

# Embodied-R1: Reinforced Embodied Reasoning for General Robotic Manipulation

Yifu Yuan<sup>♡</sup> Haiqin Cui<sup>♡</sup> Yaoting Huang<sup>♡</sup> Yibin Chen<sup>♡</sup> Fei Ni<sup>♡</sup> Zibin Dong<sup>♡</sup>

Pengyi Li<sup>♡</sup> Yan Zheng<sup>♡</sup> Jianye Hao<sup>♡</sup>

<sup>♡</sup>Tianjin University

Generalization in embodied AI is hindered by the “seeing-to-doing gap”, stemming from data scarcity and embodiment heterogeneity. To address this, we pioneer “pointing” as a unified, embodiment-agnostic intermediate representation, defining four core embodied pointing abilities that bridge high-level vision-language comprehension with low-level action primitives. We introduce **Embodied-R1**, a 3B Vision-Language Model (VLM) specifically designed for embodied reasoning and pointing. We use a wide range of embodied and general visual reasoning datasets as sources to construct a large-scale dataset, *Embodied-Points-200K*, which supports key embodied pointing capabilities. Then we train Embodied-R1 using a two-stage Reinforced Fine-tuning (RFT) curriculum with specialized multi-task reward design. Embodied-R1 achieves state-of-the-art performance on 11 embodied spatial and pointing benchmarks. Critically, it demonstrates robust zero-shot generalization by achieving a 56.2% success rate in the SIMPLEREnv and 87.5% across 8 real-world XArm tasks without any task-specific fine-tuning, representing a 62% improvement over strong baselines. Furthermore, the model exhibits high robustness against diverse visual disturbances. Our work shows that a pointing-centric representation, combined with an RFT training paradigm, offers an effective and generalizable pathway to closing the perception-action gap in robotics.

*Keywords:* *Embodied Reasoning, General Robotic Manipulation, Reinforcement Learning*

 **Projects:** <https://embodied-r1.github.io/>

 **Code Repository:** <https://github.com/pickxiguapi/Embodied-R1>

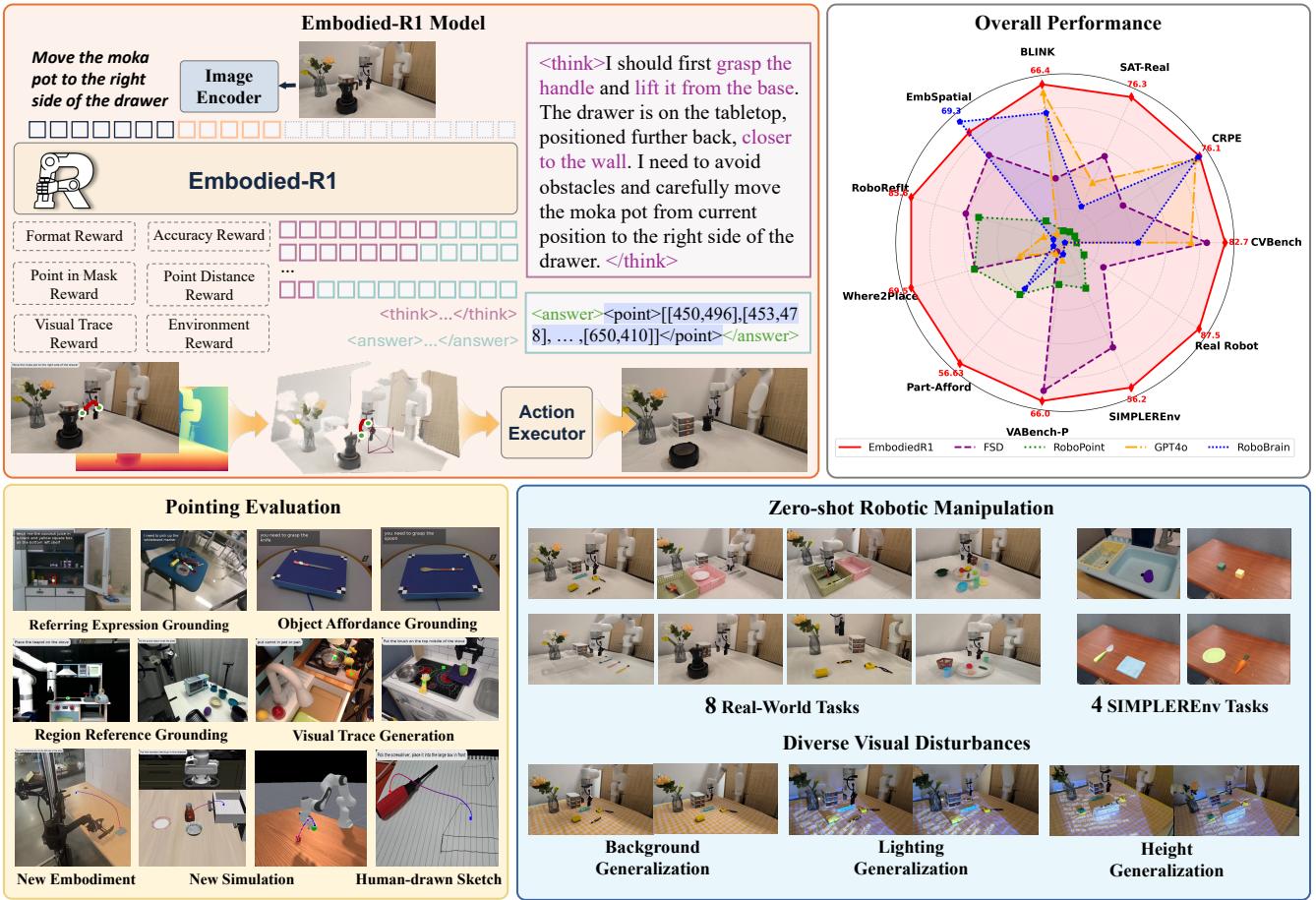
 **Datasets:** <https://huggingface.co/IffYuan>

 **Contact:** [yuanyf@tju.edu.cn](mailto:yuanyf@tju.edu.cn)

## 1. Introduction

Recent advancements in Vision-Language Models (VLMs) Liu et al. (2024b), Bai et al. (2025b) have inspired a new wave of Vision-Language-Action (VLA) models Kim et al. (2024) aimed at enhancing generalization in robotic manipulation. While these models exhibit strong visual perception and excel at imitating expert demonstrations, their manipulation performance often degrades significantly in novel settings. This critical disparity is widely recognized as the *seeing-to-doing gap* Yuan et al. (2025): a failure to reliably translate rich perceptual understanding into effective robotic actions. This gap is largely attributed to two key challenges: (a) *data scarcity*, where limited embodied data fails to sufficiently ground language and vision with physical actions Walke et al. (2023), Lin et al. (2024), and (b) *heterogeneity*, where diverse robot morphologies hinder the knowledge transferability.

The community has explored several paradigms of VLAs. End-to-end VLAs Kim et al. (2024), Nasiriany et al. (2024) aim to learn a direct mapping from multimodal inputs to action spaces. However, there is a fundamental mismatch between aligning action modalities in the physical world and pre-trained cyberspace data, which can lead to knowledge forgetting and task conflicts. Training end-to-end VLAs solely on limited



**Figure 1: The Embodied-R1 framework for zero-shot robotic manipulation through “pointing”.** Embodied-R1 takes visual and textual instructions, performs explicit reasoning, and then generates a visual trace as a universal command. The other panel showcases our comprehensive evaluation, including spatial reasoning, embodied pointing benchmarks, and real-world robot tasks.

embodied datasets hinders the improvement of zero-shot generalization capability. Modular VLAs [Huang et al. \(2024c\)](#), [Liu et al. \(2024a\)](#), [Qi et al. \(2025\)](#), on the other hand, chain together powerful specialized vision models and design pipelines for atomic steps such as object and grasp detection. However, such methods are prone to cascading failures, are difficult to tune, and also suffer from relatively high inference latency. Furthermore, such disaggregated systems often lack a holistic understanding of scene-level spatial relationships. Affordance VLAs [Li et al. \(2024b\)](#), [Yuan et al. \(2025, 2024b\)](#) offer a promising way towards more integrated solutions by training specialized embodied VLMs to predict intermediate visual aids. Despite their potential, they lack support for the comprehensive visual aids required for decision-making. Robobrain [Ji et al. \(2025\)](#) and FSD [Yuan et al. \(2025\)](#) use bounding boxes to mark objects or affordances, and they leverage visual traces to capture the dynamics of a task. Besides, Robopoint [Yuan et al. \(2024b\)](#) focuses exclusively on target regions within free space. However, all of these methods can only meet limited task requirements, as different tasks often demand a richer variety of visual aids and more comprehensive embodied grounding. FSD provides a crucial insight: embodied reasoning can effectively anchor task instructions to the correct semantic entities. However, FSD is trained on fixed Chain-of-Thought (CoT) reasoning templates through SFT; its thought process is inflexible, which limits its ability to generalize to new tasks.

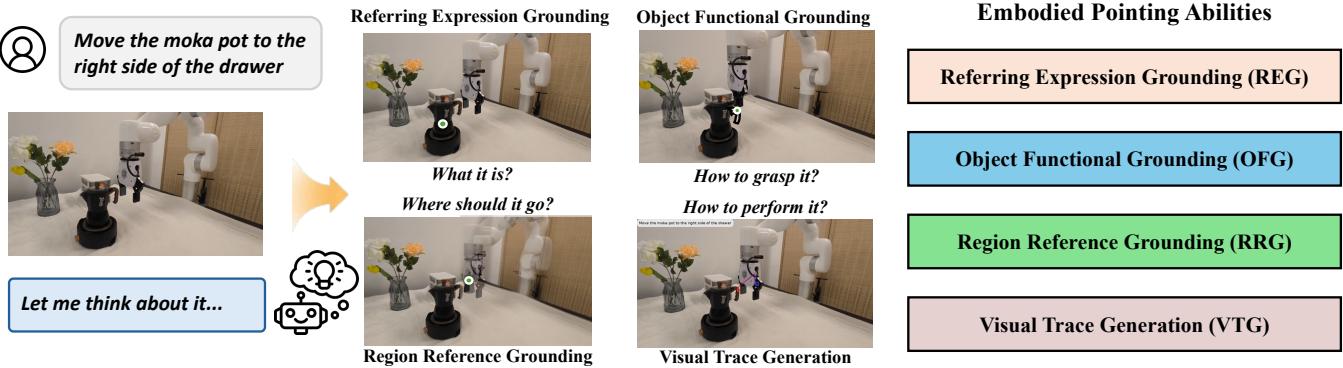


Figure 2: Overview of four embodied pointing abilities.

Summarizing observations from prior work Deitke et al. (2024), Xu et al. (2025), Yuan et al. (2025), we propose **pointing** as a highly intuitive and effective choice that bridges high-level comprehension with generalizable action prediction. A point-centric representation Yuan et al. (2024b), Deitke et al. (2024) unifies rich semantic and spatial information into a compact and accurate representation of manipulation processes, which includes object identity (*What is the object?*”), functional affordance (*How to use/grasp it?*”), desired target destination (“*Where it should be placed*”), and can even implicitly convey the execution process of actions (“*How to perform the task?*”) through a visual trace. To thoroughly assess the embodied reasoning and pointing capabilities, we define four key abilities: Referring Expression Grounding (REG), Region Referring Grounding (RRG), Object Functional Grounding (OFG), and Visual Trace Generation (VTG). We illustrate these pointing abilities in Fig. 2 and provide detailed definitions in the following sections.

Then, we propose **Embodied-R1**, an advanced embodied reasoning VLM, whose core mechanism is to achieve unified anchoring of objects and spatial concepts through “pointing”, thereby mastering general robotic manipulation. As shown in Fig. 1, Embodied-R1 first generates a detailed reasoning process and then provides the answer. With only 3B model parameters, Embodied-R1 achieves state-of-the-art performance in multiple spatial understanding and precise embodied pointing benchmarks. Embodied-R1 can also directly generate pointing signals to guide efficient robotic manipulation. By using pointing as an intermediate-level representation, Embodied-R1 grounds its reasoning in a universal visual perception task. This approach avoids the pitfalls of directly predicting low-level, embodiment-specific actions, thereby preserving and fully leveraging the strong visual generalization capabilities inherent to the pre-trained VLM. This enables the robot to achieve zero-shot control across different scenarios and tasks without any fine-tuning. Embodied-R1 achieves the best performance in SIMPLEREnv Li et al. (2024d) simulation and attains an 87.5% success rate on 8 real-world tasks based on the XArm platform. Meanwhile, Embodied-R1 enhances the robustness when facing various visual disturbances such as changes in lighting and background.

We train Embodied-R1 through Reinforced Fine-tuning (RFT), structured around a two-stage curriculum. The first stage is dedicated to building robust spatial reasoning. Subsequently, the second stage employs our specially curated *Embodied-Points-200k* dataset to foster diverse multi-task embodied pointing abilities. This extensive dataset features 200,000 high-quality questions and verification methods, drawn from diverse sources. We also design reward functions that support multi-task training, ensuring that each capability is thoroughly trained. A critical aspect of embodied pointing problems is their multi-solution nature. Consider marking a point *in the empty space to the right of a drawer*, any point within that region is considered a valid solution. This poses a significant “multi-solution dilemma” for SFT, where the model tends to overfit and merely memorize the data. In contrast, the RFT paradigm can provide positive reinforcement for all correct

answers, thereby effectively encouraging the model to develop a genuine understanding of the tasks.

Our contributions include: ① pioneering “pointing” as a unified, embodiment-agnostic representation and defining core embodied pointing abilities to bridge perception and decision; ② constructing the comprehensive *Embodied-Points-200K* dataset for these capabilities; and ③ proposing Embodied-R1, a VLM designed to boost spatial reasoning and embodied pointing, thereby delivering powerful embodied reasoning performance. ④ With only 3B parameters, Embodied-R1 attains state-of-the-art performance on 11 diverse spatial and pointing benchmarks and enables robust zero-shot robotic manipulation, achieving 56.2% success in SIMPLEREnv simulation and 87.5% in 8 real-world XArm tasks, representing a 62% improvement over strong baselines, without task-specific fine-tuning.

## 2. Embodied-R1: Advancing Embodied Reasoning from RFT

In this section, we sequentially introduce the model architecture of Embodied-R1 and its various embodied pointing abilities. Subsequently, we present the dataset construction and training methodology of Embodied-R1. Finally, we describe the deployment scheme of Embodied-R1 for performing tasks in real-world scenarios.

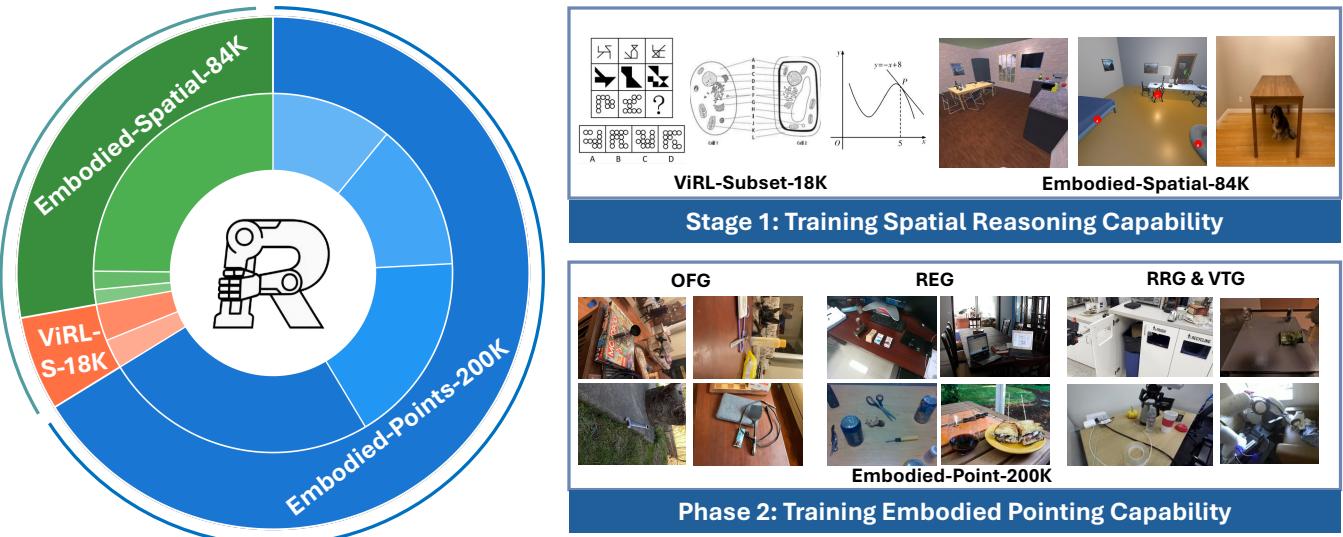
### 2.1. The Architecture and Capabilities of Embodied-R1

Embodied-R1 follows the fundamental architecture of Qwen2.5-VL [Bai et al. \(2025a\)](#), which consists of three components: a Vision Transformer (ViT) as the visual encoder, a projector, and an LLM. Given a multimodal input  $x = (I, Q)$ , where  $I$  represents the images and  $Q$  is the textual instruction, the model autoregressively predicts a textual response  $y$ . Embodied-R1 is specifically designed for embodied manipulation, enhancing spatial reasoning and embodied pointing capabilities. We define four fundamental embodied pointing abilities. We posit that these point-centric representations can serve as embodiment-agnostic intermediaries between perception and action (*overcoming heterogeneity*). This unified representation enables training on both large-scale internet datasets (from cyberspace) and embodied robotics datasets (from the physical world) (*overcoming data scarcity*), thereby promoting robust generalization to novel scenarios and tasks.

The core abilities all operate by generating a coordinate point  $p = (p, q) \in [0, w] \times [0, h]$  on the image with width  $w$  and height  $h$ . But it differs in semantic purpose and output structure:

- ① **Referring Expression Grounding (REG):** This ability localizes objects via linguistic descriptions, generating a point within its corresponding mask. It enables a robot to locate relevant objects via natural language instructions.
- ② **Region Referring Grounding (RRG):** This ability identifies a spatial region based on relational language (e.g., “the space between the cup and the bowl”) by generating a point in a suitable free-space location for object placement.
- ③ **Object Functional Grounding (OFG):** This ability identifies functionally significant part-level regions of an object (i.e., affordances). The point must lie within this functional area, such as the handle of a knife for grasping.
- ④ **Visual Trace Generation (VTG):** This ability produces an ordered sequence of points,  $\tau = \{p_t \mid t = 1, 2, \dots, T\}$ ,  $T$  denotes the sequence length, to form a complete, object-centric manipulation trajectory. This sequence provides a comprehensive spatial plan, enabling a robot to follow motion patterns dictated by instructions while avoiding obstacles. We deliberately use the visual trace of the object-centric rather than the agent-centric to achieve an agent-agnostic visual representation, ensuring a strict correspondence between the visual traces and the task instructions.

We present the visualization of each embodied pointing capability in Fig. 2.



**Figure 3: Overview of training data:** In stage 1, we focus on improving the model’s spatial reasoning capability, while incorporating a small amount of general reasoning data. In stage 2, we train the model’s embodied pointing capabilities, which comprise four distinct capability items.

## 2.2. Enhancing the Embodied Reasoning Abilities of VLM

To develop general embodied pointing capabilities, we train Embodied-R1 on three data types: embodied spatial reasoning for foundational awareness, general reasoning to preserve its existing skills, and embodied pointing to learn four key pointing abilities. We present an overview of the training data in Fig. 3.

**General and Spatial Reasoning Data** The cornerstone of this dataset is **Embodied-Spatial-84K**, specifically designed for embodied spatial awareness. These data were aggregated from two prominent benchmarks, SAT Ray et al. (2024) and WhatsUp Kamath et al. (2023). To facilitate objective performance evaluation and a verifiable reward structure, all source data were systematically converted into a unified multiple-choice format. Furthermore, to counteract the issue of catastrophic forgetting and preserve general reasoning during specialized training, we introduced a supplementary dataset, **ViRL-subset-18K**. This comprises diverse, general-knowledge instances. We strategically filtered the ViRL39K Wang et al. (2025a) dataset by removing overly difficult questions and balancing questions across different subjects and types, resulting in a final dataset of 18,000 question-answer pairs. The resulting composite dataset provides a balanced curriculum, fostering specialized spatial skills while safeguarding the model’s foundational knowledge.

**Embodied Pointing Data** To advance a suite of embodied pointing capabilities, we introduce the **Embodied-Points-200K** dataset, a high-quality, meticulously curated corpus containing about 200k samples. Due to the multi-solution dilemma inherent in embodied pointing problems, we circumvent the need to construct “question-answer” pairs typical for SFT. Instead, we structure the data as “question-verification” pairs, leveraging RFT for training. Subsequently, pre-defined reward functions for each task evaluate the response based on the verification and calculate the corresponding rewards. We briefly outline the pipeline for generating point data below. For more details on data generation, please refer to Appendix A.

- **REG Data:** In robotic manipulation, precise localization is critical. However, traditional bounding boxes suffer from inherent ambiguity and fail to meet this requirement. We constructed a point-centric REG dataset. Its data sources are diverse, integrating general web images from RefCOCO Kazemzadeh et al.

(2014) with multiple embodied datasets (RoboRefIt Lu et al. (2023), Yuan et al. (2024b) and RoboPoint Yuan et al. (2024b)) to ensure broad coverage. We critically adjusted the success criterion for the task: instead of outputting a bounding box, the model must specify a single point. The prediction is considered correct if the point falls within the object’s segmentation mask.

- **RRG Data:** To enable robots to comprehend complex spatial placement commands, we developed an automated data generation pipeline for creating relation-aware placement regions. This pipeline processes a large corpus of open-source embodied dataset (about 1 million), and after rigorous filtering, yields 33,000 high-quality samples. The core process includes: **① Region Extraction:** extracting the final position of the manipulated object from the terminal frame; **② Region Referring:** calculating the precise placement of the region relative to a reference object in the scene; and **③ Rendering:** rendering this spatial placement information back onto the initial image. To ensure data diversity and quality, we designed a heuristic filtering strategy, which covers a rich variety of spatial relationships, object configurations, and scenes. Furthermore, we leveraged the Isaac Gym simulation engine to generate a synthetic dataset of 3D object rearrangements, equipping the model with 3D spatial awareness. In this task, which takes RGB-D images as input, the model is required to place objects in the correct relative positions according to instructions. Task success is automatically determined and fed back by the simulation based on the true physical state.
- **OFG Data:** To enhance Embodied-R1’s fine-grained understanding of functional object parts, we built a dataset of 40,000 functional grasping points. We leveraged the HandAL dataset Guo et al. (2023), which contains 212 real-world objects with meticulously annotated manipulable parts. We converted these part annotations into bounding boxes to serve as verification for the OFG tasks. Meanwhile, we used the GPT-4o model to rewrite function-related questions (e.g., “Which part should be held when using a knife to cut vegetables?”), enhancing the model’s ability to generalize its knowledge of object affordances.
- **VTG Data:** We constructed an object-centric visual trace dataset that exclusively tracks the object’s movement. The extraction pipeline follows the methodology of Yuan et al. (2025) and consists of three main steps: **① Key Object Proposal:** Using GPT-4o to identify the primary object of interest for a given task. **② KeyPoint Identification:** A self-supervised keypoint extractor Huang et al. (2024c), in conjunction with Grounded-SAM Ren et al. (2024), is used to automatically identify the object’s grasping point. **③ Point Tracking and Projection:** We used Cotracker3 Karaev et al. (2024) to compute the dense temporal visual trace originating from the keypoints. Next, the trajectory is then downsampled into 8 equidistant discrete points and projected back onto the initial image, creating an “image-visual trace” pair. Notably, using multiple pre-trained vision models in the process inevitably introduces noise. We implemented rigorous rule-based filtering and continually validated our approach using a manually annotated test set. Based on this feedback, we iteratively refined the filtering criteria to improve the quality of the dataset.

**Training Strategy** Based on the collected data, Embodied-R1 adopts a two-stage training process: the first stage focuses on enhancing spatial reasoning abilities, as spatial reasoning serves as the foundation for point comprehension; the second stage further trains embodied pointing capabilities using point-centric, multi-task mixed data. At each stage, we train a policy  $\pi_\theta(y|x)$  to generate output  $y$  by maximizing expected reward  $\max_\theta \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} [r(y, x)]$ . Training is performed using the GRPO Guo et al. (2025), Shao et al. (2024) algorithm. The behavior policy  $\pi_{\theta_{\text{old}}}$  generates  $G$  candidate responses  $\{y_i\}_{i=1}^G$  following input  $x$ . The advantage for the  $i$ -th response at time step  $t$  is computed by normalizing rewards:

$$\hat{A}_{i,t} = \frac{r(y_i, x) - \text{mean}(\{r(y_1, x), \dots, r(y_G, x)\})}{\text{std}(\{r(y_1, x), \dots, r(y_G, x)\})}. \quad (1)$$

Then we incorporate a clipped surrogate loss with the clip function:

$$\mathcal{L}(\theta) = \frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|y_i|} \min \left[ \frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x, y_{i,<t})} \hat{A}_{i,t}, \text{clip} \left( \frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x, y_{i,<t})}, 1 - \epsilon, 1 + \epsilon \right) \hat{A}_{i,t} \right]. \quad (2)$$

### 2.3. Multi-task Reward Design

Embodied-R1 mixes the training of multiple tasks at each stage by uniformly sampling from a shared distribution, meaning that different tasks are included in every training batch. We aim to leverage mixed training to share general knowledge across various embodied pointing tasks, thereby achieving better alignment for pointing coordinates and improving training efficiency. However, due to the tendency of reward optimization in RL, simpler tasks are more likely to receive higher rewards and thus dominate policy training. To address this issue in multi-task mixed training, we have carefully designed multiple verifiable reward functions that support various types of question answering and pointing annotations. Each task has a distinct total reward function  $\mathcal{R}$ , which is composed of several primary rewards  $r$  weighted differently.

**Format Rewards:** To encourage structured outputs  $y$ , we define a unified format reward  $r_{\text{format}}(y)$ . The reasoning process is enclosed within the `<think></think>` tags, and the answer is wrapped in `<answer></answer>`. For pointing reasoning tasks, we further require that the model's output must follow the `<point>[[x1,y1],[x2,y2],...]</point>` format, where all predicted coordinates are in the original pixel image coordinate system. By adopting a unified point space, we facilitate the sharing of common knowledge across different tasks. If all the above conditions are satisfied, the output is marked as tags `valid`.  $r_{\text{format}}(y) = \mathbb{I}(\text{tags valid}(y))$ .

**Accuracy Rewards:** For the multiple-choice questions for general QA tasks, we adopt an accuracy reward to evaluate whether the answer extracted from the response matches the standard answer  $g$ . The model only receives a positive incentive when the answers are consistent.  $r_{\text{acc}}(y, g) = \mathbb{I}(y = g)$ .

**Point in Mask Reward:** For pointing tasks, point in mask reward function  $r_{\text{mask}}$  is determined by whether the predicted output point  $p$  lies within the ground-truth answer mask  $M_{\text{gt}}$ . The reward function can be formally expressed as  $r_{\text{mask}}(p, M_{\text{gt}}) = \mathbb{I}(p \in M_{\text{gt}})$ .

**Point Distance Reward:** To improve learning efficiency, we also designed a dense auxiliary reward  $r_{\text{dis}}$ , which is used to guide the predicted point to approach the target region  $M_{\text{gt}}$ . The Euclidean distance is  $d = \|p - g\|_2$ , where  $g$  is the center of the  $M_{\text{gt}}$ . Given pixel distance thresholds  $D_{\text{min\_thresh}}$  and  $D_{\text{max\_thresh}}$ ,  $r_{\text{dis}}$  is then defined as  $r_{\text{dis}}(p, M_{\text{gt}}) = \min \left( 1.0, \max \left( 0.0, 1.0 - \frac{d - D_{\text{min\_thresh}}}{D_{\text{max\_thresh}} - D_{\text{min\_thresh}}} \right) \right)$ , with limiting the scope of this reward to 0-1.

**Visual Trace Reward:** For evaluating generated visual trace, rewards are derived from trajectory similarity metrics comparing the predicted trajectory  $\tau$  with the ground-truth trajectory  $\tau_{\text{gt}}$ . First, we compare the number of points in the  $\tau$  and  $\tau_{\text{gt}}$ . Using the longer one as the reference, we interpolate both trajectories to have the same number of points and then proceed with the calculation Root Mean Square Error (RMSE):  $d_{\text{RMSE}}(\tau, \tau_{\text{gt}})$ . Similarly, we use the  $D_{\text{RMSE\_min}}$  and  $D_{\text{RMSE\_max}}$  hyperparameters to ensure that the reward remains between 0 and 1. The reward is calculated as:  $r_{\text{trace}}(\tau, \tau_{\text{gt}}) = \min \left( 1.0, \max \left( 0.0, 1.0 - \frac{d_{\text{RMSE}}(\tau, \tau_{\text{gt}}) - D_{\text{RMSE\_min}}}{D_{\text{RMSE\_max}} - D_{\text{RMSE\_min}}} \right) \right)$ .

**Environment Reward:** The environment reward  $r_{\text{env}}$  provides a direct signal of task completion based on feedback from a simulated environment. This reward is used for a portion of the training data originating from the Isaac Gym simulator in the RRG task. It parses and executes the model's output in the simulator

**Table 1:** Performance comparison on spatial reasoning benchmarks. **Bold** indicates the highest value among open-source models, and underlined values show the second-highest scores.

	CVBench					CRPE			SAT		BLINK				EmbSp.	Rank	
	Count	2DRel	3DDep	3DDis	Avg.	Subj.	Pred.	Obj.	Avg.	Val	Real	MV	RelDepth	SpRel	Obj	Avg.	Test
<i>Closed-source models</i>																	
GPT4V	62.4	71.1	79.8	68.3	70.4	76.7	65.1	68.5	70.1	44.8	50.7	55.6	59.7	72.7	54.9	60.7	36.1
GPT4o	65.9	85.5	87.8	78.2	79.4	81.9	71.8	73.6	75.8	49.4	57.5	60.2	74.2	69.2	59.8	65.9	49.1
<i>Open-source models</i>																	
LLaVA-1.5-13B	58.2	46.6	53.0	47.8	51.4	57.4	54.2	55.2	55.6	51.4	41.6	41.4	53.2	69.9	52.5	54.2	35.1
SAT-Dynamic-13B	61.5	89.7	80.7	73.0	76.2	60.6	57.6	65.2	61.1	87.7	54.9	44.4	73.4	66.4	45.9	57.5	51.3
RoboPoint-13B	56.5	77.2	81.5	57.7	68.2	66.3	62.4	70.9	66.5	53.3	46.6	44.4	62.1	65.7	56.6	57.2	51.4
ASMV2-13B	58.9	68.9	68.9	68.9	66.4	69.2	59.0	65.3	64.5	63.9	46.7	44.4	56.5	65.0	63.9	57.5	57.4
FSD-13B	62.4	86.5	<b>88.0</b>	<b>86.7</b>	80.9	75.2	65.1	70.4	70.2	<u>73.2</u>	63.3	46.6	70.2	78.3	46.7	60.5	63.3
RoboBrain-7B	64.3	76.6	84.0	72.0	74.2	<u>81.3</u>	<b>71.8</b>	74.8	<u>76.0</u>	45.3	52.2	<b>55.6</b>	75.8	<b>81.8</b>	45.1	64.6	<b>69.3</b>
Qwen2.5VL-3B	68.4	72.8	77.0	68.2	71.6	80.7	71.0	<b>76.1</b>	<u>76.0</u>	48.7	45.1	44.4	66.9	79.7	55.7	61.7	62.8
Embodied-SFT	66.4	<b>92.3</b>	<b>85.8</b>	83.8	<u>82.1</u>	74.7	<b>71.3</b>	73.8	73.3	59.3	65.5	50.4	<b>81.5</b>	78.3	54.9	66.3	63.1
Embodied-R1 w/o CS	70.4	90.2	84.5	81.0	81.5	80.3	69.9	<u>75.4</u>	75.2	70.0	<u>73.9</u>	47.4	72.6	79.7	56.6	64.1	65.4
Embodied-R1	<b>70.6</b>	<u>90.8</u>	84.7	<u>84.8</u>	<b>82.7</b>	82.2	70.7	<u>75.4</u>	<b>76.1</b>	70.0	<b>76.3</b>	<u>51.1</u>	<u>76.6</u>	<b>80.4</b>	<u>57.4</u>	<b>66.4</b>	<b>67.4</b>

and then returns a binary outcome indicating success or failure. The reward  $r_{\text{env}}$  is formally defined as an indicator function:  $r_{\text{env}}(y) = \mathbb{I}(\text{Simulate}(y))$ .

**Total Reward:** The total reward  $\mathcal{R}$  for each task is formulated as a combination of these individual reward components. We define the reward function library  $\mathcal{F} = \{r_{\text{format}}, r_{\text{acc}}, r_{\text{mask}}, r_{\text{dis}}, r_{\text{trace}}, r_{\text{env}}\}$ . Each component function evaluates a specific aspect of the model’s performance. Since we conduct mixed training on multiple tasks, in order to ensure that each task is equally and sufficiently trained, we constrain the total reward  $\mathcal{R}$  for each task to the range of 0 to 1, which is implemented as follows.  $\mathcal{R}$  is formulated as a weighted-sum combination:  $\mathcal{R} = \sum_{r \in \mathcal{F}} w_r \cdot r$ . The task-specific weights  $w_r$  are normalized to sum to one ( $\sum_{r \in \mathcal{F}} w_r = 1$ ). This structure guarantees that the total reward  $\mathcal{R}$  is also bounded within the range [0, 1] and allows us to tailor the reward signal for each task’s specific needs. For example, the RRG task requires simultaneously satisfying the format requirements, ensuring that the predicted points are within the specified region, and accelerating training by employing dense distance rewards. At this point,  $\mathcal{R}_{\text{RRG}}$  is defined as  $\mathcal{R}_{\text{RRG}} = 0.1r_{\text{format}} + 0.2r_{\text{dis}} + 0.7r_{\text{mask}}$ . This consistent scaling of rewards across different tasks is crucial for stabilizing. We refer to Appendix B for detailed hyperparameters.

## 2.4. Action Executor of Embodied-R1

Through simple pointing, Embodied-R1 can be flexibly integrated with various downstream action executors. This allows it to reason from any step, freely select the necessary pointing abilities, and combine them with a motion planner to achieve zero-shot robot control. We offer two primary decision-making approaches for this: the Affordance Points Branch (-P) and the Visual Traces Branch (-V). **Affordance Points Branch:** Embodied-R1 is capable of predicting multiple key grasping and placement points through abilities such as RRG and OFG. We then utilize CuRobo [Sundaralingam et al. \(2023\)](#) as the motion planner. CuRobo is responsible for generating collision-free paths to guide the robot’s end effector to the inferred target affordance points. **Visual Traces Branch:** Leveraging object-centric visual traces from VTG, we first map 2D visual traces  $\tau$  to 3D Cartesian coordinates using the Pinhole camera model and initial depth information. We interpolate these discrete points to form a complete motion trajectory in SE(3) space. Then, the robot follows the visual trace for execution like FSD [Yuan et al. \(2025\)](#).

### 3. Experiments

To validate Embodied-R1’s generalization in robotic manipulation, we conducted extensive experiments evaluating its *Seeing* (spatial reasoning and pointing capabilities) and *Doing* (manipulation tasks) dimensions. Our evaluation encompassed 11 QA benchmarks, 4 simulated tasks (SIMPLEREnv Li et al. (2024d), and 8 real-world robot (xArm platform) tasks. We used the Qwen2.5-VL-3B-Instruct Bai et al. (2025a) model as the initial model. First, we trained it for two epochs using the first-stage Embodied-Spatial-84K and ViRL-subset-18K datasets. Then, we continued training for one epoch with the second-stage EmbodiedPoints-200K dataset. For all experiments, we focus on comparing SFT models trained with the same batch size and data, which we refer to as Embodied-SFT. For training details, please refer to Appendix B and Appendix C.

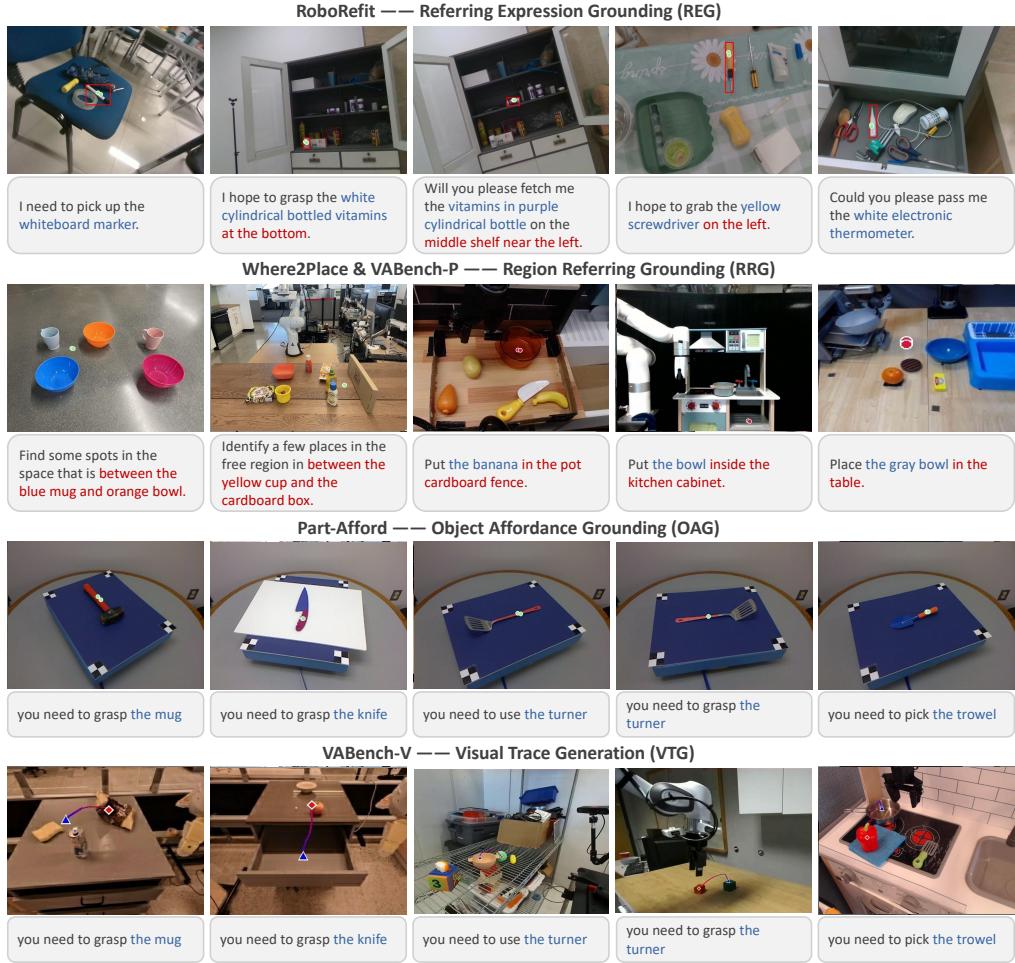
#### 3.1. Evaluation of Spatial Reasoning Capabilities

**Setup:** To verify the effect of the first stage, we first evaluated Embodied-R1’s spatial perception and understanding capabilities. We chose five widely-adopted spatial reasoning benchmarks: CVBench Tong et al. (2024), BLINK Fu et al. (2024), CRPE Wang et al. (2025b), SAT Ray et al. (2024), and EmbSpatial-Bench Du et al. (2024), which collectively comprise 15 subtasks evaluating diverse spatial competencies. We included two leading closed-source models, GPT-4o Hurst et al. (2024) and GPT-4V. We also compared open-source spatial-enhanced models, such as SAT-Dynamic Ray et al. (2024), RoboPoint Yuan et al. (2024b), ASMV2 Wang et al. (2025b), RoboBrain Ji et al. (2025) and FSD Yuan et al. (2025). Notably, both RoboBrain and FSD enhance reasoning abilities and ultimately apply them to embodied tasks. We also specifically compared the Embodied-R1 w/o CS model, which does not incorporate common sense knowledge, that is, the training dataset excludes ViRL.

**Results:** Table 1 showcases Embodied-R1’s strong performance across a range of spatial reasoning benchmarks with **only 3B parameters**. Embodied-R1 achieves the best results among all evaluated open-source models, consistently outperforming other spatially enhanced models with an average rank of 2.1. Incorporating common-sense data further boosts performance, with Embodied-R1 outperforming its own variant trained without this data (Rank 3.4). We hypothesize that more diverse data can stimulate Embodied-R1’s exploratory reasoning capabilities. Additionally, Embodied-R1 surpasses models trained solely with SFT, i.e., Embodied-SFT (Rank 3.7). Embodied-R1 demonstrates superior spatial reasoning abilities compared to other specialized embodied spatial VLMs, including RoboBrain-7B and FSD-13B.

#### 3.2. Evaluation of Pointing Capabilities

**Setup:** Regarding Embodied-R1’s Embodied Point Reasoning and Generation Abilities, we conducted a comprehensive evaluation across the defined capabilities. ① For Referring Expression Grounding (REG) capacity, we selected the RoboRefIt Lu et al. (2023) test dataset. Unlike human-centered datasets like Ref-Coco Kazemzadeh et al. (2014), RoboRefIt features images containing similar-looking objects distinguishable primarily through relational references, presenting a more challenging scenario for embodied perception. ② Region Referring Grounding (RRG) capacity was evaluated using both the Where2Place Yuan et al. (2024b) and VABench-Point Yuan et al. (2025) benchmarks. The VABench-Point (VABench-P) benchmark includes more complex task descriptions that are closer to real-life scenarios and require further reasoning. ③ For Object Affordance Grounding (OFG) capacity, we constructed a benchmark named Part-Afford Benchmark by filtering 2000 grasp-related affordances from the RGBD-Part-Affordance Myers et al. (2015) dataset. This benchmark encompasses 105 types of kitchen, workshop, and gardening tools, designed to evaluate the generalization capability of affordance prediction in OOD scenarios. ④ For Visual Trace Generation (VTG)



**Figure 4: Visualizing Embodied-R1’s Performance on Various Pointing Tasks.** The model can follow diverse text instructions and generalize its capabilities to novel, unseen environments.

capacity, we followed the VABench-VisualTrace (VABench-V) Yuan et al. (2025) evaluation methodology, measuring MAE, RMSE, and LLM Scores. ⑤ In addition, we fine-tuned Embodied-R1 on the synthetic object dataset from YCB Xiang et al. (2017) and ObjaverseXL Deitke et al. (2023), which contains both RGB and depth images. This model is named Embodied-R1-RGBD, where RGB and depth images are input separately to predict the target region’s position and depth of the object. For comparison, the model that only takes RGB images as input is referred to as Embodied-R1-RGB. This 3D capability was tested on the Open6DOR-Position benchmark Qi et al. (2025) and compared against other VLM-based methods.

**Results:** We present the results of RoboRefit, Where2Place, VABench-P, and the Part-Afford benchmark in Table 2, the results of VABench-V in Table 3, and the results of Open6DoR in Table 4. Some visualizations are provided in Fig. 4. The following are several observations regarding the experimental results:

(O1) **Powerful general VLMs perform poorly on pointing tasks**, such as GPT-4o and Qwen2.5-VL, indicating the necessity of specialized training with additional data for these capabilities. This indicates that it is necessary to train embodied VLMs with strong spatial reasoning and pointing abilities.

(O2) **Embodied-R1 demonstrates superior performance across key benchmarks.** In Table 2 and Table 3, on tasks such as REG (RoboRefit), RRG (Where2Place, VABench-P), OFG (Part-Afford) and VTG (VABench-V),

Embodied-R1 achieves state-of-the-art results compared to various specialized and general VLM baselines. Compared with FSD and RoboPoint, which also focus on pointing tasks, Embodied-R1 demonstrates stronger embodied reasoning capabilities and achieves significant improvements across multiple tasks. We present the prediction results for different tasks in Fig. 4, demonstrating that Embodied-R1 is capable of mastering a wide range of abilities with a single model. Embodied-R1 also demonstrates high accuracy for small objects and complex spatial relationships in cluttered scenes.

**(O3) Embodied-R1 generates highly accurate visual traces for robotic manipulation.** On the VABench-V benchmark, Embodied-R1 achieves the lowest RMSE and MAE, indicating its ability to produce precise point sequences for traces, a crucial aspect for reliable action execution. Embodied-R1 also demonstrates significant improvement, indicating that R1 focuses on reasoning about the ideal visual trajectory rather than rote memorization. We refer to more visualization in Fig. 4 and Appendix D.

**(O4) Embodied-R1 exhibits strong capabilities in generating precise depth information for embodied tasks.** As shown in Table 4, on the Open6DOR-Position benchmark, the Embodied-R1-RGBD variant achieves high performance, highlighting the effectiveness of our model and the value of incorporating RGB and Depth images. However, in the complex relations of level 1, the performance of Embodied-R1 is slightly lower compared to the 2D version. We believe that at higher levels of relational complexity, depth map understanding may be more prone to hallucinations, leading to incorrect answers. Here, we have only conducted a preliminary exploration of the RGBD version, and we consider this to be a promising direction for further improvement.

**(O5) Embodied-R1 significantly outperforms models trained solely with SFT.** Compared to the Embodied-SFT, Embodied-R1 demonstrates substantial improvements across these tasks, validating the benefits of RFT in developing strong generalization capabilities for embodied point reasoning and generation.

**Table 2: Performance on 4 Pointing benchmarks.** The score is the accuracy of points falling within the target region.

Model	RoboRefit	Where2Place	VABench-P	Part-Afford
GPT4o	15.28	29.06	9.30	10.15
ASMV2	48.40	22.00	10.07	13.75
RoboBrain	10.10	16.60	7.00	25.25
RoboPoint	49.82	46.01	19.09	27.60
FSD	56.73	45.81	61.82	9.55
Qwen2.5VL	74.90	31.11	9.89	23.42
Embodied-SFT	83.85	41.25	50.46	40.20
Embodied-R1	<b>85.58</b>	<b>69.50</b>	<b>66.00</b>	<b>56.63</b>

**Table 3: Performance on VABench-V.** Lower values are better for RMSE/MAE, higher is better for LLM Score.

Model	RMSE ↓	MAE ↓	LLM Score ↑
GPT-4o	136.1	113.5	4.4
DINOv2 Predictor	128.3	117.5	4.0
RoboBrain	121.6	103.8	4.5
FSD	78.3	63.4	6.2
Embodied-SFT	109.4	65.2	6.2
Embodied-R1	<b>77.8</b>	<b>45.0</b>	<b>7.3</b>

**Table 4: Performance on Open6DOR-Position Benchmark**

Benchmark	Level0	Level1	Overall
GPT-4V	46.8	39.1	45.2
Qwen2.5-VL <a href="#">Bai et al. (2025b)</a>	59.5	36.2	54.9
VoxPoser <a href="#">Huang et al. (2023)</a>	35.6	21.7	32.6
SoFar <a href="#">Qi et al. (2025)</a>	<u>96.0</u>	<b>81.5</b>	<b>93.0</b>
Embodied-SFT	62.4	44.7	58.9
Embodied-R1-RGB	68.5	<u>59.4</u>	66.8
Embodied-R1-RGBD	<b>99.8</b>	50.9	<u>90.2</u>

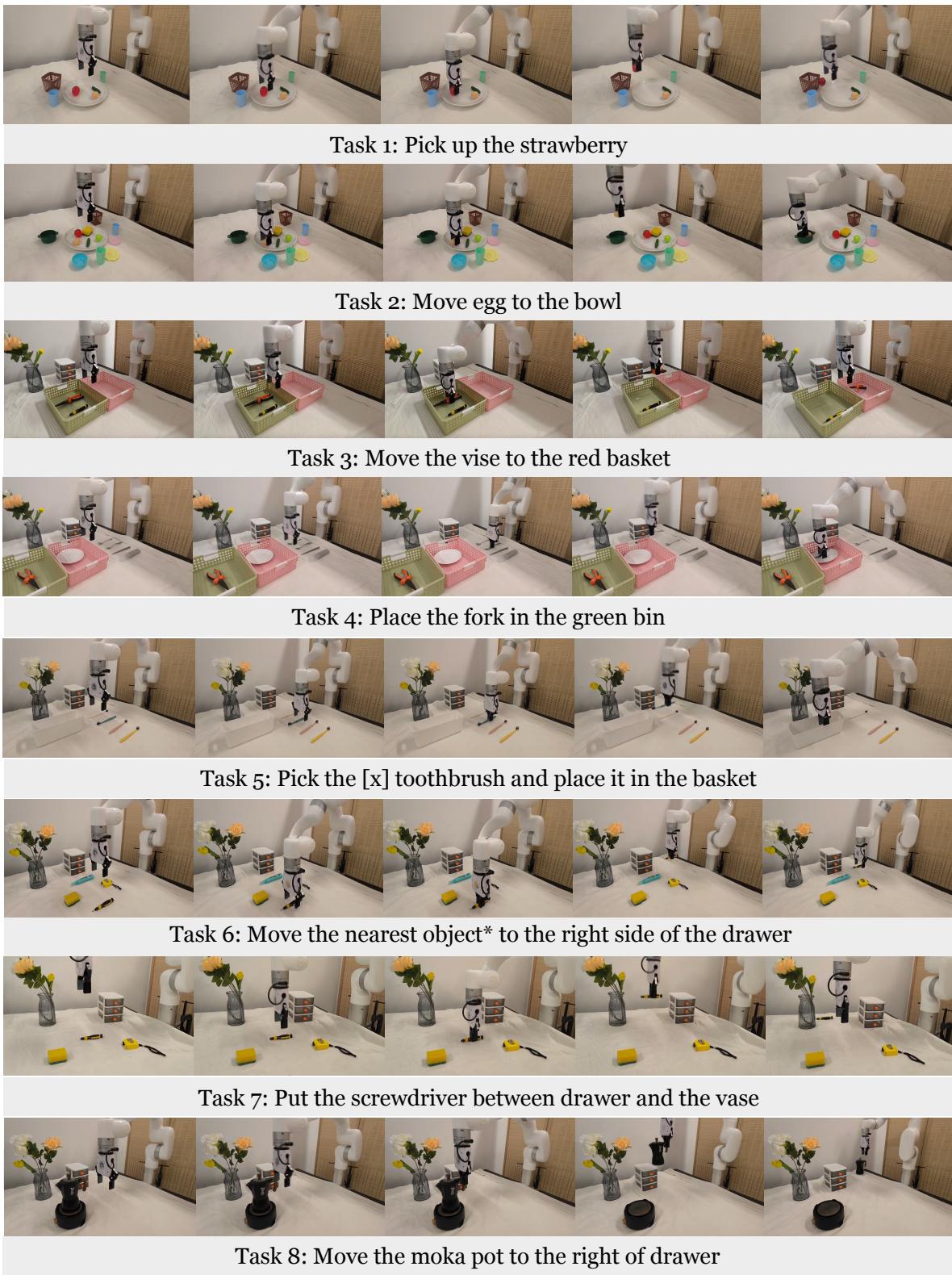
**Table 5: SimplerEnv Evaluation on WidowX Robot.** The results of baselines are derived from Qu et al. (2025). ZS: zero-shot, FT: fine-tuning using BridgeData. The results of Embodied-R1 and SoFar are averaged over three rounds of experiments, with each round consisting of 24 runs. The other results are derived from Yuan et al. (2025). **Spoon→Towel:** Put Spoon on Towel, **Carrot→Plate:** Put Carrot on Plate, **Green→Yellow:** Stack Green Block on Yellow Block, **Eggplant→Basket:** Put Eggplant in Yellow Basket.

Model	Spoon→Towel		Carrot→Plate		Green→Yellow		Eggplant→Basket		Avg.
	Grasp	Succ.	Grasp	Succ.	Grasp	Succ.	Grasp	Succ.	
<i>End-to-end VLAs</i>									
RT-1-X O’Neill et al. (2023)	16.7%	0.0%	20.8%	4.2%	8.3%	0.0%	0.0%	0.0%	1.1%
Octo-S Team et al. (2024)	77.8%	47.2%	27.8%	9.7%	40.3%	4.2%	87.5%	56.9%	30.0%
OpenVLA Kim et al. (2024)	4.1%	0.0%	33.3%	0.0%	12.5%	0.0%	8.3%	4.1%	1.0%
RoboVLM (ZS) Li et al. (2024c)	37.5%	20.8%	33.3%	25.0%	8.3%	8.3%	0.0%	0.0%	13.5%
RoboVLM (FT) Li et al. (2024c)	54.2%	29.2%	25.0%	25.0%	45.8%	12.5%	58.3%	58.3%	31.3%
SpatialVLA (ZS) Qu et al. (2025)	25.0%	20.8%	41.7%	20.8%	58.3%	25.0%	79.2%	70.8%	34.4%
SpatialVLA (FT) Qu et al. (2025)	20.8%	16.7%	29.2%	25.0%	62.5%	29.2%	100.0%	100.0%	42.7%
<i>Modular VLAs</i>									
MOKA Liu et al. (2024a)	75.0%	45.8%	64.0%	41.6%	83.3%	33.3%	50.0%	12.5%	33.3%
SoFar Qi et al. (2025)	69.4%	55.5%	73.6%	56.9%	87.5%	62.5%	68.0%	40.2%	53.8%
<i>Affordance VLAs</i>									
RoboPoint Yuan et al. (2024b)	58.3%	16.7%	41.7%	20.8%	54.2%	8.3%	66.7%	25.0%	17.7%
FSD Yuan et al. (2025)	58.3%	41.6%	58.3%	50.0%	91.6%	33.3%	37.5%	37.5%	40.6%
<b>Embodied-R1</b>	65.2%	62.5%	81.9%	68.0%	93.0%	36.1%	62.5%	58.3%	56.2%

### 3.3. Evaluation of Embodied-R1 for Robot Manipulation

**SimplerEnv Simulation** We utilized Embodied-R1 to generate object affordance points and target region points with CuRobo Sundaralingam et al. (2023) planner, then performed zero-shot deployment on the WidowX arm. We compared its performance with three types of VLAs. For end-to-end VLAs, we selected Octo Team et al. (2024), OpenVLA Kim et al. (2024), RoboVLM Li et al. (2024c), and SpatialVLA Qu et al. (2025). For modular VLAs, we chose SoFar Qi et al. (2025) and MOKA Liu et al. (2024a). The SoFar pipeline integrates Florence-2 Xiao et al. (2024), SAM Kirillov et al. (2023), and GPT-4o to accomplish the tasks. MOKA utilizes multiple models to obtain keypoints for task execution. We also compared RoboPoint Yuan et al. (2024b) and FSD Yuan et al. (2025) as affordance VLA baselines. The results are presented in Table 5. Embodied-R1 surpasses the baseline in most tasks with a 56.2% average success rate, demonstrating the strong generalization ability. Notably, Embodied-R1 achieves the best average performance, even outperforming specially fine-tuned VLA models. We believe that, compared to end-to-end VLAs, perception-centered VLAs (modular and affordance-based methods) generally exhibit better zero-shot generalizability.

**Real World Robot Evaluation** We conducted zero-shot real-world evaluations of different methods using an xArm 6 robot across eight tabletop manipulation tasks. The experimental setup uses an Intel RealSense L515 LiDAR camera positioned at a third-person perspective, with an image resolution of  $640 \times 480$ . The experimental objects, scenes, and tasks have never been seen in the training data and are directly used to test the performance of the OOD generalization. Table 6 presents the complete results for all tasks, including stage success rates (Grasp./Correct Obj.) and final success rates (Succ.). Fig. 5 illustrates the execution process of all real-world tasks. Compared to the RoboPoint and FSD baselines, Embodied-R1 achieved an improvement of over 60%, reaching a zero-shot success rate of 87.5%. We found that this improvement mainly stems from the poor performance of baseline algorithms on tasks requiring spatial reasoning (such as moving the nearest object). Additionally, the baselines exhibited low success rates when manipulating rigid objects that are difficult to grasp, such as the screwdriver and moka pot. In contrast, the Embodied-R1 correctly identified



**Figure 5: The process of Embodied-R1 performing real-world tasks.**

**Table 6: Real-world experimental evaluation results.** The first two tasks were conducted 5 times each, while the other tasks were conducted 6 times each. The best results are highlighted in bold. **Embodied-R1-P:** the version that predicts grasp and placement points; **Embodied-R1-V:** the version that predicts visual traces. [x]: The instruction for each trial is a randomly selected color, including white/green/red/yellow. **Nearest object\***: grasping the object closest to the camera’s viewpoint. *Grasp.* and *Correct Obj.* represent stage success rates, i.e., successfully grasping the target object. *Succ.* denotes the overall success rate.

	Pick up the strawberry	Move the egg to the bowl	Move the vase to the red basket		Place the fork in the green bin		Pick the [x] toothbrush and place it to the bucket		Move the nearest object* to the right side of the drawer		Put the screwdriver between drawer and the vase		Move the moka pot to the right of drawer		Avg
	Succ.	Succ.	Grasp.	Succ.	Grasp.	Succ.	Correct Obj.	Succ.	Correct Obj.	Succ.	Grasp.	Succ.	Grasp.	Succ.	Succ.
RoboPoint	40.0%	60.0%	50.0%	0.0%	0.0%	0.0%	16.7%	0.0%	0.0%	0.0%	66.7%	0.0%	0.0%	0.0%	12.5%
FSD	20.0%	80.0%	66.7%	33.3%	16.7%	16.7%	16.7%	0.0%	0.0%	0.0%	66.7%	33.3%	16.7%	16.7%	25.0%
<b>Embodied-R1-P</b>	<b>100.0%</b>	<b>100.0%</b>	<b>66.7%</b>	<b>50.0%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>83.3%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>50.0%</b>	<b>33.3%</b>	<b>83.3%</b>
<b>Embodied-R1-T</b>	<b>100.0%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>66.7%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>33.3%</b>	<b>33.3%</b>	<b>87.5%</b>

these cases and achieved high success rates, demonstrating the effectiveness of its reasoning process. Overall, Embodied-R1-V generated more accurate annotations than Embodied-R1-P, resulting in a slightly higher average success rate.

We also selected the task of "moving the nearest object to the right side of the drawer" to test the model's robustness under the zero-shot setting by introducing visual disturbances such as changes in background, lighting, and height. As shown in Table 7, Embodied-R1 demonstrates outstanding generalization when facing various visual disturbances. Surprisingly, the results indicate that our method exhibits strong adaptability to different lighting conditions, accurately locating objects and completing the task even under the poorest lighting. In addition, changing the background has no effect on task performance at all. This experiment confirms that pointing serves as a general representation capable of maintaining policy performance and robustness under visual disturbances. We present the experimental demo in Fig. 6.

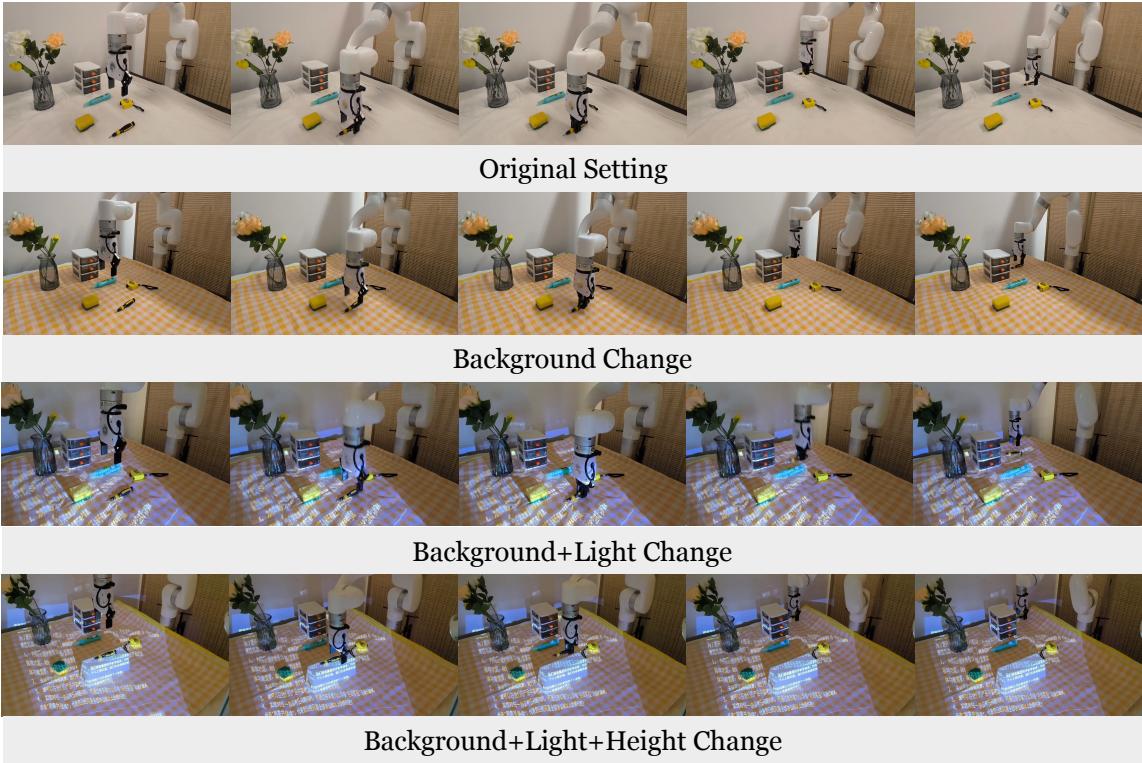
**Table 7: Performance under different visual disturbances for the task “Move the nearest object\* to the right side of the drawer”.** Each task runs 6 tests. Visualizations are in Fig. 6.

Model	Task	Grasp.	Succ.
Embodied-R1	Original	100%	100%
	Background Change	100%	100%
	Background+Light Change	83%	83%
	Background+Light+Height Change	83%	83%

### 3.4. Further Analysis of Embodied-R1

**Embodied-R1 already acquired embodied reasoning capabilities.** As shown in Fig. 7, we demonstrated the reasoning pathways of Embodied-R1 when facing different tasks. Even without any SFT stage, Embodied-R1 exhibited a human-like and rational reasoning process: it first infers the target object to focus on based on the task goal, then analyzes the relative spatial relationship between the object and the surrounding environment, and subsequently performs step-by-step reasoning to determine the target location (in RRG and VTG tasks), strictly adhering to a structured process of reasoning before providing the final answer. In some cases, Embodied-R1 still demonstrates a clear and accurate reasoning process even when confronted with cluttered scenarios. Appendix D presents several case comparisons between RL and SFT.

**Embodied-R1 exhibits strong generalization capabilities.** As shown in Fig. 8, we specifically designed tests

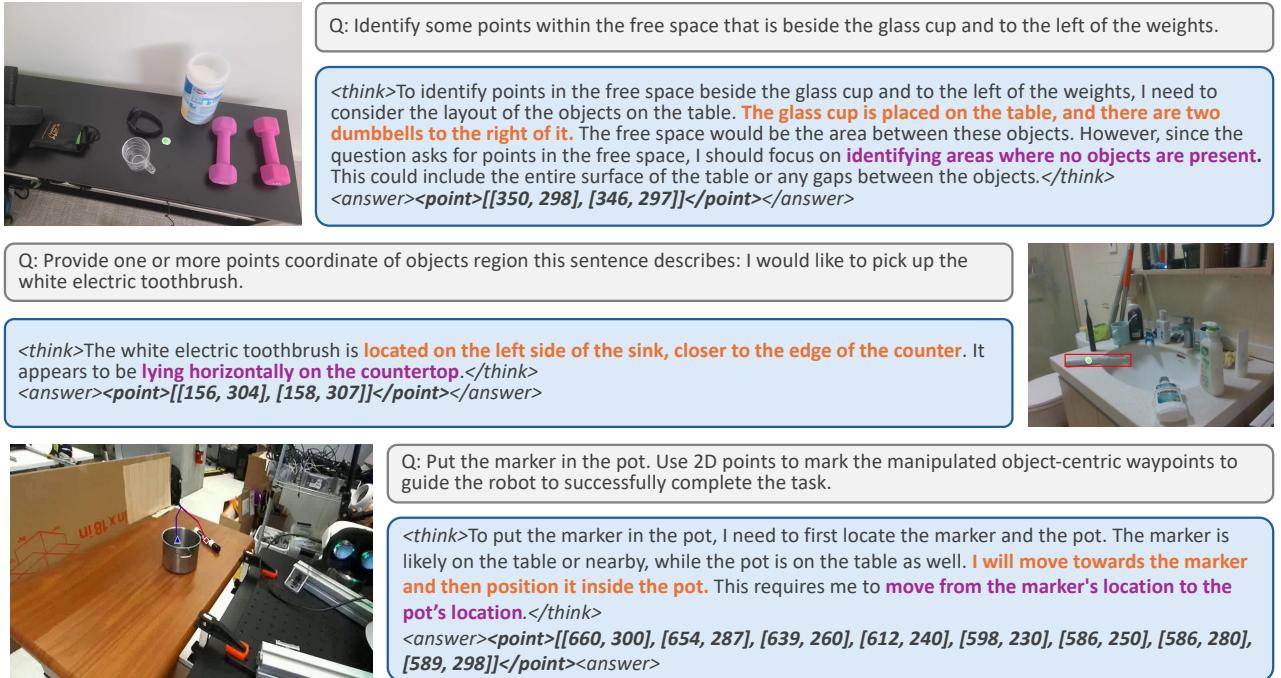


**Figure 6:** The process of Embodied-R1 performing Task 6 under different visual disturbances.

using VTG tasks in entirely unseen scenarios (simulation, novel embodiment, and hand-drawn sketches). We were pleasantly surprised to find that Embodied-R1 can generalize to vastly different and previously unseen environments. Embodied-R1 was trained solely on real-world data, yet it still made accurate predictions on LIBERO Liu et al. (2023) and Maniskill Tao et al. (2024) simulations, which may suggest a promising direction for sim-to-real transfer. Also, Embodied-R1 adapts to different embodiments, demonstrating the advantage of object-centric visual traces—an embodiment-agnostic representation. The right figure presents a particularly interesting case: even when using a real screwdriver and a hand-drawn box, the predicted visual trajectory remains accurate.

### 3.5. Ablations

**Performance Comparison between SFT and RL:** We further investigate the differences between SFT and RL, and analyze whether the performance improvement brought by RL stems from the reasoning process or the reward-driven paradigm. To this end, we selected two benchmarks for evaluating RRG capabilities and trained four models corresponding to the combinations of (SFT, RL)  $\times$  (Think, Not Think). RL w/ Think corresponds to the method used in this paper; however, for a fair comparison, we did not use mixed training with all datasets. In RL w/o Think, we modified the prompt to “*Do not think, just output the answer*” to force the model to output answers directly. SFT w/o Think uses standard question-answer pair data, while in SFT w/ Think, we first designed manual templates and examples, then used GPT-4o to complete the chain-of-thought section. The relevant results are presented in Table 8. We observed that the full RL model (RL w/ Think mechanism) achieved the highest scores on both benchmarks, demonstrating the effectiveness of combining reinforcement learning with the reasoning mechanism, while directly outputting answers leads



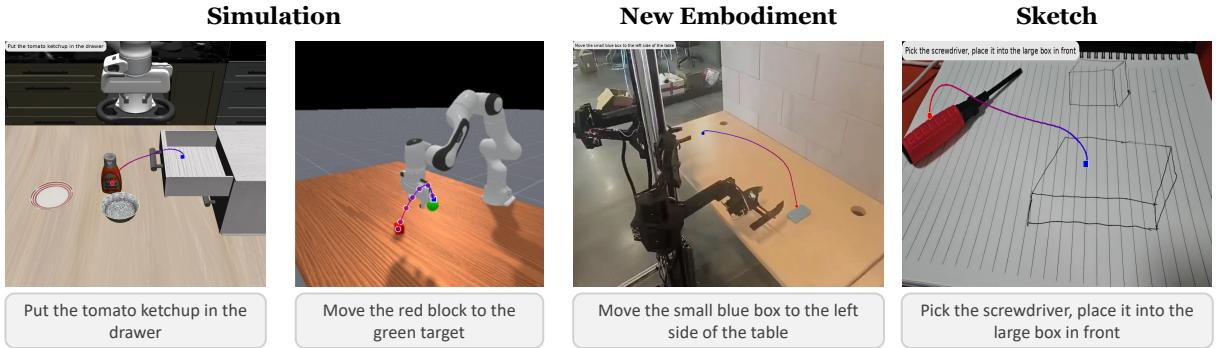
**Figure 7: Case Analysis:** Embodied-R1 possesses embodied reasoning capabilities. It can progressively locate relevant objects and infer spatial relationships according to task instructions, and ultimately provide coordinates through pointing based on embodied scene analysis.

to some performance drop. Models utilizing RL consistently outperformed those without (SFT), indicating that RL plays a key role in OOD generalization. Incorporating COT into supervised learning did not yield significant performance improvements and even resulted in decreased performance on VABench-Point. We believe this may be because inappropriate reasoning processes hinder the learning process, making the learning objectives more complex.

**Advantage of Mixed Training:** We performed multi-task joint training of the various abilities required for embodied pointing and reasoning in the second stage. The advantage of this approach is that all abilities can share general knowledge of point coordinates and semantic space alignment during training, thereby achieving better performance. To validate this idea, we conducted multiple sets of experiments comparing the performance of mixed training and unmixed training. In the unmixed training setting, only the data corresponding to the benchmark ability is used; for example, training for one epoch only on the HandAL dataset and then testing on the Part-Afford dataset. As shown in Table 9, joint training consistently improves the success rate across multiple tasks compared to unmixed training. We believe that mixed training enables knowledge sharing among multiple abilities, enhances semantic understanding of spatial information, and thus leads to better generalization.

## 4. Related Work

**Embodied Reasoning in Robotic Manipulation** Enhancing the reasoning capabilities of LLM has become an important research direction Chu et al. (2025). State-of-the-art models OpenAI et al. (2024), Guo et al. (2025), Team et al. (2025) have demonstrated outstanding performance on complex reasoning tasks, including mathematics Yu et al. (2025), STEM Team et al. (2025), and code generation Jimenez et al. (2024).



**Figure 8: Embodied-R1 exhibits strong generalization capabilities.** We specifically designed tests using VTG tasks in entirely unseen scenarios (simulation, novel embodiment, and hand-drawn sketches), where the model must reason about objects in the images based on common knowledge: (left) LIBERO and ManiSkill simulator tasks, (middle) AhaRobot dual-arm robot tasks, and (right) human-drawn sketches.

**Table 8: Performance Comparison between SFT and RL on RRG benchmarks.**

Model	RL	Think	Where2Place	VABench-P
RL w/ Think	✓	✓	<b>65.50</b>	<b>65.39</b>
RL w/o Think	✓	✗	63.00	60.50
SFT w/ Think	✗	✓	41.25	47.67
SFT w/o Think	✗	✗	36.85	50.46

	Mixed Training	Unmixed Training
Part-Afford	<b>56.63</b>	51.25
Where2Place	<b>69.50</b>	65.50
VABench-P	<b>66.00</b>	65.39

Recent studies in the field of embodied AI have also explored reasoning-driven robotic manipulation. Some works based on SFT involve various forms of templated CoT, such as language prompts Zawalski et al. (2024), Clark et al. (2025), visual subgoals Zhao et al. (2025), depth map Lee et al. (2025), and spatial relation graphs Yuan et al. (2025) to guide action execution. More recent efforts have begun to incorporate RFT for VLA, though these are generally limited to robotic manipulation in simulation environments Lu et al. (2025), Liu et al. (2025a) or online learning in real-world Chen et al. (2025). Besides, ThinkAct Huang et al. (2025) introduces latent visual planning with action-aligned RL. In this paper, we propose an embodied reasoning VLM tailored for general robotic manipulation. Compared to SFT-based approaches, we do not constrain the reasoning within fixed templates; instead, we stimulate free-form reasoning by integrating pointing with RL. In addition, Embodied-R1 utilizes pointing to precisely anchor the reasoning within the scene, directly guiding manipulation rather than relying on indirect instructions through language or sub-goal image.

**Spatial Reasoning with VLMs** Developing spatial intelligence Yang et al. (2024), Song et al. (2024), Du et al. (2024), Zhou et al. (2025), Liao et al. (2024), Ray et al. (2024) in open-world environments is essential for enabling generalizable embodied AI. Robots must grasp spatial concepts and object relationships to perform precise manipulation Yuan et al. (2024b) and navigation Song et al. (2024), Hong et al. (2023), Li et al. (2024a). Recent research has focused on enhancing the spatial reasoning abilities of VLMs through methods such as SFT with specially customized datasets Du et al. (2024), Ray et al. (2024) and embedding spatial knowledge into training data Cai et al. (2024). These methods address tasks such as recognizing spatial relationships Fu et al. (2024), distance Chen et al. (2024), and counting Cai et al. (2024). SpatialRGPT Cheng et al. (2024) improves spatial cognition by more accurately modeling spatial relationships, while Yuan et al. (2025) and Liu et al. (2025b) enhance the spatial reasoning ability of VLMs by spatial chain-of-thought for

step-by-step reasoning. SAT Ray et al. (2024) uses simulation engines and 3D assets to generate complex, configurable spatial reasoning scenarios. Notably, Embodied-R1 employs RL to stimulate model reasoning, achieving stronger OOD generalization compared to the SFT approach.

**Visual Auxiliary Signals for Robotic Manipulation** Utilizing visual auxiliary cues Bharadhwaj et al. (2024), Wen et al. (2023), Xu et al. (2024), Zheng et al. (2024), Yuan et al. (2024a) is a promising approach to enhance robotic manipulation performance. Previous studies have explored various auxiliary signals, such as keypoints Yuan et al. (2024b, 2025), affordance map Huang et al. (2024a, 2023), Li et al. (2024e), bounding boxes Liu et al. (2024a), Huang et al. (2024b), optical flow Xu et al. (2024), Wen et al. (2023), and visual trajectories Yuan et al. (2025), Ji et al. (2025), Li et al. (2025), Gu et al. (2023). These structured and spatially grounded annotations serve as a bridge between visual perception and physical interaction. This approach abstracts out embodiment-specific control mechanisms, thereby enhancing cross-embodiment generalization. In contrast to previous methods that generate a single type of visual signal, we propose a unified “pointing” definition to express diverse and multi-granular manipulation capabilities, and adopt an RL paradigm to improve zero-shot generalization in novel environments.

## 5. Conclusion

We propose Embodied-R1, a powerful embodied reasoning VLM that effectively addresses the “seeing-to-doing” gap in robotic manipulation by enhancing spatial reasoning and embodied pointing abilities. We first constructed a large-scale, meticulously designed dataset and trained Embodied-R1 using a two-stage RFT paradigm, incorporating a multi-task mixed reward design. Embodied-R1 possesses an understanding of the physical world and, through pointing, realizes a suite of capabilities including grounding, spatial region reference, object affordance marking, and visual trace generation, which are further applied to downstream robotic manipulation tasks. Embodied-R1 achieves state-of-the-art results on multiple benchmark tests and demonstrates robust zero-shot generalization in robotic manipulation tasks, offering a promising direction for the development of more capable embodied AI. We discuss detailed limitations in Appendix E.

## 6. Acknowledgment

We would like to thank Zhongwen Xu, Hongyao Tang, Liang Wang, Shuyang Gu, and Chen Li for their participation in the discussions of this paper and for providing valuable insights. In addition, we would especially like to thank Yiyang Huang for the constructive suggestions on improving the figures in the manuscript.

## References

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, 2025a. URL <https://arxiv.org/abs/2502.13923>.

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, 2025b.

Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point

tracks from internet videos enables generalizable robot manipulation. [arXiv preprint arXiv:2405.01527](#), 2024.

Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. [arXiv preprint arXiv:2406.13642](#), 2024.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In [Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition](#), pages 14455–14465, 2024.

Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. Conrft: A reinforced fine-tuning method for vla models via consistency policy. [arXiv preprint arXiv:2502.05450](#), 2025.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision language model. [arXiv preprint arXiv:2406.01584](#), 2024.

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.

Jaden Clark, Suvir Mirchandani, Dorsa Sadigh, and Suneel Belkhale. Action-free reasoning for policy generalization. [arXiv preprint arXiv:2502.03729](#), 2025.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforet, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. [Advances in Neural Information Processing Systems](#), 36:35799–35813, 2023.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. [arXiv preprint arXiv:2409.17146](#), 2024.

Mengfei Du, Biniao Wu, Zejun Li, Xuanjing Huang, and Zhongyu Wei. Embspatial-bench: Benchmarking spatial understanding for embodied tasks with large vision-language models. [arXiv preprint arXiv:2406.05756](#), 2024.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In [European Conference on Computer Vision](#), pages 148–166. Springer, 2024.

Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. [arXiv preprint arXiv:2311.01977](#), 2023.

Andrew Guo, Bowen Wen, Jianhe Yuan, Jonathan Tremblay, Stephen Tyree, Jeffrey Smith, and Stan Birchfield. HANDAL: A dataset of real-world manipulable object categories with pose annotations, affordances, and reconstructions. In [IROS](#), 2023.

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.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *arXiv preprint arXiv:2507.16815*, 2025.

Siyuan Huang, Haonan Chang, Yuhang Liu, Yimeng Zhu, Hao Dong, Peng Gao, Abdeslam Boularias, and Hongsheng Li. A3vlm: Actionable articulation-aware vision language model. *arXiv preprint arXiv:2406.07549*, 2024a.

Siyuan Huang, Iaroslav Ponomarenko, Zhengkai Jiang, Xiaoqi Li, Xiaobin Hu, Peng Gao, Hongsheng Li, and Hao Dong. Manipvqa: Injecting robotic affordance and physically grounded information into multi-modal large language models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7580–7587. IEEE, 2024b.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024c.

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.

Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, Xinda Xue, Qinghang Su, Huaihai Lyu, Xiaolong Zheng, Jiaming Liu, Zhongyuan Wang, and Shanghang Zhang. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. *arXiv preprint arXiv:2502.21257*, 2025.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues?, 2024. URL <https://arxiv.org/abs/2310.06770>.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? investigating their struggle with spatial reasoning, 2023. URL <https://arxiv.org/abs/2310.19785>.

Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1086. URL <https://aclanthology.org/D14-1086>.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

- Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In Proceedings of the IEEE/CVF international conference on computer vision, pages 4015–4026, 2023.
- Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. Molmoact: Action reasoning models that can reason in space. arXiv preprint arXiv:2508.07917, 2025.
- Chengzu Li, Caiqi Zhang, Han Zhou, Nigel Collier, Anna Korhonen, and Ivan Vulić. Topviewrs: Vision-language models as top-view spatial reasoners. arXiv preprint arXiv:2406.02537, 2024a.
- Xiang Li, Cristina Mata, Jongwoo Park, Kumara Kahatapitiya, Yoo Sung Jang, Jinghuan Shang, Kanchana Ranasinghe, Ryan Burgert, Mu Cai, Yong Jae Lee, et al. Llara: Supercharging robot learning data for vision-language policy. arXiv preprint arXiv:2406.20095, 2024b.
- Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. arXiv preprint arXiv:2412.14058, 2024c.
- Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. arXiv preprint arXiv:2405.05941, 2024d.
- Yi Li, Yuquan Deng, Jesse Zhang, Joel Jang, Marius Memmel, Raymond Yu, Caelan Reed Garrett, Fabio Ramos, Dieter Fox, Anqi Li, et al. Hamster: Hierarchical action models for open-world robot manipulation. arXiv preprint arXiv:2502.05485, 2025.
- Yicong Li, Na Zhao, Junbin Xiao, Chun Feng, Xiang Wang, and Tat-seng Chua. Laso: Language-guided affordance segmentation on 3d object. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14251–14260, 2024e.
- Yuan-Hong Liao, Rafid Mahmood, Sanja Fidler, and David Acuna. Reasoning paths with reference objects elicit quantitative spatial reasoning in large vision-language models. arXiv preprint arXiv:2409.09788, 2024.
- Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. arXiv preprint arXiv:2410.18647, 2024.
- Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. Advances in Neural Information Processing Systems, 36: 44776–44791, 2023.
- Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-world robotic manipulation through mark-based visual prompting, 2024a. URL <https://arxiv.org/abs/2403.03174>.
- Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024b. URL <https://llava-vl.github.io/blog/2024-01-30-llava-next/>.
- Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. What can rl bring to vla generalization? an empirical study. arXiv preprint arXiv:2505.19789, 2025a.

Yuecheng Liu, Dafeng Chi, Shiguang Wu, Zhanhuang Zhang, Yaochen Hu, Lingfeng Zhang, Yingxue Zhang, Shuang Wu, Tongtong Cao, Guowei Huang, et al. Spatialcot: Advancing spatial reasoning through coordinate alignment and chain-of-thought for embodied task planning. [arXiv preprint arXiv:2501.10074](#), 2025b.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. [arXiv preprint arXiv:2505.18719](#), 2025.

Yuhao Lu, Yixuan Fan, Beixing Deng, Fangfu Liu, Yali Li, and Shengjin Wang. Vi-grasp: a 6-dof interactive grasp policy for language-oriented objects in cluttered indoor scenes. In [2023 IEEE/RSJ International Conference on Intelligent Robots and Systems \(IROS\)](#), pages 976–983. IEEE, 2023.

Austin Myers, Ching L Teo, Cornelia Fermüller, and Yiannis Aloimonos. Affordance detection of tool parts from geometric features. In [2015 IEEE international conference on robotics and automation \(ICRA\)](#), pages 1374–1381. IEEE, 2015.

Soroush Nasiriany, Sean Kirmani, Tianli Ding, Laura Smith, Yuke Zhu, Danny Driess, Dorsa Sadigh, and Ted Xiao. Rt-affordance: Affordances are versatile intermediate representations for robot manipulation. [arXiv preprint arXiv:2411.02704](#), 2024.

Jake O’Neill, Abraham Arthurs, Fábio Avila Belbute-Peres, Julian Balaguer, Sarah Bechtle, Gemma Bidoia, Kyle Burden, Erwin Chang, Sheila Chen, Todor Davchev, et al. Open x-embodiment: Robotic learning datasets and rt-x models. [arXiv preprint arXiv:2310.08864](#), 2023.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O’Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singh, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese,

Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL <https://arxiv.org/abs/2412.16720>.

Zekun Qi, Wenyao Zhang, Yufei Ding, Runpei Dong, Xinqiang Yu, Jingwen Li, Lingyun Xu, Baoyu Li, Xialin He, Guofan Fan, et al. Sofar: Language-grounded orientation bridges spatial reasoning and object manipulation. [arXiv preprint arXiv:2502.13143](#), 2025.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. [arXiv preprint arXiv:2501.15830](#), 2025.

Arijit Ray, Jiafei Duan, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A Plummer, Ranjay Krishna, Kuo-Hao Zeng, et al. Sat: Spatial aptitude training for multimodal language models. [arXiv preprint arXiv:2412.07755](#), 2024.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. [arXiv preprint arXiv:2401.14159](#), 2024.

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.

Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. [arXiv preprint arXiv:2411.16537](#), 2024.

Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, et al. Curobo: Parallelized collision-free robot motion generation. In [2023 IEEE International Conference on Robotics and Automation \(ICRA\)](#), pages 8112–8119. IEEE, 2023.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. [arXiv preprint arXiv:2410.00425](#), 2024.

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

Octo Team, RT-X Team, Anthony Brohan, Noah Brown, Lauren Chen, Michael Cheng, Krzysztof Choromanski, Eamonn Cullina, Gabe Dalal, Chelsea Fu, Florian Golemo, et al. Octo: An open-source generalist robot policy. [arXiv preprint arXiv:2403.10164](#), 2024.

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.

Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In [Conference on Robot Learning](#), pages 1723–1736. PMLR, 2023.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhui Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning, 2025a. URL <https://arxiv.org/abs/2504.08837>.

Weiyun Wang, Yiming Ren, Haowen Luo, Tiantong Li, Chenxiang Yan, Zhe Chen, Wenhai Wang, Qingyun Li, Lewei Lu, Xizhou Zhu, et al. The all-seeing project v2: Towards general relation comprehension of the open world. In [European Conference on Computer Vision](#), pages 471–490. Springer, 2025b.

Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. [arXiv preprint arXiv:2401.00025](#), 2023.

Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. [arXiv preprint arXiv:1711.00199](#), 2017.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In [Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition](#), pages 4818–4829, 2024.

Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. [arXiv preprint arXiv:2407.15208](#), 2024.

Rongtao Xu, Jian Zhang, Minghao Guo, Youpeng Wen, Haoting Yang, Min Lin, Jianzheng Huang, Zhe Li, Kaidong Zhang, Liqiong Wang, et al. A0: An affordance-aware hierarchical model for general robotic manipulation. [arXiv preprint arXiv:2504.12636](#), 2025.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. [arXiv preprint arXiv:2412.14171](#), 2024.

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.

Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. [arXiv preprint arXiv:2401.11439](#), 2024a.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. [arXiv preprint arXiv:2406.10721](https://arxiv.org/abs/2406.10721), 2024b.

Yifu Yuan, Haiqin Cui, Yibin Chen, Zibin Dong, Fei Ni, Longxin Kou, Jinyi Liu, Pengyi Li, Yan Zheng, and Jianye Hao. From seeing to doing: Bridging reasoning and decision for robotic manipulation, 2025. URL <https://arxiv.org/abs/2505.08548>.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. [arXiv preprint arXiv:2407.08693](https://arxiv.org/abs/2407.08693), 2024.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In [Proceedings of the Computer Vision and Pattern Recognition Conference](https://openaccess.thecvf.com/content/CVPR2025/papers/Zhao_Cot-vla:_Visual_chain-of-thought_reasoning_for_vision-language-action_models_CVPR_2025_paper.pdf), pages 1702–1713, 2025.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. [arXiv preprint arXiv:2412.10345](https://arxiv.org/abs/2412.10345), 2024.

Yuchen Zhou, Jiayu Tang, Xiaoyan Xiao, Yueyao Lin, Linkai Liu, Zipeng Guo, Hao Fei, Xiaobo Xia, and Chao Gou. Where, what, why: Towards explainable driver attention prediction. [arXiv preprint arXiv:2506.23088](https://arxiv.org/abs/2506.23088), 2025.

## Appendix

### A. Automatic Data Generation Pipeline

In this section, we provide additional explanations regarding the generation of certain datasets. The generation processes of both the RRG and VTG datasets are improved based on [Yuan et al. \(2025\)](#).

**3D RRG Data Generation using Isaac Gym Simulation** The dataset comprises 10,028 tasks, each situated in a tabletop scene containing multiple objects. The input for each task consists of processed RGB and depth images, accompanied by a language instruction describing the desired target position of an object within the scene (e.g., "place the cup between the book and the spoon"). Based on these instructions, the model is required to output the 3D position of the target object in the camera coordinate system, specified by its pixel coordinates ( $X, Y$ ) and a depth value  $D$  in millimeters. The  $D$  value is obtained either through monocular depth estimation or by reasoning from the scene geometry. The generation process of our dataset is informed by the methodology of Open6Dor [Qi et al. \(2025\)](#). The object set utilized contains over 200 items spanning more than 70 distinct categories, originally sourced from the YCB [Xiang et al. \(2017\)](#) and Objaverse-XL [Deitke et al. \(2023\)](#) datasets. These objects underwent a rigorous selection process to ensure their physical integrity and semantic suitability for tabletop arrangements. All selected objects were subsequently scale-normalized and uniformly represented using a consistent mesh format.

In terms of scene configuration, between two and five objects were randomly selected from the object set and placed on a tabletop with random initial poses. For each configured scene, we rendered both RGB and depth images. The depth values represent ground truth measurements, with the scene's depth range spanning from 600 mm to 1700 mm. For subsequent processing convenience, the depth images were normalized to an 8-bit grayscale format (0-255). We filtered out low-quality scenes, such as those exhibiting implausible object placements or severe occlusions. To augment the dataset's quality and volume, we expanded a subset of high-quality, filtered data by algorithmically generating variations in task descriptions, such as substituting directional prepositions or altering object relations. Then, the task instructions are formulated in two primary types: basic directional commands (e.g., left, right, top, behind, front) and relational commands (e.g., "between," "center of"). All instructions adhere to a standardized template, for instance, "Place object A in front of object B," where A and B are objects present in the scene. During training, the model receives RGB and depth image inputs and is required to output the target's coordinates ( $X, Y$ ) and depth value  $D$ . The simulated environment then executes and evaluates the predicted position, giving positive rewards for correct predictions.

**VTG Dataset Generation Pipeline** For each video sequence, we first process the initial frame to perform instance segmentation on the manipulated object, thereby obtaining its pixel-wise mask. Instead of relying on a single tracking point, which is susceptible to tracking failure from occlusion or rapid motion, we sample a set of three distinct query points from within this mask. This multi-point initialization serves as a redundancy measure, significantly enhancing the robustness of the tracking process. These points are strategically chosen to represent the object's initial state before they are passed, along with the full video sequence, to the tracking model for trajectory prediction. The core of the trajectory generation is handled by the CoTracker model [Karaev et al. \(2024\)](#), which takes the initialized query points and video as input. The model concurrently tracks each point throughout the sequence, yielding a set of three candidate trajectories. As these trajectories may vary in quality and completeness due to transient tracking errors, a selection criterion is required to identify the single most representative path. We employ a path length heuristic for this purpose, calculating the total Euclidean distance of each trajectory. The trajectory with the longest path is selected as the definitive motion path for the object. The rationale behind this criterion is that the longest

trajectory is most likely to have successfully tracked the object through its entire course of motion without premature termination. Following the selection of the representative trajectory, a two-stage refinement process is applied to produce the final visual trace. The raw trajectory, composed of discrete frame-by-frame coordinates, is first smoothed using cubic spline interpolation. This step transforms the discrete points into a continuous, smooth curve, effectively filtering out high-frequency noise and jitter inherent in the tracking process. From this smoothed curve, we then uniformly sample eight equidistant points. This final set of eight points constitutes the visual trace—a structured, discretized representation of the object’s motion, suitable for downstream analysis and model consumption.

We found that this process inevitably faces prediction errors from pre-trained visual models, such as incorrect object grounding or incomplete motion trajectory tracking. To mitigate these issues, we employ stringent rule-based filtering methods using hyperparameters such as size thresholds and trajectory length thresholds. Before annotating each dataset, we iteratively adjust these rules and conduct manual sampling inspections. Only when the filtering rules are robust enough do we apply them to the full data generation pipeline, ensuring the high quality of the data.

## B. Implementation Details of Embodied-R1

**Training Hyperparameters:** We conducted model training on eight NVIDIA A100 40G GPUs. The first phase was trained for 2 epochs, and the second phase for 1 epoch, with each phase taking approximately 48 hours. The backbone model used is Qwen2.5-VL-3B-Instruct<sup>1</sup>, with a maximum context length of 4096 and a maximum response length of 2048. The optimizer selected is AdamW, with a learning rate of 1e-6 and a weight decay coefficient of 1e-2. In Embodied-R1, we performed reinforcement learning training based on GRPO Shao et al. (2024), set the number of samples to 8, and introduced a KL penalty (coefficient 1e-2), with a global batch size of 128 for each step. For all experiments, we focus on comparing SFT models trained with the same batch size and data, which we refer to as Embodied-SFT. As for Embodied-SFT, we used exactly the same data but trained with a supervised learning loss, kept the batch size at 128, and trained for 3 epochs.

**Reward Hyperparameters:** To enable stable multi-task training, we constrain the total reward for each task to the range of 0 to 1 and define the total reward  $\mathcal{R}$  as a weighted combination  $\mathcal{R} = \sum_{r \in \mathcal{F}} w_r \cdot r$ . Each task utilizes a different combination of reward terms. Here, we provide the hyperparameters for each task in the Table 10. We would like to add two clarifying points: First, if the task output fails to meet the required parsing format, subsequent analysis cannot proceed successfully, so the reward is set directly to 0. Second, for the VTG task, we introduced an additional constraint on the format: the generated visual trace must consist of *exactly 8 points*. In practice, we found that without this constraint, the model in the VTG task was prone to reward hacking behavior. It would tend to output only two points to form a straight line, which easily yields a high reward and prematurely terminates exploration. By enforcing the 8-point constraint, we compel the model to perform more complex curve interpolation, thereby improving the performance of visual trace generation. We provide a detailed comparison in the Appendix D.

---

<sup>1</sup><https://huggingface.co/Qwen/Qwen2.5-VL-3B-Instruct>

**Table 10:** Detailed Reward Functions for Each Task

Task	Reward Function $\mathcal{R}$
General QA	$\mathcal{R} = 0.1 r_{\text{format}} + 0.9 r_{\text{acc}}$
Spatial QA	$\mathcal{R} = 0.1 r_{\text{format}} + 0.9 r_{\text{acc}}$
REG	$\mathcal{R} = 0.1 r_{\text{format}} + 0.9 r_{\text{mask}}$
RRG	$\mathcal{R} = 0.1 r_{\text{format}} + 0.6 r_{\text{mask}} + 0.3 r_{\text{dis}}$
3D RRG	$\mathcal{R} = 0.1 r_{\text{format}} + 0.9 r_{\text{env}}$
OAG	$\mathcal{R} = 0.1 r_{\text{format}} + 0.8 r_{\text{mask}} + 0.1 r_{\text{dis}}$
VTG	$\mathcal{R} = 0.1 r_{\text{format}} + 0.9 r_{\text{trace}}$

## C. Embodied-R1 Prompts for Each Task

### Referring Expression Grounding (REG) Prompt

Provide one or more points coordinate of object region this sentence describes: *Your Instruction*. The results are presented in a format `<point>[[x1,y1], [x2,y2], ...]</point>`. You FIRST think about the reasoning process as an internal monologue and then provide the final answer. The reasoning process and answer are enclosed within `<think> </think>` and `<answer> </answer>` tags. The answer consists only of several coordinate points, with the overall format being: `<think> reasoning process here </think><answer><point>[[x1, y1], [x2, y2], ...]</point></answer>`

### Region Referring Grounding (RRG) Prompt

You are currently a robot performing robotic manipulation tasks. The task instruction is: *Your Instruction*. Use 2D points to mark the target location where the object you need to manipulate in the task should ultimately be moved. You FIRST think about the reasoning process as an internal monologue and then provide the final answer. The reasoning process and answer are enclosed within `<think> </think>` and `<answer> </answer>` tags. The answer consists only of several coordinate points, with the overall format being: `<think> reasoning process here </think><answer><point>[[x1, y1], [x2, y2], ...]</point></answer>`.

### Object Functional Grounding (OFG) Prompt

Please provide the 2D points coordinates of the region this sentence describes: *Your Instruction*. The results are presented in a format `<point>[[x1,y1], [x2,y2], ...]</point>`. You FIRST think about the reasoning process as an internal monologue and then provide the final answer. The reasoning process and answer are enclosed within `<think> </think>` and `<answer> </answer>` tags. The answer consists only of several coordinate points, with the overall format being: `<think> reasoning process here </think><answer><point>[[x1, y1], [x2, y2], ...]</point></answer>`.

### Visual Trace Generation (VTG) Prompt

You are currently a robot performing robotic manipulation tasks. The task instruction is: *Your Instruction*. Use 2D points to mark the manipulated object-centric waypoints to guide the robot to successfully complete the task. You must provide the points in the order of the trajectory, and the number of points must be 8. You FIRST think about the reasoning process as an internal monologue and then provide the final answer. The reasoning process and answer are enclosed within `<think>` `</think>` and `<answer>` `</answer>` tags. The answer consists only of several coordinate points, with the overall format being: `<think>` reasoning process here `</think>` `<answer>` `<point>` `[[x1, y1], [x2, y2], ..., [x8, y8]]` `</point>` `</answer>`.

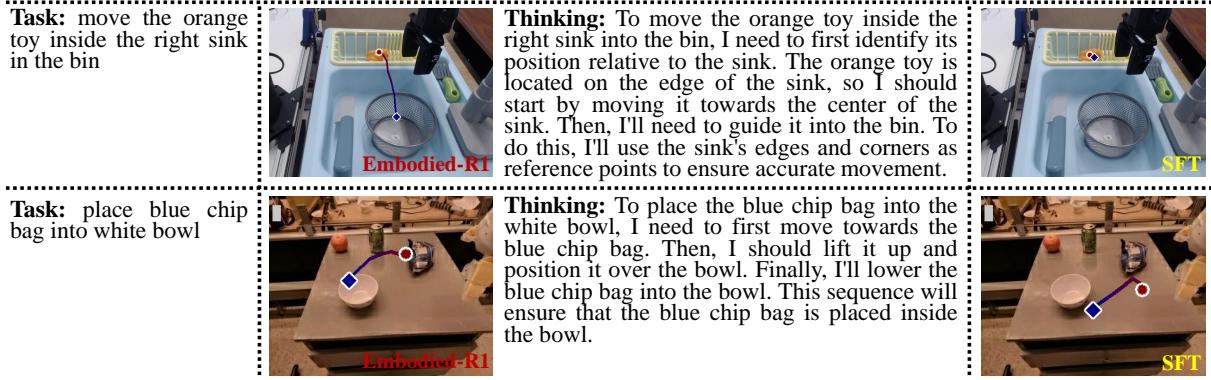
## D. Additional Experiments

**The Phenomenon of Reward Hacking in VTG Tasks** We carefully designed the reward function so that the reward for each task is only related to the final goal, thereby avoiding reward hacking caused by intermediate rewards. However, we found that in the VTG task, designing the reward solely based on the distance between the predicted trajectory and the target trajectory can still result in reward hacking. The model quickly learned that in visual trajectory generation tasks, accurately predicting the starting and ending points is both crucial and relatively easy, leading it to converge rapidly to outputs with only these two points while ignoring the generation of intermediate trajectory points. We observed that by forcing the model to output multiple points and applying reward constraints for format reward, it becomes possible to generate complete visual traces. Therefore, we explicitly require the model to output a visual trace with eight points; otherwise, all rewards are set to zero (since the format reward is not satisfied, subsequent analysis will not be performed). As shown in Table 11, we present the results on VABench-V, demonstrating the performance differences with and without the trajectory point number constraint. After modifying the reward function, the model is better able to fit the visual trace, achieving lower RMSE and a higher GPT score.

Table 11: Comparasion of w/ and w/o Point Num Reward. **Bolds** are better.

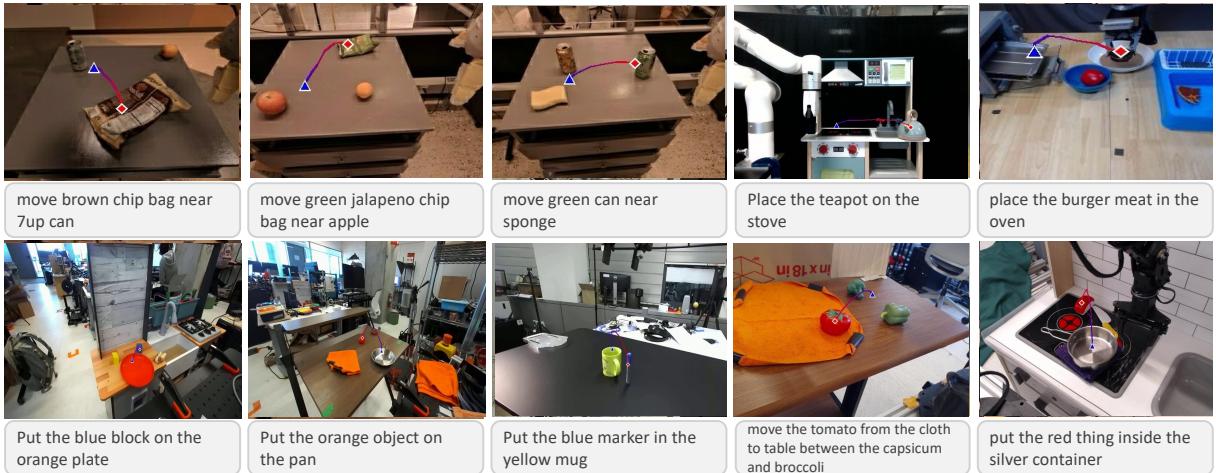
VABench-VisualTrace	RMSE↓	MAE↓	GPT Score↑
w/ Point Num Constraint	<b>77.83</b>	<b>44.97</b>	<b>7.27</b>
w/o Point Num Constraint	105.2	59.7	5.57

**Qualitative Comparison: Embodied-R1 vs. SFT** To provide a deeper insight into the performance gains of our model, we conducted a qualitative analysis comparing Embodied-R1 with the SFT baseline. As illustrated in Fig. 9, the difference in capabilities is stark. In the first task, “move the orange toy inside the right sink in the bin,” Embodied-R1 first articulates a clear plan: it identifies the toy’s initial position, determines the need to move it towards the sink’s center, and then guides it into the bin. This logical reasoning translates into a precise and successful visual trajectory. The SFT baseline, however, produces an erroneous trajectory that fails to place the toy correctly. Similarly, for the second task, “place blue chip bag into white bowl,” Embodied-R1 correctly reasons that the task requires moving to the bag, lifting it, positioning it over the bowl, and then lowering it. This step-by-step plan underpins the generated trajectory, which successfully completes the task. The SFT model again fails, generating a flawed path that does not accomplish the goal. These examples strongly suggest that the explicit chain-of-thought reasoning in Embodied-R1 is critical for understanding complex instructions and planning accurate, multi-stage visual trajectories, a capability that the standard SFT approach evidently lacks.



**Figure 9: Qualitative comparison of Embodied-R1 and the SFT baseline.** Our model Embodied-R1, leverages chain-of-thought reasoning (middle column) to generate a successful visual trace (left column). In contrast, the SFT baseline, which lacks an explicit reasoning process, produces incorrect trajectories (right column) for the same tasks.

**More Visualization of VTG Task** We provide additional visualization examples of Embodied-R1’s predicted visual traces in Fig. 10. It can be seen that Embodied-R1 achieves accurate visual trajectory prediction across various scenarios.



**Figure 10: Visualizing Embodied-R1’s Prediction on VTG Tasks across Various Scenarios**

## E. Limitation and Future Work

Despite the state-of-the-art performance achieved by Embodied-R1 across numerous benchmarks and real-world tasks, this work has several limitations that present avenues for future research.

- **Untapped Potential of Integrating with Learning-based Policies.** The current work primarily combines the perception and reasoning capabilities of Embodied-R1 with a classical motion planner. A natural and promising future direction is to integrate the model as a high-level front-end for a learning-based conditional policy. Such integration could potentially enhance execution efficiency and reactivity, especially in dynamic environments. Many existing studies Bharadhwaj et al. (2024), Gu et al. (2023), Xu et al.

(2024) have explored using visual traces as conditional inputs for policies, which improves the learning efficiency and performance of the policies; however, they only focus on designing downstream policies and do not provide a general visual trajectory predictor. Therefore, this integration approach is promising; however, it remains unexplored in the current study.

- **Untapped Potential in Long-horizon Tasks.** The current framework is designed to process single-step instructions and does not natively include a mechanism for decomposing long-horizon commands (e.g., "prepare a meal"). However, this could be addressed through a modular, hierarchical approach. Embodied-R1 is well-suited to act as a robust execution module for individual sub-tasks. A high-level embodied planner could first decompose a complex instruction into a sequence of simpler steps, which would then be passed to Embodied-R1 for execution, enabling the system to tackle complex, multi-stage problems.
- **Inherent Limitations of the “Pointing” Representation.** While the pointing representation is effective for localization, placement, and trajectory generation, it may be insufficient for the full spectrum of complex robotic manipulation. Tasks requiring precise force control, twisting, wiping, or intricate interactions with deformable objects demand a richer representation than 2D coordinate points. We believe this issue can be mitigated by coupling the high-level “pointing” commands with a learnable downstream policy that can translate these targets into complex, dynamic actions. The design for Embodied-R1 reflects our primary focus on providing a promising solution for zero-shot generalization, for which a simplified, embodiment-agnostic intermediate representation is a key advantage.
- **Preliminary Integration of 3D Information.** The exploration of an RGB-D version of the model is still in its early stages. The paper notes that in tasks with complex spatial relations, the performance of the RGB-D variant can be slightly lower than its 2D counterpart. It is hypothesized that “depth map understanding may be more prone to hallucinations,” indicating that robustly fusing 3D information into the model requires further development.