Fine-grained VLMs Within Reinforcement Learning Frameworks

Bogdan Perju[1]

Vrije Universiteit Amsterdam, Amsterdam, Netherlands
b.perju@student.vu.nl

Abstract. Recent developments in reasoning-based vision language models have shown significant improvements through reinforcement learning optimization, in particular with Group Relative Policy Optimization (GRPO). This work investigates the application of GRPO to spatial tasks within the LEGO-ARTA dataset.

This paper evaluates three models: a few-shot baseline (Qwen2.5VL-7B-Instruct), an SFT trained reference model, and a LEGO-ARTA trained model using GRPO optimization. We implement custom reward functions for each separate task, targeting reasoning format, spatial understanding and alignment, and object detection accuracy.

Results demonstrate performant reward dynamics, with GRPO displaying improved IoU scores over both the baseline (65.49 vs 60.82 for the base model) and the SFT method. While IoU optimization maintained strong spatial performance, careful reward balancing was required to preserve language precision alongside spatial accuracy.

The findings provide valuable insights into applying reinforcement learning to vision-language tasks, especially in domains requiring spatial understanding.

1 Introduction and Motivation

Recent advances in the field of LLMs have demonstrated remarkable capabilities in reasoning-based tasks, with systems like DeepSeek-R1 [5] achieving state-of-the-art performance across complex mathematical, logical, and analytical tasks. These reasoning models have shown that explicit chain-of-thought [17] thinking processes, when implemented through structured reasoning templates and reinforcement learning optimization, can enhance model performance beyond supervised fine-tuning approaches.

Building upon this foundation, the field has witnessed an evolution toward multi-modal reasoning capabilities with the rise of projects such as VLM-R1 [14] and Visual-RFT [10] that incorporate similar reasoning mechanics to handle complex vision-language tasks. This presents major opportunities for addressing challenges within computer vision tasks, particularly in domains that require both visual and spatial understanding.

Object detection is a task in which a Vision-Language Model (VLM) must determine the coordinates of a bounding box of a named object. Observations from previous developments suggest that a failure mode in such tasks is a low Intersection over Union (IoU) between the ground truth and predicted bounding boxes. Such cases signal a clear problem in assembly tasks due to the dependence on communicating clear object locations in situations where users require special aid.

Beyond object detection, assembly tasks require models to understand procedural instructions through grounding (interpreting temporal/step instructions from visual context) and state validation (determining whether assembly steps have been completed correctly). These additional capabilities form a comprehensive framework for integrating VLMs as tools for assembly assistance. We aim to unveil which method is most suitable for these tasks.

The working hypothesis in this paper is that VLMs exhibit a low IoU due to a lack of pre-training on domain data and high-complexity cases in which SFT (Supervised Fine-Tuning) may not be enough. This paper proposes to address this problem by applying RL (Reinforcement Learning) [16] with custom reward functions to fine-tune an R1-Style VLM using GRPO (Group Relative Policy Optimization) [13]. The motivation is that a GRPO-trained model can learn better spatial and contextual patterns in domain data, leading to improved IoU and F1 scores.

Additional observations show that VLMs exhibit high FPR rates in state detection tasks and overall low scene understanding scores. Our aim is to develop a unified reward system through which we can improve all three tasks.

2 Research Questions

1. Can GRPO outperform SFT for fine-grained assembly tasks?
2. What is the trade-off between using SFT and GRPO?
3. How does reward modeling influence learning dynamics of GRPO-trained VLMs?

3 Preliminary

Reinforcement Learning (RL) [16] is an approach in which agents learn by interacting with an environment and receiving feedback in the form of scalar rewards. The agent's goal is to learn a policy $\pi_\theta(a|s)$ that maximizes the expected cumulative reward over time based on its actions. Policy gradient methods are a class of RL algorithms where the agent directly learns a policy by adjusting its parameters to increase expected returns.

Among policy gradient methods, Group Relative Policy Optimization (GRPO) [13] introduces a novel concept where the value baseline used to evaluate actions is replaced with a comparative baseline computed across multiple sampled outputs (group-relative). In GRPO, multiple outputs are generated for each query, and the agent learns by comparing the rewards against each other rather than relying on an absolute value estimation. This approach has proven particularly effective in training reasoning models such as DeepSeek-R1 [5], whose rewards are explicitly designed to encourage reasoning and formatting via format and language consistency rewards.

The VLM-R1 project [14] leverages this methodology and applies it to Vision-Language reasoning tasks. For the object detection task in particular, the VLM-R1 project implements GRPO with 2 reward functions: format and accuracy. The format reward encourages the model to output rationale in <thinking> tags, while the accuracy reward optimizes the IoU score of bounding box predictions.

4 Methodology

4.1 Dataset

Within the scope of this project, the dataset used will be the LEGO-ARTA dataset [7], a synthetic multimodal dataset designed to evaluate and improve vision-language models (VLMs) in augmented reality (AR) assembly tasks. The dataset introduces three tasks: scene understanding (grounding) , object detection, and state detection, each targeting a different aspect of instruction-following in assembly scenarios.

The focus of this work is on improving the IoU score in the **object detection task**, which requires the model to identify specific LEGO pieces within an image and output their bounding box coordinates using the format:

```
<p>object</p>{<Xleft><Ytop><Xright><Ybottom>}
```

with coordinates normalized to $[0, 100]$.

The LEGO-ARTA dataset contains 35,612 total instances, of which 19,136 are dedicated to object detection, 5,612 to grounding and 10,864 to state detection. Given an instruction Q and corresponding image I, the VLM must correctly name the target object and provide its position in the specified format. Each query includes a task-specific token (e.g.,`[object]`), an instruction describing the object to locate (e.g.,`"Find a 1x2 white beam"`) and a fixed format directive to ensure structured output. The format directive used for the base model inference in the source paper is

"Providing the positions in the format:**[Format]** with X and Y coordinates normalized to [0,100]. <Xleft> and <Ytop> for the top-left corner. <Xright> and <Ybottom> for the bottom-right corner."

For the reasoning models' inference, the following instruction also gets prepended:

"Question First output the thinking process in <think> </think> tags and then output the final answer in <answer> </answer> tags."

This instruction nudges the model to conform to the reasoning template. Other publications in the field suggest that in structured output tasks models perform better when asked to output json format; likewise, such format is easier to validate, parse and design rewards for [15]. For the object detection task, we ask the model to output valid json, and we modified the original dataset to use the following format response:

```
['bbox_2d': [x1, y1, x2, y2], 'label': 'object name']
```

At inference time the model input is prepended with an example. This work introduces a dynamic demo selection as opposed to the original implementation such that the examples are chosen from the same manual to give the model contextual clues. Similarly, if there is a same-manual example that predicts the same number of bounding boxes as the ground truth answer, such an example is chosen to reduce instruction ambiguity.

For completeness, here is a brief description of the other two tasks:

Grounding Task Asks the model to describe what should be done in the current step based on the provided image and previous instruction. Queries are marked with the token `[grounding]`, and responses must align with the step-by-step manual instructions.

State Detection Task Evaluates whether the current image reflects a correct assembly state. Models receive a question asking if the step was completed correctly and must respond with "Yes" or "No". It uses the token `[state]` and helps assess the model's ability to assess procedural correctness.

During training, data is subsampled from a 75% split of the dataset. Evaluation is conducted on a held-out 25% test set.

4.2 Experimental Setup

The experimental setup presents three models evaluated on the LEGO-ARTA object detection task.

The first model serves as a few-shot baseline: Qwen2.5VL-7B-Instruct [1], tested without further fine-tuning on all three LEGO-ARTA tasks. This establishes initial performance metrics in terms of mean IoU and F1, while also providing benchmarks for the grounding and state detection tasks using F1, ROUGE, and BLEU for the grounding task, and F1 and FPR for the state detection task.

The second model, Qwen2.5VL-7B-Instruct was trained on the LEGO-ARTA dataset using LLaMA-Factory [19] via LoRA and PEFT [6] [18] on the 75% training partition. Trained in a similar way to the original paper, it provides a solid comparison between the methods.

Finally, the main experimental model, Qwen2.5VL-7B-Instruct trained on LEGO-ARTA via GRPO, is initialized from the Qwen base, then fine-tuned via LoRA and PEFT on the same partition as the SFT training. The implementation approach of the VLM-R1 project integrates two types of reward functions in this stage: a format reward (to encourage explicit chain-of-thought output in <think></think> tags) and an IoU-based spatial reward customized for the dataset-specific version of the task.

4.3 Reward Function Design

In order to adapt the VLM-R1 project for the dataset-specific object detection task, two custom reward function modules were written. Both the format reward and the IoU reward were adapted to accept multiple bounding box predictions, while 2 versions of the IoU reward were tested.

In early experiments, the model failed to output bounding box coordinates within the expected [0,100] normalized range when instructed. To address this, the following reward system was designed:

If any coordinate is outside [0,100], a penalty gets applied: the IoU is multiplied by 0.5. For every coordinate that is within [0,100], a bonus of +0.1 is added. This explicitly encourages normalization.

Since the dataset was synthetically generated via a different model (MiniGPT-v2 [3] [7]), the model sometimes predicted a different number of object boxes than expected. To handle this, **Hungarian matching** [9] was applied between predicted and ground truth boxes, which returned a +0.1 reward if the number of predicted boxes matched the ground truth count.

In order to improve the textual accuracy of the predicted labels, we formulated a ROUGE reward. This is meant to reward the model for predicting a label that matches closely to the ground truth on a token level.

All predictions are always post-processed to be within the [0,100] range during reward calculation via normalization. However, the reward is scaled based on whether the model's raw output was normalized before post-processing.

In the second reward configuration, the normalization penalty was removed entirely. The model received full IoU-based reward regardless of coordinate range. This allowed the observation of the effects of removing normalization pressure on learning dynamics.

For the state and grounding tasks, the following rewards were implemented:

State Detection: A binary accuracy reward system was implemented, providing +1.0 for correct yes/no predictions and 0.0 otherwise.

Grounding: A composite reward combining BLEU-4 and ROUGE-1 F1 scores was designed to capture both fluency and content accuracy in natural language generation. The scores are summed (maximum reward of 2.0) to provide stronger learning signals given the complexity of the task. The implementation uses NLTK [11] for tokenization and smoothing functions for BLEU calculation.

All reward functions implement automatic task type inference from solution format in order to enable dynamic reward selection within mixed training batches such as the multiple tasks in the LEGO-ARTA dataset. Each function activates only for its target task type and returns 0.0 for others, thus, preventing cross-task reward signal interference.

4.4 Technical Configuration

Inference Inference was conducted on a single A40 GPU. The model configuration used Flash-Attention-2 [4] and mixed precision (bf16) to save VRAM; batch processing with batch size 3 was used. Due to large image sizes often exceeding 3600 pixels, the model preprocessor max pixel size was set to 2600 pixels to balance quality and inference speed. The image input was subsequently pre-processed by the qwen-vl-utils [1] `process_vision_info` module.

Training All training was run on a single A100 GPU. LoRA [6] with rank 8 was used for parameter-efficient fine-tuning. Mixed precision (bf16) and Flash-Attention-2 were enabled to reduce memory usage. The vision backbone was frozen to focus adaptation on language and fusion modules. Training lasted 500 steps, using DeepSpeed ZeRO Stage 2 optimization for memory management and performance.

5 Results

Results in the baseline paper [8]	T1: Scene understanding						T2: Object detection				T3: State detection			
	F1-Theme ↑		ROUGE ↑		BLEU ↑		F1-Object ↑		IOU ↑		F1-State ↑		FPR ↓	
PEFT (LoRA)	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/	w/o	w/
mPLUG-OWL2	23.16	29.70$^{\blacktriangle}$	25.23	32.75$^{\blacktriangle}$	3.04	8.46$^{\blacktriangle}$	77.94	93.05$^{\blacktriangle}$	14.23	34.88$^{\blacktriangle}$	36.82	15.00$^{\triangledown}$	63.41	35.00$^{\blacktriangle}$
BLIP2	27.85	32.65$^{\blacktriangle}$	29.55	40.35$^{\blacktriangle}$	6.40	12.50$^{\blacktriangle}$	77.35	84.48$^{\blacktriangle}$	34.25	40.57$^{\blacktriangle}$	24.71	28.50$^{\blacktriangle}$	100.00	94.87$^{\blacktriangle}$
LlaVA	32.10	35.88$^{\blacktriangle}$	13.63	**42.97**$^{\blacktriangle}$	0.99	**17.73**$^{\blacktriangle}$	54.39	72.32$^{\blacktriangle}$	∅	**60.98**$^{-}$	25.04	30.00$^{\blacktriangle}$	99.41	50.00$^{\blacktriangle}$
QWEN-VL	**34.84**	37.00$^{\blacktriangle}$	**29.57**	39.08$^{\blacktriangle}$	5.19	13.13$^{\blacktriangle}$	78.56	89.15$^{\blacktriangle}$	25.60	30.08$^{\blacktriangle}$	39.77	**39.53**$^{\triangledown}$	97.99	96.82$^{\blacktriangle}$
InstructBLIP	32.85	32.85^{-}	29.20	36.87$^{\blacktriangle}$	4.91	11.52$^{\blacktriangle}$	79.76	**98.16**$^{\blacktriangle}$	∅	47.20^{-}	0.00	0.02$^{\blacktriangle}$	**1.22**	**0.00**$^{\blacktriangle}$
MiniGPT-v2	33.06	**37.52**$^{\blacktriangle}$	34.72	32.56$^{\triangledown}$	**9.81**	8.28$^{\triangledown}$	84.95	85.91$^{\blacktriangle}$	**26.98**	25.94$^{\triangledown}$	36.76	38.64$^{\blacktriangle}$	60.42	80.72$^{\triangledown}$
Otter	11.49	–	12.12	–	1.28	–	72.39	–	∅	–	35.33	–	75.88	–
MiniGPT-4	34.11	–	15.05	–	1.88	–	**87.09**	–	30.20	–	37.19	–	67.37	–
GPT-4o	25.81	–	18.67	–	2.00	–	66.67	–	21.68	–	**40.54**	–	43.06	–
Main results in the work														
Qwen2.5VL-7B-Instruct (Base)	31.55	–	27.37	–	4.94	–	59.35	–	**60.82**	–	38.55	–	30.19	–
Qwen2.5VL-7B-Instruct (SFT)	–	**57.16**$^{\blacktriangle}$	–	**52.99**$^{\blacktriangle}$	–	**24.35**$^{\blacktriangle}$	–	81.14$^{\blacktriangle}$	–	14.16$^{\triangledown}$	–	**100.00**$^{\blacktriangle}$	–	**0.00**$^{\blacktriangle}$
Qwen2.5VL-7B-Instruct (GRPO)	–	22.21$^{\triangledown}$	–	18.99$^{\triangledown}$	–	2.72$^{\triangledown}$	–	62.96$^{\blacktriangle}$	–	**65.49**$^{\blacktriangle}$	–	39.66$^{\blacktriangle}$	–	36.18$^{\triangledown}$

Table 1: Benchmarking VLM on dataset, without (/wo) and with (/w) PEFT using LoRA. The bold font indicates the highest score in each column. Symbols ↑ and ↓ denote that higher and lower values are better, respectively. Symbol "–" indicates the model is not applicable for fine-tuning. Symbol "∅" denotes a meaningless zero as the model fails to generate output as instructed. The superscripts "$^{\blacktriangle}$", "$^{\triangledown}$", and "$^{-}$" indicate an increase, decrease, or inapplicability in the evaluation score after fine-tuning, respectively.

5.1 Comparison of GRPO vs SFT Model Performance

The experimental evaluation across three LEGO-ARTA tasks in Table 1 reveals clear performance differences between the GRPO-trained model, SFT-trained model, and baseline Qwen2.5VL-7B-Instruct, as well as an improvement over the results in the baseline paper [12].

Object Detection Performance For the object detection task, the baseline model achieved an F1-Object score of 59.35 and a mean IoU of 60.82. The SFT-trained model showed substantial improvement in label accuracy, reaching an F1-Object score of 81.14, representing a 36.7% increase over the baseline. However, the SFT model's spatial localization performance declined significantly, with the mean IoU dropping to 14.16, a 76.7% decrease from the baseline. The GRPO-trained model demonstrated superior spatial accuracy with a mean IoU of 65.49, achieving a 7.7% improvement over the baseline model's 60.82. This represents the highest spatial localization performance among all three models. However, the GRPO model's object identification capability measured by F1-Object score reached 62.96, which exceeded the baseline by 6.1% but remained substantially lower than the SFT model's 81.14.

Scene Understanding Performance In scene understanding tasks, the baseline model established the initial performance with an F1-Theme score of 31.55, ROUGE score of 27.37, and BLEU score of 4.94. The SFT-trained model achieved marked improvements across all scene understanding metrics, reaching an F1-Theme score of 57.16, ROUGE score of 52.99, and BLEU score of 24.35. These results represent increases of 81.2%, 93.6%, and 393.1% respectively over the baseline performance. In contrast, the GRPO-trained model showed decreased performance in the scene understanding task compared to the baseline. The F1-Theme score dropped to 22.21, the ROUGE score decreased to 18.99, and the BLEU score fell to 2.72. These represent decreases of 29.6%, 30.6%, and 44.9% respectively from the baseline model's performance.

State Detection Performance For state detection tasks, the baseline model achieved an F1-State score of 38.55 and a false positive rate of 30.19. The SFT-trained model demonstrated optimal performance in this domain, achieving a perfect F1-State score of 100.00 and eliminating false positives entirely with an FPR of 0.00. The GRPO-trained model showed modest improvement over the baseline in state detection, with an F1-State score of 39.66 and an FPR of 36.18. While the F1-State score increased by 2.9% compared to the baseline, the false positive rate increased by 19.8%, indicating mixed performance in this task.

5.2 Effect of Reward Design on Training Dynamics

The training results in Figure 1 show a clear distinction between the two reward curves; mainly, the first reward module performs significantly worse. This is likely due to the model being unable to consistently output values in the correct normalized range of [0-100]. Respectively, the IoU reward signal was diminished both by its small magnitude and the penalty factor of 0.5, resulting in a reward gradient too weak for effective learning. In addition, the bonus reward of 0.1 for each value in the range of [0-100] seems to further confuse the model.

In contrast, the second module performs significantly better. After approximately 100 training steps, the IoU reward experiences an upward shift, moving from a range predominantly between 0 and 0.4 to a more elevated band between 0.2 and 0.6. This suggests improved learning dynamics under the second configuration.

Moreover, the reward curve under the second configuration shows higher variance, indicating periods of both exploration and exploitation. The reward further increases over the remaining 400 steps with a slower convergence towards the optimal policy. Observations show a clear capacity to optimize the reward signal, and the statistically significant gap between the two curves reinforces the observation that reward design is crucial in RL tasks.

The results for the state and grounding task signal a stronger problem within RL frameworks aimed at optimizing natural text outputs. While the state reward was expressive enough to steer the policy in the right direction and offer improvement over the baseline, the grounding task rewards simply worsened the model. Reports from other developments confirm that while some tasks are advantaged by the flexibility of a reward system, others are better off with SFT implementations that use causal loss functions.

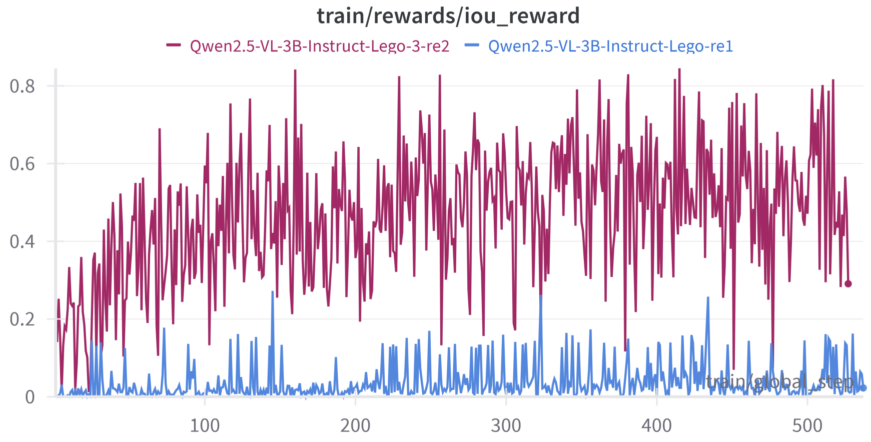

Fig. 1: Comparison of reward signals across 500 GRPO steps for the two reward configurations.

5.3 Tradeoff between GRPO and SFT

The tradeoff between GRPO and SFT displays a limitation in how different loss functions impact task-specific objectives. While SFT shows superior performance in textual accuracy tasks (achieving 81.14% F1-Object score compared to GRPO's 62.96%), it fails catastrophically in spatial localization, thus the mean IoU drops to 14.16% compared to the baseline's 60.82%. This discrepancy appears because of the traditional causal loss functions in SFT, which are optimized for next-token prediction and cannot effectively capture the positional relationships present in spatial tasks like object detection. Furthermore, GRPO's reward-based optimization directly targets spatial accuracy through IoU rewards, achieving 65.49% mean IoU but at the cost of reduced textual precision. The VLAA-Thinking paper confirms this pattern, showing that, when asked to provide a similar spatial reasoning output, SFT induces "pseudo reasoning paths" that hinder genuine spatial understanding [2], while RL incentivizes a more adaptive behavior through its reward signals. This further proves the

VLAA-Thinking paper, which suggests that the optimal approach for multimodal tasks requiring both spatial and textual accuracy may be hybrid reward systems that optimize both IoU for spatial understanding and textual similarity metrics for label accuracy, rather than relying solely on either SFT's token-level supervision or GRPO's single-objective rewards. Additionally, the reward system implemented in this work for the grounding task might be less expressive for complex natural text prediction than SFT, which might require a more semantically rich reward that allows the model to converge to an optimal policy.

5.4 Reproducibility

All resources related to the experiments and evaluation, including source code, are available at https://github.com/Bperju/LEGO-R1.

6 Conclusion

This research set out to determine whether Group Relative Policy Optimization could enhance object detection performance in Vision-Language Models for assembly tasks, specifically within the LEGO-ARTA dataset. The results demonstrate that GRPO-based fine-tuning successfully improved spatial localization performance, achieving a mean IoU of 65.49 compared to the baseline model's 60.82, representing a 7.7% improvement in spatial accuracy.

Regarding whether GRPO can outperform SFT for fine-grained assembly tasks, the results paint a nuanced picture: GRPO demonstrates clear superiority in spatial localization tasks, achieving the highest mean IoU of 65.49 across all evaluated models, while SFT does better in textual accuracy with an F1-Object score of 81.14. This difference in performance highlights the core trade-off between these approaches, where GRPO's reward-based optimization directly targets spatial understanding through IoU rewards, whereas SFT's causal loss functions are better suited for next-token prediction accuracy.

The reward function results confirm this by showcasing a story about the importance of careful reward design. The first reward configuration, which penalized coordinate outputs outside the normalized range and added bonuses for correct formatting, actually hindered learning by creating weak gradient signals. The second configuration, which removed these penalties, showed remarkably better learning dynamics with clear upward trends in IoU rewards after 100 training steps. This demonstrates that even well-intentioned reward models can fail if they dilute the primary learning signal.

However, the results also exposed limitations in applying GRPO to certain task types. While the method showed promise for object detection and modest improvements in state detection (F1-State score of 39.66 versus baseline's 38.55), it demonstrated poor performance in scene understanding tasks. The F1-Theme, ROUGE, and BLEU scores all declined compared to the baseline, which suggests that complex natural language generation tasks may require more semantically rich reward functions in order to outperform SFT compared to the current implementation.

These findings contribute to the larger understanding of reinforcement learning reasoning applications in vision-language models. They highlight the importance of task-specific reward design and the present trade-offs between spatial accuracy and textual precision in multimodal learning frameworks.

7 Future Work

There are several promising directions emerging from this research that could advance the field of reinforcement learning for VLMs.

Firstly, future work should investigate adapting models trained on established object detection datasets before domain-specific fine-tuning. Pre-training on popular datasets such as RefCOCO-g and RefCOCO+ could provide a stronger foundation for spatial understanding before applying GRPO to adapt to downstream task-specific domains like LEGO-ARTA. This approach would enable the exploration of better initial policies for further reinforcement learning optimization.

Secondly, while this research utilized Qwen2.5VL-7B-Instruct as the base model, other vision-language architectures could be investigated for gathering different results. InternVL, a model compatible with the VLM-R1 training framework, should be evaluated to compare performance across tasks. Additionally, other projects that implement RL frameworks for VLMs such as Visual-RFT could be investigated. Similarly, the VLAA-Thinking project directly addresses the tradeoff of GRPO and SFT via a hybrid learning mechanism that should be explored in the context of the LEGO-ARTA dataset.

Lastly, the trade-off encountered between accuracy and textual fidelity identified in this work suggests the need for more sophisticated reward functions that maintain and improve both capabilities. Future work should explore semantic similarity comparison approaches in order to develop a more expressive and robust reward for natural text.

These future directions point toward a better understanding of the capabilities and limitations of reinforcement learning within the context of vision-language models for practical applications, while addressing the challenges identified in the current work.

References

1. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report (2025), https://arxiv.org/abs/2502.13923
2. Chen, H., Tu, H., Wang, F., Liu, H., Tang, X., Du, X., Zhou, Y., Xie, C.: Sft or rl? an early investigation into training r1-like reasoning large vision-language models (2025), https://arxiv.org/abs/2504.11468v1

3. Chen, J., Zhu, D., Shen, X., Li, X., Liu, Z., Zhang, P., Krishnamoorthi, R., Chandra, V., Xiong, Y., Elhoseiny, M.: Minigpt-v2: large language model as a unified interface for vision-language multi-task learning (2023), https://arxiv.org/abs/2310.09478
4. Dao, T.: Flashattention-2: Faster attention with better parallelism and work partitioning (07 2023). https://doi.org/10.48550/arXiv.2307.08691, https://arxiv.org/abs/2307.08691
5. DeepSeek-AI: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning (2025), https://arxiv.org/abs/2501.12948
6. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv:2106.09685 [cs] (10 2021), https://arxiv.org/abs/2106.09685
7. Huang, H., Pei, J., Aliannejadi, M., Sun, X., Ahsan, M., Cesar, P., Yu, C., Ren, Z., Wang, J.: Fine-grained vision-language modeling for multimodal training assistants in augmented reality (2025), https://arxiv.org/abs/2507.05515
8. Huang, Haochen; Pei, J.A.M.S.X.A.M.C.P.Y.C.R.Z.W.J.: Fine-grained vision-language modeling for multimodal training assistants in augmented reality. arXiv preprint arXiv:2507.05515 (2025)
9. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1–2), 83–97 (1955). https://doi.org/10.1002/nav.3800020109
10. Liu, Z., Sun, Z., Zang, Y., Dong, X., Cao, Y., Duan, H., Lin, D., Wang, J.: Visual-rft: Visual reinforcement fine-tuning (2025), https://arxiv.org/abs/2503.01785
11. Loper, E., Bird, S.: Nltk: The natural language toolkit. arXiv (Cornell University) (01 2002). https://doi.org/10.48550/arxiv.cs/0205028
12. Pei, J., Viola, I., Huang, H., Wang, J., Ahsan, M., Ye, F., Yiming, J., Sai, Y., Wang, D., Chen, Z., Ren, P., Cesar, P.: Autonomous workflow for multimodal fine-grained training assistants towards mixed reality (2024), https://www.arxiv.org/abs/2405.13034
13. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., K, L.Y., Wu, Y., Guo, D.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models (2024), https://arxiv.org/abs/2402.03300
14. Shen, H., Liu, P., Li, J., Fang, C., Ma, Y., Liao, J., Shen, Q., Zhang, Z., Zhao, K., Zhang, Q., Xu, R., Zhao, T.: Vlm-r1: A stable and generalizable r1-style large vision-language model (2025), https://arxiv.org/abs/2504.07615
15. Shorten, C., Pierse, C., Smith, T.B., Cardenas, E., Sharma, A., Trengrove, J., Luijt, v.: Structuredrag: Json response formatting with large language models (2024), https://arxiv.org/abs/2408.11061
16. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, second edn. (2018), http://incompleteideas.net/book/the-book-2nd.html
17. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., Zhou, D.: Chain of thought prompting elicits reasoning in large language models. arXiv:2201.11903 [cs] (10 2022), https://arxiv.org/abs/2201.11903
18. Xu, L., Xie, H., Qin, S.Z.J., Tao, X., Wang, F.L.: Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment (12 2023), https://arxiv.org/abs/2312.12148
19. Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., Feng, Z., Ma, Y.: Llamafactory: Unified efficient fine-tuning of 100+ language models. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations). Association for Computational Linguistics, Bangkok, Thailand (2024), http://arxiv.org/abs/2403.13372