + CE = BC + BE$$

Given that AB = 6 cm and AC = BC, and since AC = BC, we can infer that the segments are symmetrically divided. Therefore, the perimeter of triangle DBE is the sum of the segments that can be derived from the given information.

Since AC = BC and the segments are symmetric, the perimeter of triangle DBE is:

$$DB + BE + DE = 6 \text{ cm}$$

Thus, the perimeter of triangle DBE is 6 cm.
</think>
\boxed{A}

Vanilla GRPO

<think>
Given that AC = BC and AD bisects angle CAB, triangle ACD is congruent to triangle ABD by the SAS (Side-Angle-Side) congruence criterion. This means that AD = AD (common side), angle CAD = angle BAD (bisected angle), and AC = AB (given). Therefore, triangle ACD is congruent to triangle ABD.

Since AD bisects angle CAB, it also means that D is the foot of the perpendicular from A to BC, and E is the foot of the perpendicular from D to AB. This implies that DE is parallel to AC and BC, and thus DE is the harmonic mean of AC and BC, but more importantly, it means that DE is the altitude of triangle ABD and also the median from D to AB.

The perimeter of triangle DBE is the sum of the lengths of its sides: DB + BE + DE. Since D is the foot of the perpendicular from A to BC, and E is the foot of the perpendicular from D to AB, we can use the properties of the right triangles to find the lengths.

Given that AC = BC = 6 cm and AD bisects angle CAB, we can use the fact that the altitude from the vertex of the right angle in a right triangle splits the hypotenuse into segments that are proportional to the squares of the legs. However, a simpler approach is to recognize that since D is the foot of the perpendicular from A to BC, and E is the foot of the perpendicular from D to AB, the segments DB and BE are equal to half the hypotenuse of the right triangle ABD, which is 3 cm each.

Therefore, the perimeter of triangle DBE is DB + BE + DE = 3 + 3 + 3 = 9 cm.

So, the perimeter of triangle DBE is 9 cm.
</think>
\boxed{D}

Figure 11: Case study illustrating enhanced reasoning capability of NoisyRollout over vanilla GRPO.

H Templates

Reasoning Template from EasyR1

SYSTEM: You FIRST think about the reasoning process as an internal monologue and then provide the final answer.The reasoning process MUST BE enclosed within <think> </think> tags. The final answer MUST BE put in \boxed{}.
USER: {question}

Direct-Answer Template

SYSTEM: You are a helpful assistant
USER: {question}. Answer yes or no directly.

I Detailed Methodology for Gradient Contribution Analysis

To quantitatively assess the impact of noisy rollouts on the reinforcement learning (RL) optimization process, we partition the rollouts associated with each training sample into two distinct subgroups based on their input type: `Clean` and `Noisy`. Specifically, `Clean` rollouts are generated from original inputs $(I, \mathbf{q})$, whereas `Noisy` rollouts originate from distorted inputs $(\tilde{I}, \mathbf{q})$. In this experimental setup, each training sample comprises $n_1 = 6$ `Clean` rollouts and $n_2 = 6$ `Noisy` rollouts. For computational efficiency, all gradient calculations and parameter differences discussed below ($\mathbf{g}^t$, $\mathbf{g}^t_{\text{clean}}$, $\mathbf{g}^t_{\text{noisy}}$) are performed using only the parameters from specific model modules. The exact modules used are "`lm_head.weight`", "`model.layers.27.self_attn.o_proj`", "`model.layers.27.self_attn.q_proj`", "`model.layers.27.self_attn.k_proj`", and "`model.layers.27.self_attn.v_proj`".

To quantify the contribution of each subgroup to the optimization, we first define an **anchor gradient**, denoted $\mathbf{g}^t$. This quantity represents the effective overall update to the selected model parameters θ at a given training stage t. It is calculated as the difference in these parameters between checkpoints at training steps t and $t + \Delta t$:

$$\mathbf{g}^t = \theta^{t+\Delta t} - \theta^t,$$

where θ^t represents the selected model parameters at training step t, and we use $\Delta t = 5$ steps. This $\mathbf{g}^t$ reflects the actual change in these parameters resulting from the standard training procedure which utilizes losses derived from both `Clean` and `Noisy` rollouts from each training sample.

Subsequently, starting from the same parameter state θ^t, we isolate the influence of each subgroup. This involves performing Δt actual optimization steps under two modified conditions, using the same batch of training samples that contributed to the standard update from θ^t to $\theta^{t+\Delta t}$:

1. To obtain $\theta^{t+\Delta t}_{\text{clean}}$: Starting from θ^t, we performed Δt optimization steps. During these steps, the loss components arising from the `Noisy` rollouts were masked (i.e., their corresponding loss was set to zero). Thus, the gradients and subsequent parameter updates were derived solely from the `Clean` rollouts. This procedure yielded the updated parameters $\theta^{t+\Delta t}_{\text{clean}}$.
2. To obtain $\theta^{t+\Delta t}_{\text{noisy}}$: Similarly, starting from θ^t, we performed Δt optimization steps. In this case, the loss components arising from the `Clean` rollouts were masked. The gradients and subsequent parameter updates were therefore derived solely from the `Noisy` rollouts. This yielded updated parameters $\theta^{t+\Delta t}_{\text{noisy}}$.

From these updates, we define the subgroup-specific effective gradients (or parameter deltas) with respect to the selected modules:

$$\mathbf{g}^t_{\text{clean}} = \theta^{t+\Delta t}_{\text{clean}} - \theta^t$$

and

$$\mathbf{g}^t_{\text{noisy}} = \theta^{t+\Delta t}_{\text{noisy}} - \theta^t.$$

We then quantify the contribution of each subgroup by projecting its effective gradient onto the anchor gradient. The projection ratios are computed as follows:

$$r^t_{\text{clean}} = \frac{\mathbf{g}^t_{\text{clean}} \cdot \mathbf{g}^t}{\|\mathbf{g}^t\|^2} \quad \text{and} \quad r^t_{\text{noisy}} = \frac{\mathbf{g}^t_{\text{noisy}} \cdot \mathbf{g}^t}{\|\mathbf{g}^t\|^2}.$$

These ratios, r^t_{clean} and r^t_{noisy}, represent the estimated proportion of the anchor gradient $\mathbf{g}^t$ that can be attributed to the `Clean` and `Noisy` subgroups, respectively, at training stage t. To ensure robust estimates, all effective gradient quantities ($\mathbf{g}^t$, $\mathbf{g}^t_{\text{clean}}$, and $\mathbf{g}^t_{\text{noisy}}$) are determined by averaging results from 5 independent runs of the Δt-step update processes described above. Each run starts with the same parameters θ^t.

J Supplementary Implementation Details

This section provides the detailed hyperparameter configurations for our experiments that were omitted from Section 3. In Table 13, we summarize our experimental settings across different model sizes and datasets, with specific focus on image distortion parameters and noise annealing schedules.

Table 13: Summary of hyperparameter configurations.

Parameter	**Configuration**
General Settings (All Experiments)	
Model Base	Qwen2.5-VL-Instruct
Vision Encoder	Frozen
Global Batch Size	128
Rollout Batch Size	512
Rollout Temperature	1.0
Learning Rate	$1e-6$
Optimizer	AdamW
Policy Loss Aggregation	`token-mean`
Image Distortion Strategy	Gaussian noise
Noise Annealing Schedule	Sigmoid-shaped
CPU Memory	1TB
GPU	A100-SXM4-40/80GB
Qwen2.5-7B-VL-Instruct on Geometry3K (2.1K samples)	
Initial Noise (α_0)	500
Training Episodes	15
Total Optimization Steps (t_{max})	60
Sigmoid Midpoint (γ)	40
Sigmoid Steepness (λ)	30
Rollout Number	$n_1 = n_2 = 6$
Time Cost per Step	about 1100s
Qwen2.5-7B-VL-Instruct on MMK12 (6.4K samples)	
Initial Noise (α_0)	450
Training Episodes	12
Total Optimization Steps (t_{max})	120
Sigmoid Midpoint (γ)	40
Sigmoid Steepness (λ)	60
Rollout Number	$n_1 = n_2 = 6$
Time Cost per Step	about 1500s
Qwen2.5-32B-VL-Instruct on Geometry3K (2.1K samples)	
Initial Noise (α_0)	450
Training Episodes	10
Total Optimization Steps (t_{max})	40
Sigmoid Midpoint (γ)	35
Sigmoid Steepness (λ)	30
Rollout Number	$n_1 = n_2 = 4$
Time Cost per Step	about 3300s
Qwen2.5-32B-VL-Instruct on MMK12 (6.4K samples)	
Initial Noise (α_0)	450
Training Episodes	7
Total Optimization Steps (t_{max})	70
Sigmoid Midpoint (γ)	35
Sigmoid Steepness (λ)	30
Rollout Number	$n_1 = n_2 = 4$
Time Cost per Step	about 3300s

K Licenses

We use standard licenses from the community. We include the following licenses for the codes, datasets and models we used in this paper.

Datasets & Benchmarks:

- Geometry3K [47]: MIT
- MMK12 [53]: Apache License 2.0
- MathVerse [93]: MIT
- MathVision [74]: MIT
- MathVista [48]: Creative Commons Attribution Share Alike 4.0 International
- WeMath [57]: CC BY-NC 4.0

Codes:

- verl [65]: Apache License 2.0
- EasyR1 [96]: Apache License 2.0

Models:

- Qwen2.5-VL-7B-Instruct [4]: Apache License 2.0
- Qwen2.5-VL-32B-Instruct [4]: Apache License 2.0
- Gemini API [18]: Gemini API Additional Terms of Service