# RoboRefer: Towards Spatial Referring with Reasoning in Vision-Language Models for Robotics

**Enshen Zhou**[1,3]*, **Jingkun An**[1]*, **Cheng Chi**[3]*‡†, **Yi Han**[1,3], **Shanyu Rong**[2,3], **Chi Zhang**[2], **Pengwei Wang**[3], **Zhongyuan Wang**[3], **Tiejun Huang**[2,3], **Lu Sheng**[1]†, **Shanghang Zhang**[2,3]†

[1]School of Software, Beihang University

[2] State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University [3]Beijing Academy of Artificial Intelligence

zhouenshen@buaa.edu.cn anjingkun02@gmail.com chicheng@baai.ac.cn lsheng@buaa.edu.cn shanghang@pku.edu.cn


Figure 1: Spatial referring in complex 3D environments demands not only precise *single-step spatial understanding* but also *multi-step spatial reasoning* to resolve intricate references step-by-step, thereby enabling efficient control of diverse robots across tasks (*e.g.*, manipulation, navigation).

## Abstract

Spatial referring is a fundamental capability of embodied robots to interact with the 3D physical world. However, even with the powerful pretrained vision language models (VLMs), recent approaches are still not qualified to accurately understand the complex 3D scenes and dynamically reason about the instruction-indicated locations for interaction. To this end, we propose *RoboRefer*, a 3D-aware VLM that can first achieve precise spatial understanding by integrating a disentangled but dedicated depth encoder via supervised fine-tuning (SFT). Moreover, *RoboRefer* advances generalized multi-step spatial reasoning via reinforcement fine-tuning (RFT), with metric-sensitive process reward functions tailored for spatial referring tasks. To support SFT and RFT training, we introduce *RefSpatial*, a large-scale dataset of 20M QA pairs (2× prior), covering 31 spatial relations (*vs.* 15 prior) and supporting complex reasoning processes (up to 5 steps). In addition, we present *RefSpatial-Bench*, a challenging benchmark filling the gap

* Equal contribution † Corresponding author ‡ Project leader

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

in evaluating spatial referring with multi-step reasoning. Experiments show that SFT-trained *RoboRefer* achieves state-of-the-art spatial understanding, with an average success rate of 89.6%. RFT-trained *RoboRefer* further outperforms all other baselines by a large margin, even surpassing Gemini-2.5-Pro by 17.4% in average accuracy on *RefSpatial-Bench*. Notably, *RoboRefer* can be integrated with various control policies to execute long-horizon, dynamic tasks across diverse robots (*e.g.*, UR5, G1 humanoid) in cluttered real-world scenes. See the project page at https://zhoues.github.io/RoboRefer.

## 1 Introduction

Open-world spatial intelligence is crucial for embodied AI, as robots must understand and reason about 3D scenes to interact effectively in complex environments. As one vital topic in this field, *spatial referring*, which bridges spatial intelligence and embodied AI by formalizing how agents interpret and act upon spatially constrained instructions, has received increasing attention. Specifically, given sensor observations (*e.g.*, RGB or RGB-D) and a spatially constrained instruction, the spatial referring task aims to predict a precise point that satisfies complex spatial relations within the instruction. This predicted point can serve various downstream embodied functions as navigation waypoints, manipulation targets, or placement locations, enabling wide robotic applications, as shown in Fig 1.

Spatial referring task comprises two distinct levels of complexity: **(1)** *Single-step spatial understanding*, which forms the foundation of spatial perception by accurately recognizing objects' spatial properties (*e.g.*, position, orientation) and their spatial relations (*e.g.*, distance, direction). This level, where most current research [1–7] concentrates, provides the essential perceptual basis for complex spatial referring. **(2)** *Multi-step spatial reasoning*, which transcends basic understanding through compositional reasoning to resolve complex spatial references sequentially. Despite its importance for sophisticated spatial intelligence, this capability remains underexplored. Thus, this work attempts to address this gap by integrating both levels for comprehensive spatial referring. In Fig 1, one must first identify the plate closest to the observer and locate the desired soy sauce dish, then determine the free space between them, which is increasingly challenging as more spatial constraints are introduced.

Specifically, existing vision-language models (VLMs) [8–11]-based methods mainly attempt to enhance the first level, *i.e.*, *single-step spatial understanding* by integrating 3D inputs. However, they either demand costly 3D reconstruction of multi-view images [12, 13], causing modality gaps , or treat depth as RGB-like inputs [1, 3, 14] via a shared image encoder, risking modality interference and degrading pretrained image encoders, requiring additional co-training data for compensation. In contrast, the second level, *i.e.*, *multi-step spatial referring with reasoning*, remains underexplored due to the scarcity of suitable datasets, limiting current models' capability and preventing exploration of how single-step understanding might support it. Moreover, current VLMs depend heavily on supervised fine-tuning (SFT) for implicit reasoning, risking memorizing answers over explicit reasoning and thereby hindering generalization and accuracy in open-world spatial referring.

In this work, we propose *RoboRefer*, a 3D-aware VLM that not only acquires precise spatial understanding via SFT but also exhibits generalized strong reasoning capabilities for spatial referring via reinforcement fine-tuning (RFT). Specifically, for single-step spatial understanding, *RoboRefer* employs a dedicated depth encoder to enhance precise spatial perception without interfering RGB branch. To enable multi-step spatial reasoning, we design an RFT stage after SFT with explicitly annotated reasoning processes. This stage allows *RoboRefer* to break down complex spatial referring tasks into sequential analytical steps. In each step, *RoboRefer* can leverage the spatial understanding gained in SFT and refine the intermediate reasoning precision with our proposed metric-sensitive process reward functions, thus making more accurate point predictions. To our best knowledge, *RoboRefer* is the first 3D-aware reasoning VLM for multi-step spatial referring with explicit reasoning.

To advance spatial referring, we introduce *RefSpatial*, a large-scale dataset of 2.5M high-quality examples with 20M QA pairs (2× prior [3]). Leveraging diverse data sources from 2D/3D/Simulation, this dataset can teach a general VLM to achieve spatial referring in a bottom-up manner. Specifically, 2D web images provide fundamental spatial concepts and broad depth perception (indoor and outdoor), 3D embodied videos refine fine-grained spatial understanding of indoor scenes for robotics, and simulated data with ground-truth reasoning processes encourage multi-step spatial referring (up

to 5 steps). Notably, *RefSpatial* includes 31 spatial relations, far exceeding 15 found in previous datasets [2, 3], and each sample contains RGB-D data to support depth alignment in SFT stage.

We evaluate our SFT-trained model on existing single-step spatial reasoning benchmarks (*e.g.*, CV-Bench [15], BLINK [16]), achieving SOTA performance with an average success rate of 89.6%. To address the lack of multi-step spatial referring benchmarks, we introduce *RefSpatial-Bench*, comprising 200 real-world images with manually annotated tasks for object location and placement. Over 70% of the samples require multi-step reasoning (up to 5 steps) and are annotated with precise masks. Our model consistently outperforms all baselines on this benchmark, even surpassing Gemini-2.5-Pro by an average of 17.4%. Moreover, in Fig. 1 and Sec. 4.4, *RoboRefer* can execute long-horizon, dynamic tasks in cluttered real-world scenes with various control policies, exhibiting strong generalization across robots (*e.g.*, UR5, G1 humanoid) and tasks (*e.g.*, manipulation, navigation).

Our contributions are summarized as follows: **(1)** We propose *RoboRefer*, a 3D-aware reasoning VLM trained using a sequential SFT-RFT strategy with metric-sensitive process reward functions to achieve spatial referring. **(2)** We construct *RefSpatial*, a well-annotated dataset tailored for spatial referring, facilitating both SFT and RFT training, and introduce *RefSpatial-Bench*, a benchmark that fills the gap in evaluating spatial referring with multi-step reasoning. **(3)** Extensive experiments show that *RoboRefer* generalizes well, surpasses baselines in spatial understanding and referring with reasoning, and efficiently controls diverse robots across tasks in the real world.

## 2 Related work

**Spatial Understanding with VLMs.** Spatial understanding [16–24] focuses on object-centric properties (*e.g.*, position, orientation) and inter-object relations (*e.g.*, distance, direction), while spatial reasoning [25–36] draws higher-level inferences over such information. Recent advances in VLMs [8–11, 37–57] enhance these two abilities via two paradigms: **(1)** tool-based approaches [7, 14, 58–64] that integrate VLMs with vision foundation models [65–78] to extract and reason spatial cues and **(2)** data-driven methods, which fine-tune VLMs using pseudo-3D annotations [1, 6], real-world 3D datasets [2, 3], or simulated data [4, 79]. However, existing datasets lack multi-step reasoning annotations critical for spatial referring tasks, and a benchmark for evaluating such abilities remains unavailable. We thus introduce a new dataset and benchmark specifically tailored for spatial referring.

**Referring with VLMs for Robotics.** Referring, also known as Referring Expression Comprehension (REC) [80–87], leverages unambiguous descriptions to localize a unique region/point in an image, and has seen great progress via VLMs [88–93]. Unlike Phrase Localization [94–96] and Generalized Visual Grounding [97–101], which address ambiguous or multiple referents, REC focuses on one single target—an emphasis crucial for robotics, especially in pick-and-place tasks requiring precise object identification and destination [102–106]. While 2D REC relies on object attributes (*e.g.*, color) and image-plane localization (*e.g.*, top right of the image), real-world scenarios for robotics require 3D spatial reasoning to localize (*e.g.*, "near" *vs.* "far"). Although efforts [107–109] like RoboPoint [5] incorporate basic spatial cues via images to meet such expectations, they often struggle with complex environments and instructions required for spatial referring. Thus, we propose *RoboRefer*, a 3D-aware framework that employs multi-step reasoning to ensure precise spatial referring for robotics.

**Reinforcement Fine-tuning for VLMs.** Reinforcement Fine-tuning (RFT) [110–114] is a post-training strategy that aligns models with human preferences or specific goals via feedback, complementing SFT [115, 116], which adapts pre-trained models using task-oriented data. Recent advances in LLM-based reasoning [114, 117–120] have shifted RL in VLMs toward visual reasoning [121–127], grounding [128–130], segmentation [131] and trajectory prediction [132]. However, most methods rely solely on 2D perception, limiting their ability to handle spatial referring tasks that require 3D spatial reasoning. To address this, we propose a two-stage training strategy: (1) incorporate depth information during SFT to strengthen spatial understanding; (2) RFT stage then leverages intermediate perception outputs powered by SFT to enable multi-step spatial referring with reasoning.

## 3 Method

We first formulate the spatial referring task (Sec. 3.1). Then, we elaborate on *RoboRefer*, including its architecture and training strategies (Sec. 3.2). Finally, we describe the construction of the *RefSpatial* dataset (Sec. 3.3) and necessary training details about *RoboRefer* (Sec. 3.4).

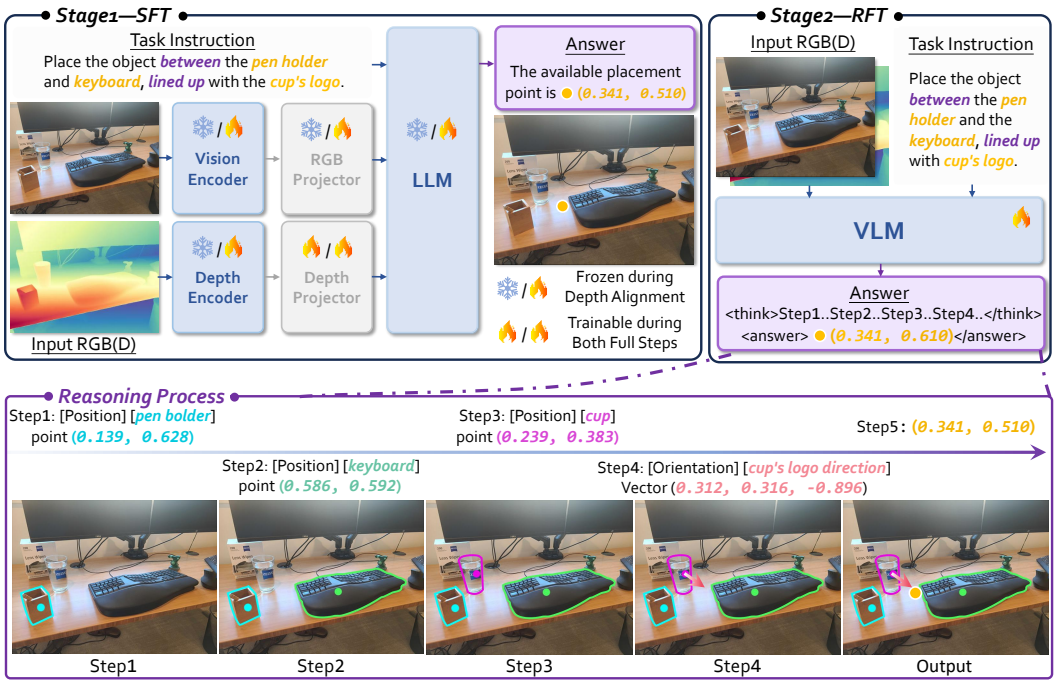



Figure 2: Overview of *RoboRefer*. *RoboRefer* can perform *single-step precise spatial understanding* from RGB(D) inputs with spatially constrained instructions (enabled by the SFT stage introducing depth modality), and *multi-step spatial referring* with explicit reasoning (powered by the reinforcement fine-tuning stage and leveraging spatial understanding within each step learned in SFT).

### 3.1 Problem Formulation

We formulate spatial referring as predicting a single 2D point $(x, y)$ in image space to specify a target location or destination, given visual inputs $\mathcal{O}$ (*e.g.*, RGB or RGB-D) from the sensor and a textual instruction $\mathcal{L}$. This instruction encodes not only single-step spatial properties such as size (*e.g.*, large, small), position (*e.g.*, relative or ordinal location), orientation (*e.g.*, front-facing), and spatial relations (*e.g.*, distance, direction), but also requires multi-step spatial reasoning. For example, "Place the object between the pen holder and keyboard, lined up with the cup's logo." (See Fig. 2) becomes more complex as multiple spatial constraints are combined.

Unlike region-based 2D referring methods [88, 93, 101], *this point-based formulation is more suitable and generalizable for robotics.* Compared to the 2D bbox, point can naturally map to 3D coordinates via depth, providing accurate spatial anchors. By leveraging predicted points for navigation, grasping, or placement, this formulation enables multi-task learning and execution. Moreover, it can accurately localize a visible object part in occlusion scenarios, while 2D bbox often includes irrelevant objects.

### 3.2 *RoboRefer*: A 3D-aware reasoning VLM for spatial referring

**VLM Architecture.** In Fig. 2, *RoboRefer* employs separate RGB and depth encoders to extract features, which are then aligned via projectors with the LLM for VQA or point prediction. As 3D cues are vital for spatial understanding, 2D VLMs pretrained solely on RGB lack accurate 3D perception. Recent methods [1, 3, 14] avoid explicit 3D representations by treating depth as an image-like modality and sharing the RGB encoder, but this causes modality interference, degrading the pretrained encoder and requiring additional RGB co-training to compensate. To address this, we propose a simple yet effective approach: a dedicated depth encoder and projector, initialized from their RGB counterparts. Notably, during joint RGB and RGB-D training, the image encoder remains unaffected by depth input, while the depth encoder is updated independently. This design not only avoids modality interference and preserves general VQA performance without extensive RGB-only co-training, but it also improves spatial understanding through enhanced perception of depth cues (*e.g.*, distance, near–far relations, and perspective-based size variations). See Appx. D.1 for details.

**Supervised Fine-tuning.** We adopt NVILA [38] as our base VLM; however, its 2D-only pretraining limits spatial understanding. To address this, we propose a two-step SFT. **(1)** Depth alignment. In Fig. 2, we first train a depth projector to align the newly introduced depth space with the textual space, leveraging RGB-D annotations of the *RefSpatial* (see Sec. 3.3). In this step, only the depth projector is updated. **(2)** Spatial understanding enhancement. We fine-tune all parameters on the *RefSpatial*, including single-step fine-grained annotations and multi-step reasoning data with explicit reasoning processes, and additional instruction-following datasets [87, 133, 134]. Therefore, the model is jointly optimized on RGB and RGB-D inputs, with separate updates for the image and depth encoders. This process not only enhances single-step spatial understanding via the new depth modality but also bolsters implicit multi-step reasoning through data with explicit reasoning processes, providing a "cold start" for the subsequent RFT stage. As a result, this SFT-trained model exhibits improved capability for multi-step spatial referring tasks. Please check Appx. D.3 for details.

**Reinforcement Fine-tuning.** Though SFT employs data with precise reasoning, it tends to memorize answers rather than generalize to novel spatial constraints. We thus design a subsequent RFT stage using Group Relative Policy Optimization (GRPO [114]) with multi-step reasoning data from *RefSpatial*. To guide RFT for more accurate point predictions, we first define two outcome reward (*i.e.*, only care about whether the output answer is correct) functions: **(1)** Outcome Format Reward ($R_{OF}$) for structured reasoning and clarity; and **(2)** Point L1 Reward ($R_P$) granting a score of 1 if the final prediction falls within a specific range near the ground-truth point, and 0 otherwise. To enhance intermediate reasoning precision, we exploit key-step perception annotations from *RefSpatial* and design specialized metric-sensitive process reward functions: **(1)** Process Format Reward ($R_{PF}$), enforcing the format "[Perception Type] [Target Object]:"; **(2)** Accuracy Reward ($R_{Acc}$), which applies to steps included in the key-step perception annotations. For each relevant step, we measure the prediction error using a specific metric, according to the perception type (*e.g.*, L1 distance for positions between ground-truth points and predicted points). Notably, this design is order-invariant and does not constrain the reasoning trajectory to a fixed sequence. We sample $N$ responses $\{a_1, \ldots, a_N\}$ from the current policy (initialized from the SFT model) to encourage exploration. Each response receives a combined reward ($r_i = R_{OF}(a_i) + R_P(a_i) + \alpha R_{PF}(a_i) + \alpha R_{Acc}(a_i)$). where $\alpha$ is set to 0.25. Rewards are normalized within each group to compute relative advantages ($A_i = \frac{r_i - \text{mean}(\{r_j\})}{\text{std}(\{r_j\})}$), which are then used to update the policy, reinforcing high-quality responses and suppressing suboptimal ones. A KL-divergence regularization term stabilizes updates by constraining them near the reference policy. Notably, the SFT initialization provides a strong prior, enabling rapid adaptation to output formats and supporting accurate, step-wise spatial reasoning by using the spatial understanding learned from SFT. Fig 2 shows that the RFT-trained model generalizes well to tasks like 4-step spatial referring, progressively handling intricate spatial relations, and yielding precise point predictions. For more details about the RFT training and reward design, please see Appx. D.4.

### 3.3 *RefSpatial* dataset

#### 3.3.1 Overview

*RefSpatial* is a comprehensive dataset integrating 2D images from OpenImages [135], 3D embodied videos from CA-1M [136], and simulated scenes from Infinigen [137] using Objaverse [138] assets (See Fig. 3 (a)). *RefSpatial*'s key features are: **(1) Fine-Grained Annotations.** While prior spatial datasets [2, 3] simplify object reference by limiting each category to a single instance per scene, *RefSpatial* includes multiple objects of the same category. Moreover, each object is annotated with hierarchical captions—from broad categories (*e.g.*, "cup") to precise spatial referents (*e.g.*, "the third cup from the left", "the cup closest to the camera")—enabling unambiguous spatial referring in cluttered environments. **(2) Multi-Dimensionality.** Beyond basic spatial concepts, relations, point coordinates, and point depth predictions, the dataset supports multi-step spatial reasoning by annotating detailed reasoning processes (all simulated data), addressing limitations in existing datasets. **(3) High Quality.** We rigorously filter data to maintain quality. Retain 466k OpenImages containing text-referable, spatially relevant objects (down from 1.7M); sample 100k frames from CA-1M with text-identifiable 3D bounding boxes (down from 2M); and manually check and annotate 3k Objaverse-LVIS assets with semantic orientation labels (down from 46k). **(4) Large Scale.** Comprising 2.5M samples and 20M QA pairs, our dataset spans qualitative VQA, quantitative queries on object attributes/relations, and point coordinate prediction (Fig. 3(b)). **(5) Rich Diversity.** *RefSpatial* spans indoor and outdoor scenes, covers common embodied scenes and integrates 31

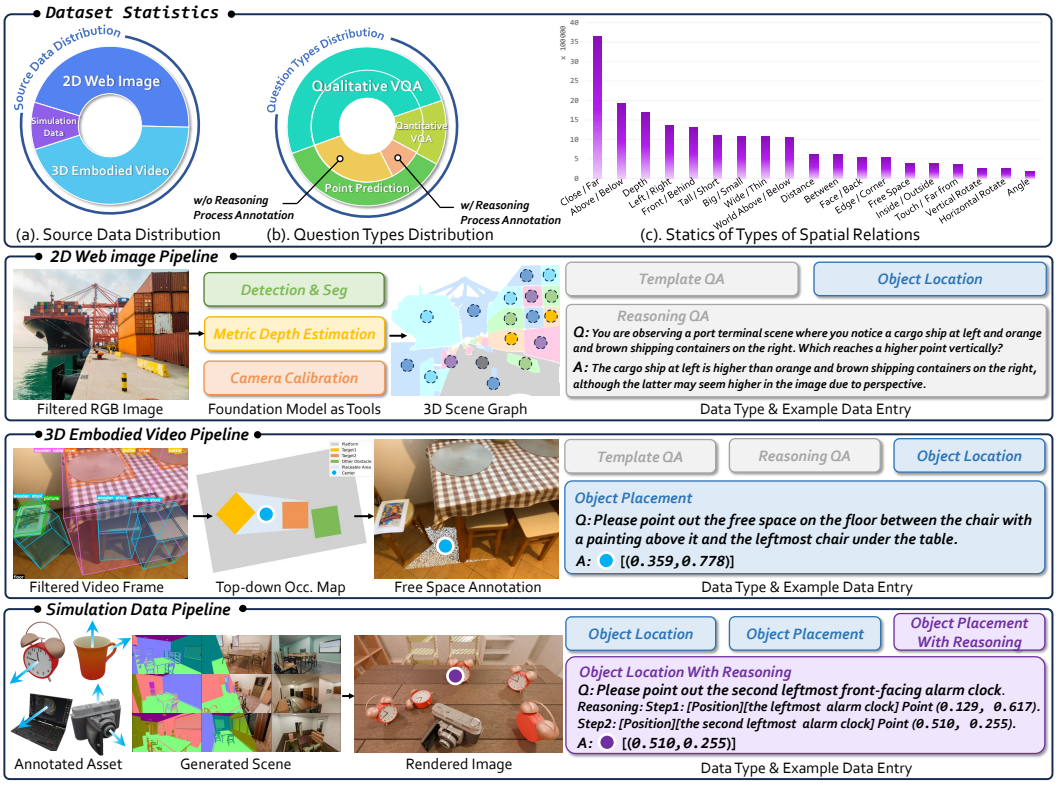



Figure 3: *RefSpatial*: 2.5M data samples from 2D/3D/Simulated sources, with 31 spatial relations.

distinct spatial relations (See Fig. 3 (c)), fostering precise spatial understanding during SFT. **(6) Easy Scalability.** Our pipeline seamlessly scales spatial referring data using diverse sources, including 2D images, 3D videos with bounding boxes, and simulation assets. See Appx. B for more dataset details.

### 3.3.2 Data Recipe

In Fig. 3, we present the dataset recipe that progressively integrates 2D, 3D, and simulated data to enable general VLMs to adapt to spatial referring tasks, thereby enhancing bottom-up spatial understanding and reasoning. **(1) 2D Web Images** aim to endow the model with core spatial concepts and comprehensive depth perception across both indoor and outdoor scenes. To mitigate depth scale and category discrepancies between indoor and outdoor scenes, we leverage the large-scale, diverse 2D web image dataset, OpenImage [135]. However, directly extracting 3D-aware spatial information is challenging. Inspired by prior work [1, 36], we transform 2D images into pseudo-3D scene graphs. In detail, after high-quality filtering (from 1.7M to 466K images), we further enhance the data using Qwen2.5-VL [11] and a heuristic for generating hierarchical region captions, capturing both coarse labels and fine-grained spatial references, differentiating our approach from previous methods. We then construct scene graphs via object detection/segmentation, depth estimation, and camera intrinsic estimation, using object captions as nodes and spatial relations as edges. Finally, we generate QA pairs via template-based or LLM-based approaches, augmented by object-location QA derived from the annotated captions. **(2) 3D Embodied Videos** want to provide the model with a focused spatial understanding of indoor scenes, with a finer-grained perception of spatial relations and concepts. We therefore leverage the richly annotated CA-1M [136]. After fine-grained filtering (from 2M to 100K frames), we construct 3D scene graphs with more diverse spatial relations, enabled by precise 3D bounding boxes compared to 2D approaches. Moreover, we generate top-down occupancy maps that encode the object positions, orientations, and metric distances (*e.g.*, "10cm right of the chair"), enabling accurate spatial referring for placement. **(3) Simulation Data** arms the model with multi-step referring capabilities with spatial reasoning. While 2D and 3D data enable single-step spatial understanding, they are less scalable for multi-step spatial referring with reasoning. Therefore, we leverage procedurally generate scene layouts [137], using manually verified assets [138] (from 46k to 3k) with semantic orientation annotations [7]. Tasks are purposefully designed to foster multi-step

Table 1: Performance on the *single-step spatial understanding* benchmarks across different model types. Top-1 & Top-2 accuracies are represented using **bold text**, and underlines.

| Method | Input | CV-Bench [15] | | | BLINK$_{val}$ [16] | | RoboSpatial [2] | SAT [4] | EmbSpatial [22] |
|---|---|---|---|---|---|---|---|---|---|
| | | 2D-Relation | 3D-Depth | 3D-Distance | 2D-Relation | 3D-Depth | | | |
| *Proprietary Models* | | | | | | | | | |
| GPT-4o [8] | RGB | 84.62 | 86.50 | 83.33 | 82.52 | 78.23 | 77.20 | 68.67 | 63.38 |
| Gemini-2.5-Pro [9] | RGB | 93.54 | 91.00 | 90.67 | **91.61** | 87.90 | 77.24 | 70.59 | **76.67** |
| Claude-3.7-Sonnet [37] | RGB | 74.15 | 85.83 | 84.17 | 74.83 | 67.74 | 60.73 | 40.67 | 33.33 |
| *Open-Source Vison-Language Models* | | | | | | | | | |
| NVILA-2B [38] | RGB | 70.15 | 79.67 | 60.00 | 67.83 | 62.10 | 51.79 | 31.33 | 47.34 |
| NVILA-8B [38] | RGB | 91.54 | 91.83 | 90.67 | 76.92 | 76.61 | 59.35 | 63.33 | 67.72 |
| Qwen-2.5-VL-7B [11] | RGB | 82.15 | 60.17 | 69.00 | 64.34 | 60.98 | 49.59 | 30.00 | 40.20 |
| Qwen-2.5-VL-72B [11] | RGB | 84.15 | 86.17 | 84.15 | 78.32 | 73.55 | 70.73 | 65.33 | 57.69 |
| *Spatial Specialist Models* | | | | | | | | | |
| SpatialBot-3B [14] | RGB-D | 69.38 | 77.33 | 60.83 | 67.83 | 67.74 | 72.36 | 63.33 | 50.66 |
| SpatialRGPT-8B [1] | RGB-D | 91.00 | 89.8 | 88.50 | 81.12 | 89.51 | 66.67 | 64.00 | 59.62 |
| SpaceLLaVA-13B [6] | RGB | 63.69 | 66.83 | 70.17 | 72.73 | 62.90 | 61.00 | 62.67 | 49.40 |
| RoboPoint-13B [5] | RGB | 75.85 | 77.83 | 44.50 | 60.84 | 61.29 | 69.90 | 46.60 | 49.31 |
| *RoboRefer Variants* | | | | | | | | | |
| *RoboRefer*-2B-SFT | RGB | 96.15 | 95.83 | 90.67 | 83.92 | 88.71 | 82.93 | 71.33 | 70.66 |
| *RoboRefer*-2B-SFT | RGB-D | 96.31 | 97.17 | 90.83 | 87.41 | 91.13 | 82.93 | 82.00 | 71.10 |
| *RoboRefer*-8B-SFT | RGB-D | **96.90** | **98.33** | **93.50** | **91.61** | **92.74** | **84.55** | **86.67** | 72.53 |

Table 2: Performance on current referring and *multi-step spatial referring* benchmarks. L. and P. denote our benchmark's Location and Placement parts; U. indicates unseen compositional spatial relations during SFT/RFT. Top-1 & Top-2 accuracies are represented using **bold text**, and underlines.

| Benchmark | *Proprietary Models* | *Referring Specialist Models* | | | | *RoboRefer Variants* | | |
|---|---|---|---|---|---|---|---|---|
| | Gemini-2.5-Pro [9] | SpaceLLaVA [6] | RoboPoint [5] | Molmo-7B [15] | Molmo-72B [15] | 2B-SFT | 8B-SFT | 2B-RFT |
| RoboRefIt | - | 21.3 | 49.8 | - | - | 72.8 | **75.9** | 74.2 |
| Where2Place | 61.9 | 11.8 | 46.8 | 45.0 | 63.8 | 66.0 | 70.0 | **71.0** |
| RoboSpatial | 40.2 | 16.0 | 41.3 | 38.0 | 40.9 | 66.4 | 70.8 | **71.3** |
| *RefSpatial-Bench*-L. | 46.96 | 5.82 | 22.87 | 21.91 | 45.77 | 47.00 | **52.00** | **52.00** |
| *RefSpatial-Bench*-P. | 24.21 | 4.31 | 9.27 | 12.85 | 14.74 | 48.00 | 53.00 | **54.00** |
| *RefSpatial-Bench*-U. | 27.14 | 4.02 | 8.40 | 12.23 | 21.24 | 33.77 | 37.66 | **41.56** |

spatial referring and generate corresponding data. We assume that the generated code reflects optimal reasoning, with each line translated into textual form and intermediate results filled into structured formats (*e.g.*, coordinates, distances), as shown in Fig. 2, Fig. 3, and Appx. D.4.2, yielding QA pairs with reasoning annotations. For more demonstrations about *RefSpatial*, please refer to Appx. F.

### 3.4 Training Details

We adopt NVILA [38] (2B/8B) as the base model and apply SFT to obtain *RoboRefer*-SFT. Due to computational limits, RFT is applied only to the 2B model, yielding *RoboRefer*-RFT. SFT has two steps: the first uses only the *RefSpatial*; the second trains on a mixture of *RefSpatial*, instruction tuning (1/20 the size of *RefSpatial* QA)[133, 134], and referring datasets[87]. Notably, *RefSpatial* is reused with both RGB and RGB-D inputs in the second step to enforce the image encoder to learn spatial understanding beyond depth cues. Thus, the model supports both RGB-only and RGB-D inference, with depth optionally inferred via a relative depth estimation model [139]. Finally, RFT stage uses the multi-step reasoning data from *RefSpatial* to train. See Appx. D for details.

## 4 Experiments

### 4.1 Single-step Spatial Understanding

We evaluate on public *single-step spatial understanding benchmarks*, including CV-Bench [15], the BLINK [16] validation split, RoboSpatial [2] configuration part, SAT [4], and EmbSpatial [22]. Check Appx. E.2 for more evaluation details. The following parts present our analyses.

Table 3: Performance on general referring benchmarks. B. and P. denote Bounding Box and Point. Top-1/2 accuracies are indicated by **bold**/underlined text.

| Method | Output | RefCOCO val / testA / testB | RefCOCO+ val / testA / testB | RefCOCOg val / test |
|---|---|---|---|---|
| *Grounding Specialist Models* | | | | |
| GroundingDINO | BBox | 90.6 / 93.2 / 88.2 | 88.2 / 89.0 / 75.9 | 86.1 / 87.0 |
| *Open-Source Vision-Language Models* | | | | |
| Qwen2.5-VL-72B | BBox | 92.7 / 94.6 / 89.7 | 88.9 / 92.2 / 83.7 | 89.9 / 90.3 |
| Qwen2.5-VL-72B | Point | 95.2 / 96.5 / 93.8 | 90.4 / 93.5 / 86.7 | 91.5 / 92.0 |
| Qwen2.5-VL-72B | B. → P. | 95.4 / 97.0 / 94.2 | 91.5 / 94.9 / **88.2** | 92.5 / 92.5 |
| *RoboRefer Variants* | | | | |
| *RoboRefer*-2B-SFT | Point | 95.5 / 96.7 / 93.8 | 91.8 / 94.3 / 86.3 | 93.3 / 93.6 |
| *RoboRefer*-8B-SFT | Point | **96.6** / **97.7** / **94.7** | 91.9 / **95.2** / 87.5 | **94.3** / **94.1** |
| *RoboRefer*-2B-RFT | Point | 95.6 / 96.6 / 93.9 | **92.0** / 94.2 / 86.3 | 93.2 / 93.63 |

Table 4: Performance on general VLM benchmarks. We also show the advantage of dedicated depth encoder (E. = Encoder). We use the same evaluation protocol of CV-Bench/BLINK in Sec. 4.1.

| Benchmark | NVILA-2B [38] | *RoboRefer*-2B-SFT Shared E. | *RoboRefer*-2B-SFT Dedicated E. (Ours) |
|---|---|---|---|
| *Public Vision-Language Benchmarks* | | | |
| $MME_{test}$ | 1547 | 1541 | **1553** |
| $MMBench_{dev}$ | **78.63** | 76.23 | 77.73 |
| OK-VQA | 61.93 | 64.9 | **65.81** |
| POPE | 81.96 | 81.93 | **83.44** |
| *Single-step Spatial Understanding Benchmarks* | | | |
| CV-Bench | 69.94 | 93.99 | **94.22** |
| $BLINK_{val}$ | 64.97 | 80.02 | **85.26** |

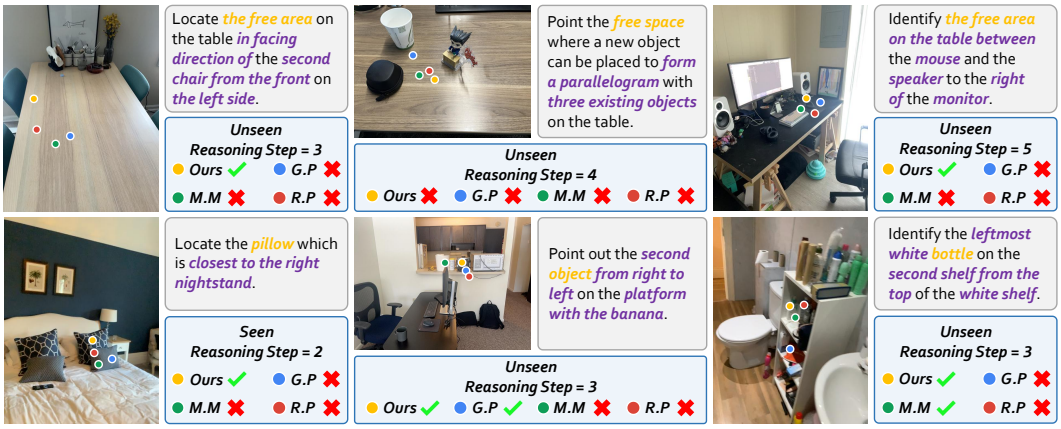



Figure 4: *RefSpatial-Bench* results. G.P., M.M., and R.P. donate Gemini-2.5-Pro [9], Molmo-72B [15], and RoboPoint [5]. *RoboRefer*-RFT excels in unseen and multi-step cases.

**SFT stage enables strong spatial understanding.** In Tab 1, trained solely on *RefSpatial*, *RoboRefer*-SFT surpasses all spatial specialist models on these benchmarks, even surpassing Gemini-2.5-Pro by 5% (absolute) on average. Moreover, our 2B variant outperforms NVILA-2B by 21.7% (absolute).

**Depth input improve 3D spatial understanding during inference.** In Tab. 1, we find that incorporating depth information during inference leads to relative improvements in 3D benchmarks compared to 2D ones by 1.5%, although our model exhibits strong spatial understanding with RGB input alone by reusing the *RefSpatial* dataset with both RGB and RGB-D inputs during SFT's second step.

### 4.2 Multi-step Spatial Referring

We first evaluate current robotic referring benchmarks, namely RoboRefIt [140] (location) and Where2Place [5]/RoboSpatial [2] (placement), all limited to 2 reasoning steps. To evaluate more complex multi-step spatial referring, we propose *RefSpatial-Bench*, a challenging benchmark based on real-world cluttered scenes. It contains two subsets, Location and Placement, each with 100 images. Notably, 77 images involve spatial relation combinations unseen in *RefSpatial*. Over 70% requires multi-step reasoning (up to 5 steps), including precise ground-truth masks. More details about *RefSpatial* can be found in Appx. C. For metrics, we report the average success rate of predicted points within the mask. We evaluate *RoboRefer* using RGB-D inputs by default, with depth maps generated from RGB images via DepthAnything V2 [139]. See Appx. E.3 for more details.

**RFT stage fosters better reasoning ability.** As shown in Tab. 2, the 2B-RFT variant outperforms all baselines, exceeding the prior SOTA (Gemini-2.5-Pro [9]) by 17.4% (absolute) on *RefSpatial-Bench*. We find that although Gemini-2.5-Pro excels in 2D referring (*e.g.*, color, image-space localization), it struggles with 3D spatial relations involving distance (*e.g.*, identifying the second-farthest object),

Table 5: Simulation Results

| Method | Success Rate(%) ↑ L.1 | L.2 | L.3 | Avg. | Execution Time(s) ↓ |
|---|---|---|---|---|---|
| Octo | 51.2 | 12.7 | 0.0 | 43.2 | - |
| OpenVLA | 51.6 | 13.1 | 0.0 | 43.6 | - |
| SoFar | 75.3 | 65.6 | 50.0 | 72.4 | 40 |
| **Ours** | **81.4** | **73.1** | **80.0** | **79.2** | **29** |

Table 6: Real-world robot evaluation requiring spatial referring.

| Manipulation or Navigation tasks with spatial referring | Success Rate(%) ↑ OpenVLA | RoboPoint | **Ours** |
|---|---|---|---|
| Pick the specific hamburger closest to the mug nearest the camera and place it in front of the teddy bear. | 0.00 | 0.00 | **80.00** |
| Pick the apple in front of the leftmost cup's logo side, navigate to the nearest table, and place it aligned with the apple row. | 0.00 | 0.00 | **60.00** |

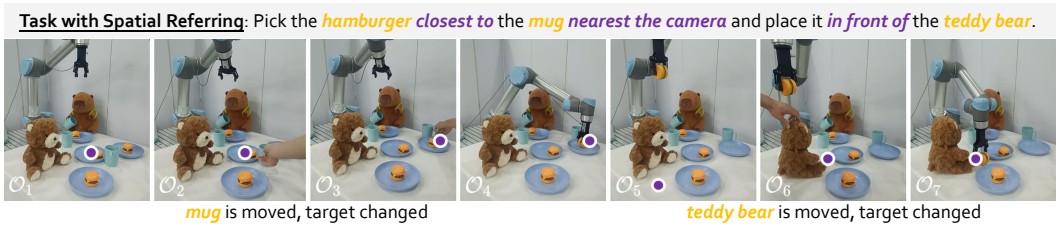



Figure 5: Real-World Evaluation. The purple point denotes the current target predicted by our model.

reducing overall performance when multiple spatial constraints are combined. Fig. 4 shows complex multi-step spatial reasoning cases from *RefSpatial-Bench* with model comparisons.

**RFT stage provides powerful generalization ability.** In the *RefSpatial-Bench*-Unseen row of Tab. 2, we evaluate novel spatial relation combinations omitted during our SFT/RFT training. The 2B-RFT model exceeds 2B-SFT by 9.1% (absolute), indicating that SFT overfits the training distribution, whereas the RFT model better generalizes by leveraging learned spatial knowledge, consistent with prior findings [141]. Fig. 4 shows results on these unseen combinations across different models.

### 4.3 Public vision-language benchmarks

***RefSpatial* enhance 2D general referring ability.** We also evaluate 2D referring capability on the ReCOCO/+/g [87]. Since our model predicts a single point, we deem a prediction correct if the point lies within the ground-truth bounding box. As this evaluation differs from standard visual grounding protocols, we additionally assess Qwen-2.5VL-72B [11] by using either its predicted point or the center of its predicted box as baselines. In Tab. 3, our method surpasses baselines, indicating that our dataset not only supports 3D spatial referring but also enhances 2D referring performance.

**Joint RGB and RGB-D training preserves commonsense knowledge.** In Tab. 4, we assess how spatial and depth information influences overall VQA performance by comparing *RoboRefer*-2B-SFT with the baseline NVILA-2B [38], trained on standard VQA datasets. Our model achieves comparable or slightly superior results, corroborating insights from SpatialVLM [6] and SpatialRGPT [1]. These findings indicate that although VLMs often struggle with spatial reasoning, targeted spatial VQA training, especially with combined RGB and RGB-D data enriched by general visual instruction datasets, can enhance spatial understanding without compromising overall VQA performance.

### 4.4 Simulator and Real-world Evaluation for Robotics

***RoboRefer* can be integrated into the system as a useful tool.** We evaluate our model on the Open6DOR [59] V2 position track, comparing against VLA-based baselines (pretrained Octo [142], LIBERO-finetuned OpenVLA [143]) and SoFar [7], which integrates Florence-2 [93], SAM [78], GPT-4o, and GSNet. *RoboRefer* serves as a lightweight alternative to Florence-2 and GPT-4o for object localization and placement. By using a single target point predicted by *RoboRefer*, the system can generate more accurate masks and corresponding grasp poses than those from 2D boxes under occlusion in cluttered scenes, yielding a 6.8% absolute improvement in success rate (Tab. 5). Its compact size also reduces execution time by 27.5% relative to GPT-4o. See Appx. E.4 for details.

**Spatial referring from *RoboRefer* is crucial for real-world robots.** In Tab. 6, only our method can handle long-horizon tasks requiring complex multi-step spatial referring in cluttered and dynamic environments. These tasks are challenging, as the robot must precisely identify objects and their placement to satisfy spatial constraints that may change over time. In Fig. 5, integrating *RoboRefer* with an open-loop policy enables rapid updates at 2.5 Hz. Thus, when the mug nearest the camera

is moved, the robot adapts by grasping the hamburger closest to the mug's new position and also readjusts placement after the teddy bear's 90° rotation, preserving correct spatial alignment. Notably, spatial referring unifies both manipulation and navigation under a single formulation. This allows the G1 humanoid to navigate while performing spatially constrained pick-and-place actions (Fig. 1), thereby enabling more complex, long-horizon tasks. Check Appx. E.5 for more details.

### 4.5 Ablation Study

**Data recipe is critical for SFT training.** Ablation results in Tab. 7 reveal that combining 2D, 3D, and simulated data yields optimal performance. As noted in Sec. 3.3, 2D data spans indoor/outdoor scenes, enabling depth learning across scales; its removal severely degrades performance on outdoor-centric BLINK [16]. Meanwhile, 3D data captures embodied indoor environments and mitigates the Sim2Real gap, benefiting indoor-focused CV-Bench [15]. Finally, simulated data broadens spatial diversity. This tripartite data composition is thus key to effective SFT training.

**Dedicated depth encoder preserves image understanding.** We compare dedicated and shared image-depth encoders during SFT. In Tab. 4, the dedicated encoder better maintains image understanding under limited RGB-only data (1/20 *RefSpatial* QA), while the shared encoder harms general performance. Though prior work[1] adopts a shared encoder, it **(1)** requires over twice as much RGB-only data compared to spatial-related data for co-training; **(2)** *targets region-level depths, differing from our full-image approach*.

Table 7: Ablation Studies. S.D. means simulated data. P.R. denotes process reward. We use the same evaluation protocol in Sec. 4.1 and Sec. 4.2.

| Data Recipe | | | Depth | Spatial Understanding | |
|---|---|---|---|---|---|
| 2D | 3D | S.D. | Encoder | CV-Bench | BLINK$_{val}$ |
| *SFT Variants (2B)* | | | | | |
| ✗ | ✓ | ✓ | ✓ | 84.17 | 74.48 |
| ✓ | ✗ | ✓ | ✓ | 81.83 | 74.61 |
| ✓ | ✓ | ✗ | ✓ | 83.96 | 75.10 |
| ✓ | ✓ | ✓ | ✗ | 91.24 | 85.27 |
| ✓ | ✓ | ✓ | ✓ | **94.77** | **89.27** |

| Reward | Depth | Spatial Referring |
|---|---|---|
| P.R. | Encoder | *RefSpatial-Bench* |
| *RFT Variants (2B)* | | |
| ✓ | ✗ | 40.00 |
| ✗ | ✓ | 48.00 |
| ✓ | ✓ | **53.00** |

**Depth encoder improves both spatial understanding and reasoning.** Recent VLMs [3, 107] show that large-scale spatial training enables implicit 3D understanding (*e.g.*, depth, distance, 3D boxes) from images alone. To assess this, we fine-tune NVILA-2B [38] on *RefSpatial* without the depth encoder, followed by continued RFT. Results indicate that depth improves single-step spatial understanding, consistent with MM-Spatial [3], and yields greater gains in multi-step spatial referring. We attribute this to: **(1)** the need for precise coordinate prediction in spatial referring, unlike VQA's multiple-choice; **(2)** cumulative reasoning across steps, amplifying the utility of depth cues.

**Process reward advances the accuracy of intermediate perception.** Tab 7 shows a 5-point improvement with process reward, which leverages key step annotations from *RefSpatial* to refine step-wise perception, thereby predicting more accurate points with complex spatial relations.

## 5 Conclusion and Future work

In this paper, we introduce *RoboRefer*, a novel 3D-aware VLM that addresses spatial referring through the combination of both single-step accurate understanding and multi-step spatial reasoning. In detail, we enhance 3D perception with a separate depth encoder via SFT, and enable generalized multi-step spatial referring via RFT with our proposed metric-sensitive process reward functions. We also present *RefSpatial*, a large-scale, well-designed dataset for SFT and RFT training, with *RefSpatial-Bench*, a benchmark tailored to evaluate spatial referring. Extensive experiments show the effectiveness of *RoboRefer* and highlight its potential for a broad range of robotic applications.

Our future work will focus on two main directions. **(1)** Enhancing the model's understanding of human priors and intent. As discussed in Appx. G, human instructions are often brief and ambiguous, even when the correct location is unique. Potential solutions include exploring procedural synthesis of intent-aware data or improving model performance through co-training with intent-rich datasets. **(2)** Improving the model's 3D perception capabilities. Our current models predominantly rely on qualitative spatial relations (*e.g.*, left, right) and predict 2D image-plane coordinates, necessitating depth-based conversion to 3D, as discussed in Appx. A. Future directions include directly modeling quantitative geometry to enable precise 3D reasoning or direct prediction of 3D points and visual traces, which are more challenging if combined with spatially constrained instructions.

## Acknowledgement

This work was supported by the National Natural Science Foundation of China (62132001, 62476011), Capital's Funds for Health Improvement and Research (CFH 2024-2-40611), and the Fundamental Research Funds for the Central Universities. We sincerely thank Jiayuan Zhang, Jiawei He for their valuable discussions and insightful feedback about the method. We also sincerely appreciate Yusu Deng's excellent figure design (*e.g.*, teaser figure, pipeline overview) and demo reviewing work.

## References

[1] 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 models. *NIPS*, 2024.

[2] 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. *CVPR*, 2025.

[3] Erik Daxberger, Nina Wenzel, David Griffiths, Haiming Gang, Justin Lazarow, Gefen Kohavi, Kai Kang, Marcin Eichner, Yinfei Yang, Afshin Dehghan, et al. Mm-spatial: Exploring 3d spatial understanding in multimodal llms. *arXiv*, 2025.

[4] 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*, 2024.

[5] 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*, 2024.

[6] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *CVPR*, 2024.

[7] 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*, 2025.

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

[9] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv*, 2023.

[10] 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*, 2024.

[11] 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*, 2025.

[12] 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. *NIPS*, 2023.

[13] Yining Hong, Chunru Lin, Yilun Du, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. 3d concept learning and reasoning from multi-view images. In *CVPR*, 2023.

[14] Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. Spatialbot: Precise spatial understanding with vision language models. *ICRA*, 2025.

[15] Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *NIPS*, 2024.

[16] 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 *ECCV*, 2024.

[17] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *CVPR*, 2022.

[18] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019.

[19] Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. In *ECCV*, 2024.

[20] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[21] Emilia Szymanska, Mihai Dusmanu, Jan-Willem Buurlage, Mahdi Rad, and Marc Pollefeys. Space3d-bench: Spatial 3d question answering benchmark. In *ECCV 2024 Workshop*, 2024.

[22] Mengfei Du, Binhao Wu, Zejun Li, Xuan-Jing Huang, and Zhongyu Wei. Embspatial-bench: Benchmarking spatial understanding for embodied tasks with large vision-language models. In *ACL (Volume 2: Short Papers)*, 2024.

[23] Santhosh Kumar Ramakrishnan, Erik Wijmans, Philipp Kraehenbuehl, and Vladlen Koltun. Does spatial cognition emerge in frontier models? *ICLR*, 2025.

[24] Enshen Zhou, Cheng Chi, Yibo Li, Jingkun An, Jiayuan Zhang, Shanyu Rong, Yi Han, Yuheng Ji, Mengzhen Liu, Pengwei Wang, et al. Robotracer: Mastering spatial trace with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2512.13660*, 2025.

[25] Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's" up" with vision-language models? investigating their struggle with spatial reasoning. In *EMNLP*, 2023.

[26] Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 2023.

[27] Navid Rajabi and Jana Kosecka. Towards grounded visual spatial reasoning in multi-modal vision language models. *arXiv*, 2023.

[28] Kanchana Ranasinghe, Satya Narayan Shukla, Omid Poursaeed, Michael S Ryoo, and Tsung-Yu Lin. Learning to localize objects improves spatial reasoning in visual-llms. In *CVPR*, pages 12977–12987, 2024.

[29] Fatemeh Shiri, Xiao-Yu Guo, Mona Far, Xin Yu, Reza Haf, and Yuan-Fang Li. An empirical analysis on spatial reasoning capabilities of large multimodal models. In *EMNLP*, 2024.

[30] Phillip Y Lee, Jihyeon Je, Chanho Park, Mikaela Angelina Uy, Leonidas Guibas, and Minhyuk Sung. Perspective-aware reasoning in vision-language models via mental imagery simulation. *arXiv*, 2025.

[31] Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Sharon Li, and Neel Joshi. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. *NIPS*, 2024.

[32] Yihong Tang, Ao Qu, Zhaokai Wang, Dingyi Zhuang, Zhaofeng Wu, Wei Ma, Shenhao Wang, Yunhan Zheng, Zhan Zhao, and Jinhua Zhao. Sparkle: Mastering basic spatial capabilities in vision language models elicits generalization to composite spatial reasoning. *arXiv*, 2024.

[33] Yuecheng Liu, Dafeng Chi, Shiguang Wu, Zhanguang 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*, 2025.

[34] Yang Liu, Ming Ma, Xiaomin Yu, Pengxiang Ding, Han Zhao, Mingyang Sun, Siteng Huang, and Donglin Wang. Ssr: Enhancing depth perception in vision-language models via rationale-guided spatial reasoning. *arXiv preprint arXiv:2505.12448*, 2025.

[35] Wufei Ma, Luoxin Ye, Celso de Melo, Alan L Yuille, and Jieneng Chen. Spatialllm: A compound 3d-informed design towards spatially-intelligent large multimodal models. In *CVPR*, 2025.

[36] Wufei Ma, Yu-Cheng Chou, Qihao Liu, Xingrui Wang, Celso de Melo, Jieneng Chen, Jianwen Xie, and Alan Yuille. Spatialreasoner: Towards explicit and generalizable 3d spatial reasoning. *arXiv*, 2025.

[37] AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 1:1, 2024.

[38] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. *arXiv*, 2024.

[39] Zhenfei Yin, Jiong Wang, Jianjian Cao, Zhelun Shi, Dingning Liu, Mukai Li, Xiaoshui Huang, Zhiyong Wang, Lu Sheng, Lei Bai, et al. Lamm: Language-assisted multi-modal instruction-tuning dataset, framework, and benchmark. *NIPS*, 2023.

[40] Yiran Qin, Enshen Zhou, Qichang Liu, Zhenfei Yin, Lu Sheng, Ruimao Zhang, Yu Qiao, and Jing Shao. Mp5: A multi-modal open-ended embodied system in minecraft via active perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16307–16316, 2024.

[41] Yiran Qin, Ao Sun, Yuze Hong, Benyou Wang, and Ruimao Zhang. Navigatediff: Visual predictors are zero-shot navigation assistants. *arXiv preprint arXiv:2502.13894*, 2025.

[42] Yiran Qin, Zhelun Shi, Jiwen Yu, Xijun Wang, Enshen Zhou, Lijun Li, Zhenfei Yin, Xihui Liu, Lu Sheng, Jing Shao, et al. Worldsimbench: Towards video generation models as world simulators. *arXiv preprint arXiv:2410.18072*, 2024.

[43] Zeren Chen, Zhelun Shi, Xiaoya Lu, Lehan He, Sucheng Qian, Zhenfei Yin, Wanli Ouyang, Jing Shao, Yu Qiao, Cewu Lu, et al. Rh20t-p: A primitive-level robotic dataset towards composable generalization agents. *arXiv preprint arXiv:2403.19622*, 2024.

[44] Enshen Zhou, Yiran Qin, Zhenfei Yin, Yuzhou Huang, Ruimao Zhang, Lu Sheng, Yu Qiao, and Jing Shao. Minedreamer: Learning to follow instructions via chain-of-imagination for simulated-world control. *arXiv preprint arXiv:2403.12037*, 2024.

[45] Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. *arXiv preprint arXiv:2502.21257*, 2025.

[46] Yuzhou Huang, Yiran Qin, Shunlin Lu, Xintao Wang, Rui Huang, Ying Shan, and Ruimao Zhang. Story3d-agent: Exploring 3d storytelling visualization with large language models. *arXiv preprint arXiv:2408.11801*, 2024.

[47] Lijun Li, Zhelun Shi, Xuhao Hu, Bowen Dong, Yiran Qin, Xihui Liu, Lu Sheng, and Jing Shao. T2isafety: Benchmark for assessing fairness, toxicity, and privacy in image generation. *arXiv preprint arXiv:2501.12612*, 2025.

[48] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv*, 2025.

[49] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *Robotics: Science and Systems*, 2024.

[50] Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks, 2024.

[51] Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Lily Lee, Kaichen Zhou, Pengju An, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. Robomamba: Multimodal state space model for efficient robot reasoning and manipulation. *arXiv e-prints*, pages arXiv–2406, 2024.

[52] BAAI RoboBrain Team, Mingyu Cao, Huajie Tan, Yuheng Ji, Xiansheng Chen, Minglan Lin, Zhiyu Li, Zhou Cao, Pengwei Wang, Enshen Zhou, et al. Robobrain 2.0 technical report. *arXiv preprint arXiv:2507.02029*, 2025.

[53] Yuheng Ji, Huajie Tan, Cheng Chi, Yijie Xu, Yuting Zhao, Enshen Zhou, Huaihai Lyu, Pengwei Wang, Zhongyuan Wang, Shanghang Zhang, et al. Mathsticks: A benchmark for visual symbolic compositional reasoning with matchstick puzzles. *arXiv preprint arXiv:2510.00483*, 2025.

[54] Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. *arXiv preprint arXiv:2506.10967*, 2025.

[55] Rui Li, Zixuan Hu, Wenxi Qu, Jinouwen Zhang, Zhenfei Yin, Sha Zhang, Xuantuo Huang, Hanqing Wang, Tai Wang, Jiangmiao Pang, et al. Labutopia: High-fidelity simulation and hierarchical benchmark for scientific embodied agents. *arXiv preprint arXiv:2505.22634*, 2025.

[56] Shaoan Wang, Jiazhao Zhang, Minghan Li, Jiahang Liu, Anqi Li, Kui Wu, Fangwei Zhong, Junzhi Yu, Zhizheng Zhang, and He Wang. Trackvla: Embodied visual tracking in the wild. *arXiv preprint arXiv:2505.23189*, 2025.

[57] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 2023.

[58] Chenyang Ma, Kai Lu, Ta-Ying Cheng, Niki Trigoni, and Andrew Markham. Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors. In *NIPS*, 2024.

[59] Yufei Ding, Haoran Geng, Chaoyi Xu, Xiaomeng Fang, Jiazhao Zhang, Songlin Wei, Qiyu Dai, Zhizheng Zhang, and He Wang. Open6dor: Benchmarking open-instruction 6-dof object rearrangement and a vlm-based approach. In *IROS*, 2024.

[60] Yiran Qin, Li Kang, Xiufeng Song, Zhenfei Yin, Xiaohong Liu, Xihui Liu, Ruimao Zhang, and Lei Bai. Robofactory: Exploring embodied agent collaboration with compositional constraints. *arXiv preprint arXiv:2503.16408*, 2025.

[61] Kefei Zhu, Fengshuo Bai, YuanHao Xiang, Yishuai Cai, Xinglin Chen, Ruochong Li, Xingtao Wang, Hao Dong, Yaodong Yang, Xiaopeng Fan, et al. Dexflywheel: A scalable and self-improving data generation framework for dexterous manipulation. *arXiv preprint arXiv:2509.23829*, 2025.

[62] Huajie Tan, Xiaoshuai Hao, Minglan Lin, Pengwei Wang, Yaoxu Lyu, Mingyu Cao, Zhongyuan Wang, and Shanghang Zhang. Roboos: A hierarchical embodied framework for cross-embodiment and multi-agent collaboration. *arXiv preprint arXiv:2505.03673*, 2025.

[63] Enshen Zhou, Qi Su, Cheng Chi, Zhizheng Zhang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, and He Wang. Code-as-monitor: Constraint-aware visual programming for reactive and proactive robotic failure detection. *arXiv preprint arXiv:2412.04455*, 2024.

[64] Yi Han, Cheng Chi, Enshen Zhou, Shanyu Rong, Jingkun An, Pengwei Wang, Zhongyuan Wang, Lu Sheng, and Shanghang Zhang. Tiger: Tool-integrated geometric reasoning in vision-language models for robotics. *arXiv preprint arXiv:2510.07181*, 2025.

[65] Linyi Jin, Jianming Zhang, Yannick Hold-Geoffroy, Oliver Wang, Kevin Blackburn-Matzen, Matthew Sticha, and David F Fouhey. Perspective fields for single image camera calibration. In *CVPR*, 2023.

[66] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. In *CVPR*, 2024.

[67] Mengzhen Liu, Mengyu Wang, Henghui Ding, Yilong Xu, Yao Zhao, and Yunchao Wei. Segment anything with precise interaction. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3790–3799, 2024.

[68] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *arXiv*, 2025.

[69] Aleksei Bochkovskii, AmaÃĢl Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv*, 2024.

[70] Chengbo Yuan, Rui Zhou, Mengzhen Liu, Yingdong Hu, Shengjie Wang, Li Yi, Chuan Wen, Shanghang Zhang, and Yang Gao. Motiontrans: Human vr data enable motion-level learning for robotic manipulation policies. *arXiv preprint arXiv:2509.17759*, 2025.

[71] Zhe Li, Weihao Yuan, Yisheng He, Lingteng Qiu, Shenhao Zhu, Xiaodong Gu, Weichao Shen, Yuan Dong, Zilong Dong, and Laurence T Yang. Lamp: Language-motion pretraining for motion generation, retrieval, and captioning. *arXiv preprint arXiv:2410.07093*, 2024.

[72] Zhe Li, Yisheng He, Lei Zhong, Weichao Shen, Qi Zuo, Lingteng Qiu, Zilong Dong, Laurence Tianruo Yang, and Weihao Yuan. Mulsmo: Multimodal stylized motion generation by bidirectional control flow. *arXiv preprint arXiv:2412.09901*, 2024.

[73] Zhe Li, Cheng Chi, Yangyang Wei, Boan Zhu, Yibo Peng, Tao Huang, Pengwei Wang, Zhongyuan Wang, Shanghang Zhang, and Chang Xu. From language to locomotion: Retargeting-free humanoid control via motion latent guidance. *arXiv preprint arXiv:2510.14952*, 2025.

[74] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv*, 2025.

[75] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *CVPR*, 2023.

[76] Shengjie Zhu, Abhinav Kumar, Masa Hu, and Xiaoming Liu. Tame a wild camera: in-the-wild monocular camera calibration. *NIPS*, 2023.

[77] 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 *ICCV*, 2023.

[78] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *ICLR*, 2025.

[79] Xingrui Wang, Wufei Ma, Tiezheng Zhang, Celso M de Melo, Jieneng Chen, and Alan Yuille. Pulsecheck457: A diagnostic benchmark for 6d spatial reasoning of large multimodal models. *CVPR*, 2025.

[80] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016.

[81] Kees van Deemter, Ielka van der Sluis, and Albert Gatt. Building a semantically transparent corpus for the generation of referring expressions. In *CVPR*, 2006.

[82] Dave Golland, Percy Liang, and Dan Klein. A game-theoretic approach to generating spatial descriptions. In *EMNLP*, 2010.

[83] Margaret Mitchell, Kees van Deemter, and Ehud Reiter. Natural reference to objects in a visual domain. In *Proceedings of the 6th international natural language generation conference*, 2010.

[84] Margaret Mitchell, Kees Van Deemter, and Ehud Reiter. Generating expressions that refer to visible objects. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013.

[85] Nicholas FitzGerald, Yoav Artzi, and Luke Zettlemoyer. Learning distributions over logical forms for referring expression generation. In *EMNLP*, 2013.

[86] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.

[87] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016.

[88] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *ICLR*, 2024.

[89] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv*, 2023.

[90] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv*, 2023.

[91] Yang Zhao, Zhijie Lin, Daquan Zhou, Zilong Huang, Jiashi Feng, and Bingyi Kang. Bubogpt: Enabling visual grounding in multi-modal llms. *arXiv*, 2023.

[92] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *NIPS*, 2023.

[93] 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 *CVPR*, 2024.

[94] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015.

[95] Mingzhe Wang, Mahmoud Azab, Noriyuki Kojima, Rada Mihalcea, and Jia Deng. Structured matching for phrase localization. In *ECCV*, 2016.

[96] Bryan A Plummer, Arun Mallya, Christopher M Cervantes, Julia Hockenmaier, and Svetlana Lazebnik. Phrase localization and visual relationship detection with comprehensive image-language cues. In *CVPR*, 2017.

[97] Chi Xie, Zhao Zhang, Yixuan Wu, Feng Zhu, Rui Zhao, and Shuang Liang. Described object detection: Liberating object detection with flexible expressions. *NIPS*, 2023.

[98] Shuting He, Henghui Ding, Chang Liu, and Xudong Jiang. Grec: Generalized referring expression comprehension. *arXiv*, 2023.

[99] Chang Liu, Henghui Ding, and Xudong Jiang. Gres: Generalized referring expression segmentation. In *CVPR*, 2023.

[100] Yutao Hu, Qixiong Wang, Wenqi Shao, Enze Xie, Zhenguo Li, Jungong Han, and Ping Luo. Beyond one-to-one: Rethinking the referring image segmentation. In *CVPR*, 2023.

[101] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, 2024.

[102] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *CoRL*, 2021.

[103] Weiyu Liu, Chris Paxton, Tucker Hermans, and Dieter Fox. Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects. In *ICRA*, 2022.

[104] Weiyu Liu, Yilun Du, Tucker Hermans, Sonia Chernova, and Chris Paxton. Structdiffusion: Language-guided creation of physically-valid structures using unseen objects. In *RSS*, 2023.

[105] Wentao Yuan, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. M2t2: Multi-task masked transformer for object-centric pick and place. *arXiv*, 2023.

[106] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *CoRL*, 2022.

[107] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv*, 2025.

[108] Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. In *ICML*, 2024.

[109] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, et al. Magma: A foundation model for multimodal ai agents. *arXiv*, 2025.

[110] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv*, 2022.

[111] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv*, 2022.

[112] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *NIPS*, 2023.

[113] Jingkun An, Yinghao Zhu, Zongjian Li, Enshen Zhou, Haoran Feng, Xijie Huang, Bohua Chen, Yemin Shi, and Chengwei Pan. Agfsync: Leveraging ai-generated feedback for preference optimization in text-to-image generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 1746–1754, 2025.

[114] 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*, 2024.

[115] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv*, 2021.

[116] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *NIPS*, 2023.

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

[118] 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*, 2025.

[119] Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*, 2025.

[120] Heng Zhou, Hejia Geng, Xiangyuan Xue, Li Kang, Yiran Qin, Zhiyong Wang, Zhenfei Yin, and Lei Bai. Reso: A reward-driven self-organizing llm-based multi-agent system for reasoning tasks. *arXiv preprint arXiv:2503.02390*, 2025.

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

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

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

[124] Huajie Tan, Yuheng Ji, Xiaoshuai Hao, Minglan Lin, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Reason-rft: Reinforcement fine-tuning for visual reasoning. *arXiv*, 2025.

[125] Li Kang, Xiufeng Song, Heng Zhou, Yiran Qin, Jie Yang, Xiaohong Liu, Philip Torr, Lei Bai, and Zhenfei Yin. Viki-r: Coordinating embodied multi-agent cooperation via reinforcement learning. *arXiv preprint arXiv:2506.09049*, 2025.

[126] Huajie Tan, Sixiang Chen, Yijie Xu, Zixiao Wang, Yuheng Ji, Cheng Chi, Yaoxu Lyu, Zhongxia Zhao, Xiansheng Chen, Peterson Co, et al. Robo-dopamine: General process reward modeling for high-precision robotic manipulation. *arXiv preprint arXiv:2512.23703*, 2025.

[127] Jiayuan Zhang, Kaiquan Chen, Zhihao Lu, Enshen Zhou, Qian Yu, and Jing Zhang. Prune4web: Dom tree pruning programming for web agent. *arXiv preprint arXiv:2511.21398*, 2025.

[128] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. *arXiv*, 2025.

[129] Yufei Zhan, Yousong Zhu, Shurong Zheng, Hongyin Zhao, Fan Yang, Ming Tang, and Jinqiao Wang. Vision-r1: Evolving human-free alignment in large vision-language models via vision-guided reinforcement learning. *arXiv*, 2025.

[130] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

[131] Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv*, 2025.

[132] Zirui Song, Guangxian Ouyang, Mingzhe Li, Yuheng Ji, Chenxi Wang, Zixiang Xu, Zeyu Zhang, Xiaoqing Zhang, Qian Jiang, Zhenhao Chen, et al. Maniplvm-r1: Reinforcement learning for reasoning in embodied manipulation with large vision-language models. *arXiv preprint arXiv:2505.16517*, 2025.

[133] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Mitigating hallucination in large multi-modal models via robust instruction tuning. In *ICLR*, 2024.

[134] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024.

[135] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

[136] Justin Lazarow, David Griffiths, Gefen Kohavi, Francisco Crespo, and Afshin Dehghan. Cubify anything: Scaling indoor 3d object detection. *arXiv preprint arXiv:2412.04458*, 2024.

[137] Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, et al. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *CVPR*, 2024.

[138] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023.

[139] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

[140] Yuhao Lu, Yixuan Fan, Beixing Deng, Fangfu Liu, Yali Li, and Shengjin Wang. Vl-grasp: a 6-dof interactive grasp policy for language-oriented objects in cluttered indoor scenes. In *IROS*, 2023.

[141] 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*, 2025.

[142] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv*, 2024.

[143] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In *CoRL*, 2024.

[144] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025.

[145] Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. Foundationstereo: Zero-shot stereo matching. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5249–5260, 2025.

[146] Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025.

[147] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv*, 2021.

[148] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.

[149] Johanna Wald, Torsten Sattler, Stuart Golodetz, Tommaso Cavallari, and Federico Tombari. Beyond controlled environments: 3d camera re-localization in changing indoor scenes. In *ECCV*, 2020.

[150] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[151] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, 2019.

[152] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, 2019.

[153] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017.

[154] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[155] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *CVPR*, 2024.

[156] Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *arXiv*, 2025.

[157] Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *arXiv*, 2025.

[158] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *ICML*, 2023.

[159] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: Please check Sec .1.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Please check Appx. F.

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Please see Sec. 3.4 and Appx. A/B/C/D for details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

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

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

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

   Answer: [No]

   Justification: Since we are still organizing the dataset and related code, we will open source them as soon as possible. In Appx. A/B/C/D, we try to explain how to access the raw data, preprocessed data, generated data, and details about all experiments.

   Guidelines:

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

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Please see Sec. 4 and Appx. D

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: All our experiments involve VLMs. With the temperature set to 0, the output becomes deterministic, eliminating variability and thus making error bars unnecessary. Meanwhile, measuring the inference cost of various VLMs is computationally expensive.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Please check Appx. D.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Please check Appx. C and Appx. D.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Please check Appx. G.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper poses no such risks.

Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Please see Appx. I.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: Please see Appx. C and Appx. I.

    Guidelines:

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

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

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

Answer: [No]

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

Guidelines:

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

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

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

Answer: [No]

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

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: We declare the LLM usage in Appx .A.

Guidelines:

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

## Appendix Table of *RoboRefer*

# A Discussion

**Distinction from SpatialRGPT.** Our approach differs from SpatialRGPT in 4 key aspects: **(1) Task Setting**: Our model addresses a more challenging spatial referring task, where takes a spatially constrained textual instruction as input. It requires multi-step spatial reasoning with learned spatial knowledge to precisely localize the referred object as a 2D point step-by-step. In contrast, SpatialRGPT addresses a simpler VQA task and relies on externally provided region information as input for specific object referring. **(2) Model Usage**: Unlike SpatialRGPT, which needs additional masks or detection tools to generate masks or 2D boxes as inputs for object reference and simplify referring tasks, our model can use textual descriptions for object referencing (see L283), which better aligns with real-world robotic applications. **(3) Data Pipeline**: Our data pipeline adopts a more structured, progressive design than SpatialRGPT. It first uses 2D image data to teach core spatial concepts and general depth perception across diverse indoor and outdoor scenes. Next, accurate 3D data enhances fine-grained spatial understanding in indoor settings for robotics. Finally, simulation data introduces multi-step spatial referring with reasoning. This staged approach yields stronger spatial understanding and reasoning than SpatialRGPT, which relies solely on data generated from 2D images and lacks precise spatial perception for more complex spatial referring tasks. **(4) Training Pipeline**: Our training pipeline includes process-based RFT after SFT, further to improve multi-step reasoning and generalization for spatial referring tasks, whereas SpatialRGPT is trained with SFT only.

**Justification for RFT.** RFT brings two main benefits: **(1) Generalization to Unseen Cases**: In Tab. 2 in the main paper (RefSpatial-Bench Unseen raw), which features novel combinations of spatial relations absent from *RefSpatial* dataset, our 2B-RFT model surpasses 2B-SFT by 9.1% in accuracy, showing the strong generalization enabled by the RFT stage. **(2) Enhance Multi-Step Reasoning Ability**: In the Tab. 8, the RFT-based model consistently outperforms the SFT-based model across varying reasoning steps, especially at larger steps, showing the RFT stage's effectiveness in enhancing multi-step reasoning.

**Why is NVILA chosen as the backbone?** In Tab 1 in the main paper, NVILA outperforms other open-source VLMs under comparable model scales, such as Qwen 2.5-VL (even 72B), in spatial understanding. Enhancing a strong baseline with our dataset and training strategy further validates their effectiveness. Notably, our dataset is model-agnostic and transferable to other backbones. Despite partial training on the *RefSpatial* dataset, Qwen2.5-VL-7B still shows notable improvements on spatial understanding benchmarks in the Tab. 9 below.

**Depth-to-3D mapping assumption.** The depth-to-3D mapping assumption is essential in our real-world evaluation, as our model predicts only 2D image-plane points, while real-world tasks typically require 3D coordinates for grasping, placement, or navigation. While depth noise and partial observations are important real-world challenges, our setting follows prior work [5, 45], which assumes that accurate depth-to-3D mapping is feasible given known camera intrinsics and extrinsics—sufficient for common manipulation and navigation tasks. Moreover, these challenges can be effectively mitigated via the following strategies: (1) Depth noise can be mitigated by recent advances in monocular depth estimation [68], monocular geometry prediction [144], and stereo methods [145]. In cases of severe noise, we employ FoundationStereo [145] in real-world settings to mitigate this issue. (2) Partial views are mitigated in our method by leveraging pixel-level target points. Further improvement is possible by incorporating RoboRefer as a spatially-aware planner for active perception.

**The effect of depth noise on the model's accuracy and robustness.** In real-world experiments, we utilize a relative depth estimation model, DepthAnything v2, to obtain relative depth as the model's depth input, thereby effectively reducing depth noise from a real camera. We also evaluate success rates under depth noise in real-world settings (see Tab below). Depth maps generated from the strong monocular relative depth estimation model (*i.e.*, DepthAnything V2) offer the highest robustness and success. Despite depth noise from a real camera, RoboRefer maintains great performance by leveraging RGB priors due to mixed RGB and RGB-D training during the SFT stage.

Table 8: We report the success rates (%) of 2B-SFT and 2B-RFT model at each reasoning step on *RefSpatial-Bench*.

| Benchmark | Reasoning Step Num. | 2B-SFT | 2B-RFT | Gain |
|---|---|---|---|---|
| RefSpatial-Bench-Location | Step 1 | 63.33 | 66.67 | +3.34 |
| | Step 2 | 39.58 | 43.75 | +4.17 |
| | Step 3 | 27.27 | 36.36 | +9.09 |
| | **Total** | **47.00** | **52.00** | **+5.00** |
| RefSpatial-Bench-Placement | Step 2 | 55.56 | 55.56 | +0.00 |
| | Step 3 | 41.67 | 41.67 | +0.00 |
| | Step 4 | 41.67 | 45.83 | +4.16 |
| | Step 5 | 0.00 | 25.00 | +25.00 |
| | **Total** | **48.00** | **54.00** | **+6.00** |

# B Implementation Details and Samples of *RefSpatial* Dataset

In this section, we present a comprehensive exposition of the implementation details and representative data samples underpinning the construction of the *RefSpatial* dataset. As this dataset is intended to equip general VLMs with the ability to adapt to spatial referring tasks, thereby enhancing spatial understanding and reasoning in a bottom-up manner, we meticulously design a multi-data-source generation pipeline. We elaborate on the three core components of this pipeline as follows:

• **2D Web Image (Appx. B.1)**: We present a 2D data pipeline comprising image filtering, pseudo-3D scene graph construction, hierarchical referential description generation—from coarse categories to fine-grained spatial referents—and diverse QA pair creation.

• **3D Embodied Video (Appx. B.2)**: This section outlines the 3D data selection process from CA-1M [136], discusses its limitations and mitigation strategies, and presents methods for enriched scene graph construction compared to the 2D data source. We further describe a QA generation framework that leverages detailed 3D annotations (*e.g.*, depth maps, oriented 3D bounding boxes) to capture richer spatial relations. Finally, we detail how to generate QA pairs for the problem of "feasibility assessment for object placement in free space".

• **Synthetic Data from Simulator (Appx. B.3)**: We describe how to synthesize 3D scenes, select and annotate digital assets, efficient scene assembly and rendering, and the generation of QA pairs grounded in these simulated scenes.

In the following subsections of each section, we detail the employed models, prompt design rationale, data processing steps, filtering criteria, and illustrative examples, providing a clear and thorough overview of the construction pipeline and core technical details of the *RefSpatial* dataset.

## B.1 2D Web Image

### B.1.1 Multi-Stage Image Filtering

2D Web Images aim to endow the model with basic spatial concepts and comprehensive depth perception across both indoor and outdoor scenes. Here we use OpenImage [135] as 2D data source.

**Overall Motivation and Goals for Filtering.** The OpenImages dataset offers a vast collection of 2D internet images (1.7 M images in training split) with extensive visual diversity, but many images, such as text-only graphics, QR codes, medical scans, or abstract art, lack relevance for spatial

https://storage.googleapis.com/openimages/web/index.html

Table 9: Performance on the *single-step spatial understanding* benchmarks across different model types. Top-1 & Top-2 accuracies are represented using **bold text**, and underlines.

| Method | CV-Bench [15] | | | BLINK$_{val}$ [16] | | RoboSpatial [2] | SAT [4] | EmbSpatial [22] |
|---|---|---|---|---|---|---|---|---|
| | 2D-Relation | 3D-Depth | 3D-Distance | 2D-Relation | 3D-Depth | | | |
| Qwen-2.5-VL-7B (base) | 82.15 | 60.17 | 69.00 | 64.34 | 60.98 | 49.59 | 30.00 | 40.20 |
| Qwen-2.5-VL-7B (finetuned) | 95.85 | 95.00 | 90.83 | 83.22 | 84.68 | 69.92 | 85.75 | 76.32 |
| NVILA-8B (base) | 91.54 | 91.83 | 90.67 | 76.92 | 76.61 | 59.35 | 63.33 | 67.72 |
| RoboRefer-8B-SFT (finetuned) | 96.90 | 98.33 | 93.50 | 91.61 | 92.74 | 84.55 | 86.67 | 72.53 |

Table 10: We report the success rates (%) of real-world evaluation performance when using depth from DepthAnything V2 and Real Camera.

| **Real-world Task** | **Depth from DepthAnything V2** | **Depth from a Real Camera** |
|---|---|---|
| Pick the specific hamburger closest to the mug nearest to the camera. | 80 | 70 |
| Place the hamburger in front of the teddy bear. | 90 | 90 |
| Pick the apple in front of the leftmost cup's logo side. | 80 | 80 |
| Place the apple aligned with the existing apple row. | 60 | 40 |

understanding and referring with reasoning. To curate a dataset tailored for these tasks, we employ a two-stage filtering pipeline: a coarse pre-filtering using SigLIP2 [74], followed by fine-grained selection via Qwen2.5-VL [11], to retain images rich in spatial semantics.

**Stage 1: Initial Coarse Filtering. We employ the `siglip2-giant-opt-patch16-384` model for initial filtering to efficiently discard low-quality or off-theme images (*e.g.*, irrelevant scenes or content lacking multiple everyday objects).** This step greatly reduces data volume, streamlining subsequent processing. Specifically, the SigLIP2 model is guided by predefined positive and negative textual labels. Positive labels represent desired image characteristics, while negative labels describe undesired content. For each image, the model computes cosine similarity between its embedding and all label embeddings. The label with the highest similarity is selected; if it belongs to the positive set, the image is retained, otherwise discarded. Label sets are manually refined iteratively to balance recall and precision, ensuring relevance while excluding noise. These labels act as semantic anchors for visual-text alignment. In this stage, only 934k images are qualified to be retained. Positive and negative label lists are provided in Listings 1 and 2. For more details about the compute resources needed in this stage, please see Appx. E.1.

Listing 1: Positive Labels used during SigLIP2 filtering.

```
Positive Labels = [
    "Mid-distance observation of some objects on a table",
    "Some objects on the desktop",
    "Distant view of some animals",
    "Mid-distance observation of some animals",
    "Distant view of one object",
    "Mid-distance observation of one object",
    "Distant view of some objects",
    "Mid-distance observation of some objects",
    "Distant view of a person",
    "Mid-distance observation of a person",
    "Distant view of some people",
    "Mid-distance observation of some people",
    "Distant view of indoor scene",
    "Distant view of outdoor scene",
    "Distant view of traffic",
    "Distant view of Urban architecture"
]
```

Listing 2: Negative Labels used during SigLIP2 filtering.

```
Negative Labels = [
    "Macro shot of an animal",
    "Macro shot of one object",
    "Macro shot of a person",
    "Macro shot of flowers",
```

```
    "A piece of text",
    "A person displayed in front of a white background",
    "A product displayed in front of a white background",
    "A screenshot of the graphics user interface",
    "A dimly lit environment"
]
```

SigLIP2 preserves images rich in object diversity, depth cues, and scene context (indoor/outdoor) through the above labeling process, but its performance declines on certain image types, including:

1. **Paintings/Artworks**: Especially those with visible brushstrokes or canvas textures.
2. **Low-Light Scenes**: Images with minimal illumination and strong shadows.
3. **Grayscale Photographs**: Black-and-white imagery lacking color cues.
4. **Distorted Images**: Those exhibiting warping, mirroring, or other geometric anomalies.
5. **Multi-Scene Collages**: Images containing three or more distinct scenes with hard borders.

SigLIP2 struggles to detect and interpret these categories reliably, highlighting the need for a secondary, fine-grained filtering stage.

**Stage 2: Fine-grained Filtering** Due to SigLIP2's limitations in handling certain visual content mentioned above, **we introduce a fine-grained filtering stage using the Qwen2.5-VL-7B model to further improve dataset quality. This step ensured the final images are clear, authentic, and well-suited** for spatial understanding and reasoning required for spatial referring tasks. Qwen2.5-VL processed 934k images filtered by SigLIP2, retaining 846k. Although Qwen2.5-VL offers higher filtering precision, its slower speed necessitated the use of SigLIP2 for initial fast filtering, significantly improving overall efficiency. To ensure accurate and consistent fine-grained filtering, we adopt a structured prompt engineering strategy for the Qwen2.5-VL model. The process begins with a system prompt (See Listing 3) that defines the model's role as an image analysis expert, specifying key visual attributes to assess and negative categories to detect, and enforcing a strict workflow. For each image, a corresponding user prompt (See Listing 4) instructs the model to determine whether the image belongs to any predefined negative categories. The model's response follows a structured format: if the segment after the pipe symbol (|) is "Yes", the image is classified as negative and discarded; otherwise, it is retained. This prompting scheme ensures that the model adheres to a consistent output, enhancing the reliability of filtering outcomes. Please refer to Appx. E.1 for details on the computational requirements of this stage.

Listing 3: System Prompt for Qwen2.5-VL-7B filtering.

```
system_prompt = """
You are an image analysis expert. Follow this workflow rigidly:

1. **Content Analysis**:
   - Inspect: Main subjects, artistic style, visual characteristics
   - Check: Lighting intensity, color channels, geometric integrity,
      composition structure

2. **Category Verification** (YES if matches ANY):
   a) Painting/Artwork - Visible brushstrokes/canvas texture
   b) Dim Lighting - Very low brightness, heavy shadows
   c) B&W Photo - Grayscale only (0 color channels)
   d) Distorted Image - Warping/mirroring anomalies
   e) Multi-image Collage - >=3 distinct scenes with hard borders

3. **Structured Response**:
   Output EXACTLY in this format:
   "[Analysis sentence]. | Yes/No"
   - Analysis must contain observable evidence
   - Final answer MUST use pipe separator

Examples of VALID responses:
    "This image is a composite created by stitching together multiple
       smaller images, with distinct white borders visible between
       the individual components. | Yes"
```

```
    "This image features vibrant colors, is neither an artistic
       painting nor a composite of multiple images, and does not
       conform to any of the specified categories. | No"
"""
```

Listing 4: User Prompt for Qwen2.5-VL-7B filtering.

```
user_prompt = """
Analyze if this image belongs to ANY of these categories:
1. Painting/artwork
2. Dim lighting
3. Black-and-white
4. Geometric distortion
5. Multi-image collage

Respond EXACTLY FORMATTED as:
"[Your evidence-based analysis]. | Yes/No"
"""
```

**Visualizing Overall Filtering Results.** Fig. 6 presents a visual overview, showing the effectiveness of our multi-stage filtering pipeline. The first row shows the output of SigLIP2, which effectively removes images lacking spatial semantics, such as macro shots of animals, people, or flowers, textual content, and GUI screenshots. The second row demonstrates how Qwen2.5-VL-7B further eliminates unsuitable categories, including artworks, dimly lit scenes, black-and-white images, geometrically distorted content, and image collages. The third row displays the retained images after both stages, which exhibit rich spatial relationships, confirming their suitability for spatial understanding and reasoning and the overall quality of our dataset.

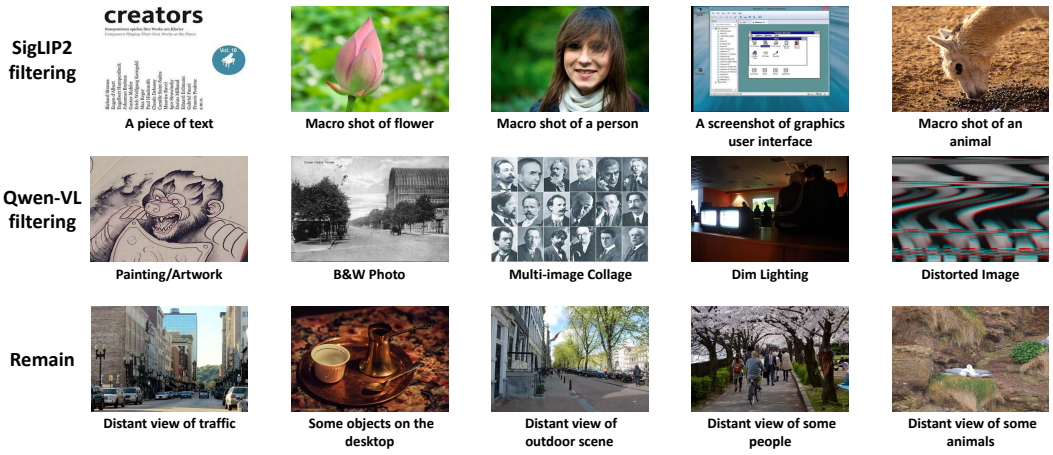



Figure 6: Visual overview of the multi-stage filtering results. Row 1: Images discarded by SigLIP2 due to insufficient spatial context (*e.g.*, close-ups, text). Row 2: Additional filtering by Qwen2.5-VL removes non-natural content (*e.g.*, artwork, collages). Row 3: Remaining high-quality images suitable for spatial understanding and referring.

### B.1.2 Pseudo-3D Scene Graphs Construction

Although filtered 2D images contain some spatial cues, QA pairs containing sufficient 3D spatial information (*e.g.*, "near" *vs.* "far", distances) derived directly from these 2D images are challenging. Inspired by prior work [1, 6], we construct pseudo-3D scene graphs from 2D images to enhance the generation of QA pairs with rich 3D spatial semantics. In these graphs, nodes represent object attributes, while edges encode inter-object spatial relations. We detail the process of converting 2D images into pseudo-3D scene graphs below.

**Object Detection and Annotation.** Although the OpenImages dataset provides annotations, its limited vocabulary and coarse labeling are insufficient for open-world scenarios. To address this, we

leverage state-of-the-art foundation models for enhanced object detection and annotation. Specifically, our scene graph construction pipeline integrates the Recognize Anything Model (RAM) [66] and GroundingDINO [101] to assign semantic labels and bounding boxes to key objects in filtered raw 2D images. The workflow is broadly as follows:

1. **Semantic Labeling via RAM**: RAM analyzes each image to generate category labels for all recognized objects. Its broad recognition capability ensures comprehensive semantic coverage, guiding subsequent localization.
2. **Bounding Box Localization via GroundingDINO**: The labels from RAM are used as text prompts for GroundingDINO, an open-vocabulary detector that localizes the corresponding objects and outputs precise bounding boxes.

**Although recent VLM, *i.e.*, Florence-2-Large supports instruction-based recognition and detection simultaneously, we find that combining RAM with GroundingDINO yields superior results.** In Fig. 7, Florence-2 (top) often produces ambiguous or redundant detections (*e.g.*, vague labels, multiple boxes for a single object, or single boxes covering multiple objects), which are unsuitable for precise object referring. In contrast, GroundingDINO+RAM (bottom) generates concise labels and one-to-one object-bounding-box mappings, better satisfying the requirements of referring tasks.

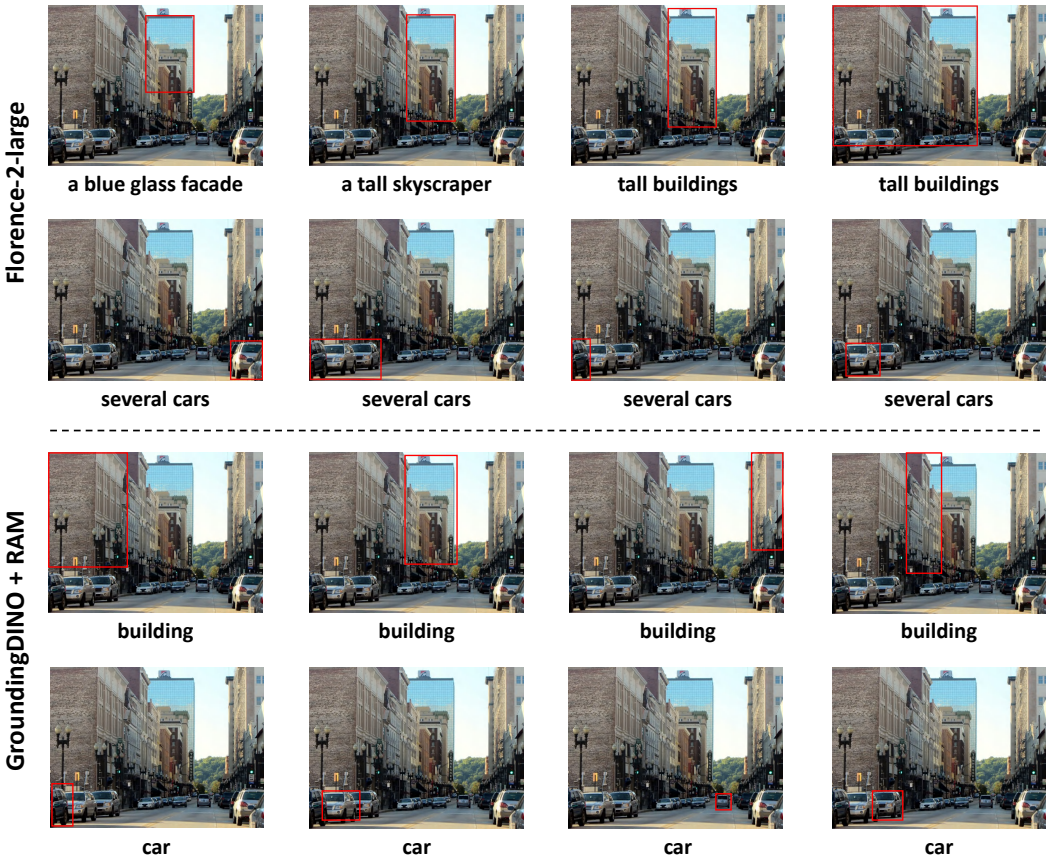



Figure 7: Object detection comparison: Florence-2 (above), GroundingDINO+RAM (below)

**3D-aware information Extraction.** To further extract 3D-aware information from 2D images, we adopt UniDepth V2 [68] for metric depth estimation due to its recent state-of-the-art performance, surpassing models such as DepthPro and Metric3Dv2 across multiple benchmarks. For camera intrinsic prediction, we employ WildeCamera [76]. Together, these models enable robust 3D point cloud reconstruction of the scene. Based on previously annotated object bounding boxes, we apply SAM 2.1 [78] to generate instance masks. Each resulting Pseudo-3D scene graph comprises object labels (via RAM), 2D bounding boxes (via GroundingDINO), instance masks (via SAM 2.1),

and object-level point clouds (via UniDepth v2 with WildeCamera), resulting in **axis-aligned 3D bounding boxes**. A visualization is provided in Fig. 8.

Figure 8: Scene graph visualization with image, detected objects, and corresponding point clouds.

### B.1.3 Hierarchical Object Description Generation

While 3D scene graphs encode basic object categories, real-world scenes often contain multiple instances per category. Prior datasets [2, 3] simplify reference by assuming a single object per category, limiting their utility in spatial referring tasks. To overcome this, **we augment object descriptions with attributes and spatial relations, enabling finer-grained disambiguation among similar instances of the same category.** We present our two-stage generation pipeline below.

**Stage 1: Generating Object and Image Dense Descriptions in image space.** We first generate detailed descriptions for each detected object and the entire image, employing the Qwen2.5-VL-7B model. This stage involves both object-level captioning and comprehensive image-level captioning. The global captions are essential for providing contextual grounding to downstream large language models (QwQ-32B) during LLM QA generation (detailed in Appx. B.1.4), enhancing the relevance and accuracy of the outputs. Prompt templates are detailed in Listings 5 and 6. In particular, the `object_caption_user_text_prompt` uses a dynamic placeholder `[class_name]`, which is filled with the object category predicted by the RAM model (See Appx. B.1.2).

Listing 5: Prompts for Image Caption Generation with Qwen-VL.

```
image_caption_system_text_prompt =  """
    You are an expert image analysis assistant. Your task is to
        generate a detailed and comprehensive description of the image
        .
    Please focus on accurately capturing all visual elements present
        in the image, including objects, scenery, colors, shapes,
        textures, and lighting.
    Your description should be clear, precise, and professional.
        Additionally, ensure that your description begins with either
        'this image' or 'the image'.
"""

image_caption_user_text_prompt = """
    Please carefully examine the provided image and generate a
        detailed description.
    Include all visible elements such as objects, scenery, colors,
        shapes, textures, and lighting.
    Ensure that your description is thorough, accurate, and complete,
        and that it starts with either 'this image' or 'the image'.
"""
```

Listing 6: Prompts for Object Caption Generation with Qwen-VL.

```
object_caption_system_text_prompt = """
    You are a visual localization analyzer working with TWO distinct
        images:
    1. [POSITION-REFERENCE] (First Image):
    - Contains ONLY location clues with background
    - Strictly use ONLY for determining spatial position (left/right/
        upper/lower/center)
    - Ignore all visual features except object placement

    2. [DETAIL-SOURCE] (Second Image):
    - Shows the object's TRUE APPEARANCE without background
    - Extract EXCLUSIVELY from this: color, texture, material, shape
    - Never infer details from the first image

    Generate phrase in pattern: [Color][Material][Object] at [Position
        ]
    Example: "Matte black laptop on the left" NOT "Red-boxed laptop"
"""

object_caption_user_text_prompt = """
    For the [class_name] marked by red box in FIRST image and fully
        shown in SECOND image:
    -> COLOR/MATERIAL: Must come from SECOND image
    -> POSITION: Only from FIRST image's placement
    Forbidden actions:
    x Mention 'red box' or background elements
    x Use location terms in second image
    x Combine features across images

    Describe the [class_name] marked by red box in FIRST image and
        fully shown in SECOND image with this format:
    [Color][Material/Texture][Object] at [Position]
    Samples:
    - "Brushed metal water at bottle left"
    - "Glossy ceramic mug at upper center"
    - "Faded denim jacket at lower right"
"""
```

**Stage 2: Generating Object Description with Spatial Cues.** To enhance referential specificity in object captions, particularly when multiple instances of the same category coexist, we adopt a heuristic strategy that appends spatially indicated relative information, such as "the third chair from the front". This method leverages 3D object positions from the scene graph (See Appx. B.1.2). By comparing same-category objects along the three principal axes (front–back, left–right, top–bottom), we identify the axis with the largest spatial variation as the primary arrangement direction to guide relative spatial reference generation. Once the main sorting axis is identified, we retrieve appropriate templates from a predefined library (see Listing 7) to augment the initial object descriptions. These templates are designed to capture diverse natural language patterns. For instance, for a row of chairs arranged left to right, templates may include: "{dense_caption}, which is the {ordinal} {class_name} from left to right," or "{dense_caption}, the {ordinal} {class_name} in the left-to-right sequence". Here, `dense_caption` denotes the initial description generated by the Qwen2.5-VL model, `ordinal` indicates the object's position in the sorted sequence, and `class_name` is the category label predicted by RAM. This spatially-aware enhancement is applied only when multiple instances of the same category are detected to avoid redundancy. If an object appears only once, its original dense caption is used directly. To ensure spatial diversity, we set a variance threshold across the three principal axes; images with multiple same-category objects but low variance on all axes are discarded, resulting in a final set of 466k images. By integrating spatial ordering with visual descriptions, this heuristic enables the generation of precise and discriminative referential expressions, essential for producing high-quality, unambiguous question-answer pairs.

Listing 7: Templates for Spatial Order Description Enhancement.

```
TEMPLATES = {
```

```
    "left_to_right": [
        "{dense_caption}, which is the {ordinal} {class_name} from
           left to right",
        "{dense_caption}, marked as the {ordinal} {class_name} in a
           left-to-right arrangement",
    ],
    "right_to_left": [
        "{dense_caption}, the {ordinal} {class_name} viewed from the
           right",
        "{dense_caption}, the {ordinal} {class_name} from the right",
    ],
    "front_to_back": [
        "{dense_caption}, which appears as the {ordinal} {class_name}
           when viewed from the front",
        "{dense_caption}, positioned as the {ordinal} {class_name} in
           front-to-back order",
    ],
    "back_to_front": [
        "{dense_caption}, which is counted as the {ordinal} {
           class_name}, starting from the back",
        "{dense_caption}, the {ordinal} {class_name} in the back-to-
           front sequence",
    ],
    "top_to_bottom": [
        "{dense_caption}, the {ordinal} {class_name} viewed from the
           top",
        "{dense_caption}, placed as the {ordinal} {class_name} when
           sorted from top to bottom",
    ],
    "bottom_to_top": [
        "{dense_caption}, which ranks as the {ordinal} {class_name} in
            bottom-to-top order",
        "{dense_caption}, arranged as the {ordinal} {class_name} when
           ordered from the bottom",
    ]
}
```

**Examples of Object and Image Descriptions** This part qualitatively shows the representative examples with the generated descriptions. As shown in Fig. 9, we present two types of object captions. The top row shows simple captions produced by Qwen2.5-VL for single-instance object categories, where spatial ordering is unnecessary. The bottom row includes captions augmented with spatial order information to distinguish multiple instances of the same category. Additionally, Fig. 10 demonstrates Qwen2.5-VL's ability to generate detailed global descriptions of entire images used in the following.

### B.1.4 Generating Diverse QA Pairs via Pseudo-3D Scene Graphs

After constructing scene graphs and generating hierarchical object descriptions, we can leverage this information to generate diverse QA pairs from pseudo-3D scene graphs to support SFT training for improved single-step spatial understanding.

**Template, Choice and Fact QA Generation.** We first adopt a template-based method to generate structured preliminary QA pairs, multiple-choice questions, and factual statements. The templates are derived from scene graph information (*e.g.*, object attributes, positions) and refined hierarchical object descriptions. When designing QA templates using pseudo-3D scene graphs from 2D images, we explicitly account for the spatial ambiguity inherent in single-view pseudo-3D representations (*e.g.*, inaccuracies in monocular depth, lack of object orientation, absence of 3D oriented bounding boxes). Consequently, **our QA templates from 2D data source focus mainly on qualitative spatial understanding and reasoning**, while incorporating quantitative cues only when they can be inferred reliably from 2D or pseudo-3D signals. The spatial concepts covered in the QA templates fall into the following categories:

1. **Relative position relations**: capture spatial layouts (left/right, above/below, front/behind).


Figure 9: Generated object descriptions. Top: Unique-category captions. Bottom: Spatially-aware captions for the same categories. Red boxes indicate referenced objects.

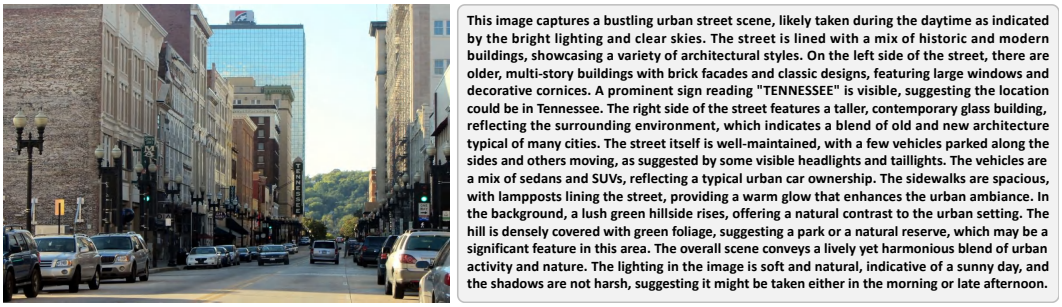



Figure 10: Visualization of generated image detailed descriptions.

2. **Relative size comparisons**: describe object attributes (*e.g.*, bigger/smaller, taller/shorter, wider/thinner) often inferred from image-plane projections.
3. **Quantitative information from 2D or pseudo-3D label**: include spatial reasoning based on estimated depth maps, 2D object coordinates, and coarse monocular depth approximations.

Accordingly, we design diverse QA templates in these types for spatial understanding:

1. Templates assess spatial and size relations:
   - Position relation: ‘‘`Is [A] to the left of [B]?`’’
   - Size comparison: ‘‘`Which object is larger, [A] or [B]?`’’
2. Templates query 2D point of a uniquely identified object ‘[A]’ (defined in Appx. B.1.3):
   - ‘‘`Where is [A] located? Provide its 2D coordinates.`’’ (*e.g.*, “*Where is the red apple at left, which is the second apple from left to right, located? Provide its 2D coordinates.*”)
3. Templates query attributes at a specific 2D point ‘[X]’ (formatted as ‘(x, y)’):
   - Depth retrieval: ‘‘`What is the depth at point [X]?`’’ (*e.g.*, “*What is the depth at (0.528, 0.317)*?”)
   - Object identification: ‘‘`Which object is at point [X]?`’’ (*e.g.*, *“Which object is at (0.753, 0.839)?”*)

We further design fact templates to generate declarative statements, forming a structured basis for prompting Reasoning LLM to produce richer and more natural QA pairs. Example templates include:

1. Approximate depth: ‘‘Point [A] and the camera are [X] apart.’’ (based on depth estimation).
2. Precise 2D object location: ‘‘[A] is located at point [X].’’, ‘‘[A] is to the right of [B].’’

**Reasoning QA Generation.** To generate more natural, complex, and diverse QA pairs beyond templated formats, we leverage a powerful reasoning LLM, QwQ-32B [146]. It takes the factual statements, initial QA pairs, and multiple-choice questions (if available) as input, along with global image captions and precise object descriptions. QwQ-32B then produces more challenging and conversational spatial reasoning QA. The prompt design is shown in Listing 8.

Listing 8: Prompt for QwQ-32B QA Diversification and reasoning QA Generation.

```
"""
You are a helpful assistant tasked with generating spatial reasoning-
    based questions and answers from provided descriptions of scenes.

Rules:
1. **We have three types of input information**:
- **[Scene]**: A general description of the entire image, which
    provides context for the objects and their surroundings.
     **Example:**
     [Scene]: The image shows a tranquil lakeside with a small wooden
         dock on the right and calm, reflective water in the center.
         The sky is overcast.

- **[Objects]**: A list containing one or more object labels separated
     by "|".
     **Example:**
     [Objects]: teal glossy water at lower center | green bamboo dock
         at lower right.

- **[Objects Description]**: Provides spatial or comparative details
    between those objects.
     **Example:**
     [Objects Description]: teal glossy water at lower center is taller
          than green bamboo dock at lower right.


2. **When crafting a Question**:
- **Always use the provided [Scene] description as context** to ensure
     the question aligns with the overall image.
- **Mention all object labels from [Objects]** in the question.
- **Do not modify or paraphrase the object labels**; they must appear
    **exactly** as given in ‘[Objects]’.
- **Do not assume or invent additional scene details** beyond what is
    provided in ‘[Scene]’.
- **Do not reveal the specific details in [Objects Description]** (
    like which object is taller, shorter, wider, etc.).
- Always generate questions related to the description using the
    object labels from [Objects].
- Each object label in ‘[Objects]’ **must appear exactly once** in the
     Question.
- The question should read from **an observer’s perspective**.
- The description should always be used to answer and not leak into
    the question.

3. **When crafting an Answer**:
- **Mention at least one object label from [Objects]** in the answer.
- **Use the ‘[Objects Description]’ to provide a correct answer**.
- **Ensure the answer is concise, factual, and directly related to the
     provided ‘[Scene]’ and ‘[Objects]’**.
- **You may restate or summarize the relevant details from ‘[Objects
    Description]’, but do not introduce new assumptions**.
```

```
Here's several examples:

[Scene]: The image depicts a modern living room with a large window
    allowing warm sunlight to enter. The room has a wooden floor, a
    patterned rug in the center, and a coffee table with a few
    magazines neatly stacked on it. A yellow leather sofa is
    positioned centrally, facing the television mounted on the
    opposite wall. To the left of the sofa, a black metal chair with a
     cushioned seat is placed beside a tall bookshelf filled with an
    assortment of books and decorative items. The furniture
    arrangement leaves an open pathway between the sofa and the chair.
[Objects]: yellow leather sofa at lower center, black metal chair on
    the left.
[Objects Description]: The path between yellow leather sofa at lower
    center and black metal chair on the left is 1.5 meters.
"Question": You are a cleaning robot that is 1 meter wide. Now you are
     standing in a living room and see the image; you want to move
    from here to the door that leads to the backyard. Do you think you
     can go through the path between the yellow leather sofa at lower
    center and the black metal chair on the left?
"Answer": The path between the yellow leather sofa at lower center and
     the black metal chair on the left is 1.5 meters, so yes, the
    robot can go through the path between the yellow leather sofa at
    lower center and the black metal chair on the left since it is
    wider than the robot's width.

[Scene]: The image showcases a modern kitchen with a wooden countertop
     that extends across the space, separating the cooking area from
    the dining area. On the left side of the countertop, a fruit bowl
    holds a variety of fresh produce. A red fresh apple is placed on
    the left side of the bowl, while a bright fresh orange sits neatly
     on the right side. Behind the fruit bowl, a glass pitcher filled
    with orange juice and a stack of white ceramic plates are visible.
     Natural light streams in from a large window above the sink,
    reflecting off the stainless steel appliances and giving the space
     a bright, clean feel.
[Objects]: red fresh apple on the left, fresh orange on the right.
[Objects Description]: red fresh apple on the left is positioned on
    the left side of fresh orange on the right.
"Question": You see two fruits, a red fresh apple on the left and a
    fresh orange on the right. Which one is more on the left side?
"Answer": The red fresh apple on the left is more on the left.

Now its your turn!
"""
```

**Training data visualization.** For specific examples of training data generated from 2D web images and their visualizations, please refer to Appx. F, which contains detailed sample presentations.

### B.2 3D Embodied Video

#### B.2.1 Why Use CA-1M and How to Pre-process It

**Rationale for Selecting CA-1M as the 3D Data Source.** To enable fine-grained spatial reasoning in indoor environments, we adopt Apple's open-source CA-1M [136] dataset as our primary 3D data source. CA-1M aligns closely with our objectives due to the following key attributes:

1. Dense 2D/3D Annotations: CA-1M provides per-frame 2D/3D oriented bounding boxes, enabling spatial localization (*e.g.*, 3D spatial occupancy) and accurate interaction modeling.
2. Comprehensive Camera and Depth Data: The inclusion of camera intrinsics, extrinsics, and depth maps supports accurate 3D reconstruction and geometric reasoning.

https://github.com/apple/ml-cubifyanything

3. Large-Scale Coverage: Its extensive volume enables training/evaluation of VLMs at scale.

**These features offer a strong foundation for constructing 3D scene graphs and generating spatial reasoning data involving 3D geometry, object interactions, and egocentric understanding.**

**Comparative Analysis with Alternative 3D Datasets.** While several high-quality 3D datasets exist, they fall short of meeting the specific requirements of our project, motivating our selection of CA-1M:

1. **ARKitScenes** [147]: As a predecessor to CA-1M, it provides only global 3D bounding boxes without per-frame 3D or 2D annotations. Projecting these 3D boxes to 2D yields oversized and contain irrelevant objects. Additionally, its annotations are less comprehensive, and image resolution is also much lower compared to CA-1M.
2. **ScanNet V2** [148]: Lacks 3D bounding boxes, making object orientation estimation infeasible. Although EmbodiedScan introduces per-frame 3D bounding boxes, it still lacks 2D annotations, and projected 2D boxes remain imprecise and contain irrelevant objects.
3. **3RScan** [149]: Suffers from low image quality, hindering spatial information extraction and limiting its usability.

In summary, despite some limitations (See Appx. B.2.2), CA-1M provides large-scale egocentric video, per-frame 2D/3D annotations, depth, and camera parameters, **making CA-1M the most suitable choice for generating rich and complex 3D spatial data from an embodied perspective**.

**Pre-processing.** Video datasets capturing continuous activities, such as CA-1M, exhibit high temporal redundancy, as consecutive frames often contain near-identical visual content with minor variations. Processing all frames is computationally intensive and yields redundant samples with limited informational gain for model training. To mitigate this, we adopt a frame sampling strategy, selecting one frame every 20 frames. This reduces redundancy while preserving meaningful scene and viewpoint transitions. The resulting subset maintains scene diversity and supports efficient downstream processing, including 3D scene graph construction and question-answer generation. In Fig. 11, the top row shows four consecutive frames before sampling, revealing minimal variation; the bottom row, sampled at 20-frame intervals, demonstrates significantly greater scene variation.


Figure 11: Comparison of frame sequences before and after sampling. Top: Four consecutive frames from the original video. Bottom: Corresponding frames sampled every 20 frames, exhibiting increased temporal variation between adjacent frames.

### B.2.2 Inherent Challenges and Limitations in CA-1M

Although the CA-1M dataset is chosen for its large scale, egocentric perspective, and rich annotations, it presents inherent limitations that undermine scene graph construction if unaddressed.

**Ambiguous or Meaningless Object Annotations.** A major issue is the prevalence of ambiguous or semantically insignificant object annotations. Many instances are difficult to interpret, even for

humans, due to unclear boundaries or a lack of identifiable semantics. For example, some bounding boxes enclose small, indiscernible regions such as a patch of wall or vague background elements (see Fig. 12). Incorporating such annotations into model training introduces noise, hampers spatial understanding, and may mislead the model's perception of object relationships.

**Widespread Absence of Semantic Labels.** Another major limitation of CA-1M is the lack of semantic labels for most annotated objects. Unlike datasets (*e.g.*, ARKitScenes and ScanNet V2), which provide object category annotations, CA-1M includes only a few structural categories (*e.g.*, floors, walls, doors), leaving the majority of object instances unlabeled. Although bounding boxes indicate the presence of objects (*i.e.*, labeled as "object"), their categories (*e.g.*, "chair", "table") remain unknown. This absence of semantic information hinders spatial understanding, making it impossible to generate category-dependent queries such as *"Is the red chair to the left of the table?"*

Figure 12: Example of some meaningless annotations in CA-1M.

### B.2.3 Addressing Limitations: Object Annotation and Bounding Box Filtering

To address the limitations of the CA-1M dataset, particularly the lack of semantic labels and the prevalence of noisy bounding boxes (See Appx. B.2.2 above), we develop a dedicated annotation and filtering pipeline. This enhances the dataset's usability for downstream tasks such as 3D scene graph construction and spatial reasoning. We detail the two-stage pipeline below.

**Stage 1: Initial Annotation and 2D Bounding Box Prediction.** We employ a multi-model combination strategy to annotate unlabelled objects and refine bounding boxes in CA-1M video frames. Specifically, we integrated GroundingDINO, RAM, and Florence-2 to perform object semantic labeling and 2D bounding box prediction. **Our approach yields semantically meaningful and visually coherent object bounding boxes compared to the often difficult-to-discern or ambiguous bounding boxes from the original CA-1M annotations.** To maximize recall, we intentionally lower the confidence thresholds of GroundingDINO and RAM, ensuring the inclusion of low-confidence but potentially relevant objects. These candidates are retained for further validation in subsequent matching and filtering stages.

**Stage 2: Bidirectional 2D Bounding Box Matching.** To associate model-predicted 2D bounding boxes (with semantic labels) with unlabeled boxes in CA-1M, **we propose a bidirectional matching and refinement strategy. This not only enables semantic annotation of meaningful objects but also filters out noisy or ambiguous CA-1M bounding boxes**, thus enhancing the utility of its 3D annotations (*e.g.*, 3D oriented bounding boxes). The strategy consists of two steps:

1. **Matching CA-1M Bounding Boxes to Model Predictions.** We first match original CA-1M 2D bounding boxes to model-predicted bounding boxes based on the IoU metric. Due to the sparsity, occlusions, and fragmentation of annotation, multiple CA-1M boxes may correspond to a single prediction, resulting in many-to-one matches.
2. **Refining Model Predictions via One-to-One Mapping.** To resolve many-to-one matches, we retain only predicted boxes that match at least one CA-1M bounding box. For each, we assign the CA-1M bounding box with the highest IoU as its unique match. This enforces a one-to-one correspondence, eliminating redundant or weakly aligned CA-1M boxes. The result is a refined set of original bounding boxes with strong semantic alignment.

**Visualizing Matching Results.** Fig. 13 shows the effectiveness of our bounding box matching procedure. The first row shows successfully matched CA-1M bounding boxes annotated with RAM-predicted object labels, while the second row highlights unmatched bounding boxes, typically corresponding to ambiguously annotated objects. This demonstrates that our method reliably filters

out uncertain annotations and assigns semantic labels to clearly identifiable objects. For subsequent scene graph generation, we adopt the model-predicted bounding boxes instead of the original CA-1M ones, as they better align with the visible object extents, facilitating more accurate instance mask extraction via models like SAM2.

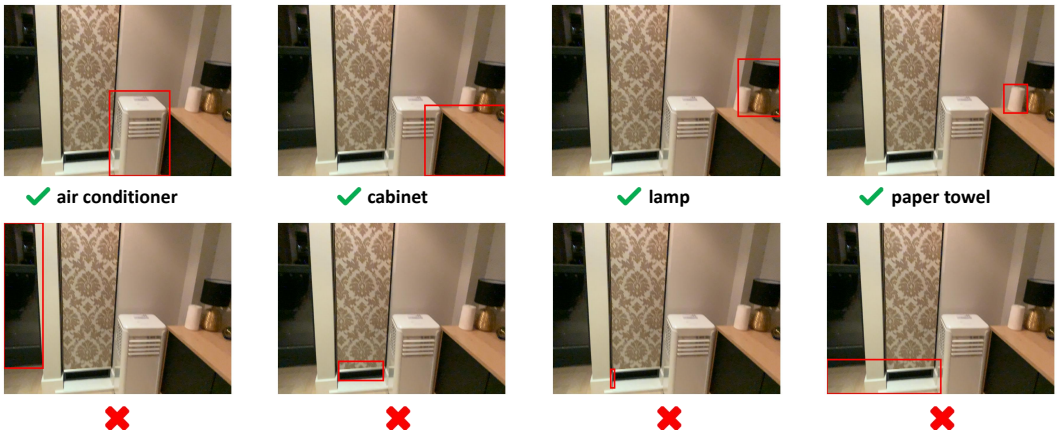



Figure 13: Visual comparison of 2D bounding box matching. The top row shows matched CA-1M bounding boxes with object labels; the bottom row displays unmatched or ambiguous cases.

#### B.2.4 3D Object Description Generation and Scene Graph Construction.

The process of 3D scene graphs construction largely follows the 2D scene graph pipeline in Appx. B.1.2 and the description generation in Appx. B.1.3. The resulting graphs are structurally similar to those from OpenImages in Fig. 8. The key distinction is CA-1M's focus on indoor environments and its provision of high-precision geometric data (*e.g.*, ground-truth depth, camera intrinsics/extrinsics, 3D oriented bounding boxes). **The enhanced 3D scene graph outperforms 2D-based counterparts in object localization and spatial relation accuracy, enabling structurally rich and quantitatively grounded 3D QA data across 28 spatial relation types.**.

#### B.2.5 Free Space QA Generation for Object Placement

A key challenge in spatial referring is identifying unoccupied regions suitable for object placement. We propose a multi-step pipeline to address this.

**Step 1: Detecting Viable Platforms via Qwen2.5-VL.** We first filter out scenes lacking plausible placement surfaces (*e.g.*, only walls or ceilings). To this end, we employ the Qwen2.5-VL-7B, initialized with a system prompt (Listing 9) that specifies its role and procedure. For each image, a user prompt (Listing 10) directs the model to detect candidate images with surfaces such as tables, floors, or shelves. Fig. 14 shows the filtering results. The first row shows scenes without platforms, while the second row shows scenes with platforms. This filtering enables subsequent computationally intensive analysis to focus on semantically relevant scenes for placement queries.

Listing 9: System Prompt for Qwen2.5-VL to identify images with suitable platforms.

```
image_have_platform_system_text_prompt = """
You are an expert visual scene understanding assistant.

Your task is to analyze an image and determine whether it contains **
    any obvious flat horizontal surfaces** where physical objects can
    be placed. These include **floors, tabletops, bed surfaces, or
    other flat and stable areas**.

IMPORTANT:
- If you can see any part of the **floor**, **tabletop**, **bed**, or
    **similar flat surfaces**, you MUST assume it can support physical
     objects (\eg, books, boxes, pillows).
```

```
- Do NOT consider whether the surface is cluttered, partially visible,
     or obstructed. If the platform exists and is horizontal, assume
    it can hold objects.
- Your answer must be based strictly on visible surfaces.

You must provide a short reasoning based on visual evidence in the
    image, followed by a final conclusion.

Your response MUST strictly follow this format:
"[Your analysis]. | Yes/No"

Examples of valid responses:

Example 1:
The image shows a wooden floor that is flat and unobstructed. And it
    could potentially support physical objects. | Yes

Example 2:
There is a bed clearly visible in the scene with a flat top surface
    where items like pillows or books can be placed. | Yes

Example 3:
A rectangular table is visible in the center of the image, providing a
     flat surface suitable for placing objects. | Yes

Example 4:
The image contains mostly a wall with a window and no visible floor,
    table, or other flat surfaces. | No

Do NOT provide any extra commentary or formatting outside this exact
    format.
"""
```

Listing 10: User Prompt for Qwen2.5-VL to identify images with suitable platforms.

```
image_have_platform_user_text_prompt = """
Please examine the image and determine whether it contains any **clear
     horizontal platforms** where physical objects can be placed.
    These platforms include: **floors, tabletops, bed surfaces, or
    other flat and stable horizontal areas**.

Important notes:
- As long as **any part** of a **floor**, **tabletop**, **bed surface
    **, or similar platform is visible in the image, you must assume
    it is capable of supporting physical objects (such as books, boxes
    , pillows, etc.).
- Do NOT consider whether the surface is messy, partially blocked, or
    whether there's enough space. If the platform exists and is
    horizontal, you must assume it can hold objects.
- Your answer must be based entirely on visible visual evidence in the
     image.

You must respond in the following exact format:
"[Your analysis]. | Yes/No"

Refer to the following examples to guide your response:

Example 1:
The image shows a wooden floor that is flat and unobstructed. And it
    could potentially support physical objects. | Yes

Example 2:
There is a bed clearly visible in the scene with a flat top surface
    where items like pillows or books can be placed. | Yes
```

```
Example 3:
A rectangular table is visible in the center of the image, providing a
    flat surface suitable for placing objects. | Yes

Example 4:
The image contains mostly a wall with a window and no visible floor,
   table, or other flat surfaces. | No

Please strictly follow the format and do not add any extra commentary.
"""
```


Figure 14: Visualization of platform filtering results. Row 1: Examples of images filtered out (lacking platforms). Row 2: Examples of images retained (containing platforms).

**Step 2: Gravity Alignment and Top-Down Projection.** To enable top-down (orthographic) scene projection, we apply gravity alignment matrices from the CA-1M to transform point clouds and object 3D bounding box into a consistent frame where gravity uniformly points downward. This normalization allows for projection onto a plane orthogonal to the gravity vector, revealing object layouts and spatial relations more clearly. Fig. 17 shows the point cloud and coordinate axes before and after alignment.

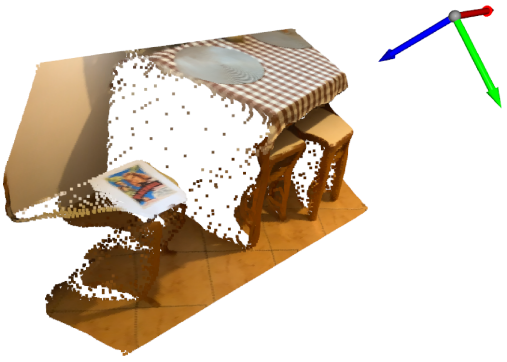


Figure 15: *
(a) Before gravity alignment

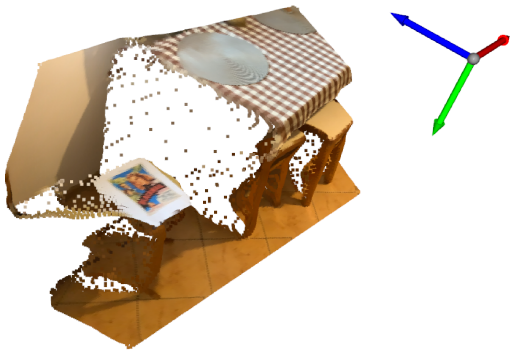


Figure 16: *
(b) After gravity alignment

Figure 17: Visualization of gravity alignment. (a) The scene before alignment. (b) The scene after alignment, with the Y-axis oriented along the gravity vector.

**Step 3: Programmatic Platform Association.** After gravity alignment, we associate objects with supporting platforms based on their spatial relationships. For relationships such as **"front/behind/left-**

**/right"**, we identify the platform supporting a target object by checking the top surfaces of candidate platforms. A platform qualifies if its top surface is within $0.05$ meters of the object's bottom surface and overlaps at least $70\%$ of the object's bottom area in top-down view. For the **"below"** relationship, when an object may be suspended in mid-air, we identify the supporting platform as the one located beneath it, whose top surface is nearest to the object's bottom, and whose top-down intersection with the object exceeds $70\%$ of the object's area. For the **"above"** relationship, the object's own top surface is treated as the reference platform, eliminating the need for additional platform search. For the **"between"** relationship involving two objects, each object's supporting platform is determined independently using the same procedure as in the **"front/back/left/right"** cases. The space between them is considered valid only if both share the same supporting platform, ensuring spatial reasoning occurs within a unified physical context.

**Step 4: Identifying Other Objects on the Platform.** After identifying the target object and its supporting platform, we locate other relevant objects co-occurring on the same platform. For **"front/behind/left/right/between"**, candidate objects are selected based on the following criteria:

1. Their bottom must not be significantly higher than the top of the target object.
2. Their top must remain above the platform surface.
3. Their XZ-plane footprint must intersect with that of the target object.
4. Their volume must not exceed $4.236$ times that of the target object.

To mitigate the over-filtering of solid objects during visible point selection via depth matching, we introduce a volumetric constraint (thresholded at approximately $4.236$, *i.e.*, $(1/0.618)^3$, the cube of the reciprocal of the golden ratio). This prevents large, potentially hollow objects—such as tables with substantial under-space—from being misclassified as fully occluding based solely on their bounding box volume, which would mislead placement reasoning on the primary platform. Instead, the filter targets small to medium-sized objects, where the bounding box volume more accurately reflects actual occupancy. For such objects, even if hollow, the limited under-space is typically negligible for placement purposes. This behavior is shown in the top-view occupancy maps in Fig. 18 and Fig. 26, where a large table is excluded due to exceeding the volume threshold. Despite being present in the scene, its hollow geometry allows for usable space beneath, justifying its omission.

For the **"below"** relation, an object on a platform is considered below the target object (when the target might be suspended or on a higher tier) if:

1. Its footprint (XZ-plane projection) intersects with that of the target;
2. Its bottom is no higher than the top of the target;
3. Its top is not below the top surface of its supporting platform.

For the **"above"** relation, an object is considered above the platform of target object if:

1. Its bottom is within 20 cm above the platform's top surface;
2. Its top is not below the platform's top surface;
3. Its footprint (projection on the XZ plane) overlaps with that of the platform.

**Step 5: Sampling Unoccupied Points in the Top-View Occupancy Map.** After identifying the target object, its supporting platform, and adjacent objects, we determine the surrounding free space. This includes regions in front, behind, left, right, above, or below the target. To locate free space in the horizontal directions (**"front/behind/left/right"**), we define a $90^\circ$ sector centered on the target and oriented in the respective direction. The sector radius is set to the maximum of either the diagonal of the object's footprint or a fixed 20 cm, ensuring adequate coverage. For vertical directions (**"above/below"**), we project the object's top or bottom surface onto the supporting platform. To mitigate overestimation from coarse 3D bounding boxes, we shrink the projection to $80\%$ of its original size (centered), reducing overlap with nearby objects and better approximating usable space. To identify free space **"between"** two target objects, we define the search region as the planar area enclosed by the projections of both objects onto their shared supporting surface. This region is evaluated for occupancy by other objects. Across all spatial contexts (**"above/below/between"**), we enforce a minimum free area constraint: the unoccupied region must exceed $0.036m^2$ (half an A4 sheet) in a top-down view. This threshold filters out trivial cases and ensures the queried space

can accommodate small objects such as a book or cup. Free space is computed by analyzing object footprints in the gravity-aligned top-down view. In Fig. 18, 22, 26, blue-shaded regions indicate the final sampling areas after accounting for bounding box scaling and occlusions. These visualizations highlight viable unoccupied regions for subsequent placement analysis.

**Step 6: Projection and Visibility Filtering.** Given candidate points sampled in the top-down view (XZ-plane) as free space or target locations, we project them into the original 2D camera image to assess visibility. Each point is assigned the $y$ coordinate of the platform's top surface, forming full 3D coordinates. Using camera intrinsics, extrinsics, and gravity alignment, we project these 3D points onto the 2D image plane, as shown in Fig. 19, 23, 27. To determine visibility, we compare the $z$-coordinate of each 3D point with the corresponding depth value in the aligned depth image. Points are discarded as occluded if this difference exceeds $2.5$ cm. For the **front**, **behind**, **left**, and **right** directions, we sample $9,000$ points per direction and retain the direction if at least $2,000$ points remain visible. For the **above**, **below**, and **between** relations, we sample $10,000$ points and require a minimum of $6,000$ visible points. We compute the mean position of the remaining visible points as the representative target location. If its depth deviates from the depth image by more than $2.5$ cm, we instead choose the nearest point within this threshold. Fig. 20, 24, 28 illustrate this process, highlighting visible points and the final selected target (blue circle).

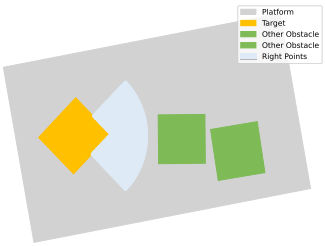



Figure 18: *
(a) Top-view occupancy map with right-side search area

Figure 19: *
(b) Sampled points projected onto the 2D image

Figure 20: *
(c) Visible points with the final placement center

Figure 21: Visualization of **right**-side free space identification. (a) Top-view occupancy map with the target's right-side search area. (b) Projection of sampled candidate points into the image plane. (c) Non-visible points are removed; the final placement center is marked with a blue circle.

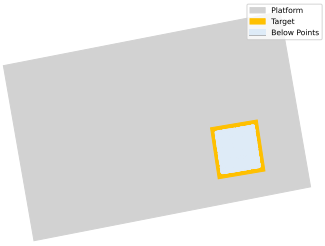



Figure 22: *
(a) Top-view occupancy map of the object's bottom surface

Figure 23: *
(b) Sampled points projected into the 2D image

Figure 24: *
(c) Visible points with the final placement center

Figure 25: Visualization of **bottom**-side free space identification. (a) Top-view occupancy map of the target's bottom surface on the platform. (b) Projection of candidate points into the image plane. (c) Non-visible points are filtered; the final placement center is indicated.

### B.2.6 Generating Diverse QA via precise 3D annotations

Building on the 2D QA generation pipeline (Appx. B.1.4), we utilize a template-based approach augmented with QwQ-32B to generate diverse QA pairs for 3D scenes. **Unlike 2D images, 3D**

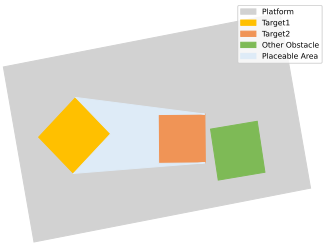



Figure 26: *
(a) Top-view occupancy map with between search area

Figure 27: *
(b) Sampled points projected into the 2D image

Figure 28: *
(c) Visible points with the final placement center

Figure 29: Visualization of **between**-object free space identification. (a) Top-view occupancy map showing the area between two target objects on the same platform. (b) Projection of sampled points into the image plane. (c) Non-visible points are filtered; the final placement center is indicated.

**datasets offer rich and precise spatial annotations—such as depth maps, camera poses, and per-object 3D bounding boxes—which enable the construction of more complex QA with spatial referring with reasoning.** Additionally, QA generation from 3D data accounts for above/below relations in both world and camera reference frames: the former reflects gravity, while the latter corresponds to vertical image orientation. Leveraging these annotations, we design both qualitative and quantitative QA templates grounded in the following spatial concept categories:

1. **Relative Position Relations:** Encompasses spatial relations such as left/right, above/below, front/behind, inside/outside, touching/separated, and near/far. These queries require accurate 3D positioning and spatial layout to infer inter-object relationships in physical space.

2. **Orientation and Rotation Reasoning:** Involves reasoning over face/back direction, orientation (horizontal/vertical), and relative angles, using 3D object or camera poses (*e.g.*, orientation vectors, rotation matrices) to infer facing direction or viewpoint shifts.

3. **Geometric Attribute Comparisons:** Covers attributes like size (big/small), height (tall/short), and width (wide/thin). These comparisons rely on true 3D dimensions, mitigating distortions from 2D projections.

4. **Quantitative Spatial Reasoning:** Involves computing depth, distance, relative angles, and spatial betweenness using precise 3D coordinates and metric reasoning.

5. **Free Space Reasoning:** Identifies free space above, below, or between objects. As illustrated in Fig. 21, 25, and 29, blue-shaded regions represent unoccupied areas computed from object footprints and platform segmentation. To mitigate overestimation from large bounding boxes, we apply a shrink factor (*e.g.*, 80%) to the projected surfaces for above/below queries.

6. **Location and Placement Prediction for Spatial Referring:** Involves predicting precise 2D coordinates from language descriptions, *e.g.*, "*Point to the second chair from the left*" — identifying a target object, or "*Indicate a free spot to the right of the white box on the second shelf*" — selecting a valid placement location. These tasks require accurate 2D-3D projection and fine-grained spatial understanding, forming a vital bridge between visual perception and physical interaction and execution.

Building on the structured templates, we design a diverse suite of 3D QA covering spatial reasoning, geometric comparison, viewpoint inference, environmental understanding, and coordinate-level localization. Leveraging QwQ-32B's powerful capabilities, our pipeline also generates complex reasoning QA pairs that are both structurally diverse and semantically rich.

**3D training data visualization.** For specific examples of 3D training data and their visualizations, please refer to Appx. F, which contains detailed sample presentations.

### B.3 Synthetic Data Generation in the simulator

We want to arm our model with multi-step referring capabilities with spatial reasoning. While 2D and 3D data enable single-step spatial understanding, they are less scalable for multi-step spatial referring with reasoning. Therefore, we generate synthetic data in the simulator.

#### B.3.1 Indoor Scene Generation

**Initial Scene Generation.** We utilize Infinigen [137] to generate a large corpus of indoor scenes. To be specific, we configure the generation process to exclude small objects by setting `compose_indoors.solve_steps_small=0` to avoid pre-existing clutter on target surfaces. This allows us to reserve space for the subsequent placement of our curated 3D assets. This initial step yields over 3k unique indoor scenes.

**Scene Filtering.** The generated scenes underwent a rigorous filtering process to ensure their suitability for our downstream tasks. The primary filtering criteria included:

- **Adequate Tabletop Area:** Selected scenes contain at least one sufficiently large, continuous tabletop surface (*e.g.*, desk, table, counter) suitable for object placement. Scenes with absent or impractically small surfaces are excluded.
- **Acceptable Lighting Conditions:** Scenes with extreme lighting issues (*e.g.*, darkness, oversaturation, unnatural hues) are discarded to ensure a viable baseline for subsequent lighting adjustments.
- **Scene Realism and Coherence:** Scenes with severe geometric inconsistencies or implausible layouts are removed to maintain physical plausibility.
- **Camera Accessibility:** Scenes are required to support feasible camera placement with clear views of the target surfaces. Highly cluttered or confined environments are deprioritized.

**Scene Adjustments.** To enhance diversity and control experimental variables, filtered scenes underwent automated modifications:

- **Lighting Randomization:** Light source intensities (*e.g.*, ceiling lights, lamps) are uniformly scaled within $[0.6I, 1.4I]$, where $I$ denotes the original intensity.
- **Camera Pose Adjustment:** For each tabletop, camera viewpoints are defined with pitch angles randomly sampled from $[-60°, -30°]$ relative to the tabletop plane, oriented toward the region center.
- **Camera Height and Distance Variation:** Camera height is uniformly sampled from $0.3$–$0.8\,m$ above the tabletop. Distance to the target area is adjusted to maintain full visibility, conditioned on surface size and field of view.

Some typical cases of scene filtering are shown in Fig.30.

#### B.3.2 3D Asset Selection and Preparation

Our 3D assets are sourced from the Objaverse [138] LVIS dataset and undergo a two-stage filtering process to ensure quality and relevance.

**Stage 1: Category Filtering.** We select objects based on LVIS annotations, following categories:

- Are typically placeable on surfaces.
- Have a maximum dimension under 1 meter, suitable for tabletop scenarios.

**Stage 2: Attribute-based Filtering.** Next, we apply fine-grained filtering using attributes from the OrienText300K [7] dataset. Retained assets satisfy the following criteria:

- **Axis Alignment:** Key features (*e.g.*, edges, handles) align with canonical camera axes.
- **Single Object:** Represents a standalone object, not a scene or object collection.
- **Color Diversity:** Contains colors beyond white or gray.
- **No Ground Plane:** Excludes auxiliary visualization ground planes.

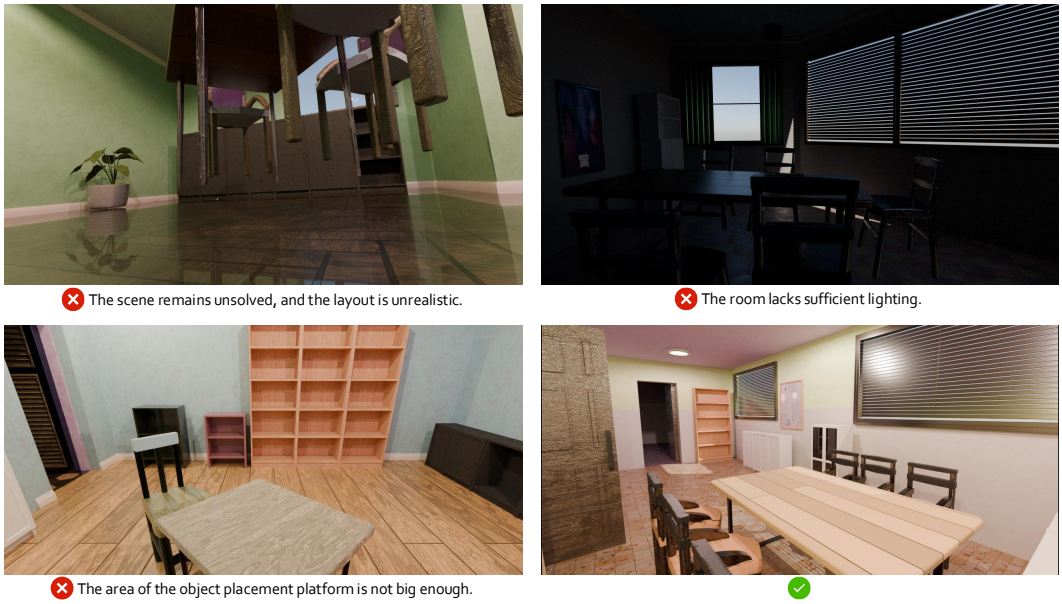



Figure 30: Cases of Scene Filtering

- **High Quality:** Clean, well-constructed geometry without artifacts.
- **Distinguishable Views:** Canonical views (front, back, top, bottom, left, right) exhibit meaningful visual or semantic differences.
- **Reasonable Object:** Represents a common, identifiable object, not an abstract shape or an unidentifiable entity.

This process yields a curated set of over 9k high-quality 3D assets.

**Stage 3: Manual Filtering.** Due to the suboptimal quality of annotations, we perform manual verification based on the seven defined rules. Since object size estimation relies on LVIS-labeled categories with a ±30% tolerance, we additionally discard instances whose irregular geometry (*e.g.*, entangled cables of a wired mouse) distorts the bounding box and hinders reliable scaling. After filtering, we retain over 3k high-quality 3D assets compliant with our criteria.

Fig.31 shows the screening results for some representative 3D assets, which encompass all the criteria mentioned above.

### B.3.3 3D Asset Annotation with LLM

To generate diverse annotations, we leverage OrienText300K's orientation and caption data, processed via GPT-4o with tailored prompts to extract structured textual attributes for each sample.

**Generated Attributes:**

- **Orientation Descriptions:** Prepositional phrases indicating the object's canonical front, salient parts, or intrinsic orientation (*e.g.*, "on the front of ...", "on the handle side of ..."), suitable for insertion into sentence templates.
- **Color Labels:** A single-word descriptor of the object's dominant color. If multiple colors are prominent, the attribute is marked as "none" (*e.g.*, "blue", "none").
- **Object Labels:** Concise noun phrases specifying the object category (*e.g.*, "coffee mug", "computer mouse"), usable as subjects or objects in templates.
- **Category Consistency:** A boolean flag indicating alignment between the object's visual category and its textual description.

**LLM Prompt:** The exact prompt used to obtain these annotations is provided in Listing 11.

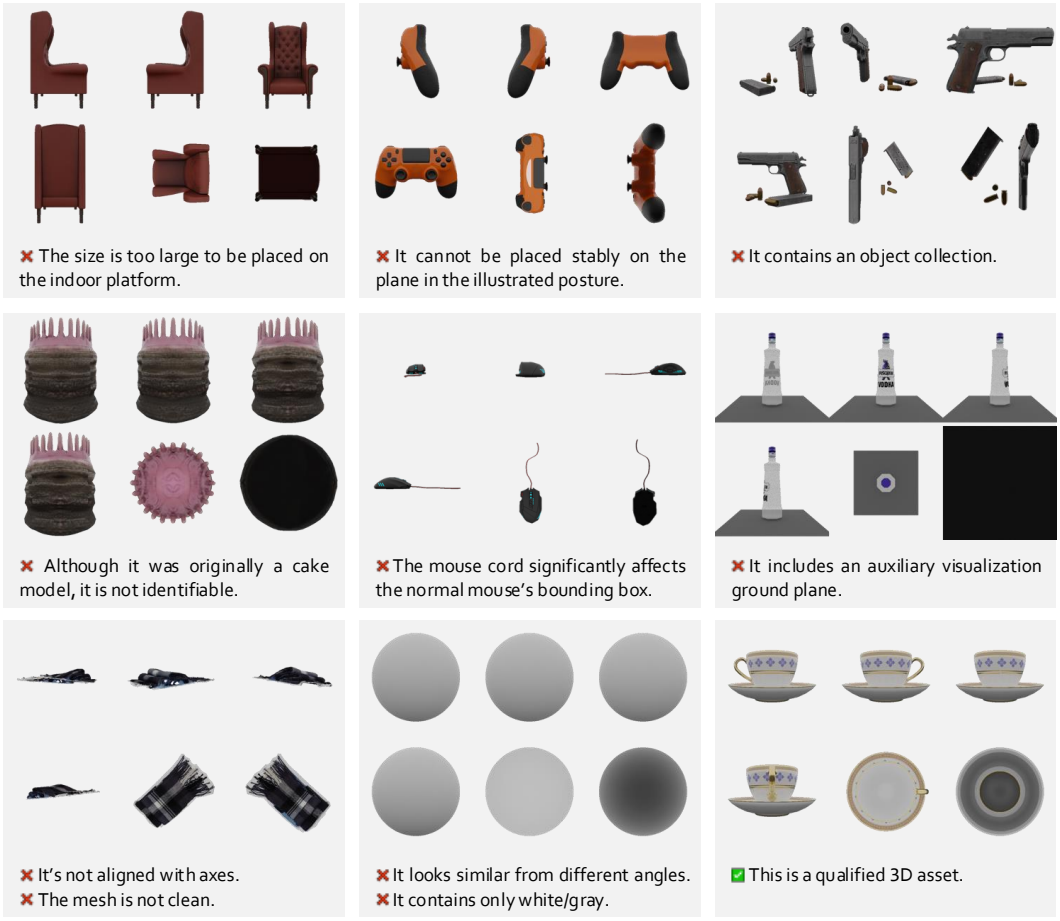



Figure 31: Cases of 3D assets Filtering

Listing 11: Prompt of generating brief description, color and orientation preposition.

```
Your objective is to generate four distinct labels derived from the
provided 3D asset information. Each asset is characterized by the
following attributes:

Category: {category}
Detailed description: {description}
Direction hint: {direction_hint}

Based on the information above, you are required to perform
the following four tasks:

Task 1: Consistency Verification
Evaluate whether the asset's specified category aligns semantically
with its detailed description. Output "True" if consistent,
and "False" otherwise.

Task 2: Object Description Phrase Generation
ormulate a concise and unambiguous object description phrase. This
phrase must encapsulate the primary characteristics of the object,
not contain any articles (\eg, "a", "an", "the"), and be
grammatically suitable for use as either the subject or object
within a template sentence.

Task 3: Simplified Directional Phrase Generation
```

```
Generate a concise and clear simplified directional description phrase
based on the provided direction_hint. This phrase must be capable of
functioning as a prepositional phrase to describe the relative
position of other objects within a template sentence. In the generated
phrase, the current object should be represented by "OBJECT".

Task 4: Color Extraction
Extract the dominant color of the asset based on the detailed
description, adhering to the following criteria:
1. The extracted color must be a single English word in lowercase.
2. If the detailed description does not explicitly state a color,
return the string "none".

Please ensure your output strictly adheres to the following JSON
format. The output must be a valid JSON object without any
supplementary explanations, comments, or introductory/closing
remarks.
{
  "consistent": (True/False),
  "simple_desc": "simplified object description phrase",
  "simple_dir": "simplified directional description phrase",
  "color": "the main color"
}
```

| Category | Bible |
|---|---|
| Description | The images depict a 3D model of an old book with reddish-brown covers and visible pages. The covers have a worn, textured appearance while the sides show the pages. |
| Direction Hint | spine |
| Direction Vec | (0, -1, 0) |

| Consistent | True |
|---|---|
| Simple Description | Bible |
| Direction Prep | on the spine side of the OBJECT |
| Color | brown |

Figure 32: A Sample of generating object phrases, colors, and orientation descriptions from 3D asset information and orientation hints.

### B.3.4 Scene Population and Data Rendering

The curated 3D assets are then programmatically placed into the filtered and adjusted indoor scenes.
**Asset Placement Strategy:**

- Assets are placed predominantly on identified tabletop surfaces.
- To enhance the model's understanding of object orientation and inter-object relationships, placement strategies included:
  - Increasing the proportion of objects with orientation vectors within the X-Y plane (*e.g.*, laptops, teddy bears, mugs) in many scenes.
  - Increasing the co-occurrence of objects from the same category that have significantly different features. (*e.g.*, a ceramic cup with a handle and a cola cup).
  - Ensuring plausible physical arrangements (*e.g.*, no excessive interpenetration, objects placed upright).
- The number of assets per scene varied from 3 to 9.

**Rendering Details:**

- **Renderer:** High-fidelity, physically-based images are rendered using the Blender Cycles.
- **Image Resolution:** Output images are rendered at $960 \times 540$ pixels.
- **Render Quality:** The `configure_render_cycles.num_samples` parameter is set to 2048 to achieve high-quality renders while maintaining reasonably controlled noise.

### B.3.5 Question-Answer Pair Generation

Based on the rendered scenes and their corresponding ground truth information (scene graphs, object properties, masks), we generate question-answering (QA) pairs. This process involves two steps:

**Step 1: Generating Unique Object Referring Expressions.** For each object in a scene, we formulate unique identifiers or referring expressions based on a combination of its attributes and relationships:

- **Feature Category:** The semantic class of the object (*e.g.*, "the mug", "the laptop").
- **Color:** The primary color of the object (*e.g.*, "the red mug").
- **Left-Right Rank:** Ordinal position from left to right (*e.g.*, "third bottle from left to right").
- **Front-Back Rank:** Ordinal position from front to back (*e.g.*, "farthest LEGO minifigure").
- **Distance Rank from Anchor:** Ordinal position based on distance from a salient anchor object (*e.g.*, "the closest plate to the blue mug").
- **Height Rank:** Ordinal position based on the height (*e.g.*, "the second tallest teddy bear").

These components are combined to create unambiguous references (*e.g.*, "the second red object from the left"). When filling objects into template structures in the subsequent process, we can randomly select from all referring expressions.

**Step 2: Applying QA Templates.** Due to the heavy reliance on spatial relationships between objects, we initially developed a template structure to formalize these relationships:

- **Position:** Location of an object relative to another (*e.g.*, "on the left of the green bottle").
- **Orientation:** Questions involving an object's intrinsic orientation (*e.g.*, "on the handle side of the red mug").
- **Distance Queries:** The precise distance (*e.g.*, "0.2 meters to the left of the plate").
- **Betweenness:** Identifying an object located between two other objects (*e.g.*, "between the stapler and the telephone").
- **Specific Surface Locations:** Locating objects relative to parts of a surface (*e.g.*, "at the far left corner of the table").

**QA Template Types:**

- **Locate from Description:** Given a unique referring expression for an object, ask for its location (*e.g.*, "*Give me the position of ...*").
- **Identify from Relations:** Provide several spatial relationships an object satisfies and ask to identify the object (*e.g.*, "*Please specify an object on the desktop that satisfies the following spatial constraints: ...*").
- **Locate Empty Space:** Define a point in an empty area on a surface based on its spatial relationships with surrounding objects, and ask to confirm this empty location (*e.g.*, "*Please provide a point in the vacant area on the desktop that simultaneously satisfies the following spatial conditions: ...*").

**Generation of Thought Processes (Reasoning Steps):** For each QA pair, a structured thought process or chain of reasoning is also generated. This involved selecting the pertinent pieces of information from the complete scene graph and ground truth data that are necessary to arrive at the answer. This recorded reasoning follows a predefined format.

## C Implementation Details and Samples of *RefSpatial-Bench*

The *RefSpatial-Bench* benchmark evaluates spatial referring with reasoning in complex 3D indoor scenes through two tasks: **Location Prediction** and **Placement Prediction**, each comprising 100 samples. Each sample includes a manually selected image, a referring caption, and precise mask annotations. Moreover, to evaluate the effectiveness of the RFT training strategy, we further select 77 samples from these 200 samples and define it as the **Unseen** set, which contains novel spatial relation combinations absent from *RefSpatial* to test the model's generalization.

**Location Task:** Given an indoor scene and a unique referring expression, the model predicts a 2D point indicating the target object. Expressions may reference color, shape, spatial order (*e.g.*, "the second chair from the left"), or spatial anchors.

**Placement Task:** Given a caption specifying a free space (*e.g.*, "the vacant area to the right of the white box on the second shelf"), the model predicts a 2D point within that region. Queries often involve complex spatial relations, multiple anchors, hierarchical references, or implied placements.

**Unseen Set:** This set comprises 77 samples from the Location/Placement task, specifically designed to evaluate model generalization after SFT/RFT training on *RefSpatial*, as it includes novel spatial relation combinations not present in *RefSpatial*.

Notably, **we introduce *reasoning steps* (*step*) for each benchmark sample, quantifying the number of anchor objects and their associated spatial relations that effectively narrow the search space.** Specifically, each *step* corresponds to either an explicitly mentioned anchor object or a directional phrase linked to an anchor that greatly reduces ambiguity (*e.g.*, "on the left of", "above", "in front of", "behind", "between"). We exclude the "viewer" as an anchor and disregard the spatial relation "on", since it typically refers to an implied surface of an identified anchor, offering minimal disambiguation. Intrinsic attributes of the target (*e.g.*, color, shape, size, or image-relative position such as "the orange box" or "on the right of the image") also do not count towards *step*.

**A higher *step* value indicates increased reasoning complexity, requiring stronger spatial understanding and reasoning about the environments. Empirically, we find that beyond 5 *steps*, additional qualifiers yield diminishing returns in narrowing the search space. Thus, we cap the *step* value at 5. Instructions with *step* $\geq$ 3 already exhibit substantial spatial complexity.** Detailed statistics on `step` distributions and instruction lengths are provided in the Tab. 11. To further show the diversity and reasoning complexity of *RefSpatial-Bench*, we present representative examples from both the Location and Placement tasks. Fig. 41, 42, 43, and 44 show Location queries with varying reasoning step counts, where *RoboRefer* accurately localizes the target object (marked by a blue dot). Similarly, Fig. 45, 46, 47, and 48 show Placement queries involving the identification of free space based on spatial relations. These examples highlight the step-wise complexity of the queries and the effectiveness of *RoboRefer* in addressing challenging spatial referring tasks.

Table 11: Statistics of the *RefSpatial-Bench* across Location/Placement tasks and unseen sets.

| ***RefSpatial-Bench*** | **Step** | **Samples** | **Avg. Prompt Length** |
|---|---|---|---|
| **Location** | Step 1 | 30 | 11.13 |
| | Step 2 | 38 | 11.97 |
| | Step 3 | 32 | 15.28 |
| | | **Avg. (All)** | 12.78 |
| **Placement** | Step 2 | 43 | 15.47 |
| | Step 3 | 28 | 16.07 |
| | Step 4 | 22 | 22.68 |
| | Step 5 | 7 | 22.71 |
| | | **Avg. (All)** | 17.68 |
| **Unseen** | Step 2 | 29 | 17.41 |
| | Step 3 | 26 | 17.46 |
| | Step 4 | 17 | 24.71 |
| | Step 5 | 5 | 23.8 |
| | | **Avg. (All)** | 19.45 |

To more comprehensive evaluate the spatial referring task, we expand the original RefSpatial-Bench in terms of difficulty and diversity, producing the manually annotated RefSpatial-Expand-Bench. It includes more indoor cases (*e.g.*, shops and factories), and also introduces outdoor scenes not present in RefSpatial-Bench (*e.g.*, streets, parking lots, and parks). Statistics of this extension are provided in Tab. 12 and Tab. 13. The detailed evaluation results of RoboRefer on this expanded benchmark are showed in Tab. 14.

Table 12: Statistics of the RefSpatial-Expand-Bench

| Task Type | Indoor | Outdoor | Total |
|---|---|---|---|
| Location | 115 | 126 | 241 |
| Placement | 120 | 80 | 200 |
| **Total** | **235** | **206** | **441** |

Table 13: Statistics of the RefSpatial-Expand-Bench by step and task.

| Task Type | Step | Samples | Avg. Prompt Length |
|---|---|---|---|
| **Location** | Step 1 | 54 | 10.61 |
| | Step 2 | 129 | 12.56 |
| | Step 3 | 58 | 16.10 |
| | **Avg. (All)** | 241 | 12.98 |
| **Placement** | Step 1 | 3 | 15.00 |
| | Step 2 | 86 | 15.14 |
| | Step 3 | 75 | 16.95 |
| | Step 4 | 29 | 22.24 |
| | Step 5 | 7 | 22.71 |
| | **Avg. (All)** | 200 | 17.11 |

# D Implementation Details for *RoboRefer*

## D.1 Architecture

We adopt NVILA [38] as base model, including a visual encoder, an LLM, and a multimodal projector.

**Visual Encoder.** We use the same image encoder as `siglip-so400m-patch14-448` [75] from NVILA [38], supporting $448 \times 448$ resolution for richer visual details. Rather than simply resizing the image to a fixed resolution and producing the same number of tokens, this image encoder processes inputs at dynamic resolutions, yielding more visual tokens from higher-resolution images via finer patch division. This enables fine-grained vision-language understanding, crucial for tasks like point prediction that require detailed perception beyond VQA. We further incorporate a dedicated depth encoder, structurally mirroring the image encoder and initialized with its weights. It encodes relative depth maps as special images, providing spatial cues to enhance 3D understanding.

**Large Language Model.** We adopt the Qwen2 LLM backbone from NVILA [38], which has been fully fine-tuned with extensive data during supervised training. This endows the model with rich visual knowledge, facilitating downstream 3D spatial understanding and reasoning tasks.

**Multi-Modal Projector.** To align multi-modal representations (*e.g.*, image to language, depth to language), we use linear connectors, the same as NVILA [38], which is better than Q-Former, to allow the LLM to focus on visual understanding and improve generalization. Separate connectors for image and depth embeddings ensure modality-specific processing and prevent cross-modal interference.

## D.2 Training Data

Here we highlight the training data used at each stage, including the number of samples per dataset and the overall total. See Tab. 15 for details.

**SFT stage.** Specifically, in the first step of the SFT stage, *i.e.*, depth alignment, we train a depth projector to align depth and language space using the *RefSpatial* (RGB-D) dataset with 2.5M samples. To increase training efficiency, we slice multi-turn conversations (up to 15 turns per sample), yielding 3.4M samples post-processing to train our model. In the second step, *i.e.*, spatial understanding enhancement via full-parameter fine-tuning—we use both *RefSpatial* (RGB) and *RefSpatial* (RGB-D) datasets, yielding 6.8M samples after slicing. To further improve instruction-following and referring capabilities, we incorporate auxiliary datasets: 965k samples from instruction-tuned data (LLaVA-1.5 [134], LRV [133]), 321k from referring datasets (RefCOCO/+/g [87]), 176k from SAT [4]

Table 14: Accuracy results of 2B SFT and 8B SFT Models on RefSpatial-Expand-Bench.

| Task | Category | 2B SFT | 8B SFT |
|---|---|---|---|
| **Location** | Over all | 50.21 | 61.00 |
| | Indoor | 49.57 | 58.26 |
| | Outdoor | 50.79 | 63.49 |
| | Step 1 | 61.11 | 72.22 |
| | Step 2 | 52.71 | 62.02 |
| | Step 3 | 34.48 | 48.28 |
| **Placement** | Over all | 48.50 | 60.00 |
| | Indoor | 50.83 | 60.00 |
| | Outdoor | 45.00 | 60.00 |
| | Step 1 | 33.33 | 33.33 |
| | Step 2 | 41.86 | 51.16 |
| | Step 3 | 54.67 | 70.67 |
| | Step 4 | 48.28 | 55.17 |
| | Step 5 | 71.43 | 85.71 |

Table 15: Details about the training datasets used in the SFT and RFT stages. D.A. and S.U.E denote the Depth Alignment and Spatial Understanding Enhancement step in the SFT stage, respectively.

| Stage | Categories | Datasets |
|---|---|---|
| SFT (D.A) | Spatial | *RefSpatial* (RGB-D) |
| SFT (S.U.E) | Spatial | *RefSpatial* (RGB), *RefSpatial* (RGB-D), SAT [4], EmbSpatial [22] |
| | General | COCO [150], GQA [18], OCR-VQA [151], TextVQA [152], VG [153], LRV [133] |
| | REC | RefCOCO/+/g [87] |
| RFT | Spatial | *RefSpatial* (RGB-D) w/ Reasoning Processing |

benchmark training sets, and 127k from EmbSpatial [22] benchmark training sets. These additions help bridge distribution gaps between *RefSpatial* and benchmark-style queries. After slicing, the total number of samples used in this stage reaches 8.5M post-slicing.

**RFT Stage.** In the RFT stage, we train the model using *RefSpatial* data annotated with detailed reasoning processes, including key intermediate steps and final answers. To ensure both training efficiency and effective learning, we use moderately difficult samples (typically involving three reasoning steps), resulting in a 100k-sample dataset.

### D.3 SFT Training Details

We formulate the SFT training stage as follows: given a dataset $\mathcal{D}$ consisting of samples in the form of triplets ($\mathcal{O}$, $\mathcal{Q}$, $\mathcal{A}$), where $\mathcal{O}$ is a sensor image (either RGB or RGB-D), $\mathcal{Q}$ is a textual question, and $\mathcal{A}$ is the corresponding answer. The answer $\mathcal{A}$ may be a direct response (*e.g.*, a point coordinate) or include intermediate reasoning steps (*e.g.*, perceptual results followed by the final answer). The training objective is to maximize the likelihood of generating the answer given the input pair ($\mathcal{Q}$, $\mathcal{A}$):

$$\mathcal{L}_{\mathrm{SFT}} = -\mathbb{E}_{(\mathcal{O},\mathcal{Q},\mathcal{A})\sim\mathcal{D}} \sum_{t=1}^{T} \log \pi_\theta(y_t \mid \mathcal{O}, \mathcal{Q}, y_{<t}), \tag{1}$$

where $\pi_\theta$ is the model's token distribution. The output model $\pi_{\mathrm{SFT}}$ serves as the initialization for the next RFT stage, ensuring a robust foundation for reinforcement learning.

To be specific, our SFT consists of two steps. In the first step, depth alignment, only the depth projector is updated by using the *RefSpatial* (RGB-D). We employ a maximum learning rate of 1e-4, a weight decay of 0, and a warm-up ratio of 0.03. The 2B variant is trained with a batch size of 7 per GPU, and the 8B variant with 3, both for one epoch. In the second step of spatial understanding enhancement, we fine-tune all model parameters using the datasets described in Sec. D.2. Training is

conducted for one epoch with a maximum learning rate of 5e-5. We use a batch size of 6 per GPU for the 2B model and 2 for the 8B model. Other hyperparameters follow those in the first step. For more details, please refer to NVILA [38] settings during alignment and SFT.

### D.4 RFT Training Details

During the RFT stage, we refine $\pi_{\text{SFT}}$ via GRPO [114], a reinforcement learning method designed for efficiency and scalability. Unlike PPO [154], which relies on a costly value network, GRPO estimates relative advantages by comparing intra-group rewards, reducing computation, and simplifying optimization. This makes it well-suited for reasoning-intensive spatial referring tasks. In detail, we modify R1-V [122] to support our 3D-aware architecture. Training is conducted for two epochs with a batch size of 1 per GPU and 8 outputs in GRPO. For details about hyperparameters, see R1-V [122].

#### D.4.1 Sampling Action Groups

Given an input state $s = (\mathcal{O}, \mathcal{Q})$, where $\mathcal{O}$ denotes the visual encoding of the RGB or RGB-D observation and $\mathcal{Q}$ the textual encoding of the question, GRPO samples a set of actions $\{a_1, a_2, \ldots, a_N\}$ from the current policy $\pi_\theta$, initialized from $\pi_{\text{SFT}}$. The sampling process is:

$$a_i \sim \pi_\theta(a \mid \mathcal{O}, \mathcal{Q}), \quad \text{for } i = 1, 2, \ldots, N \tag{2}$$

This strategy ensures diverse responses, promoting exploration and preventing premature convergence.

#### D.4.2 Reward Design and Policy Update

Each sampled action $a_i$ is assigned a reward $R(a_i)$ based on verifiable criteria, yielding a reward set $r_1, r_2, \ldots, r_N$. For spatial referring tasks, $R(a_i)$ integrates two outcome-based and our proposed two process-based components. The outcome-based reward functions are defined as follows:

**Outcome Format Reward $R_{OF}$.** This component ensures structured and interpretable outputs by requiring the model to a predefined format: reasoning within "`<think>...</think>`" and the final answer in "`<answer>...</answer>`". A reward is assigned 1 for strict compliance, 0 otherwise.

**Point L1 Reward $R_P$.** This component evaluates the accuracy of the model's final point prediction by comparing it with the ground truth from the annotations of *RefSpatial*. Following the criterion inspired by Seg-zero [131], a stricter reward of 1 is assigned if the L1 distance between the predicted and ground-truth points is within 50 pixels; otherwise, the reward is 0.

Notably, most process-based rewards depend on a Process Reward Model (PRM), typically a fine-tuned LLM or VLM tasked with providing feedback. However, applying such an approach in our setting presents two main challenges. **(1)** LLMs cannot process images, making it impossible to determine whether predicted coordinates match the target object. **(2)** Although VLMs integrate visual and textual information, prior work [155] has shown they may lack precise visual understanding when dealing with textual coordinates. Since the correct assessment of predicted coordinates is paramount for reward assignment, additional or specialized methods are needed to ensure reliable feedback.

**To address this issue, we propose a rule-based process reward for spatial referring that obviates the need for a Process Reward Model.** Our approach directly evaluates key intermediate perceptual steps using the ground-truth step-wise annotations provided in *RefSpatial*. This contrasts with most existing methods on process-based rewards [156, 157], which emphasize strictly sequential reasoning and rely on a PRM for evaluation. **In contrast, our method employs *metric-sensitive* rule-based process reward functions to assess intermediate perceptual results in an *order-invariant* manner.** Our key insight lies in two aspects: **(1) Metric-sensitivity**: Different spatial attributes require distinct metrics due to inherent differences in their representations (*e.g.*, points for positions, vectors for orientations). **(2) Order-invariance**: The reasoning process in spatial referring is not strictly sequential; for instance, identifying the position of the keyboard or the mouse first does not affect the final interpretation of "the free area between the keyboard and the mouse".

In detail, we have two process-based reward functions:

**Process Format Reward $R_{PF}$.** Similar to the Outcome Format Reward strategy, this component enforces a structured and interpretable reasoning process, thereby facilitating accurate reward

computation. In particular, the model is required to produce outputs in the following format:

```
[Perception Type] [Target Object]:  [Value]
```
(3)

where "`Perception Type`" must be one of three categories: "`Position`", "`Orientation`", or "`Size`". The "`Target Object`" corresponds to a uniquely identifiable entity (*e.g.*, "`the second largest cup`" or "`the second large cup from large to small`"). The "`Value`" depends on the selected "`Perception Type`":

- For "`Position`", the value should be a normalized 2D coordinate of the form `[(x, y)]`, where both $x$ and $y$ lie in the interval $[0, 1]$, rounded to three decimal places.
- For "`Orientation`", the value is a 3D unit vector `(x, y, z)` representing the object's semantic orientation in the camera coordinate system.
- For "`Size`", the value represents a scalar measured in meters.

Below are examples to illustrate the expected format:

- `[Position] [the second largest cup]`: [(0.245, 0.147)]
- `[Orientation] [the handle of the second largest cup]`: (1.000, 0.000, 0.000)
- `[Size] [the second largest cup]`: 0.12

**Accuracy Reward $R_{Acc}$.** The reward is computed only for steps annotated as key steps in *RefSpatial*. In detail, we use regex matching to determine whether the "`Target Object`" in the current process format appears in the key-step annotations. If not, the step receives no reward. Since the model has already undergone a cold-start phase in SFT, it can interpret instructions and identify relevant target objects. Thus, a failed match implies that the model cannot accurately refer to the object linguistically, and no reward is assigned. For each perception type, we apply a specific metric to compute the reward: (1) "`Position`": If the L1 distance between the predicted point and the ground truth is below $50$ pixels, the reward is $1$; otherwise, $0$. (2) "`Orientation`": If the cosine similarity between the predicted and ground-truth vectors exceeds $0.8$, the reward is $1$; otherwise, $0$. (3) "`Size`": If the predicted value falls within $\pm 15\%$ of the ground truth, the reward is $1$; otherwise, $0$.

We prioritize the correctness of the final outcome over intermediate steps. To prevent reward accumulation from multi-step processes, we scale the process reward by $0.25$. The final reward function is defined as:

$$r_i = R_{OF}(a_i) + R_P(a_i) + \alpha R_{PF}(a_i) + \alpha R_{Acc}(a_i) \tag{4}$$

where $\alpha$ is set to $0.25$. By normalizing the rewards within the sampled group, we obtain the set of relative advantages $\{A_1, A_2, \ldots, A_N\}$ defined as

$$A_i = \frac{r_i - \text{mean}(\{r_j\})}{\text{std}(\{r_j\})}, \tag{5}$$

which measures how each reward compares to the mean in units of standard deviation. We then update the policy based on these advantages, reinforcing actions with higher relative advantages while reducing the likelihood of those deemed less effective. To ensure stable reinforcement learning, the update is further constrained by minimizing the KL divergence between the updated policy and its reference counterpart, thereby promoting incremental and controlled policy adjustments.

# E Experimental Setting and Details

## E.1 Experiments Compute Resources

We conduct experiments on an A100 GPU cluster, with each node equipped with 8 GPUs.

**2D Web Data Coarse Filtering.** We perform the initial coarse filtering of 1.7M OpenImages using SigLIP2. The process runs on 1 node and takes 8.5 hours and yields 933k high-quality samples.

**2D Web Data Fine-grained Filtering.** We further filter 933K samples using Qwen 2.5-VL 7B to ensure high visual quality and spatially relevant QA pairs. The process is conducted on 1 node (two models per GPU) over 2.5 days, yielding approximately 845k high-quality samples.

**Pseudo-3D Scene Graphs Construction.** We construct pseudo-3D scene graphs for 845k samples using 3 nodes, requiring 10 hours for depth estimation and another 10 hours for camera parameter extraction, segmentation masks, and point cloud generation. Additionally, we generate object-level captions for all instances using 4 nodes over 18 hours by using Qwen 2.5-VL.

**Reasoning QA Generation from 2D data source.** To enrich factual statements with contextual scenarios, we employ QwQ-32B to construct reasoning QA, utilizing 4 nodes over 3.75 days.

**3D Data Filtering and Scene Graphs Construction.** Given the limited amount of 3D data (100k) and the availability of precise annotations, only 2D bounding box bidirectional matching is required. This process is completed in 3 hours using 1 node.

**Reasoning QA Generation from 3D data source.** To enrich factual statements with contextual scenarios, we employ QwQ-32B to construct reasoning QA, utilizing 4 nodes over 1.5 days.

**Synthetic Data Generation in Simulator.** We use 4 RTX 4090 GPUs to generate data for one week.

**Depth Alignment in SFT.** The process is conducted on 10 nodes over 12 hours for 2B variants and 8 nodes over 40 hours for 8B variants. Both variants training use ZeRO-3.

**Spatial Understanding Enhancement in SFT.** The process is conducted on 10 nodes over 2 days for 2B variants and 10 nodes over nearly 1 week for 8B variants. Both variants training use ZeRO-3.

**Spatial Referring in RFT.** The process is conducted on 1 node over 3 days for 2B variants. However, our model is over twice as slow as other Qwen 2/2.5-VL-based methods [122, 130], mainly because they process only a single RGB image during training and can leverage vLLM for group inference acceleration. In contrast, our method requires RGB-D inputs and modifies the original NVILA architecture, making it incompatible with vLLM or SGLang acceleration.

### E.2 Spatial Understanding Benchmarks

We evaluate several public *single-step spatial understanding* benchmarks, including CV-Bench [15] (2D Spatial Relation, 3D Depth Order, 3D Distance), the BLINK [16] validation set (Spatial Relation, Relative Depth), RoboSpatial [2] (configuration), SAT [4], and EmbSpatial [22], following their official evaluation protocols. We exclude non-spatial tasks from our evaluation, such as 2D Counting in CV-Bench and Art Style or IQ Test in BLINK. Since all these benchmarks are multiple-choice tasks, we report accuracy as the evaluation metric.

We compare 3 categories of models: (1) proprietary VLMs, such as Gemini-2.5-Pro [9], which show strong spatial perception, as shown in the Gemini-Robotics [107] paper; (2) open-source VLMs trained on general VQA datasets; and (3) spatially specialized models trained on spatially relevant datasets, offering basic spatial understanding.

### E.3 Spatial Referring Benchmarks

We evaluate three recent robotic referring benchmarks—RoboRefIt [140], Where2Place [5], and RoboSpatial [2], all limited to 2 reasoning steps. Specifically, RoboRefIt concentrates on object location referring by leveraging object attributes and spatial relations with anchor objects. Where2Place further explores how to place objects relative to anchor objects in the camera's coordinate frame. RoboSpatial builds on this idea, considering the same form of placement relative to anchor objects, but many samples in benchmarks consider the object-centric coordinate frame. We also evaluate more complex multi-step spatial referring on *RefSpatial-Bench*, a challenging benchmark based on real-world cluttered scenes, as introduced in Appx. C. For all these benchmarks, we use the same evaluation protocol: we first compute the proportion of predicted points that fall within the ground truth mask for each sample, and then average the results across all samples to obtain the success rate.

We compare two main categories of models: (1) Proprietary models (*i.e.*, Gemini-2.5-Pro) with strong spatial referring capabilities, and (2) spatially specialized models trained on spatially relevant datasets, exhibiting basic spatial referring abilities.

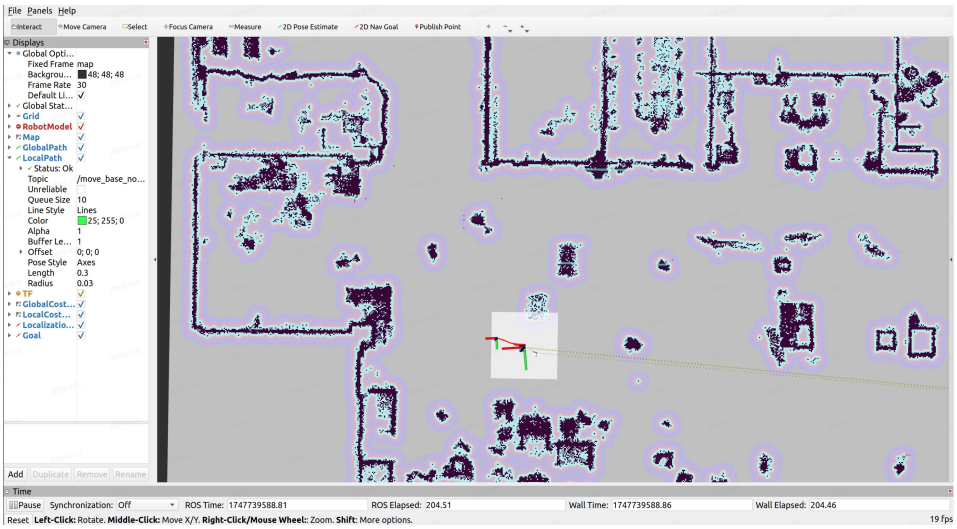



Figure 33: Map Visualization (RViz).

### E.4 Simulation Evaluation

We use the same evaluation protocol of Open6DOR V2 introduced in SoFar [7], following the official repository. Specifically, we only test the position track as this work focuses on location and placement via spatial referring rather than executing 6DOF manipulation tasks. Notably, we find that our model achieves nearly 100% success in the perception stage (*i.e.*, determining location and placement), with failures primarily attributed to motion planning errors such as IK failures or collision-prone trajectories. We show more demonstrations of simulation evaluation in Appx. F.

### E.5 Real-world Evaluation

#### E.5.1 UR5 Manipulation

We show two demos for UR5 Manipulation: human disturbance and voice interruption. In the human disturbance case, *RoboRefer* runs at 2.5Hz. Significant shifts in predicted 2D coordinates trigger motion interruption and re-planning. In the voice interruption case, incoming speech commands are continuously monitored. Upon detection, the current task is halted. We use the Whisper [158] ASR model to transcribe speech, which *RoboRefer* processes into new 2D coordinates for task redirection.

For grasping, the 2D coordinates are fed into SAM2 [78] to generate a segmentation mask, which filters the target object's point cloud from the scene captured by a third-person Intel RealSense L515 depth camera. The extracted point cloud is input to AnyGrasp [159] to predict a grasp pose in the camera coordinate frame. Using an eye-to-hand calibration method, the grasp pose is transformed into the UR5 robot's base frame for execution.

For placement, *RoboRefer* predicts the 2D placement point, which is converted to 3D coordinates using the camera's intrinsic parameters and depth data. The 3D point is then transformed into the robot's coordinate system to guide the placement action.

#### E.5.2 G1 Humanoid Mobile Manipulation

For grasping, we employ a head-mounted Intel RealSense D435 on the Unitree G1 humanoid to capture RGB-D images, which are processed by *RoboRefer* to extract 2D target coordinates. These coordinates guide SAM2 [78] to generate a segmentation mask, which filters the third-person D435 point cloud to isolate the target object. The filtered point cloud is then passed to AnyGrasp [159] to

https://github.com/Zhangwenyao1/Open6DOR_V2_Execution
https://www.universal-robots.com/products/ur5e/
https://www.intelrealsense.com/lidar-camera-l515/
https://www.intelrealsense.com/depth-camera-d435/

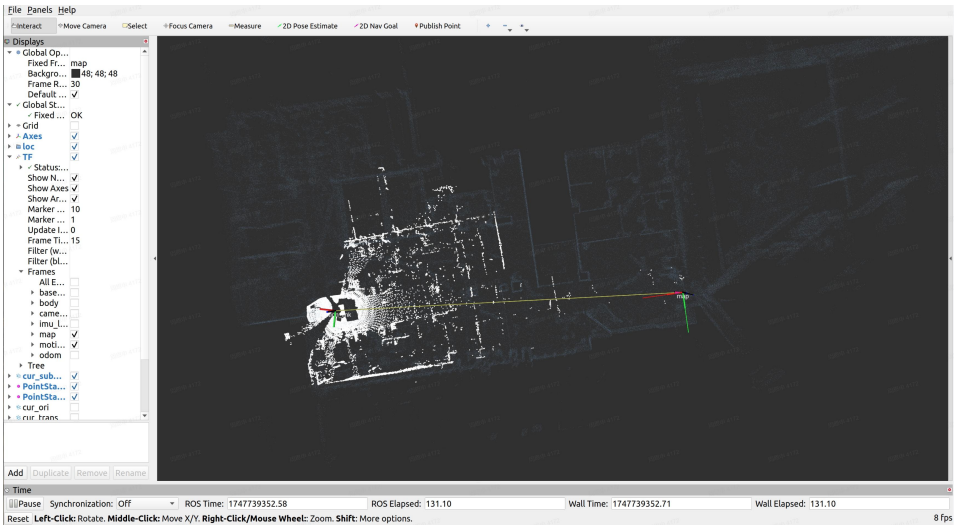


Figure 34: SLAM Navigation (RViz).

Table 16: Ablation on the combination of RGB/RGB-D from *RefSpatial* for SFT (*i.e.*, spatial understanding enhancement). Top-1/Top-2 accuracies are represented using **bold text**, and underlines.

| Method | Input | CV-Bench [15] | | | BLINK$_{val}$ [16] | |
|---|---|---|---|---|---|---|
| | | 2D-Relation | 3D-Depth | 3D-Distance | 2D-Relation | 3D-Depth |
| *Only RGB-D from RefSpatial* | | | | | | |
| *RoboRefer*-2B-SFT | RGB | 87.69 | 86.83 | 82.50 | 79.02 | 81.45 |
| *Combination of RGB and RGB-D from RefSpatial* | | | | | | |
| *RoboRefer*-2B-SFT | RGB | 96.15 | 95.83 | 90.67 | 83.92 | 88.71 |
| *RoboRefer*-2B-SFT | RGB-D | **96.31** | **97.17** | **90.83** | **87.41** | **91.13** |

predict a grasp pose in the third-person frame, which is transformed to the robot's base frame using known camera-to-robot calibration.

For navigation, the chest-mounted L515 camera continuously captures images used by *RoboRefer* to detect nearby landmarks (*e.g.*, a table near the robot). The resulting 2D locations, combined with depth and intrinsics, are projected into 3D world coordinates and integrated into a global map via FAST_LIO_LOCALIZATION_HUMANOID for SLAM-based navigation, as shown in Fig 33, 34.

For placement, the head-mounted D435 captures images processed by *RoboRefer* to localize the target placement region. The corresponding 3D coordinates, computed from depth and intrinsics, are transformed into the robot's base frame for accurate placement execution.

### E.6 More Ablation Studies

We conduct additional ablation studies to identify which design choices enhance the performance.

***RefSpatial* RGB and RGB-D combination for SFT training encourages the image encoder to learn spatial understanding beyond depth cues.** In Tab. 16, incorporating both RGB and RGB-D data from *RefSpatial* in the second stage of SFT training effectively enhances the image encoder's spatial understanding. In contrast, training solely with RGB-D may lead to over-reliance on the depth encoder, limiting the image encoder's ability to learn spatial cues from RGB images alone.

https://github.com/deepglint/FAST_LIO_LOCALIZATION_HUMANOID/tree/main

## F More Demonstrations

**Visualization of *RefSpatial*.** We present dataset examples in Fig 36, 37, 38, 39, which cover 31 distinct types of spatial relationships.

**Visualization of Simulation Evaluation.** We present example rollouts of *RoboRefer* in Fig. 40.

**Visualization of *RefSpatial-Bench*.** We present examples of location in Fig. 41, 42, 43, 44 and placement in Fig. 45, 46, 47, 48 with *RoboRefer* predictions.

**Visualization of Simulator.** We present example rollouts with *RoboRefer* predictions in Fig. 40.

**Visualization of Real-world Evaluation.** We present examples in Fig. 49, 50, 51.

## G More Discussion on Limitations and Future Work

Despite achieving promising results, our model still has limitations. In particular, it relies on precise textual descriptions to pinpoint specific object locations and placement targets, including accurate references to anchor objects. However, in practical real-world robotics scenarios, human instructions are often ambiguous. While such utterances may still imply a unique location, resolving them typically requires sophisticated visual-linguistic reasoning and a process of elimination grounded in human prior knowledge, capabilities that challenge our current model.

We show two representative but interesting examples that highlight the need for human intent understanding and visual-linguistic reasoning: **(1) Probabilistic Preference.** As shown in Fig. 35, humans may refer to a sushi plate as *"pick the one facing the drink"*. In the depicted scene, there are four drink bottles, yet only the two in the middle align with the second sushi plate from the left in the farthest row; the leftmost drink aligns with the first plate, and the rightmost with the third. Despite this ambiguity, people often judge the second plate to be the intended reference due to its higher likelihood of being aligned with two out of the four drinks, reflecting a probabilistic bias in interpretation. **(2) Spatial Compatibility.** As shown in Fig. 35, a user might instruct, *"Place another sushi between the plate and the soy sauce dish"*. Although multiple plate–soy sauce pairs exist, only the pair closest to the observer affords sufficient physical space to place another sushi plate. Thus, implicit spatial feasibility guides the correct interpretation, even without explicit constraints.

Our model struggles with these cases because *RefSpatial* lacks multi-step referring data that embeds human priors and intent understanding. As *RefSpatial* is procedurally generated, incorporating such characteristics at scale remains challenging. Future work may explore procedural synthesis of intent-aware data or improve model performance via co-training with datasets supporting intent comprehension, such as PixMo-Points [10].

## H Broader Impacts

*RoboRefer* can serve as a versatile visual assistant with advanced spatial understanding and reasoning. Due to its integration with large language models (LLMs), it inherits both potential benefits and risks, similar to other VLMs, such as output hallucinations, biases from base models, and heightened energy consumption associated with model upscaling. Beyond these considerations, *RoboRefer* can also function as a high-level planner with spatial referring abilities, guiding robots in tasks like manipulation and navigation. While such capabilities substantially enhance robotic control, they also pose safety challenges when combined with existing control policies.

Despite these concerns, releasing *RoboRefer* to the broader research community would be highly advantageous. Open access would foster continued development and refinement of spatial referring with reasoning, ultimately benefiting diverse robotics platforms (*e.g.*, robotic arms, humanoids) performing various tasks (*e.g.*, manipulation, navigation).

## I Licenses

**(1)** 2D web image data: OpenImages [135] is released under `Apache License 2.0`.

**(2)** 3D embodied video data: CA-1M [136] is released under `CC-by-NC-ND`.

Figure 35: Discussion about the limitation of human prior knowledge and intent understanding.

**(3)** Procedural scene generation: Infinigen [137] is released under `BSD 3-Clause License`.

**(4)** 3D digital assets: objaverse-xl [138] is released under `Apache License 2.0`.

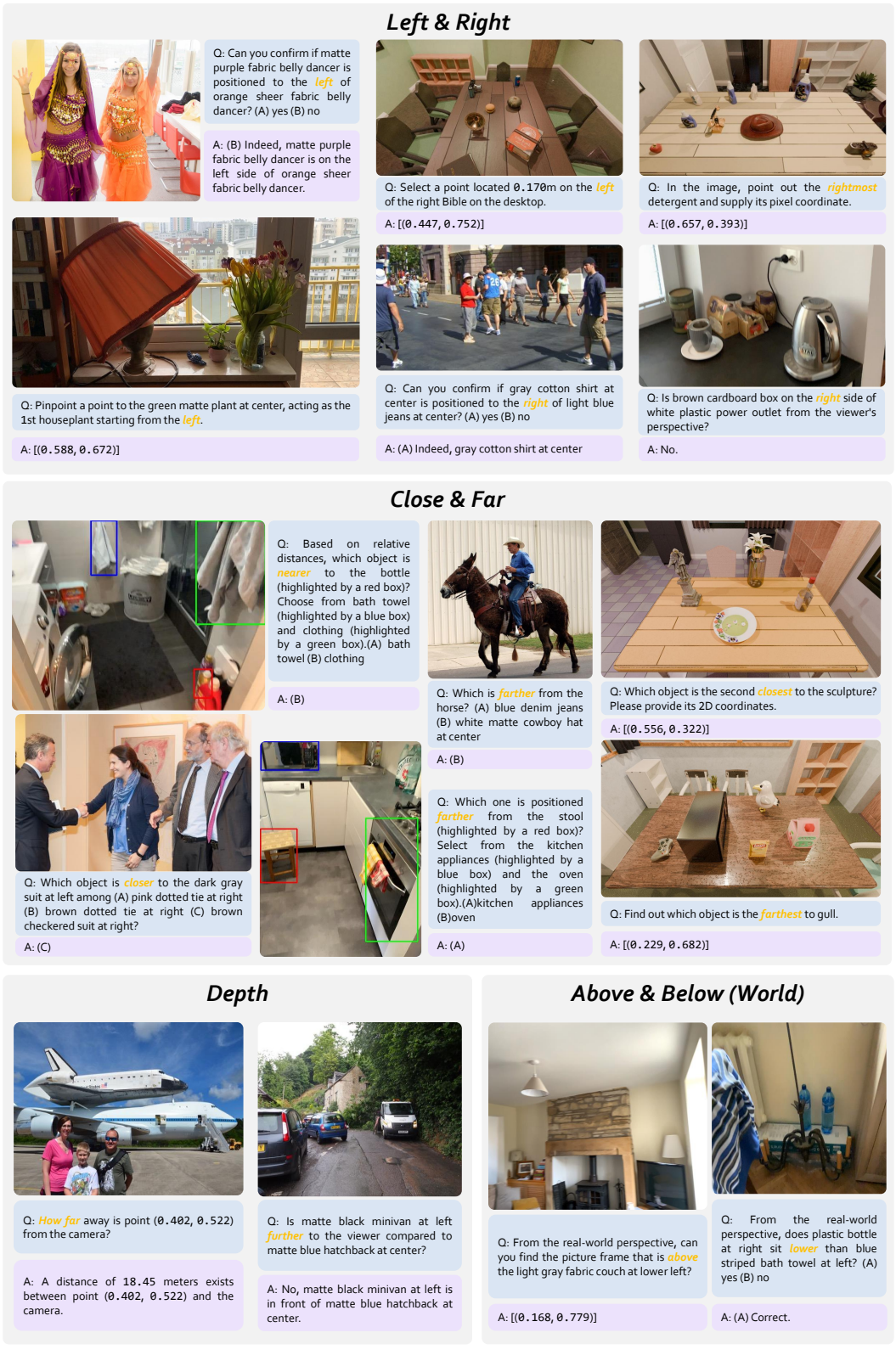



Figure 36: Visualization of *RefSpatial*.

## *Above & Below (Image)*

Q: From the image's perspective, would you say gold textured tray at center is positioned *over* glossy white cabinet at upper center?

A: Yes. Gold textured tray at center appears below white cabinet in the image.

Q: From the image's perspective, is wooden chair at right *below* gold brass saxophone at center? (A) no (B) yes

A: (B) yes.

Q: From the image's perspective, is black iron streetlight at upper left *above* white glossy car at right?

A: Yes. Gold textured tray at center appears below glossy white cabinet at upper center in the image.

Q: From the image's perspective, is wooden chair at right *below* gold brass saxophone at center? (A) no (B) yes

A: (B) Indeed, light wood/wooden chair at right is positioned under gold brass saxophone at center.

## *Tall & Short*

Q: You are organizing a display of tobacco tins and need to stack them vertically. Given that you have a space that can only accommodate items up to the height of the *taller* object between the red matte tin can at left and the matte plastic peterson university flake at center, which tin you choose to ensure it fits?

A: The matte plastic peterson university flake at center is taller than the red matte tin can at left, so choosing the matte plastic peterson university flake at center ensures it fits within the space designated for the taller object.

Q: Where is the *tallest* vodka bottle with a red label located? Please provide its 2D coordinates.

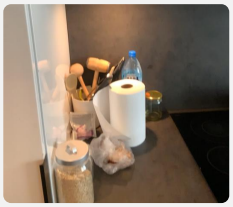
A: [(`0.616, 0.258`)]

Q: Give me position of the *shorter* bottle with a blue cap. Please provide its 2D coordinates.

A: [(`0.383, 0.398`)]

Q: Which object is *taller*, the stone fireplace at center or the wooden photo frame at on the left side?

A: The stone fireplace at center is taller than wooden photo frame.



Q: Is transparent glass jar at center, which is the second glass jar from left to right, *shorter* than transparent glass jar at lower left? (A) no (B) yes

A: (B) Yes.

Q: Which of these two, plastic recycling bin at right or dark blue quilted mattress at lower left, stands *shorter*?

A: With less height is the dark blue quilted mattress at lower left.

Figure 37: Visualization of *RefSpatial*.

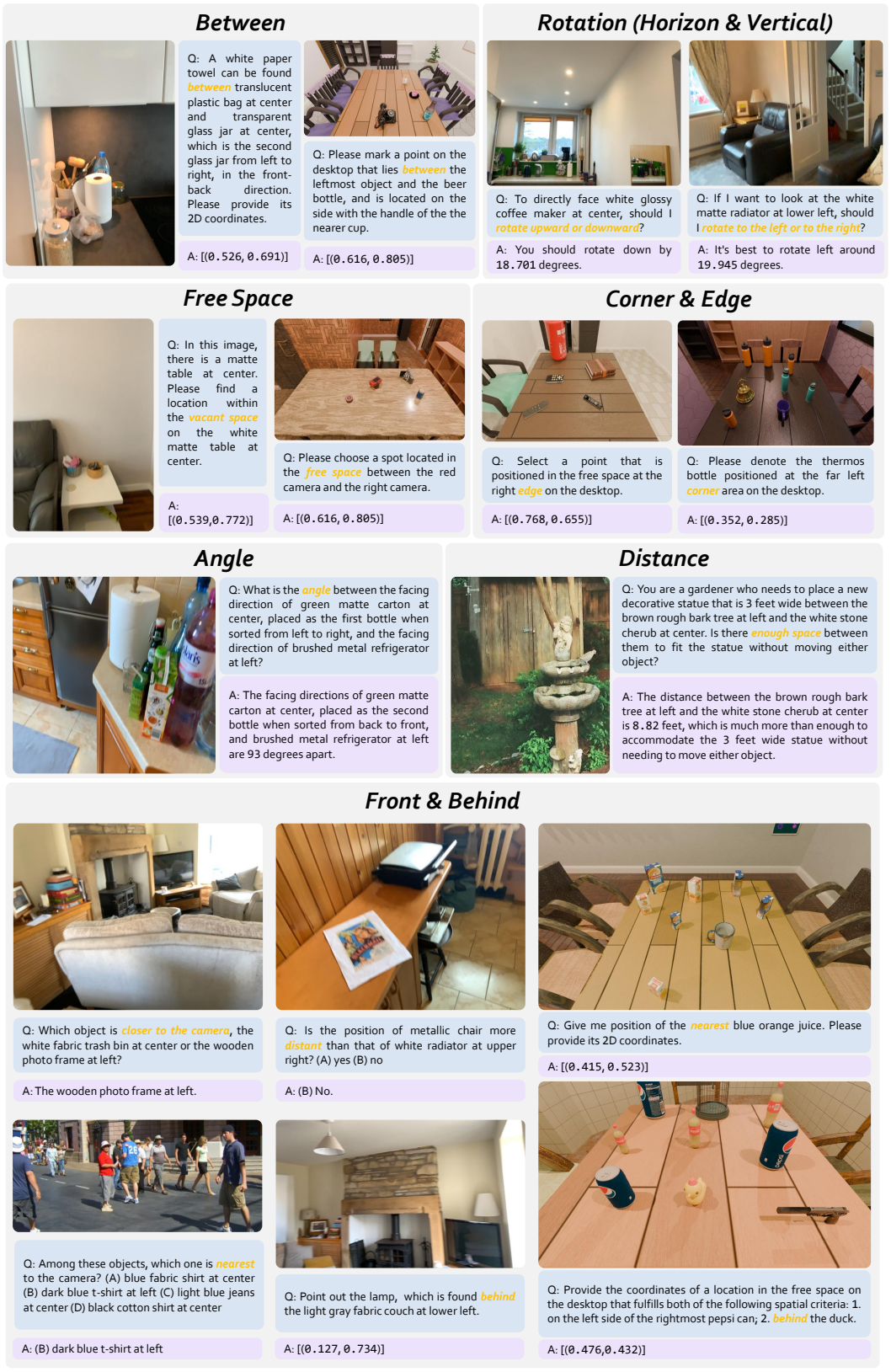



Figure 38: Visualization of *RefSpatial*.

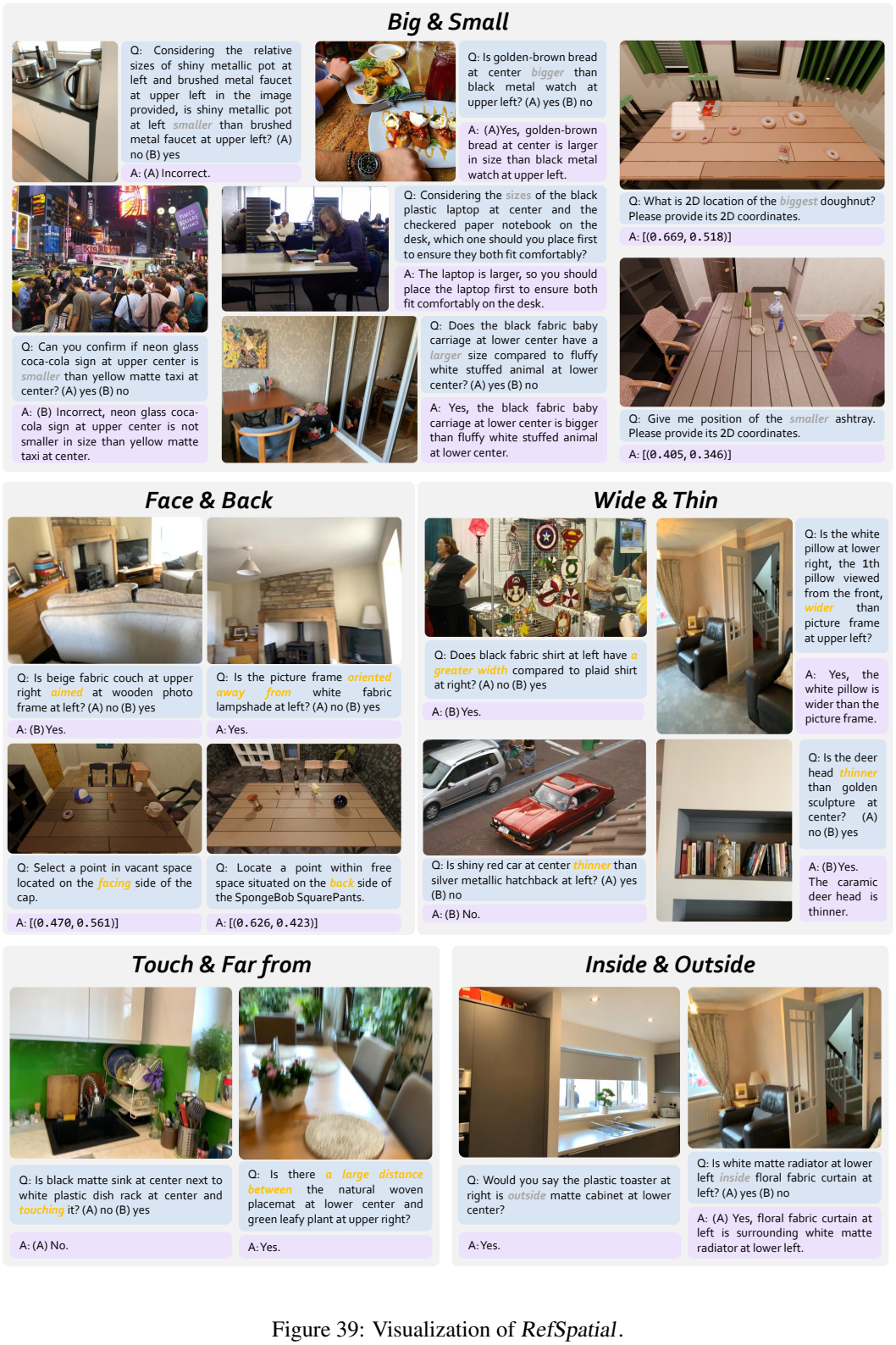
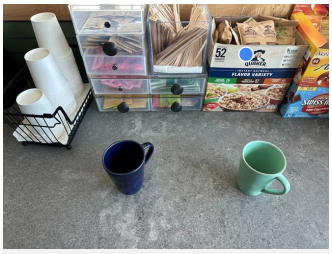



Figure 39: Visualization of *RefSpatial*.

Figure 40: Visualization of *RoboRefer*'s prediction (blue point) on Open6DOR V2 [7] benchmark.

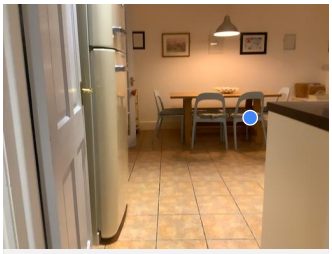


***Reasoning Step = 1***

Please point out the *box with the person logo* in the picture.

***Reasoning Step = 1***

Please point out the *monitor closest to the viewer* in the image.

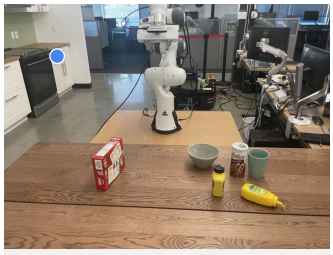


***Reasoning Step = 1***

Please point out the *third chair from the left to the right*.

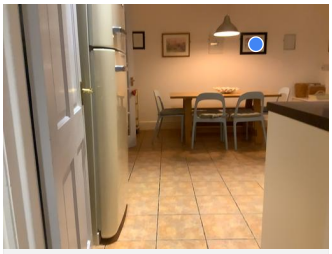
***Reasoning Step = 2***

Please point to *the top piece* of *paper* on the *white table*.



***Reasoning Step = 1***

Please point to the *farthest white cabinet* in the picture.

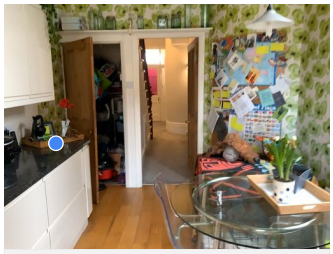


***Reasoning Step = 3***

Please point out the *black framed painting on the right of* the *lamp*.

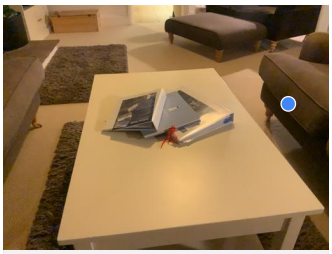
***Reasoning Step = 2***

Please point out the *third object from left to right on the closest platform*.



***Reasoning Step = 1***

Please point to the *wooden plate on the far left of the picture*.



***Reasoning Step = 1***

Please point out the *sofa on the right side of the picture* that is *closest to the viewer*.

***Reasoning Step = 1***

Please point to the *rightmost box* in the picture.

***Reasoning Step = 1***

Please point out the *second silver box from left to right* in the picture.

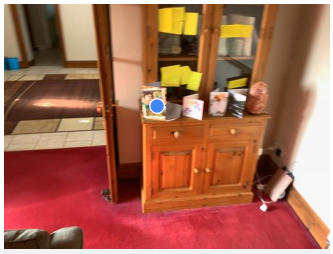
***Reasoning Step = 2***

Please point out the *painting hanging on* the *wall*.

Figure 41: Some Location Examples. The model is asked to identify the object referred to by a prompt. The blue point shows the *RoboRefer*'s prediction (all correct).

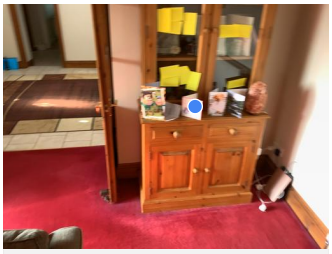
***Reasoning Step = 2***

Please point out the ***blue toothbrush farthest from the faucet***.



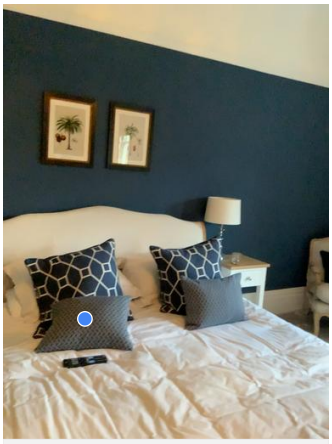
***Reasoning Step = 2***

Please point out the ***card closest to the wooden door***.



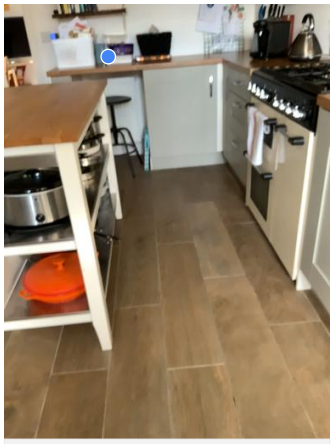
***Reasoning Step = 2***

Please point to the ***third card from right to left on the cabinet***.



***Reasoning Step = 2***

Please point to the ***pillow closest to*** the ***remote controller***.



***Reasoning Step = 3***

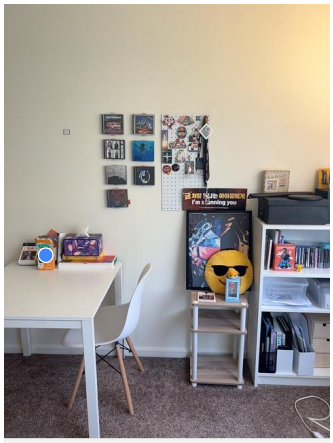
Please point out the ***object between the white box and the farthest black pot*** in the picture.

***Reasoning Step = 2***

Please point out the ***vase closest to*** the ***TV***.

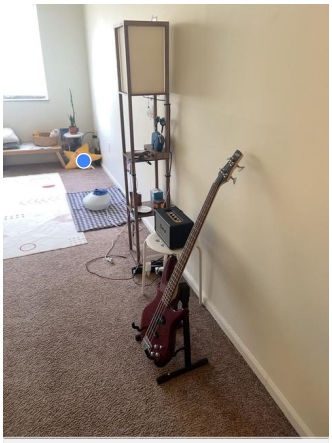
***Reasoning Step = 3***

Please point out the ***leftmost black object on the same platform*** as the ***micro-wave oven***.



***Reasoning Step = 2***

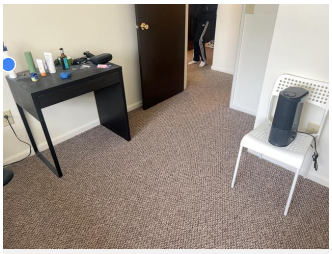
Please point out the ***orange box*** on ***the white table on the left***.



***Reasoning Step = 1***

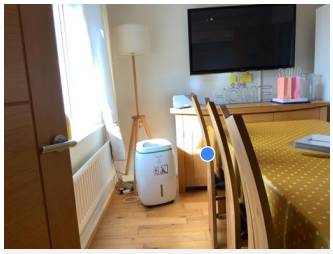
Please point out the ***yellow five-pointed star*** in the picture.

Figure 42: Some Location Examples. The model is asked to identify the object referred to by a prompt. The blue point shows the *RoboRefer*'s prediction (all correct).



***Reasoning Step = 2***

Please point out the ***tallest bottle*** on ***the black table***.

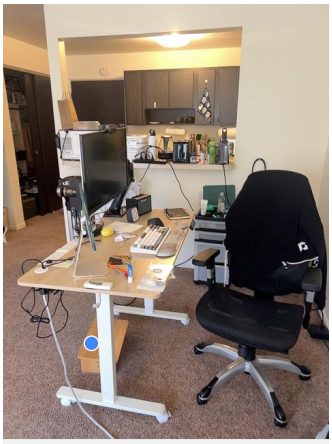


***Reasoning Step = 1***

Please point out the ***farthest chair from the viewer***.

***Reasoning Step = 2***

Please point out the ***silver bottle*** on ***the right edge*** of the ***sink***.

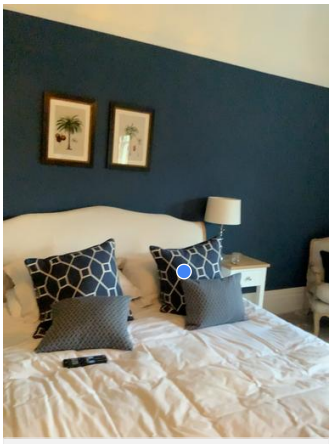


***Reasoning Step = 3***

Please point out the ***object under*** the ***table***.

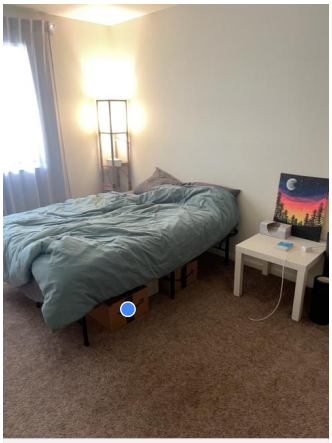
***Reasoning Step = 2***

Please point to the ***rightmost blue box*** on the ***refrigerator***.



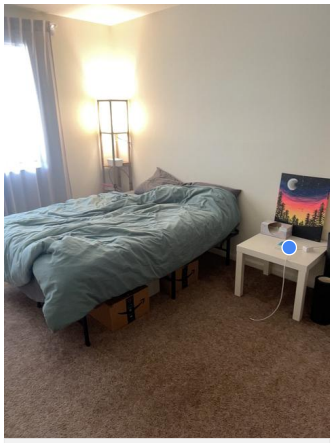
***Reasoning Step = 2***

Please point to the ***pillow closest to the right nightstand***.



***Reasoning Step = 3***

Please point out the ***cardboard box under*** the ***bed*** which is ***closest box to the viewer***.



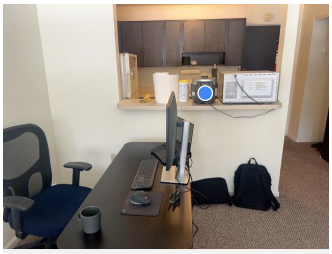
***Reasoning Step = 2***

Please point out the ***blue object*** on the ***table*** in the picture.

***Reasoning Step = 2***

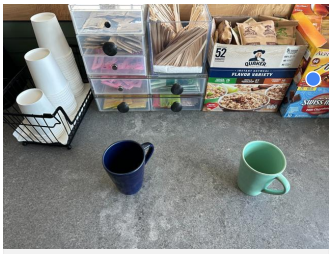
Please point to the ***white box closest to*** the ***remote control***.

Figure 43: Some Location Examples. The model is asked to identify the object referred to by a prompt. The blue point shows the *RoboRefer*'s prediction (all correct).



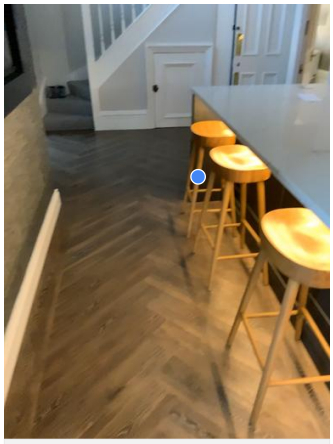
***Reasoning Step = 3***
Please point out the *second object from right to left on the platform with* the *banana*.

***Reasoning Step = 1***
Please point out the *farthest chair from the viewer*.

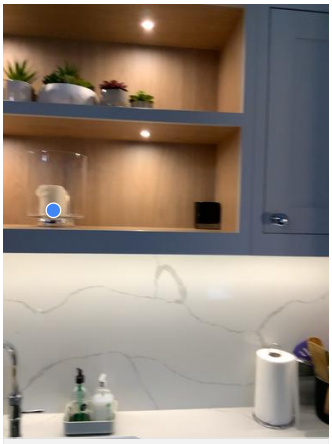


***Reasoning Step = 1***
Please point out the *orange box* in the picture.

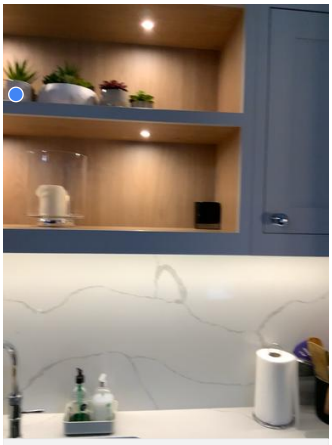
***Reasoning Step = 1***
Please point out the *remote control on the right side of the picture*.



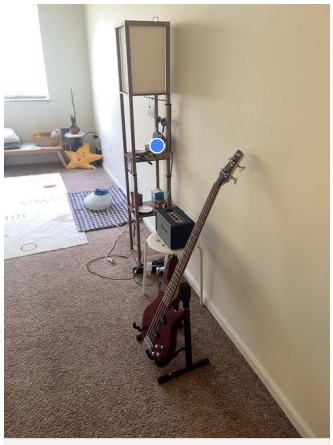
***Reasoning Step = 1***
Please point to the *stool* which is *the third stool from the front*.



***Reasoning Step = 3***
Please point out the *white object on the first shelf* of the *cabinet*.



***Reasoning Step = 3***
Please point out the *plant on the far left of the second shelf* of the *cabinet*.



***Reasoning Step = 3***
Please point out the *blue object* on the *third level* of the *wooden shelf*.

***Reasoning Step = 2***
Please point out the *paper tube closest to the viewer*.

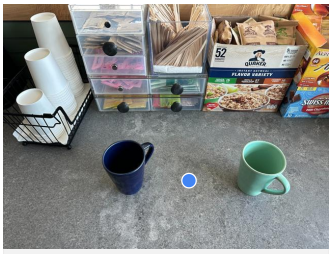
Figure 44: Some Location Examples. The model is asked to identify the object referred to by a prompt. The blue point shows the *RoboRefer*'s prediction (all correct).

***Reasoning Step = 4***

Please point out the *free space between the black water bottle*, the *pot lid*, and the *scissors*.

***Reasoning Step = 4***

Please point out the *free space between* the *black cloth box* to *the bottom-right* of the *monitor* and the *keyboard*.

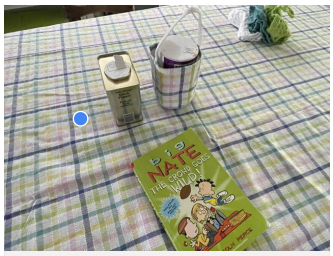


***Reasoning Step = 3***

Please point out the *free spot halfway between the blue cup* and the *green cup* to *place another identical one*.

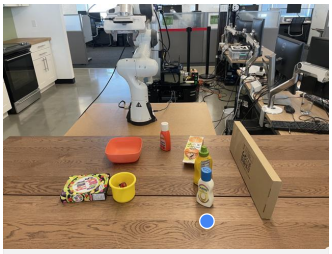
***Reasoning Step = 3***

Please point out the *free space midway between the first and second green cups from the left*.



***Reasoning Step = 4***

Please point out the *free spot to the left* of the *can*, *where an object of the same size can be placed at an equal distance between* the *can and* the *bag*.



***Reasoning Step = 2***

Please point out the *free space in the direction the logo of the closest white bottle to the viewer*.

***Reasoning Step = 4***

Please point out the *free spot*, *equidistant from both the blue bowl* and *the red bowl*, and *between them, where a new, similar-sized bowl can be placed*.

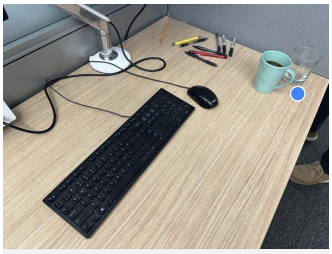
***Reasoning Step = 3***

Please point out the *free space between the red box on the left and the black box*.

***Reasoning Step = 2***

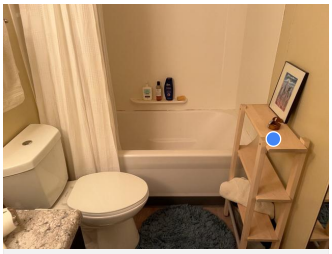
Please point out the *free space in the facing direction* of the *purple bag*.

Figure 45: Some Placement Examples. The model is asked to identify a valid free space based on spatial reference. The blue point shows the *RoboRefer*'s prediction (all correct).



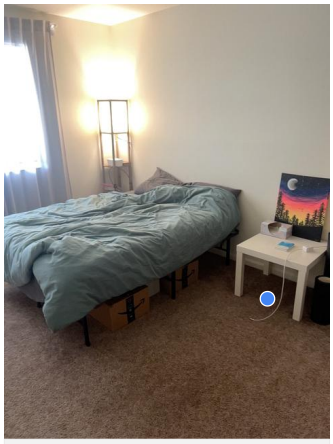
***Reasoning Step = 2***
Please point out the *free area in the direction of the handle of the rightmost green cup*.

***Reasoning Step = 2***
Please point out the *free space in the facing direction of the orange box*.

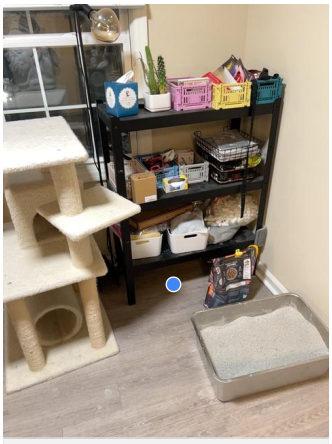


***Reasoning Step = 3***
Please point out the *free space in front of the brown object* on the *shelf*.

***Reasoning Step = 4***
Please point out the *free space* on the *table between* the *pillow and* the *brown bowl*.

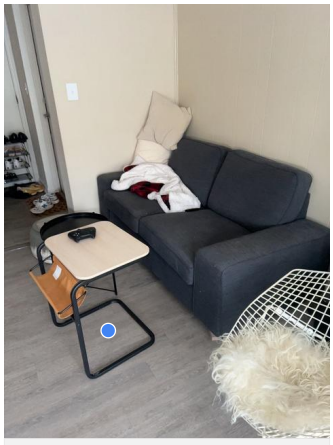


***Reasoning Step = 2***
Please point out the *free space below the white table*.

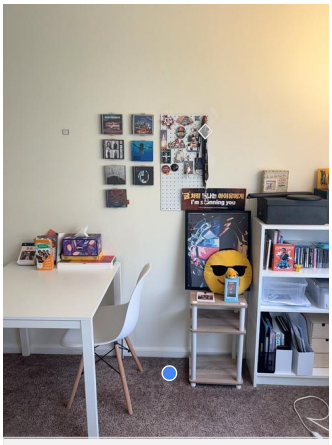


***Reasoning Step = 2***
Please point out the *free space below the black shelf*.

***Reasoning Step = 2***
Please point out the *free space inside the closest pot*.



***Reasoning Step = 2***
Please point out the *free space below* the *table*.



***Reasoning Step = 2***
Please point out the *free space between the white chair and the brown shelf*.

Figure 46: Some Placement Examples. The model is asked to identify a valid free space based on spatial reference. The blue point shows the *RoboRefer*'s prediction (all correct).

***Reasoning Step = 2***

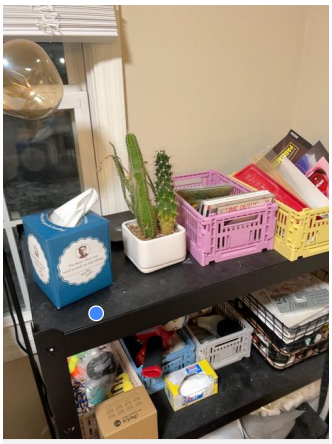
Please point out the *free space in the direction of the white side of the can*.

***Reasoning Step = 3***

Please point out the *free space between* the *scissors and* the *microwave*.

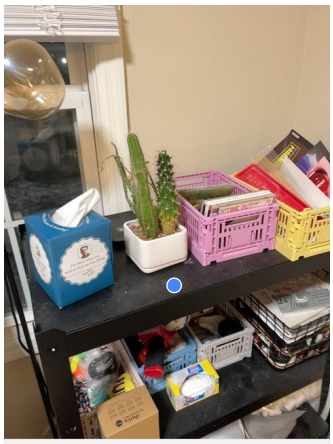
***Reasoning Step = 3***

Please point out the *free space between* the *bathtub and* the *toilet*.



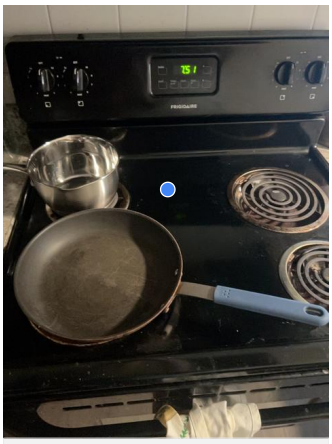
***Reasoning Step = 4***

Please point out the *free space in front of the blue box which is on the top level* of the *shelf*.



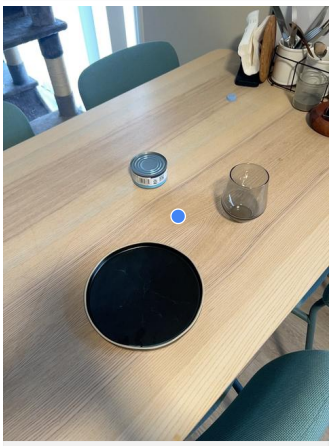
***Reasoning Step = 4***

Please point out the *free space in front of the white vase which is on the top level* of the *shelf*.



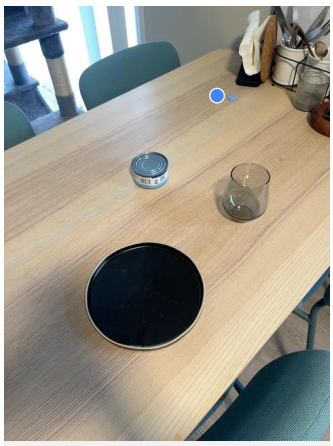
***Reasoning Step = 2***

Please point out the *free space on the right of the farthest pot*.



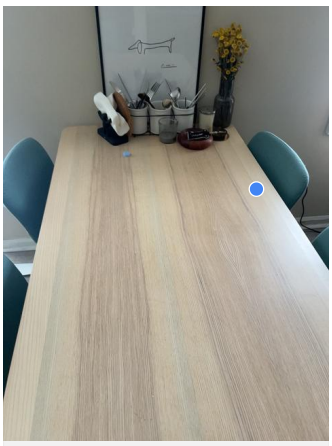
***Reasoning Step = 4***

Please point out the *free space between* the *black plate*, *blue can and closest water glass*.



***Reasoning Step = 2***

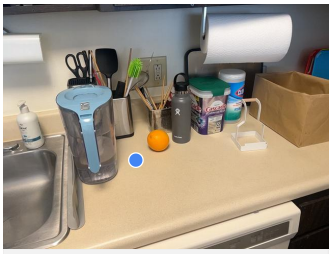
Please point out the *free space in the top corner of* the *table*.



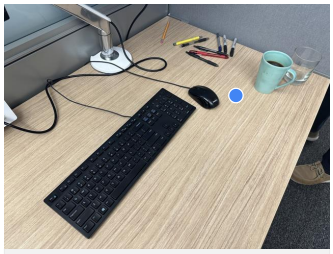
***Reasoning Step = 3***

Please point out the *free area* on the *table in facing direction of the second chair from the front on the right side*.

Figure 47: Some Placement Examples. The model is asked to identify a valid free space based on spatial reference. The blue point shows the *RoboRefer*'s prediction (all correct).

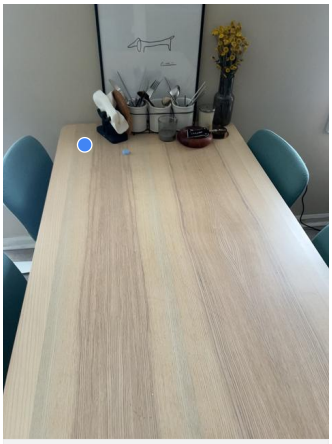


***Reasoning Step = 3***
Please point out the *free spot between the blue water kettle and* the *orange*.



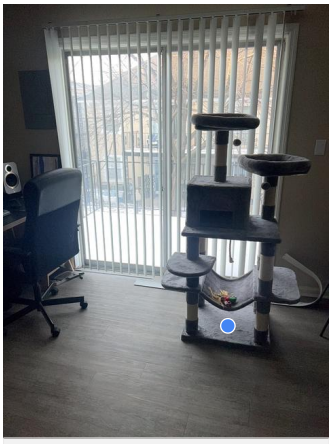
***Reasoning Step = 3***
Please point out the *free space between* the *mouse and* the *green cup*.

***Reasoning Step = 3***
Please point out the *free space between* the *pot and* the *black box* .

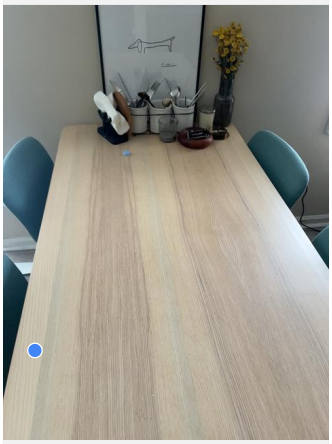


***Reasoning Step = 2***
Please point out the *free area in the top-left corner of* the *table*.

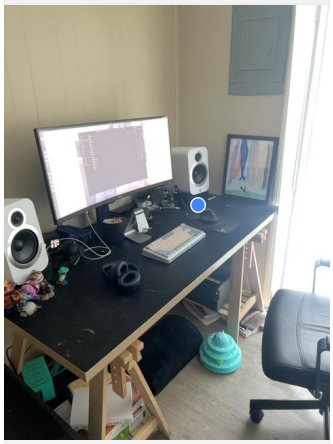
***Reasoning Step = 3***
Please point out the *free space on the third level from the top of* the *cat tree*.



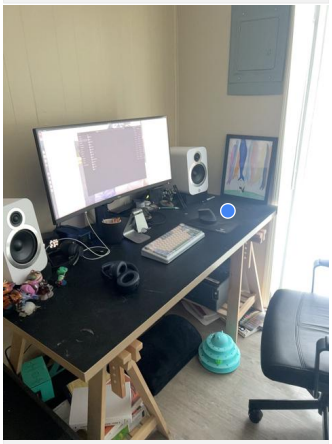
***Reasoning Step = 2***
Please point out the *free space on the lowest shelf of* the *cat tree*.



***Reasoning Step = 3***
Please point out the *free area* on the *table* that the *first chair from the front on the left side is directly facing*.

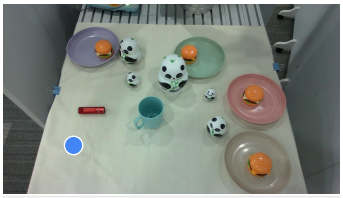


***Reasoning Step = 5***
Please point out the *free space* on the *table between* the *speaker* to *the right of* the *monitor and* the *mouse*.



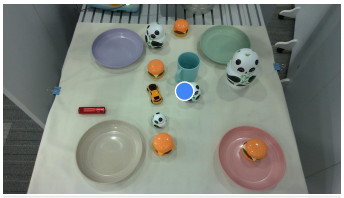
***Reasoning Step = 4***
Please point out the *free space on the right part* of the *table between* the *mouse and* the *picture frame*.

Figure 48: Some Placement Examples. The model is asked to identify a valid free space based on spatial reference. The blue point shows the *RoboRefer*'s prediction (all correct).



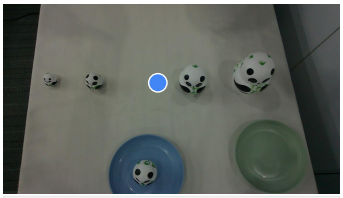
***Reasoning Step = 3***

Find a location in *blank space* located *0.2 meters to the left* of the *table*.



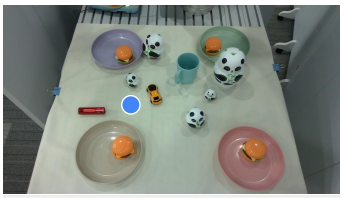
***Reasoning Step = 2***

Identify the *panda* that is the *closest* to the *cup*.



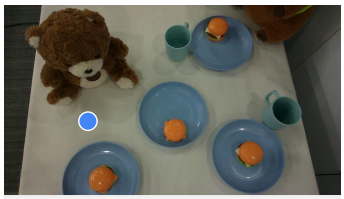
***Reasoning Step = 5***

Where should an additional *panda* be *placed in a line*? Please mark the location on the desktop.



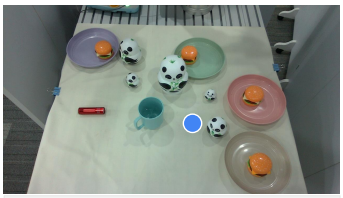
***Reasoning Step = 3***

Please choose a spot located in *the free space between* the *car and* the *flashlight*.



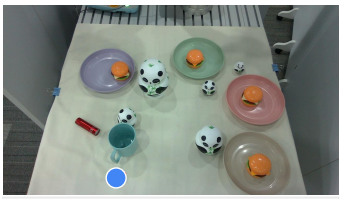
***Reasoning Step = 3***

Locate a point within *vacant space* located *0.05 meters* to the *front-facing side* of the *teddy bear*.



***Reasoning Step = 3***

Locate a point within *vacant space* located *0.13 meters* to the *right* of the *cup*.

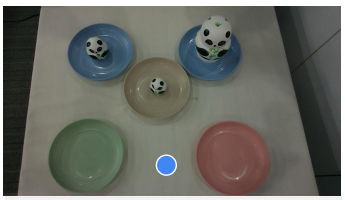


***Reasoning Step = 2***

Locate a point within the *vacant space* located *on the handle side* of the *cup*.

***Reasoning Step = 3***

Please indicate the *cup* located at the *left edge* on the *desktop*.



***Reasoning Step = 3***

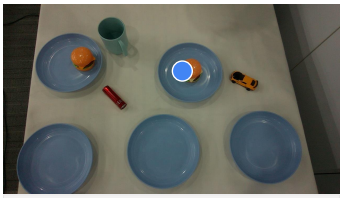
Point to the *free space between* the *green and pink plates*.

***Reasoning Step = 2***

Please indicate the *cup* located at the *near-left corner* on the *desktop*.

***Reasoning Step = 2***

Please indicate the *hamburger* located at the *far-left corner* on the *desktop*.



***Reasoning Step = 2***

Point to the *hamburger between* the *car and* the *mug*.

Figure 49: Visualization of *RoboRefer*'s prediction (blue point) in the real-world evaluation.

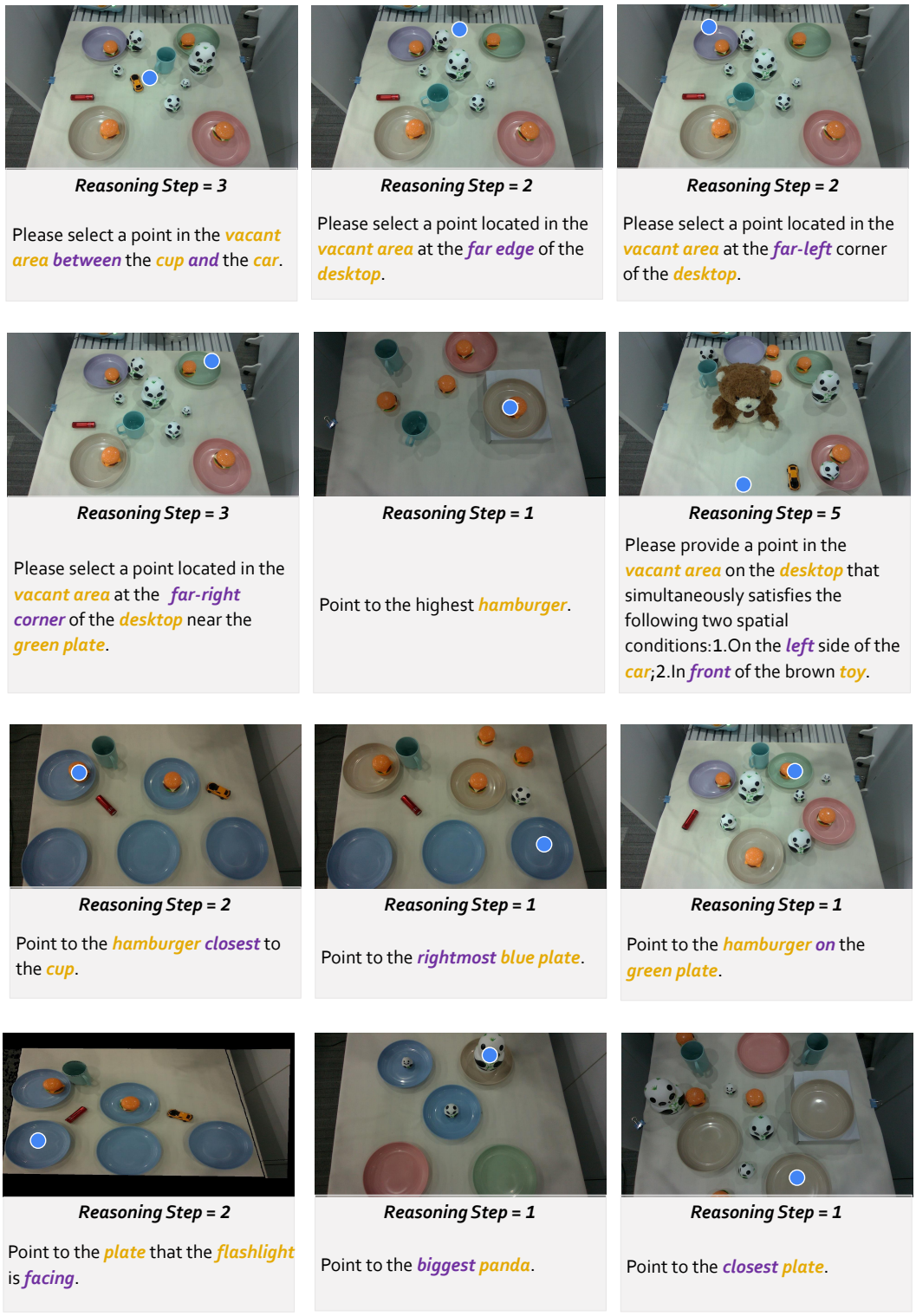
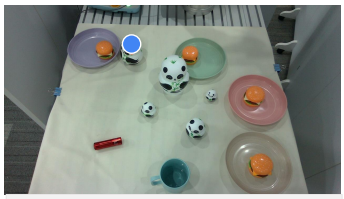



Figure 50: Visualization of *RoboRefer*'s prediction (blue point) in the real-world evaluation.

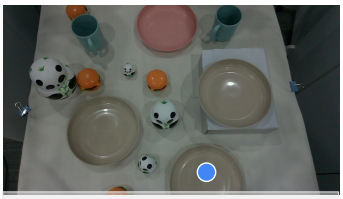


***Reasoning Step = 1***

Point to the ***farthest panda***.

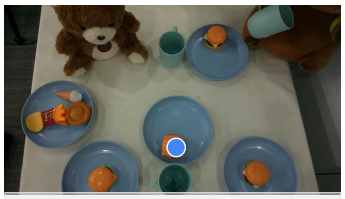
***Reasoning Step = 1***

Point to the ***biggest panda***.



***Reasoning Step = 1***

Point to the ***closest plate***.

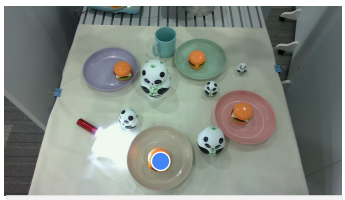


***Reasoning Step = 2***

Point to the ***hamburger*** that is ***first close*** to the ***nearest*** mug to the ***camera***.

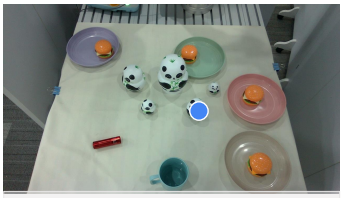
***Reasoning Step = 3***

Point to the ***free space*** on the ***front-facing*** side of the ***teddy bear***.



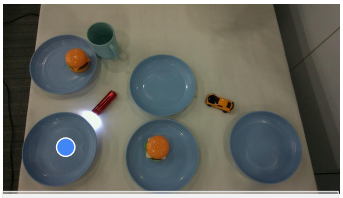
***Reasoning Step = 1***

Point to the ***hamburger*** that the ***flashlight*** is ***illuminating***.



***Reasoning Step = 1***

Point to the ***nearest panda***.



***Reasoning Step = 2***

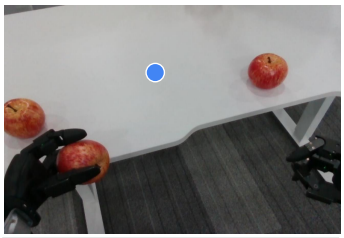
Point to the ***plate illuminated*** by the ***flashlight***.

***Reasoning Step = 1***

Point to the ***leftmost panda***.

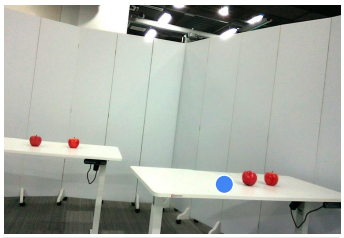
***Reasoning Step = 2***

Pick the ***apple in front of the logo*** side of the ***leftmost cup***.



***Reasoning Step = 3***

Place the ***apple*** on the table, ***aligned with the apple row***.



***Reasoning Step = 1***

Navigate to the ***nearest table***.

Figure 51: Visualization of *RoboRefer*'s prediction (blue point) in the real-world evaluation.