ASSEMBLY-R1: 3D ASSEMBLY REASONING VIA RL-BASED VISION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

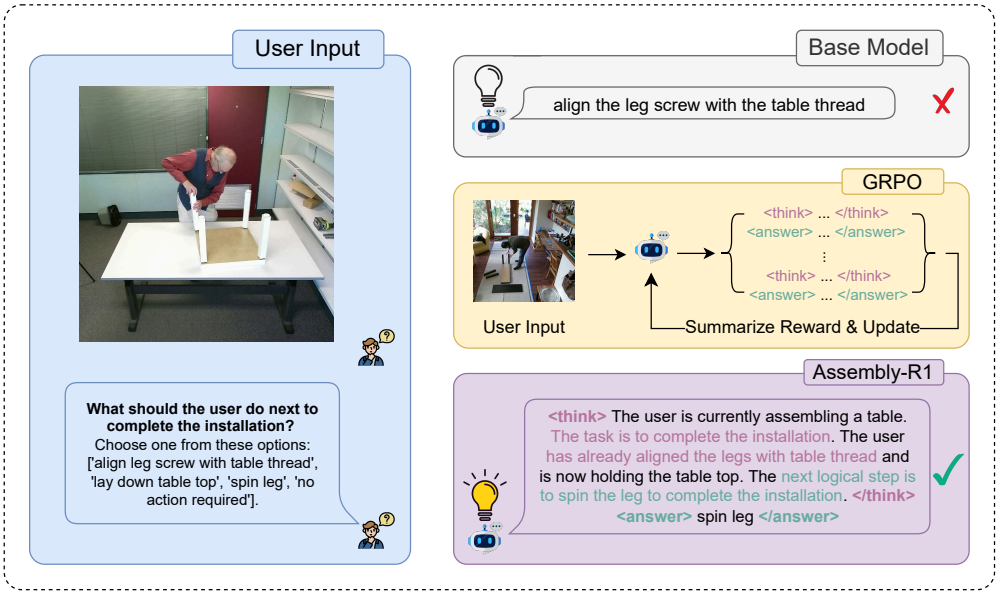

Figure 1: We train a reasoning model, Assembly-R1, which is capable of analyzing structural 3D objects. It can perform detailed 3D object/part recognition, assembly plan generation, assembly status analysis, etc. This model shows great potential for home-use and industrial intelligent robots.

ABSTRACT

3D assembly tasks require automatic agents' precise interpretations of visual scenes and structural reasoning. While large Vision-Language Models (VLMs) have shown promising capabilities in general Visual Question Answering (VQA), existing benchmarks inadequately reflect the complexities inherent in assembly reasoning. In this paper, we introduce FurniBench, an assembly-specific VQA benchmark, together with FurniQA, a large-scale dataset covering tasks such as part recognition, connection reasoning, and step ordering. Using Qwen2-VL-2B-Instruct as a base model (39.1% accuracy on FurniBench), we first establish a supervised fine-tuning (SFT) baseline, which highlights both the benefits and the limitations of SFT in this domain. Building on this, we propose Assembly-R1, trained with Group Relative Policy Optimization (GRPO), which substantially enhances reasoning ability and achieves 71.7% accuracy, outperforming the base model, the SFT baseline, and other open-source and closed-source commercial VLM candidates. Our results demonstrate that reinforcement learning offers a more robust path toward generalizable 3D assembly reasoning. We will release the dataset and code upon acceptance of this paper.

1 INTRODUCTION

Imagine a scenario where a user attempts to assemble a piece of furniture. Following a plain printed installation guide can be confusing, especially when instructions are incomplete or ambiguous. Similarly, in industrial automatic assembly, operators often face the challenge of interpreting complex 3D assembly environments under tight constraints. In both cases, intelligent robotic systems capable of understanding visual scenes and responding to natural language queries in a linguistic manner would significantly assist humans in completing the task. This capability falls within the scope of Visual Question Answering (VQA), a popular research area at the intersection of Computer Vision (CV) and Natural Language Processing (NLP) that was introduced by Agrawal et al. Antol et al. (2015).

Over the past years, VQA has evolved from answering simple, closed-form questions to addressing more complex reasoning and abstract challenges Pandey et al. (2025). Researchers have extended VQA to various application domains, such as medical imaging Bazi et al. (2023); He et al. (2020); Al-Hadhrami et al. (2023), robotics Firoozi et al. (2023); Jiang et al. (2023); Shirai et al. (2024), education and training Huynh et al. (2025); Pandey et al. (2025), etc. The applications of large vision-language models (VLMs) have further enhanced the capabilities of VQA systems by enabling deeper visual-textual alignment and contextual understanding. Brown et al. describe language models as "few-shot learners" Brown et al. (2020), indicating their potential for generalization across diverse tasks with minimal supervision. In this context, VLMs have demonstrated their promising performance across a range of VQA benchmarks Alayrac et al. (2022); Li et al. (2022); Qi et al. (2024); Li et al. (2023).

Despite these advancements, there remains a gap in applying VQA and VLM to the specific domain of 3D part assembly tasks. These tasks are challenging because they involve a mixture of closed vocabulary problems (e.g. part recognition) and open vocabulary questions (e.g. spatial understanding), where answer spaces may vary by context Eichstaedt et al. (2021); Wu et al. (2024); Ko et al. (2023). Keeping the alignment between different modalities is a prerequisite for accomplishing these tasks. In addition, these tasks require the model's deeper understanding of the scenario, such as spatial relationships, reasoning about physical constraints among components and the environment, and interpreting ambiguous human instructions in context-dependent scenarios Yan et al. (2020); Suárez-Ruiz & Pham (2015); Jia et al. (2025); Zhan et al. (2020); Cheng et al. (2023); Zhang et al. (2022). These demands go beyond what general-purpose VLMs are typically designed to handle.

To fill this gap, we introduce FurniBench, a benchmark specifically tailored for part-assembly tasks. It comprises three major categories of assembly-related queries and fifteen subcategories, designed to capture the diverse challenges of visual question answering in this domain. Alongside the benchmark, we present FurniQA, a dataset constructed for FurniBench, containing around 1.6 million high-quality QA pairs derived from the IKEA ASM Dataset Ben-Shabat et al. (2021), providing a new platform for researchers to investigate assembly-related VQAs under real-world scenarios. Given the limitations of existing VLMs in handling such domain-specific tasks, we adopt Qwen2-VL-2B-Instruct as the base model Wang et al. (2024) and first establish a supervised fine-tuning (SFT) baseline using 15k randomly sampled QA pairs from FurniQA. While SFT provides initial performance gains, it also exposes issues such as task-specific overfitting and reduced generalization. To address these challenges, and inspired by the reasoning-enhancement framework of DeepSeek-R1 Guo et al. (2025), we employ Group Relative Policy Optimization (GRPO) Shao et al. (2024) with simple rule-based rewards to foster self-reflective reasoning capabilities. This Reinforcement Learning (RL) approach equips the model with stronger Chain-of-Thought (CoT) reasoning, leading to more accurate and generalizable performance.

Regarding the answer accuracy on FurniBench, our SFT baseline, Assembly-V1, achieves 64.9%, while the reasoning model, Assembly-R1, further improves performance to 71.7%. Both models significantly outperform the base model, Qwen-2-VL-2B-Instruct, which reaches only 39.1%. These results highlight the value of task-specific fine-tuning, while also showing that RL-based optimization can deliver additional gains without requiring extra annotated data or further supervised training. Notably, our findings demonstrate that even with a relatively small-scale dataset of approximately 15k QA pairs, GRPO can significantly boost answering performance by fostering self-reflective reasoning.

Contributions:

- We propose a new benchmark called FurniBench, designed for Visual Question Answering (VQA) in part assembly scenarios, aiming to evaluate models' 3D structural and spatial reasoning abilities.
- We introduce FurniQA, a large-scale dataset comprising 1.6M diverse assembly-related visual QA pairs, spanning 3 major question categories and 15 specific task types. Derived from the IKEA ASM Dataset, FurniQA is tailored for assembly-focused VQA and, with embedded frame IDs, can be readily extended to assembly-related VideoQA tasks.
- We establish a supervised fine-tuning (SFT) baseline, Assembly-V1, based on Qwen2-VL-2B-Instruct, which demonstrates notable improvements over the base model (64.9% vs. 39.1%), while also highlighting the limitations of SFT in robustness and generalization.
- We are the first to apply Group Relative Policy Optimization (GRPO) for reasoning enhancement in VLMs targeting 3D structural understanding. The reasoning model, Assembly-R1, achieves 71.7% accuracy, outperforming both the base model and the SFT baseline, while requiring no additional annotated supervision.

2 Related Works

2.1 Vision Language Model and Visual Question Answering

Recent advancements in Vision-Language Models (VLMs) have significantly improved multimodal understanding. Models such as Flamingo, BLIP, and BLIP-2 Alayrac et al. (2022); Li et al. (2022; 2023) have demonstrated impressive performance by effectively aligning visual and textual modality. OpenAI GPT-4o OpenAI (2024a) marks a major milestone in multimodal integration, achieving state-of-the-art in various benchmarks. Meanwhile, the emergence of open-source VLMs, like QwenVL, InternVL, LLaVA, etc. Bai et al. (2023); Wang et al. (2024); Bai et al. (2025); Chen et al. (2024b); Zhu et al. (2025); Liu et al. (2023) has largely boosted the research in Visual Question Answering (VQA). At the same time, researchers have developed a variety of benchmarks to evaluate models and explore their full potential in multiple aspects Singh et al. (2019); Schwenk et al. (2022); Tong et al. (2024); Ma et al. (2023). The co-evolution of VQA benchmarks and VLMs continuously pushes forward the development of more robust and capable models.

2.2 Model reasoning with Reinforcement Learning

Following the success of large language models (LLMs) in general knowledge tasks Touvron et al. (2023); Radford et al. (2018); Brown et al. (2020), researchers have increasingly turned their attention to enhancing models' reasoning abilities, particularly for more complex domains such as science, mathematics, and logic OpenAI (2024b); Guo et al. (2025). OpenAI o1 model demonstrates that incorporating Reinforcement Learning (RL) allows models to learn from feedback on their generated responses, leading to Chain-of-Thought (CoT) reasoning patterns and more accurate answers. DeepSeek introduces R1-Zero Guo et al. (2025), a GRPO model Shao et al. (2024) to improve reasoning ability without relying on additional supervised data. With a simple rule-based reward design, it achieves competitive performance on reasoning benchmarks at only a fraction of the training cost compared to its counterparts, largely reducing the training requirement for hardware.

In the vision-language domain, SpatialVLM addresses the limitations of existing vision-language models in spatial reasoning by training on an Internet-scale multimodal dataset rich in spatial relationships Chen et al. (2024a). Inspired by DeepSeek-R1, VLM-R1 and VisualThinker-R1-Zero reproduce the 'aha' moment with non-SFT GRPO method on various VQA benchmarks Shen et al. (2025); Zhou et al. (2025).

Overall, these works demonstrate the growing impact and potential of using reinforcement learning-based methods to enhance the existing base model's reasoning capabilities with reduced reliance on annotated data and training resources.

2.3 IKEA ASM Dataset

The IKEA ASM dataset is a richly annotated, multimodal, and multiview video dataset of furniture assembly tasks Ben-Shabat et al. (2021). Originally designed for benchmarking tasks such as video

action recognition, object segmentation, part tracking, and human pose estimation, it comprises 371 video samples, including 48 unique assemblers constructing four different types of IKEA furniture in five distinct environments. Every video includes recordings from three camera views, and the primary view (denoted as 'top') contains an RGB-D stream, atomic action labels, human pose estimation, object and part tracking, etc.

3 METHOD

3.1 PROBLEM FORMULATION - FURNIBENCH

FurniBench is a VQA benchmark designed for assessing models' performance on assembly-related tasks. Given a visual input v and a textual question q, the task is to predict an answer o that matches the reference answer o_{ref}. Formally, the model learns a function: $f_\theta : (v, q) \rightarrow o$, where θ denotes the trainable parameters, optimized to minimize the discrepancy between o and o_{ref}.

3.2 DATASETS - FURNIQA

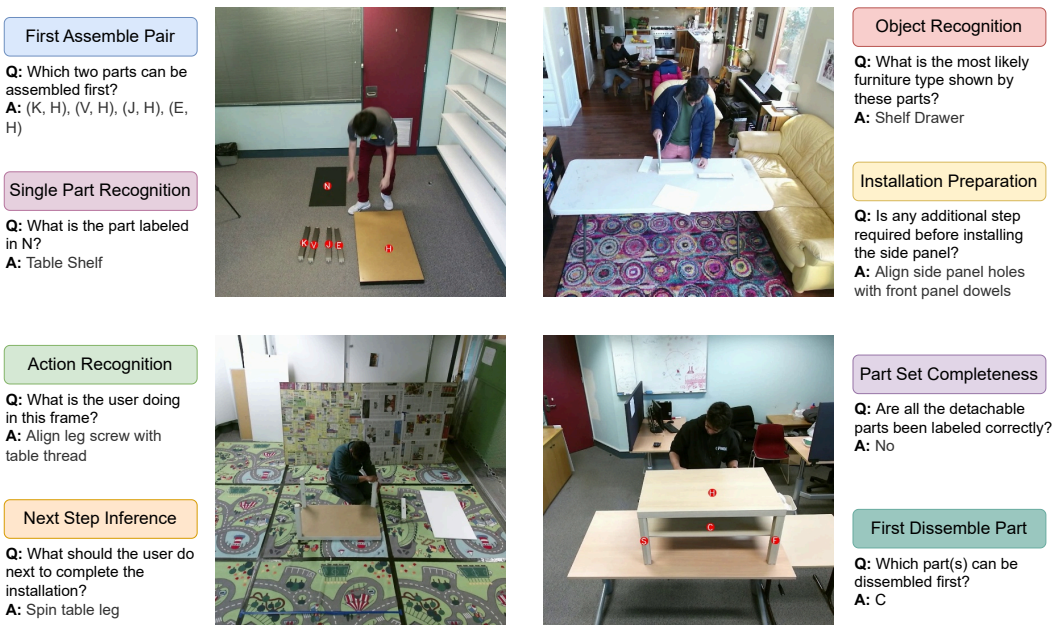

Figure 2: A demonstration of example QA pairs from FurniQA. Visual inputs are shown at the center, surrounded by corresponding textual questions and reference answers. Different QA task categories are highlighted in distinct colors, reflecting the diversity of research challenges covered in FurniBench.

To build our dataset, FurniQA, in the context of 3D assembly understanding, we utilize the RGB stream from the main camera view of the IKEA ASM video streams and combine each visual frame with its corresponding annotations. All QA pairs in FurniQA are programmatically generated using predefined rules grounded in the dataset's annotations. Importantly, the questions are manually calibrated by humans to ensure they are reasonable and aligned with real-world assembly scenarios. No generative models were used in answer generation, ensuring the validity and reliability of each QA pair.

Based on the stage of the assembly process, each scene is categorized into one of three phases: Beginning, In Progress, or Completed. QA pairs are tailored according to these phases to ensure that the questions are contextually relevant and reflective of real-world scenarios. FurniQA comprises approximately 1.6 million QA pairs and is organized into three main task categories: Part Recognition,

Part Connectivity, and General Assembly Understanding. The specific task types and corresponding quantities are detailed in the Appendix. The objectives of each main category are as follows:

Part Recognition challenges the model's capability in identifying individual furniture parts, like drawer side panels or the table shelf, understanding the part set completeness by assessing whether all required parts are present in the scene, and inferring the identity of the final object (e.g. a table or a bench) based on dispersed parts.

Part Connectivity requires the model to understand the topological and physical relationships among parts. For example, it should determine which parts can be assembled, and in what sequence. Some tasks even include reverse reasoning, such as identifying which part can be disassembled first, pushing the model to demonstrate a deeper understanding of the structural dependencies and assembly logic.

General Assembly Understanding is designed around the atomic actions of the assembly process. The model is expected to recognize the current or infer the next assembly steps, e.g., picking up a specific part or aligning two components. These questions are specifically challenging as they require the model to: (1) comprehend the current state of the scene, including parts already assembled and those remaining; (2) reason about correct assembly sequence, like which steps should be performed first ahead of a specific step; (3) differentiate between preparatory (e.g. pick up or align parts) and active assembly actions (e.g. insert or attach parts).

3.2.1 Reducing Bias and Subjectivity

- **Increasing Diversity** Questions are rephrased with GPT-4o to increase the diversity of expressions. The question expression variations are listed in the Appendix.
- **Avoiding Enforced Single Answer** IKEA-ASM includes multiple assembly demonstrations per item by different users, naturally capturing diverse valid assembly orders. We carefully consider all potential assembly steps by manually inspecting the installation videos. In preparation, we labeled the sets of all possible correction options as answers. In the training stage, models are encouraged to generate all potential options, and, during evaluation, a prediction is marked as correct if it is a subset of the ground truth answer set.
- **Dynamic Part Tagging** Letter part tags ['A'-'Z'] are randomly assigned for the parts in each frame, i.e. the same part will have different letter label in different questions, preventing the model from memorizing static associations between tags and parts, forcing it to focus on 3D structural features in the assembly context.
- **Shuffled Options** Option order is randomized to ensure the model relies on reasoning rather than positional bias

3.3 Assembly-V1: A Baseline Model Training with Supervised Fine-Tuning

Treating Assembly-V1 as a baseline, we fine-tuned the Qwen2-VL-2B-Instruct model using Supervised Fine-Tuning (SFT). The fine-tuning was performed with the help of LlamaFactory Zheng et al. (2024). In the training procedure, we use the collected question/vision-answer pairs to form the chat template, and we apply the SFT function provided by the LlamaFactory to finish the training. More training parameters and details are discussed in the Appendix

3.4 Assembly-R1: Visual Reasoning Using Reinforcement Learning

As stated before, our proposed FurniBench understanding task is challenging. This task requires the model not only to recognize object categories, but also to understand deeper information, such as geometric structures and the 3D relationships among objects in the image.

To achieve this goal, we apply the powerful RL tool, Group Relative Policy Optimization (GRPO) Guo et al. (2025); Shao et al. (2024), to train our model. The objective function of GRPO can be described as follows:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q_v \sim P(Q_V), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q_v)]$$
$$\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q_v)}{\pi_{\theta_{old}}(o_i|q_v)}A_i, \text{clip}\left(\frac{\pi_\theta(o_i|q_v)}{\pi_{\theta_{old}}(o_i|q_v)}, 1-\varepsilon, 1+\varepsilon\right)A_i\right) - \beta\mathbb{D}_{KL}(\pi_\theta||\pi_{ref})\right), \quad (1)$$

$$\mathbb{D}_{KL}(\pi_\theta||\pi_{ref}) = \frac{\pi_{ref}(o_i|q_v)}{\pi_\theta(o_i|q_v)} - \log\frac{\pi_{ref}(o_i|q_v)}{\pi_\theta(o_i|q_v)} - 1, \quad (2)$$

where q_v represents the sampled question and image set; $\{o_1, o_2, \cdots, o_i\}$ are the outputs sampled from the policy model π_θ or the old policy model π_{old}; ε and β are hyper-parameters; A_i calculated from the rewards $\{r_1, r_2, \cdots, r_G\}$ through the following formula:

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \cdots, r_G\})}{\text{std}(\{r_1, r_2, \cdots, r_G\})}. \quad (3)$$

Reward Design The reward design is key to the success of GRPO training. We basically use the following two reward functions to guide the optimization of our algorithm:

- **Format Reward:** If the model can correctly generate response with **<think>*</think> <answer>*</answer>** format, it receives a +1 reward.
- **Accuracy Reward:** If the answer in the generated response is correct, we give the model a reward of +1.

With the objectives and rewards discussed above, we can train a reasoning model that can provide both a reasoning procedure and a correct answer. We discuss the training parameters in our Appendix.

4 Experiments

4.1 Dataset and Evaluation

We randomly select 15,000 QA pairs from the FurniQA training set to fine-tune both Assembly-V1 and Assembly-R1. For evaluation, we use 1,500 QA pairs from FurniQA testing branch. In addition, to assess the generalizability of the fine-tuned model, we further evaluate both models on CVBench Tong et al. (2024). The performance of the candidate models is measured based on the accuracy of the answer responses.

4.2 Hardware and Implementation Details

In our experiments, we use 2 $\times$ NVIDIA A100 80GB GPUs to train the models, including Assembly-V1 (SFT) and Assembly-R1 (GRPO). For both training procedures, we set the per-device batch size to 1 and the gradient accumulation steps to 4. The training step is set to 1800. We tune all the parameters of the models at both the SFT and GRPO stages. Due to the page limit, we demonstrate more training details in our Appendix.

4.3 Quantitative Results

We demonstrate the quantitative comparisons in Fig. 3 and Table 1. Table 1 demonstrates that both Assembly-V1 and Assembly-R1 receive a significant improvement after the training procedure. Overall, Assembly-R1 (GRPO) outperforms the SFT model Assembly-V1, which indicates that reinforcement learning leads to better performance in in-domain testing.

Benchmarking for VLM baselines Our dataset FurniQA also provides a comprehensive evaluation of other VLMs in the 3D assembly understanding context. We show the testing results for 7 different VLMs in Fig. 3, including open-source models with similar or larger scales and SOTA closed-source commercial VLMS. In the 2B/3B level VLMs benchmarking, Qwen2-VL-2B-Instruct achieves

the best performance; while in the 7B level VLMs, Qwen2-VL-7B-Instruct performs better than LLaVA-1.5-7B-HF. Gemini-2.5-Pro outperforms other models, including GPT-4o, by a large margin.

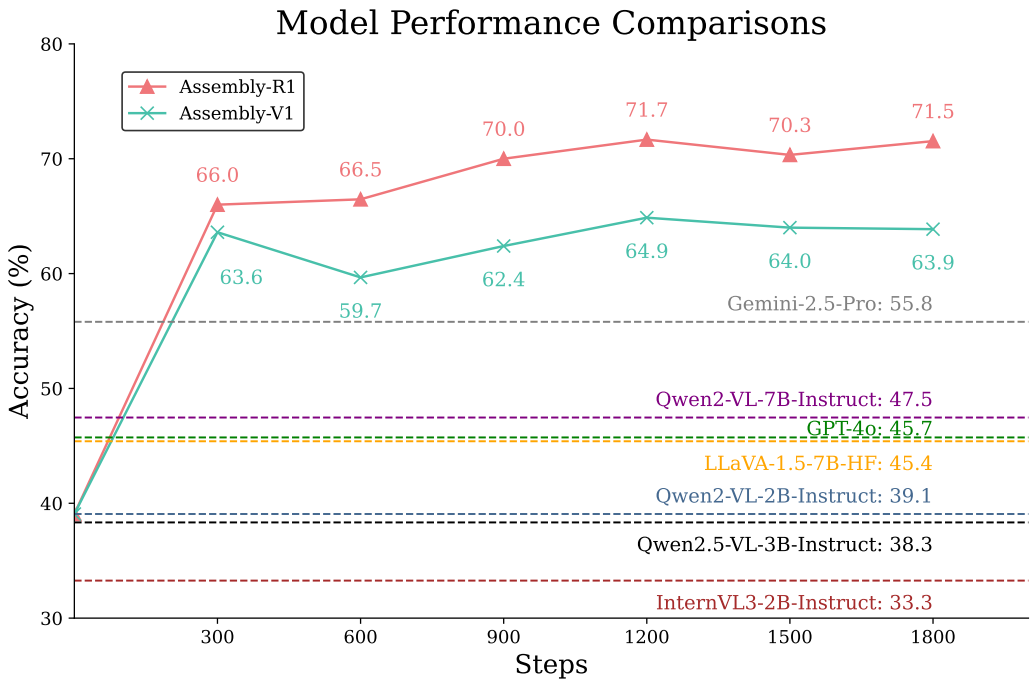

Figure 3: Performance comparison of various models on FurniBench. The green and red lines depict the progression of Assembly-V1 and Assembly-R1 performance throughout the training steps. Horizontal dashed lines indicate the benchmark performance of popular open-source vision-language models (VLMs).

Table 1: Accuracy comparison of Base Model (Qwen2-VL-2B-Instruct), SFT Baseline (Assembly-V1), and GRPO (Assembly-R1) across different task categories in FurniBench.
† PR: Part Recognition. ‡ PC: Part Connectivity. * GAU: General Assembly Understanding

Main Category	Base Model	Assembly-V1 @1800 steps	Assembly-R1 @1800 steps
PR†	37.8%	63.9%	73.4% (+35.6%, +9.5%)
PC‡	28.0%	28.0%	32.0% (+4.0%, +4.0%)
GAU*	41.0%	64.3%	73.8% (+32.8%, +9.5%)
Overall Accuracy	39.1%	63.9%	71.5% (+32.4%, +7.6%)

Detailed Evaluation Against Problem Categories We also demonstrate a detailed performance against question types in Table 1. The results in the table show that the RL model gains a significant performance boost on Part Recognition and General Assembly Understanding problems compared to the base and SFT models. The Assembly-R1 also shows a slight improvement in Part Connectivity. These results further show the superiority of the GRPO reinforcement learning algorithm.

4.4 Qualitative Results

We also present some qualitative results in Fig. 4 to showcase our trained models. Overall, the figure shows that Assembly-R1 achieves the best performance, while both the base model and Assembly-V1 struggle to provide the correct answers. Assembly-R1 can not only give the correct answers, but

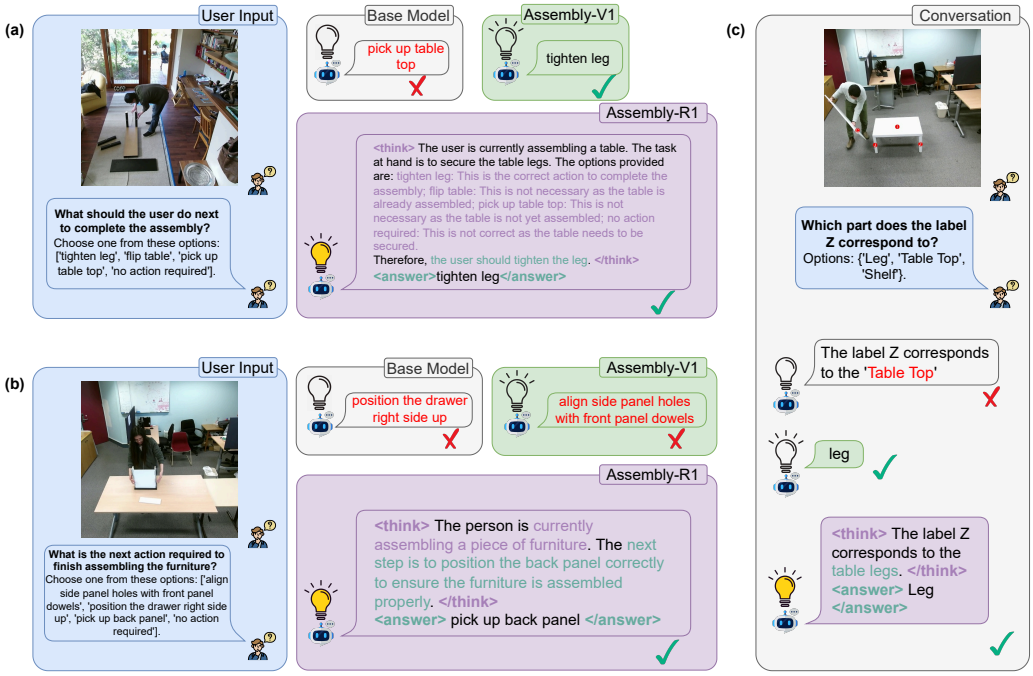

Figure 4: Qualitative results: Three example cases illustrated in (a), (b), and (c). Each example includes User Input (image + text, highlighted in blue), Base Model response (light gray), Assembly-V1 response (green), and Assembly-R1 response (purple). A tick or cross next to each model's response indicates correctness.

also outputs the detailed reasoning processes. Taking the Fig. 4 (a) as an example, Assembly-R1 can output both the reasons for selecting the correct answer and analysis for the incorrect answer. This indicates the deep analysis ability of the Assembly-R1. The SFT model Assembly-V1 cannot provide the analysis as the SFT tends to "remember" the correct answers. This phenomenon also demonstrates the superiority of the RL-based model. We can summarize that the RL training algorithm GRPO indeed can improve both the reasoning ability and the accuracy of the answer.

4.5 Out-of-Domain Results — CVBench

Table 2: Accuracy comparison of Base Model (Qwen2-VL-2B-Instruct), SFT Baseline (Assembly-V1), and GRPO (Assembly-R1) on CVBench.

Model Performance	Base Model	Assembly-V1	Assembly-R1
Overall Accuracy	62.4%	28.1%	63.5% (+1.1%, +35.4%)

We conduct Out-Of-Domain (OOD) testing on the dataset CVBench to compare the generalizability of the models. The results in Table 2 show that our RL model Assembly-R1 has a better performance compared to the base model, even on out-of-domain data, while the SFT model Assembly-V1 does not. Given the fact that the context of CVBench is quite different from FurniQA, Assembly-V1's performance drop in OOD data indicates the limitations of pure SFT. Our test results in Table 1 and 2 support the "SFT Memorizes, RL Generalizes" theory by Chu et al. (2025).

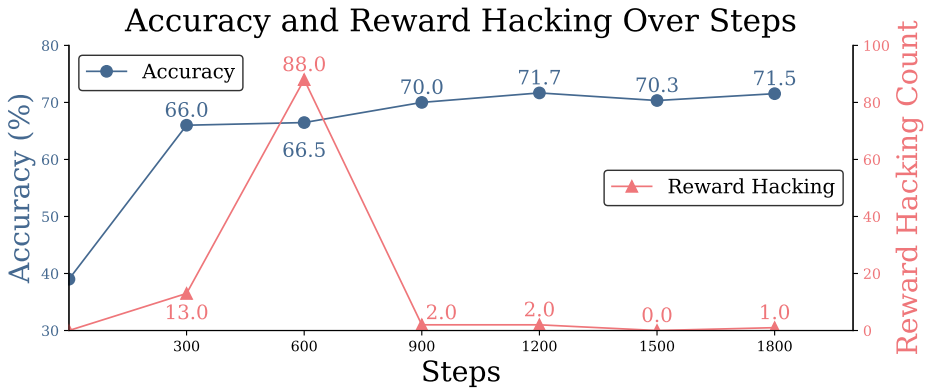

Figure 5: Visualization of model performance (blue line) and the number of reward hacking instances (red line) across training steps.

4.6 Reward Hacking

Reward hacking occurs when an agent exploits flaws in the reward design to gain rewards through unintended behaviors Shen et al. (2025). Zhou et al. (2025) show that rewarding reasoning length can lead models to generate longer outputs without improving reasoning quality.

Although we don't explicitly reward reasoning length, we still observe signs of reward hacking during training. We define the length reward hacking as a response that repeats with meaningless reasoning until reaching the output limit (1024) without a closing **</think>** tag. These incomplete responses suggest the model tries to exploit perceived reward signals without understanding the task. More analysis on the reasoning length and model performance are delivered in the Appendix.

As shown in Fig. 5, accuracy initially rises from 39.1% to 66.0% at 300 steps, with 13 reward hacking instances out of 1500 testing samples. This initial improvement in accuracy likely results from introducing the reasoning pattern, which the base model lacks. However, from 300 to 600 steps, hacking behavior increases while accuracy stagnates. In other words, the agent is optimizing for quantity over quality, generating longer but ineffective reasoning sequences. After 600 steps, rewards hacking diminishes and accuracy improves, reaching 71.7%. This is expected since our reward design does not explicitly favor long reasoning but rather meaningful thinking and accuracy. Eventually, the model shifts towards generating useful reasoning to gain more **Accuracy Reward**.

This observation highlights the importance of careful reward design in the RL-based fine-tuning framework for enhancing LLM/VLM reasoning capability.

4.7 Limitations

Despite the richness and large scale of our FurniQA, there is still room for improvement in terms of diversity. Specifically, the dataset could benefit from incorporating a broader range of QA task types and assembly objects, more diverse camera shooting angles, input modalities, like depth information, and indoor/outdoor assembly scenes. Enhancing the dataset in these aspects could assist the model to generalize better to real-world applications and unseen configurations.

5 Conclusion

In conclusion, we propose a new benchmark, FurniBench, along with a new dataset, FurniQA, to assess the 3D structural and spatial understanding of large models. We also trained new large models, Assembly-V1 and Assembly-R1, based on our dataset. We successfully established our new benchmark by testing our trained models and other open-source VLMs. In addition, we use out-of-domain experiments to demonstrate the phenomenon of "SFT Memorizes, RL Generalizes." In the future, we plan to test our models in real industrial environments, such as industrial robotic assembly scenarios.

REFERENCES

Suheer Al-Hadhrami, Mohamed El Bachir Menai, Saad Al-Ahmadi, and Ahmad Alnafessah. An effective med-vqa method using a transformer with weights fusion of multiple fine-tuned models. *Applied Sciences*, 13(17), 2023. ISSN 2076-3417. doi: 10.3390/app13179735. URL https://www.mdpi.com/2076-3417/13/17/9735.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. URL https://arxiv.org/abs/2204.14198.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: visual question answering. *CoRR*, abs/1505.00468, 2015. URL http://arxiv.org/abs/1505.00468.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. URL https://arxiv.org/abs/2308.12966.

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

Yakoub Bazi, Mohamad Mahmoud Al Rahhal, Laila Bashmal, and Mansour Zuair. Vision–language model for visual question answering in medical imagery. *Bioengineering*, 10(3), 2023. ISSN 2306-5354. doi: 10.3390/bioengineering10030380. URL https://www.mdpi.com/2306-5354/10/3/380.

Yizhak Ben-Shabat, Xin Yu, Fatemeh Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 847–859, 2021.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. URL https://arxiv.org/abs/2005.14165.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities, 2024a. URL https://arxiv.org/abs/2401.12168.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks, 2024b. URL https://arxiv.org/abs/2312.14238.

Junfeng Cheng, Mingdong Wu, Ruiyuan Zhang, Guanqi Zhan, Chao Wu, and Hao Dong. Score-pa: Score-based 3d part assembly. *British Machine Vision Conference (BMVC)*, 2023.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

Johannes C. Eichstaedt, Margaret L. Kern, David B. Yaden, and et al. Closed- and open-vocabulary approaches to text analysis: A review, quantitative comparison, and recommendations. *Psychological Methods*, 26(4):398–427, 2021. doi: 10.1037/met0000349.

Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, Brian Ichter, Danny Driess, Jiajun Wu, Cewu Lu, and Mac Schwager. Foundation models in robotics: Applications, challenges, and the future, 2023. URL https://arxiv.org/abs/2312.07843.

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.

Xuehai He, Yichen Zhang, Luntian Mou, Eric Xing, and Pengtao Xie. Pathvqa: 30000+ questions for medical visual question answering, 2020. URL https://arxiv.org/abs/2003.10286.

Ngoc Dung Huynh, Mohamed Reda Bouadjenek, Sunil Aryal, Imran Razzak, and Hakim Hacid. Visual question answering: from early developments to recent advances – a survey, 2025. URL https://arxiv.org/abs/2501.03939.

Qingxuan Jia, Guoqin Tang, Zeyuan Huang, Zixuan Hao, Ning Ji, Shihang, Yin, and Gang Chen. Perceiving, reasoning, adapting: A dual-layer framework for vlm-guided precision robotic manipulation, 2025. URL https://arxiv.org/abs/2503.05064.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts, 2023. URL https://arxiv.org/abs/2210.03094.

Dohwan Ko, Ji Soo Lee, Miso Choi, Jaewon Chu, Jihwan Park, and Hyunwoo J. Kim. Open-vocabulary video question answering: A new benchmark for evaluating the generalizability of video question answering models, 2023. URL https://arxiv.org/abs/2308.09363.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. URL https://arxiv.org/abs/2201.12086.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. URL https://arxiv.org/abs/2301.12597.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. URL https://arxiv.org/abs/2304.08485.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes, 2023. URL https://arxiv.org/abs/2210.07474.

OpenAI. Gpt-4o system card, 2024a. URL https://arxiv.org/abs/2410.21276.

OpenAI. Openai o1 system card, 2024b. URL https://arxiv.org/abs/2412.16720.

Anupam Pandey, Deepjyoti Bodo, Arpan Phukan, and Asif Ekbal. The quest for visual understanding: A journey through the evolution of visual question answering, 2025. URL https://arxiv.org/abs/2501.07109.

Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. Shapellm: Universal 3d object understanding for embodied interaction, 2024. URL https://arxiv.org/abs/2402.17766.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 3505–3506, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3406703. URL https://doi.org/10.1145/3394486.3406703.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge, 2022. URL https://arxiv.org/abs/2206.01718.

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.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. Vlm-r1: A stable and generalizable r1-style large vision-language model, 2025. URL https://arxiv.org/abs/2504.07615.

Keisuke Shirai, Cristian C. Beltran-Hernandez, Masashi Hamaya, Atsushi Hashimoto, Shohei Tanaka, Kento Kawaharazuka, Kazutoshi Tanaka, Yoshitaka Ushiku, and Shinsuke Mori. Vision-language interpreter for robot task planning, 2024. URL https://arxiv.org/abs/2311.00967.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read, 2019. URL https://arxiv.org/abs/1904.08920.

Francisco Suárez-Ruiz and Quang-Cuong Pham. A framework for fine robotic assembly, 2015. URL https://arxiv.org/abs/1509.04806.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024. URL https://arxiv.org/abs/2409.12191.

Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, Bernard Ghanem, and Dacheng Tao. Towards open vocabulary learning: A survey, 2024. URL https://arxiv.org/abs/2306.15880.

Fujian Yan, Dali Wang, and Hongsheng He. Robotic understanding of spatial relationships using neural-logic learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8358–8365, 2020. doi: 10.1109/IROS45743.2020.9340917.

Guanqi Zhan, Qingnan Fan, Kaichun Mo, Lin Shao, Baoquan Chen, Leonidas J Guibas, Hao Dong, et al. Generative 3d part assembly via dynamic graph learning. *Advances in Neural Information Processing Systems*, 33:6315–6326, 2020.

Rufeng Zhang, Tao Kong, Weihao Wang, Xuan Han, and Mingyu You. 3d part assembly generation with instance encoded transformer. *IEEE Robotics and Automation Letters*, 7(4):9051–9058, 2022.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*, 2024.

Hengguang Zhou, Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. R1-zero's "aha moment" in visual reasoning on a 2b non-sft model, 2025. URL https://arxiv.org/abs/2503.05132.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025. URL https://arxiv.org/abs/2504.10479.

A Appendix

A.1 Supervised Fine-Tuning (SFT) Configurations

The SFT training configurations are listed in Table 3. The fine-tuning was performed with the help of LlamaFactory Zheng et al. (2024). The dataset is structured in Alpaca format Taori et al. (2023) for training the model.

Parameter	Value
model_name_or_path	`Qwen/Qwen2-VL-2B-Instruct`
trust_remote_code	`true`
stage	`sft`
do_train	`true`
finetuning_type	`full`
freeze_vision_tower	`false`
freeze_multi_modal_projector	`false`
freeze_language_model	`false`
deepspeed	`LLaMA-Factory/examples/deepspeed/ds_z3_config.json`
dataset	`FurniBench_train_shuffled_selected_15000`
template	`qwen2_vl`
cutoff_len	`20480`
preprocessing_num_workers	`16`
dataloader_num_workers	`4`
output_dir	`outputs/qwen2_vl-2b_512_15000/sft`
logging_steps	`25`
save_steps	`300`
report_to	`wandb`
batch_size	`8`
learning_rate	`1.0e-4`
num_train_epochs	`3`
lr_scheduler_type	`cosine`
warmup_ratio	`0.1`
bf16	`true`
ddp_timeout	`180000000`
resume_from_checkpoint	`null`

Table 3: Supervised Fine-Tuning (SFT) & DeepSpeed training configurations.

A.2 GROUP RELATIVE POLICY OPTIMIZATION (GRPO) CONFIGURATIONS

The GRPO model fine-tuning configurations are listed in Table 4. The multi-GPU training benefits from DeepSpeed Rasley et al. (2020).

Parameter	Value
config_file	configs/zero2.yaml
model_name_or_path	Qwen/Qwen2-VL-2B-Instruct
dataset_name	FurniBench_train_shuffled_selected_15000
max_prompt_length	1024
max_completion_length	700
learning_rate	1.0e-6
batch_size	8
logging_steps	1
bf16	true
gradient_checkpointing	true
num_train_epochs	3
save_steps	300
save_only_model	true
report_to	wandb
compute_environment	LOCAL_MACHINE
distributed_type	DEEPSPEED
deepspeed_multinode_launcher	standard
zero_stage	2
zero3_init_flag	false
offload_optimizer_device	none
offload_param_device	none
mixed_precision	bf16
downcast_bf16	no
num_processes	8
num_machines	1
machine_rank	0
main_training_function	main
main_process_port	44326
rdzv_backend	static
same_network	true
use_cpu	false
tpu_use_cluster	false
tpu_use_sudo	false
tpu_env	[ ]

Table 4: GRPO & DeepSpeed training configuration.

A.3 DATASET - QUESTION REPRESENTATIVE EXPRESSIONS

FurniQA includes 15 distinct QA task types. To make the dataset more diverse, each task is associated with three representative question expressions, as illustrated in Table 5. When generating QA pairs for each assembly video frame, one of the three expressions for the corresponding question type is randomly selected.

Question Type	Representative Expressions
Single Part Recognition (MCQ)	What is the part labeled in `{id}`? Please identify the part labeled as `{id}`. Which part does the label `{id}` correspond to?
Part Set Completeness (YN)	Are the currently labeled parts sufficient to complete the assembly? Do the labeled parts cover everything needed for assembly? Are all necessary parts labeled for assembly?
Missing Part Recognition (MCQ)	What other parts are required to complete the assembly? Are there any parts not labeled that are needed? Which parts are still required to finish the assembly?
First Assemble Pair (MCQ)	Which two parts can be assembled first? Out of the listed pairs, which can be assembled at the beginning? Select the pair of parts that should be assembled first.
First Assemble Pair (YN)	Can I directly attach Part A to Part B? Are Part A and Part B ready to be connected now? Is it possible to assemble them together now?
Connection After Installation (MCQ)	What parts does Part A connect to after installation? After assembly, which parts will be connected to Part A? Select the parts that will be attached to Part A.
Disassemble First (MCQ)	Which parts can be disassembled first? Out of the listed parts, which can be removed first? Select the part(s) that should be taken apart first.
Object Recognition (MCQ)	What could be the type of furniture? What is the most likely furniture type? Which furniture category do these parts belong to?
Installation Completed (YN)	Is the installation completed? Has the assembly process finished? Are all parts fully assembled now?
Action Recognition (MCQ)	What is the user doing in this frame? Describe the action performed by the user. Which activity is the user engaged in now?
Action Recognition (YN)	Is the user manipulating a `{part}`? Is the user interacting with a `{part}`? Do you see the user handling a `{part}`?
Next Step Inference (MCQ)	What should the user do next to complete the installation? What is the next action required? Which step should be performed next?
Installation Preparation (MCQ)	What should the user do next to prepare? Which preparation is needed before continuing? What action should be taken before the next step?
Installation Assembly (MCQ)	What should the user do next to complete the assembly? Which assembly action comes next? What is the next step in the assembly process?
Ready for Installation (YN)	Are the `{part}` ready to be installed? Can the `{part}` be installed now? Is any step required before installing the `{part}`?

Table 5: Overview of all 15 question types in FurniQA with representative expressions. Each type has 3 variations to encourage language diversity. For easier evaluation, each question in the dataset comes with a list of options, either a list of different choices or Yes/No. To maintain clarity, the answer options are not shown in this table.

A.4 DATASET - STATISTICS OF FURNIQA

Table 6: Statistics of FurniQA, including the main category, sub-category, task type, and quantities of corresponding QA pairs.

Main Category	Sub Category	Type	Quantity
Part Recognition	Single Part Recognition	MCQ	176,903
	Part Set Completeness	YN	176,903
	Missing Part Recognition	MCQ	154,105
	Object Recognition	MCQ	154,105
Part Connectivity	First Assemble Pair	MCQ	3,050
	First Assemble Pair	YN	10,654
	Connection After Installation	MCQ	45,786
	First Dissemble Part	MCQ	22,798
General Assembly Understanding	Installation Completion	YN	176,903
	Action Recognition	MCQ	150,286
	Action Recognition	YN	176,903
	Next Step Inference	MCQ	176,903
	Installation Preparation	MCQ	107,972
	Installation Assembly	MCQ	45,655
	Ready For Installation	YN	35,969

A.5 Additional Results - Average Response Length

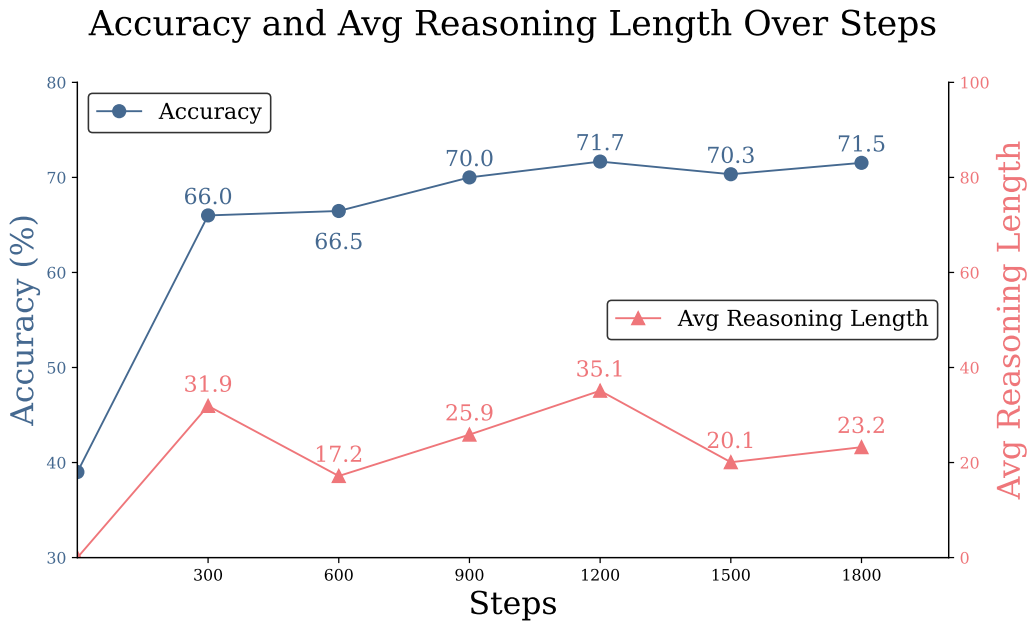

Figure 6: Visualization of model performance (blue line) and the average length of reasoning (red line) across training steps

Fig. 6 shows the relationship between the model's performance and its average reasoning length over training steps. Importantly, we exclude samples flagged as reward hacking behavior when calculating the average reasoning length per response, so the statistics reflect only valid reasoning sequences.

Since our reward function does not explicitly encourage longer reasoning, the average length does not increase monotonically during training. Instead, it fluctuates between 17 and 35 words from step 300 onward. Notably, we can observe that the improvement in accuracy is usually accompanied by longer reasoning, while the periods of stable accuracy often show a decrease in reasoning response length.

In the early training phase, from the start to step 300, accuracy improves from 39.1% to 66.0%, with the average reasoning length reaching 39.1 tokens. Between steps 300 and 600, accuracy remains steady while the average reasoning length drops to 17.2 tokens. Then, from step 600 to step 1200, the accuracy climbs further to 71.1%, accompanied by an increment in average reasoning length to 35.1 tokens. Afterward, while the model keeps the accuracy around 71%, the average length decreases by over 10 tokens per response.

In summary, while the model is not directly rewarded for longer reasoning, it learns to use a more elaborate self-reflective reasoning chain to gain reward by improving answer accuracy. At the same time, it continues to refine its reasoning pattern to avoid unnecessary verbosity.