RoboSpatial: Teaching Spatial Understanding to 2D and 3D Vision-Language Models for Robotics

Chan Hee Song[1] **Valts Blukis**[2] **Jonathan Tremblay**[2]
Stephen Tyree[2] **Yu Su**[1] **Stan Birchfield**[2]
[1]The Ohio State University, [2]NVIDIA

https://chanh.ee/RoboSpatial

Abstract: Generalist robot policies require strong spatial priors to operate reliably across diverse environments, enabling them to perceive, reason, and act within 3D space from multiple perspectives. Vision-language models (VLMs) are promising backbones for such policies but are limited by training on generic web-scale image–text datasets that lack rich, multi-frame spatial cues for manipulation. One example is *reference frame comprehension*—deciding whether to reason in egocentric, world-centric, or object-centric coordinates—which is critical for precise, context-aware actions. We introduce RoboSpatial, a large-scale dataset built from real indoor and tabletop 3D scans paired with egocentric RGB views, containing 1M images, 5k scans, and 3M annotated spatial relations spanning object–object, object–space, and object–compatibility reasoning. Its 2D/3D-ready design supports learning priors that generalize across viewpoints, scales, and task contexts. Models trained on RoboSpatial achieve significant gains in spatial reasoning benchmarks and robot manipulation, demonstrating how targeted spatial priors enhance the generalization and reliability of robot policies.

Keywords: Spatial Reasoning, Robot Manipulation, Vision-Language Models

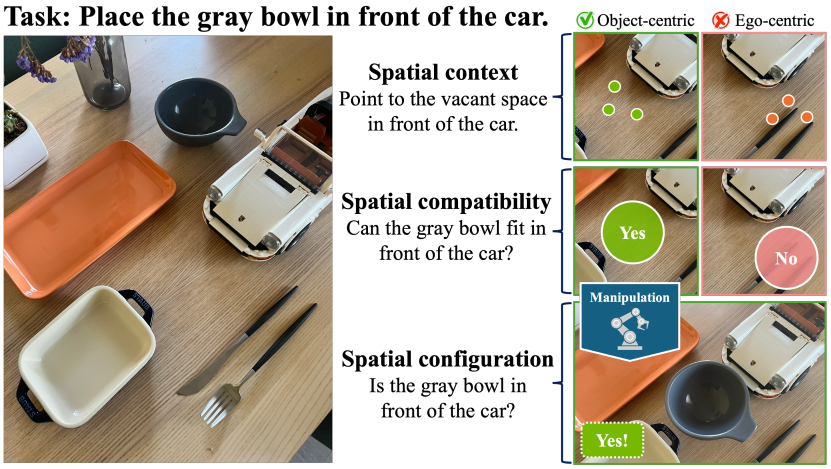


Figure 1: RoboSpatial dataset facilitates 3D spatial reasoning for robot manipulation, enabling human-aligned reasoning in the correct reference frame for grounding, planning, and detection.

1 Introduction

The rise of vision-language models (VLMs) has created new opportunities for agents to interpret and act on the visual world using natural language, with applications in robotics and augmented

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

Dataset	3D scans	Embodied	Ref. frames	Compatibility	Domain	#Scans	#Images	#Spatial QAs
EmbSpatial-Bench [21]	✓	✓	✗	✗	Indoor	277	2k	4k
Visual Spatial [22]	✗	✗	✓	✗	MSCOCO	0	10k	10k
SpatialRGPT-Bench [18]	✗	✗	✗	✓	Indoor, AV	0	1.4k	1.4k
BLINK-Spatial [23]	✗	✗	✓	✗	Generic	0	286	286
What's up [14]	✗	✗	✗	✗	Generic	0	5k	10k
Spatial-MM [15]	✗	✗	✓	✗	Generic	0	2.3k	2.3k
RoboSpatial	✓	✓	✓	✓	Indoor, tabletop	5k	1M	3M

Table 1: Comparison with other spatial reasoning datasets that include object-centric spatial relationships.

reality (AR). In robotics, VLMs enable grounded scene understanding [1, 2], manipulation [3], and policy code generation [4, 5]; in AR, they support object labeling [6], action recognition [7, 8], and temporal grounding [9].

Despite these advances, VLMs [10, 11, 12] still fall short in *spatial understanding* [13, 14, 15, 16]. They struggle with nuanced object relationships, e.g. not just recognizing a "bowl on the table" but reasoning where it should be placed for accessibility or fit. Furthermore, current datasets rarely capture *reference frame* understanding—how spatial relations shift with first-person, object-centric, or scene-level perspectives—critical for real-world interaction.

Recent works target spatial reasoning but fall short in embodied settings. SpatialVLM [17] and SpatialRGPT [18] train on web images with perception-generated annotations, limiting generalization to robot-captured views that lack absolute scale cues. Pointing models like RoboPoint [19] and Molmo [20] predict 2D coordinates for objects or free space but often ignore object-centric reference frames or real-world placement constraints (e.g. whether the gray bowl in Fig. 1 fits in front of the car).

We hypothesize that the key bottleneck is the lack of suitable training data for robotics (Table 1). To address this, we introduce **RoboSpatial**, a dataset for training VLMs in spatial reasoning for robot applications. Using existing indoor scene and tabletop RGBD datasets, we generate targeted QA pairs in three categories: **Spatial context** — predict points in free space for placing objects (e.g. "Where on the table can I put the plate?"); **Spatial compatibility** — binary check if a location fits an object; **Spatial configuration** — binary check if a spatial relation holds (e.g. "Is the mug to the left of the laptop?"). Each QA is posed from three reference frames: (a) ego-centric, (b) world-centric, and (c) object-centric, enabling flexible interpretation of spatial instructions. Applied to existing 3D datasets, this yields **1M** images, **5k** scans, and **3M** spatial relations, with paired 2D egocentric and 3D data for both 2D and 3D readiness.

We evaluate RoboSpatial on SOTA 2D and 3D VLMs. Models trained on it outperform baselines on **RoboSpatial-Home** (a manually collected indoor dataset), BLINK-Spatial [23], and SpatialBot [24]. These benchmarks test real-world skills like object rearrangement and spatial QA, showing consistent gains across tasks. Using RoboSpatial, we also compare 2D vs. 3D VLMs; while 3D shows promise, differing pretraining setups prevent a definitive conclusion.

Our contributions:

- RoboSpatial — a spatial reasoning dataset with images, 3D scans, and spatial QAs, plus RoboSpatial-Home for evaluation.
- Training on RoboSpatial yields superior spatial reasoning, surpassing SOTA baselines in robot manipulation and indoor QA.
- Comprehensive evaluation of SOTA 2D & 3D VLMs on spatial reasoning tasks in real-world contexts.

2 Related Work

VLMs for Robotics. Vision-language models (VLMs) are increasingly central to robotics, combining visual perception and language understanding for intuitive human-robot interaction and au-

tonomous decision-making. Recent advances include vision-language-action models (VLAs) [25, 26, 27] that translate instructions into executable actions, GPT-4v [10] for high-level task planning [28], and VLM-based approaches for keypoint/mask prediction [29, 30, 31], error analysis [32, 33], and grasp pose prediction [34]. However, integrating VLMs [24, 18, 19] into robotic systems remains challenging, particularly for precise spatial reasoning in dynamic environments [13, 35, 36]. RoboSpatial addresses this gap by providing large-scale pretraining and evaluation resources tailored for spatial reasoning in robotics.

Spatial Understanding with VLMs. Spatial understanding has long been studied in vision and QA tasks [23, 37, 38, 39, 40, 41, 42, 43], yet existing benchmarks have limitations: some focus on simulations [44] or generic imagery [22, 45, 18, 17, 23, 14, 15, 46], others are difficult to evaluate with free-form outputs [44, 21, 47], rely on full 3D scans [48, 49, 50, 47], or omit reference frames [48, 49, 50, 47, 17, 18, 23, 46]. Few target robotics-relevant relationships such as spatial compatibility or context [21, 23, 51, 15, 14, 47, 46].

Building on prior spatial reasoning work [22, 14], which examined reference frames and configurations in generic images [52, 43], we extend to robotics-specific, actionable relationships. We present a large-scale, 2D/3D-ready dataset generated via an automated pipeline, and demonstrate its use in training VLMs for both in-domain and out-of-domain spatial reasoning. Our goal is to lower the barrier for exploring spatial understanding directly applicable to robotic workflows such as planning and verification.

3 Approach

We begin by explaining the selection of three spatial relationships: spatial context, spatial compatibility, and spatial configuration. Next, we describe the data generation pipeline used to construct the RoboSpatial. Figure 2 provides an overview of the dataset.

3.1 Spatial Relationships

The dataset is organized around three core spatial relationships that we believe address the essential aspects of spatial reasoning for robotic tasks: spatial *context*, spatial *compatibility*, and spatial *configuration*. *Context* allows robots to assess the relationship between objects and their surrounding space, facilitating the identification of empty or occupied areas, which is relevant for downstream applications such as path planning and obstacle avoidance. *Compatibility* focuses on whether objects can coexist or interact without conflict in a given space, which is vital for object placement, assembly, and operational safety. *Configuration* enables robots to understand and interpret the relative positioning of objects, which is crucial for directing navigation, manipulation, and interaction within complex environments. Together, these spatial relationships provide a more nuanced and practical framework for robotic applications than metrics like distance—which is hard to normalize across different scales, environments, and tasks—thereby enabling robots to perform complex tasks with greater reliability.

3.2 Dataset Generation

Our pipeline generates a large-scale, high-accuracy spatial reasoning dataset with minimal human intervention, using heuristics grounded in 3D geometry and 2D image views. It takes as input a scene dataset $\mathcal{D}_s$ with RGB images, camera poses, and oriented 3D bounding boxes with semantic labels, and outputs $\mathcal{D}$, where each entry $d_i = \langle I_i, q_i, a_i, l_i \rangle$ contains an image, question, answer, and reference frame $l_i \in \textit{ego}, \textit{world}, \textit{object}$. Questions cover spatial configuration, context, or compatibility. An auxiliary grounding dataset links object descriptions to 2D bounding boxes for reliable reference resolution. We detail the process in two stages, separating 3D relation extraction from 2D image-space target generation.

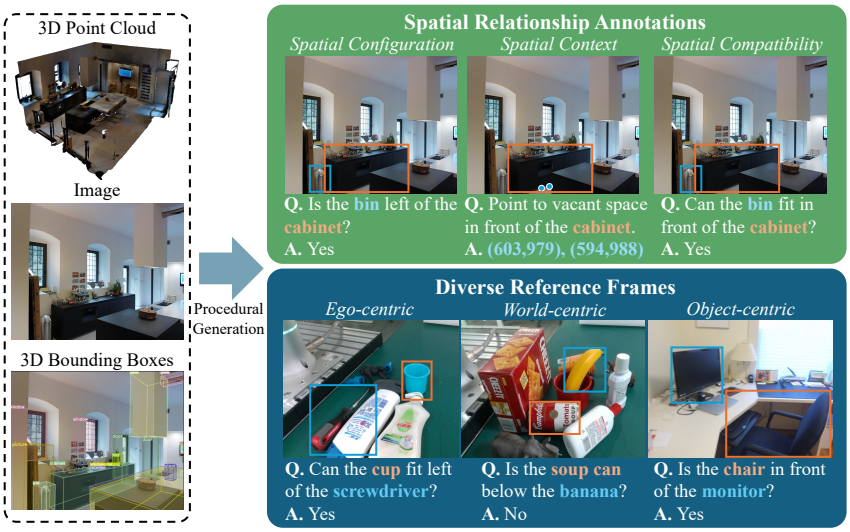


Figure 2: Overview of the RoboSpatial dataset. We automatically generate spatial relationship annotations from existing datasets with 3D point clouds, egocentric images, and 3D bounding box annotations. We create question/answer pairs covering three classes of spatial relationships, three spatial reference frames, and both binary (yes/no) and numeric (e.g., 2D image points) answers. From 1M images and 5k scans, we generate over 3M spatial question/answer pairs.

3.2.1 Stage 1: 3D Spatial Relation Extraction

The first stage involves extracting spatial relationships between objects or between objects and free space, based on 3D geometry. Each spatial relation is defined as $s_i = \langle I_i, a_i, t_i, r_i, l_i \rangle$, where I_i is the source image, a_i is the anchor object, t_i is the target object or a sampled point in free space, $r_i \in \{\textit{left}, \textit{right}, \textit{above}, \textit{below}, \textit{front}, \textit{behind}\}$ is the relation preposition, and $l_i \in \{\textit{ego}, \textit{world}, \textit{object}\}$ denotes the reference frame.

We use oriented 3D bounding boxes, provided by the source dataset, to compute spatial relationships. Each bounding box includes both the 3D location and heading of the object. The object's orientation is defined by the heading vector of the bounding box, aligned with the object's front-facing direction. Using this orientation, we determine the appropriate directional region (e.g., front, left) relative to the reference frame. For instance, a relation such as "in front of (anchor object) (object frame)" refers to the positive direction along the anchor object's heading vector. These relationships are calculated independently for each of the three reference frames: the world frame is aligned with the dataset-level coordinate system; the ego frame is defined by the camera pose (i.e., camera-centered); and the object frame is defined by the local orientation of the anchor object.

The camera extrinsics are used to transform coordinates between reference frames. Although the method does not require point clouds or meshes, it relies on camera intrinsics and extrinsics to project between 2D and 3D and to ensure consistent reference frame reasoning. For each spatial configuration task, we evaluate all visible object pairs that appear uniquely in the image, avoiding duplicate instances to minimize ambiguity. The resulting relationships are binary (True/False) and specify whether the spatial condition holds for the given object pair.

3.2.2 Stage 2: 2D Spatial Point and Region Sampling

In the second stage, we generate 2D image-space annotations for spatial context and spatial compatibility tasks. These rely on the 3D bounding box layout and calibrated camera parameters to map spatial relationships into image coordinates.

For spatial context, we construct a top-down occupancy map of the scene by marking regions occupied by 3D bounding boxes. We then randomly sample 3D points in empty space that lie in a specified directional relation to the anchor object, following the same frame-dependent heuristics as

in the configuration task. These points are projected into the image plane using the camera intrinsics. To ensure the points are valid, we filter out samples that are obstructed or occluded based on line-of-sight from the camera. Specifically, we perform raycasting from the camera center to each sampled 3D point, and discard points whose rays intersect any occupied bounding box volumes before reaching the target location. The final answer is a list of 2D (x, y) image coordinates that satisfy the spatial context constraint.

Spatial compatibility extends this idea by checking whether a target object can fit within the sampled region. We simulate placing a virtual bounding box, matching the size of the target object, at the candidate location on the ground plane. A region is considered compatible if the simulated placement does not intersect with any existing bounding boxes in the scene and provides at least a 10 cm margin along each axis. The simulation allows for translation and in-plane rotation of the object. The answer to this task is binary (True/False), indicating whether the region can accommodate the object.

3.2.3 Question-Answer Generation

From the extracted spatial relations s_i, we generate question–answer pairs d_i using templates of the form:

{TARGET} {RELATION} {ANCHOR} {REF. FRAME}

where relation and frame are defined in Section 3.2.1. Templated approach avoids ambiguity and minimizes reliance on commonsense, ensuring models learn from visual grounding.

Each relation type—context, compatibility, configuration—has its own format: configuration and compatibility yield binary (True/False) answers, while context returns valid 2D coordinate lists. To improve object reference resolution, we also generate an auxiliary grounding dataset linking object descriptions to 2D bounding boxes, obtained by projecting 3D boxes with camera parameters. This supervision is used during training (Appendix B.4).

Using this pipeline, we generate around 3 million spatial relationships and their associated question-answer pairs. This scale is an order of magnitude larger than prior spatial reasoning datasets (see Table 1).

4 Experiments

4.1 Implementation

Datasets. We construct RoboSpatial by applying our pipeline to three indoor scene datasets—ScanNet [55], Matterport3D [56], and 3RScan [57]—and two tabletop datasets—HOPE [58] and GraspNet-1B [59]. 3D bounding box annotations and embodied images are retrieved from EmbodiedScan [51], yielding **3M** spatial QA pairs from **5k** scans and **1M** images (Appendix 5 contains detailed splits).

Models. We evaluate RGB-only VLMs: VILA-1.5-8B [12], LLaVA-NeXT-8B [11], SpaceLLaVA-13B [17], RoboPoint-13B [19], Molmo-7B [20], and GPT-4o [10]. Models requiring external masks (e.g., SpatialRGPT [18]) are excluded. For 3D input, we test 3D-LLM [53] (multi-view RGB) and LEO [54] (segmented object point clouds).

Fine-tuning. Open-source models are evaluated in both zero-shot and fine-tuned settings. Fine-tuning uses RoboSpatial plus an auxiliary grounding dataset to improve object reference resolution (Appendix D.4).

In-domain evaluation. RoboSpatial-Val is a held-out split with 6k questions (2k per spatial relation type). Binary tasks are scored by accuracy; coordinate tasks are correct if predictions fall within the convex hull of ground-truth points (Table 6).

Model	RoboSpatial-Home			BLINK	SpatialBench
	Configuration	Context	Compatibility	Accuracy	Accuracy
2D VLMs					
VILA [12]	57.8	0.0	69.0	72.7	53.0
+RoboSpatial	65.9 ↑	15.6 ↑	78.0 ↑	79.7 ↑	**73.6** ↑
LLaVA-NeXT [11]	68.3	0.0	70.5	71.3	55.9
+RoboSpatial	**78.9** ↑	19.7 ↑	80.1 ↑	79.0 ↑	70.6 ↑
SpaceLLaVA [17]	61.0	2.5	61.0	76.2	47.1
+RoboSpatial	71.6 ↑	13.1 ↑	72.4 ↑	**81.8** ↑	67.7 ↑
RoboPoint [19]	69.9	19.7	70.5	63.6	44.1
+RoboSpatial	78.0 ↑	**31.1** ↑	**81.0** ↑	70.6 ↑	64.7 ↑
3D VLMs					
3D-LLM [53]	39.8	0.0	35.2	N/A	N/A
+RoboSpatial	55.2 ↑	8.2 ↑	52.3 ↑	N/A	N/A
LEO [54]	51.2	0.0	38.1	N/A	N/A
+RoboSpatial	64.2 ↑	10.0 ↑	57.1 ↑	N/A	N/A
Not available for fine-tuning					
Molmo [20]	58.6	0.1	18.1	67.1	55.9
GPT-4o [10]	77.2	5.7	58.1	76.2	70.6

Table 2: Results on an out-of-domain test split comparing prior art VLMs. The results show improved (↑) spatial understanding capabilities on similar domains. Bolded number is the best result for the column.

Cross-domain evaluation. We train on either indoor or tabletop data and test on the other to measure transfer across scene types (Table 3).

Out-of-domain evaluation. We benchmark on: (1) RoboSpatial-Home — 350 manually written questions over real RGB-D indoor scenes, (2) BLINK [23] — spatial subset only, and (3) SpatialBench [24] — position category. Additional benchmark details are in Appendix B.2.

4.2 Results

We evaluate the effectiveness of RoboSpatial in improving spatial reasoning capabilities in VLMs across held-out and out-of-domain benchmarks. In this section, we focus on analyzing the model's generalization and understanding of spatial relationships. We address the following questions:

How well does RoboSpatial training generalize to unseen spatial relationships? Although RoboSpatial consists of template-generated QA pairs with a fixed set of spatial prepositions, we observe in table 2 that models trained on it can generalize to spatial relationships not explicitly included in the training set. This is particularly evident in evaluations on the BLINK dataset [23], which contains diverse prepositions such as "under," "next to," and "far away." We attribute this generalization to the fact that RoboSpatial encompasses all six principal directions in 3D space (along the x, y, and z axes). Generalizing to new prepositions often requires mapping linguistic expressions (e.g., "on top of," "under") to these spatial primitives—a task at which LLMs are naturally proficient. For example, "on top of" often refer to "above" in a world-centric frame, while "under" maps to "below." Moreover, prepositions such as "next to" or "beside" imply proximity between objects. Because RoboSpatial includes questions that require generating points near a reference object, it implicitly teaches the concept of closeness. This enables trained models to understand these proximity-based relationships, even if they are not explicitly represented during training.

Do RoboSpatial-trained models understand nuanced perspectives? Spatial references in natural language often imply specific reference frames. For instance, "in front of the car" typically refers to the direction of the car's front hood. In RoboSpatial-Home, we omit explicit frame specifications in the questions to evaluate whether models can align with the implicit reference frame intended by the questioner. We find that models trained with RoboSpatial can often infer the

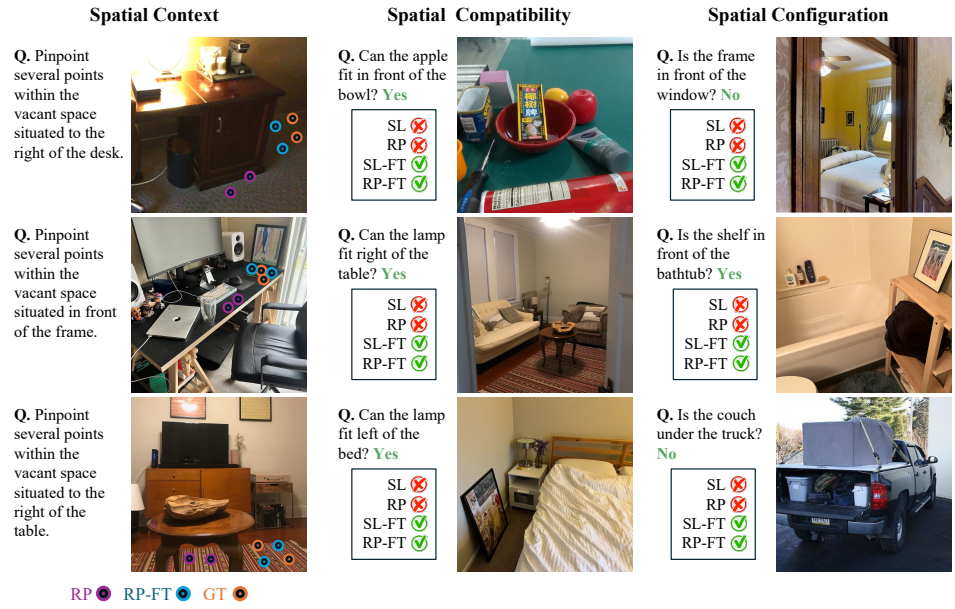


Figure 3: In-domain (RoboSpatial-Val, top) and out-of-domain (RoboSpatial-Home, BLINK [23], middle and bottom) results for RoboSpatial-trained models. Two models shown: SL (SpaceLLaVA [17]) and RP (RoboPoint [19]); the -FT suffix indicates fine-tuning on RoboSpatial. Correct answers in green. All images except bottom-right in the out-of-domain rows are from RoboSpatial-Home.

	Indoor → Tabletop	Tabletop → Indoor
RoboPoint [19]	38.7	38.2
+RoboSpatial	48.9 ↑	51.3 ↑
LEO [54]	41.9	43.7
+RoboSpatial	47.2 ↑	54.5 ↑

Table 3: Cross-dataset generalization results between indoor and tabletop environments on RoboSpatial-Val.

Model	Success (%)
Open-source	
LLaVA-NeXT [11]	23.7
+ RoboSpatial	**52.6** ↑
RoboPoint [19]	44.7
+ RoboSpatial	46.2 ↑
Not available for fine-tuning	
Molmo [20]	43.8
GPT-4o [10]	46.9

Table 4: Robot experiment results.

correct frame of reference, suggesting that they have learned to associate object geometries and orientations with spatial language. Figure 3 shows examples such as "Is the frame in front of the window?", where the model accurately identifies the intended spatial relation.

Are 3D VLMs better at learning spatial relationships than 2D VLMs? The findings in table 2 suggest that 3D VLMs tend to outperform 2D counterparts in spatial reasoning tasks, likely due to their ability to directly utilize depth information. However, this comparison is not entirely fair: models like 3D-LLM [53] and LEO [54] are pretrained on RGB-D indoor scan datasets, some of which overlap with the environments used in the source datasets (e.g., Matterport3D, ScanNet). This gives them prior exposure to scene geometry and object layouts, which may bias their performance. To support more controlled and fair comparisons in the future, we designed RoboSpatial to be compatible with both 2D and 3D modalities, allowing researchers to investigate the impact of modality, architecture, and pretraining data under unified evaluation protocols.

4.3 Real Robot Experiments

We design a suite of tabletop manipulation tasks requiring spatial reasoning. The setup includes a Kinova Jaco robot [61], paired with a ZED2 camera for RGB-D perception. The robot system implements actions to *pick* and *place* objects on the table using cuRobo [60] for motion planning.

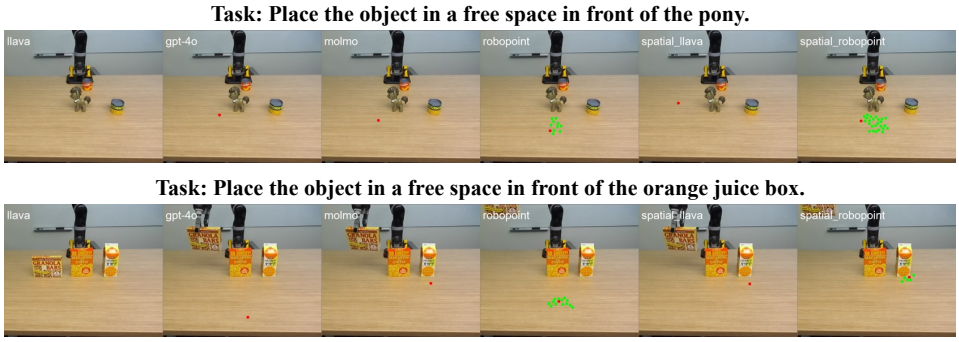


Figure 4: Robotics experiments: the red dot shows the model output (if not present, the model failed to provide a valid point in the image); green dots are used to show when a model outputs multiple points. The robot motion generator, cuRobo [60], is used to grasp the item referenced by the generated point. The *spatial-* prefix indicates model trained with RoboSpatial.

Tasks include spatial questions that require a yes/no answer, and pick-and-place instructions that require successfully controlling the robot to complete the task. We adopt a modular design, where the VLM is queried for spatial understanding, and the resulting predictions (e.g., target points) are passed to a separate motion planning system for execution. We use a range of simple, unambiguous objects—colored cubes, cylinders, food items, and toys—to ensure the challenge lies in spatial understanding rather than object recognition (Figure 4). In total, we conducted over 200 model queries. Details of the questions and scene configurations are provided in the Appendix D.5. We evaluate the following VLMs: LLaVA-NeXT [11] and RoboPoint [19], both with and without RoboSpatial training; and two strong baselines, Molmo [20] and GPT-4o [10]. Table 4 and Figure 4 present the results.

Experiments show that LLaVA-NeXT fine-tuned on RoboSpatial achieves the highest success rate across all models. Training with RoboSpatial enhances spatial understanding in 2D VLMs, enabling the model to correctly interpret instructions such as "place in front of the pony," where placement is aligned with the pony's head direction. It also demonstrates sensitivity to object scale, as in the task "place in front of the orange juice box," where the model places the object at a reasonable distance. In contrast, baseline models such as RoboPoint frequently place objects too far from the target, likely due to limited understanding of spatial proximity. We also observe that spatial failures in 2D VLMs often stem from errors in projecting 2D predictions into 3D. Even a small 2-pixel shift in image space can translate to a 5–10 cm error in the physical world, which is significant in manipulation tasks. Nonetheless, models trained on RoboSpatial produce more accurate predictions, reducing these failure cases and showing the benefit of dataset-driven improvements. Interestingly, GPT-4o performs comparably to RoboSpatial-trained RoboPoint. We attribute this to GPT-4o's broader language understanding and instruction-following ability, which partially compensates for its lack of task-specific spatial training. Looking forward, promising directions include investigating how viewpoint affects 2D spatial predictions, and developing 3D VLMs that can reason over partial point clouds—removing the need for complete 3D scans and making deployment in real-world systems more feasible.

5 Conclusion

We introduce RoboSpatial, and RoboSpatial-Home, a large-scale 2D/3D spatial understanding training and evaluation dataset tailored for robotics. Experimental results show that models trained with RoboSpatial are able to understand spatial relationships, generalize to unseen relationships, and infer nuanced reference frames, making them applicable in a wide range of tasks that require spatial understanding. We further demonstrate the real-world applicability of RoboSpatial with robot experiments. In addition, our automatic data generation pipeline can be used to extend the dataset to new data sources and spatial relations. We show that RoboSpatial has the potential to serve as a foundation for broader applications in robotics which require spatial understanding.

References

[1] K. Fang, F. Liu, P. Abbeel, and S. Levine. Moka: Open-world robotic manipulation through mark-based visual prompting. *Robotics: Science and Systems (RSS)*, 2024.

[2] A. Zook, F.-Y. Sun, J. Spjut, V. Blukis, S. Birchfield, and J. Tremblay. Grs: Generating robotic simulation tasks from real-world images, 2024. URL https://arxiv.org/abs/2410.15536.

[3] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Yang, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Bharadhwaj, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Vakil, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Mart'in-Mart'in, R. Baijal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Tulsiani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Kumar, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Dou, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[4] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530, 2023. doi:10.1109/ICRA48891.2023.10161317.

[5] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500, 2023. doi:10.1109/ICRA48891.2023.10160591.

[6] A. Suglia, C. Greco, K. Baker, J. L. Part, I. Papaioannou, A. Eshghi, I. Konstas, and O. Lemon. AlanaVLM: A multimodal embodied AI foundation model for egocentric video understand-

ing. In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11101–11122, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.findings-emnlp.649. URL https://aclanthology.org/2024.findings-emnlp.649/.

[7] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, A. Gebreselasie, C. González, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolář, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. R. Puentes, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Z. Zhao, Y. Zhu, P. Arbeláez, D. Crandall, D. Damen, G. M. Farinella, C. Fuegen, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18973–18990, 2022. doi:10.1109/CVPR52688.2022.01842.

[8] X. Gong, S. Mohan, N. Dhingra, J.-C. Bazin, Y. Li, Z. Wang, and R. Ranjan. Mmg-ego4d: Multi-modal generalization in egocentric action recognition. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6481–6491, 2023. doi:10.1109/CVPR52729.2023.00627.

[9] Y. Huang, J. Xu, B. Pei, Y. He, G. Chen, M. Zhang, L. Yang, Z. Nie, J. Liu, G. Fan, D. Lin, F. Fang, K. Li, C. Yuan, X. Chen, Y. Wang, Y. Wang, Y. Qiao, and L. Wang. An egocentric vision-language model based portable real-time smart assistant, 2025. URL https://arxiv.org/abs/2503.04250.

[10] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Łukasz Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Łukasz Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H. P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez,

N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, and B. Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[11] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306, June 2024.

[12] J. Lin, H. Yin, W. Ping, P. Molchanov, M. Shoeybi, and S. Han. VILA: On Pretraining for Visual Language Models . In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26679–26689, Los Alamitos, CA, USA, June 2024. IEEE Computer Society. doi:10.1109/CVPR52733.2024.02520. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.02520.

[13] Y. Yamada, Y. Bao, A. K. Lampinen, J. Kasai, and I. Yildirim. Evaluating spatial understanding of large language models, 2024. URL https://arxiv.org/abs/2310.14540.

[14] A. Kamath, J. Hessel, and K.-W. Chang. What's "up" with vision-language models? investigating their struggle with spatial reasoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[15] F. Shiri, X.-Y. Guo, M. G. Far, X. Yu, R. Haf, and Y.-F. Li. An empirical analysis on spatial reasoning capabilities of large multimodal models. In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21440–21455, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.emnlp-main.1195.

[16] A. Majumdar, A. Ajay, X. Zhang, P. Putta, S. Yenamandra, M. Henaff, S. Silwal, P. Mcvay, O. Maksymets, S. Arnaud, K. Yadav, Q. Li, B. Newman, M. Sharma, V. Berges, S. Zhang, P. Agrawal, Y. Bisk, D. Batra, M. Kalakrishnan, F. Meier, C. Paxton, S. Sax, and A. Rajeswaran. Openeqa: Embodied question answering in the era of foundation models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[17] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14455–14465, June 2024.

[18] A.-C. Cheng, H. Yin, Y. Fu, Q. Guo, R. Yang, J. Kautz, X. Wang, and S. Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[19] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox. Robopoint: A vision-language model for spatial affordance prediction in robotics. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=GVX6jpZOhU.

[20] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, M. Salehi, N. Muennighoff, K. Lo, L. Soldaini, J. Lu, T. Anderson, E. Bransom, K. Ehsani, H. Ngo, Y. Chen, A. Patel, M. Yatskar, C. Callison-Burch, A. Head, R. Hendrix, F. Bastani, E. VanderBilt, N. Lambert, Y. Chou, A. Chheda, J. Sparks, S. Skjonsberg, M. Schmitz, A. Sarnat, B. Bischoff, P. Walsh,

C. Newell, P. Wolters, T. Gupta, K.-H. Zeng, J. Borchardt, D. Groeneveld, J. Dumas, C. Nam, S. Lebrecht, C. Wittlif, C. Schoenick, O. Michel, R. Krishna, L. Weihs, N. A. Smith, H. Hajishirzi, R. Girshick, A. Farhadi, and A. Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024.

[21] M. Du, B. Wu, Z. Li, X. Huang, and Z. Wei. EmbSpatial-bench: Benchmarking spatial understanding for embodied tasks with large vision-language models. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 346–355, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.acl-short.33. URL https://aclanthology.org/2024.acl-short.33.

[22] F. Liu, G. Emerson, and N. Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023. doi:10.1162/tacl_a_00566. URL https://aclanthology.org/2023.tacl-1.37.

[23] X. Fu, Y. Hu, B. Li, Y. Feng, H. Wang, X. Lin, D. Roth, N. A. Smith, W.-C. Ma, and R. Krishna. Blink: Multimodal large language models can see but not perceive. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 148–166. Springer, 2024. doi:10.1007/978-3-031-73337-6_9. URL https://doi.org/10.1007/978-3-031-73337-6_9.

[24] W. Cai, Y. Ponomarenko, J. Yuan, X. Li, W. Yang, H. Dong, and B. Zhao. Spatialbot: Precise spatial understanding with vision language models. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[25] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. In P. Agrawal, O. Kroemer, and W. Burgard, editors, *Proceedings of the 8th Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

[26] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, Q. Vuong, V. Vanhoucke, H. Tran, R. Soricut, A. Singh, J. Singh, P. Sermanet, P. R. Sanketi, G. Salazar, M. S. Ryoo, K. Reymann, K. Rao, K. Pertsch, I. Mordatch, H. Michalewski, Y. Lu, S. Levine, L. Lee, T.-W. E. Lee, I. Leal, Y. Kuang, D. Kalashnikov, R. Julian, N. J. Joshi, A. Irpan, B. Ichter, J. Hsu, A. Herzog, K. Hausman, K. Gopalakrishnan, C. Fu, P. Florence, C. Finn, K. A. Dubey, D. Driess, T. Ding, K. M. Choromanski, X. Chen, Y. Chebotar, J. Carbajal, N. Brown, A. Brohan, M. G. Arenas, and K. Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In J. Tan, M. Toussaint, and K. Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

[27] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[28] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi. Gpt-4v(ision) for robotics: Multimodal task planning from human demonstration. *IEEE Robotics and Automation Letters*, 9(11):10567–10574, 2024. doi:10.1109/LRA.2024.3477090.

[29] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=9iG3SEbMnL.

[30] Y. Wi, M. Van der Merwe, P. Florence, A. Zeng, and N. Fazeli. Calamari: Contact-aware and language conditioned spatial action mapping for contact-rich manipulation. In *7th Annual Conference on Robot Learning*, 2023.

[31] S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu, Q. Vuong, T. Zhang, T.-W. E. Lee, K.-H. Lee, P. Xu, S. Kirmani, Y. Zhu, A. Zeng, K. Hausman, N. Heess, C. Finn, S. Levine, and B. Ichter. Pivot: iterative visual prompting elicits actionable knowledge for vlms. In *Proceedings of the International Conference on Machine Learning (ICML)*, ICML'24. JMLR.org, 2024.

[32] J. Duan, W. Pumacay, N. Kumar, Y. R. Wang, S. Tian, W. Yuan, R. Krishna, D. Fox, A. Mandlekar, and Y. Guo. AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=JVkdSi7Ekg.

[33] C. H. Song, J. Kil, T.-Y. Pan, B. M. Sadler, W.-L. Chao, and Y. Su. One step at a time: Long-horizon vision-and-language navigation with milestones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15482–15491, June 2022.

[34] H. Huang, F. Lin, Y. Hu, S. Wang, and Y. Gao. Copa: General robotic manipulation through spatial constraints of parts with foundation models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9488–9495, 2024. doi:10.1109/IROS58592.2024.10801352.

[35] L. Xu, S. Zhao, Q. Lin, L. Chen, Q. Luo, S. Wu, X. Ye, H. Feng, and Z. Du. Evaluating large language models on spatial tasks: A multi-task benchmarking study, 2024. URL https://arxiv.org/abs/2408.14438.

[36] J. Wang, Y. Ming, Z. Shi, V. Vineet, X. Wang, Y. Li, and N. Joshi. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=cvaSru8LeO.

[37] D. Azuma, T. Miyanishi, S. Kurita, and M. Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[38] B. Jia, Y. Chen, H. Yu, Y. Wang, X. Niu, T. Liu, Q. Li, and S. Huang. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

[39] A. Suhr, S. Zhou, A. Zhang, I. Zhang, H. Bai, and Y. Artzi. A corpus for reasoning about natural language grounded in photographs. In A. Korhonen, D. Traum, and L. Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, Florence, Italy, July 2019. Association for Computational Linguistics. doi:10.18653/v1/P19-1644. URL https://aclanthology.org/P19-1644/.

[40] L. Salewski, A. S. Koepke, H. P. A. Lensch, and Z. Akata. Clevr-x: A visual reasoning dataset for natural language explanations. In *xxAI - Beyond explainable Artificial Intelligence*, pages 85–104. Springer, 2022.

[41] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

[42] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[43] D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6700–6709, 2019.

[44] E. Szymanska, M. Dusmanu, J.-W. Buurlage, M. Rad, and M. Pollefeys. Space3D-Bench: Spatial 3D Question Answering Benchmark. In *European Conference on Computer Vision (ECCV) Workshops*, 2024.

[45] N. Rajabi and J. Kosecka. Towards grounded visual spatial reasoning in multi-modal vision language models, 2024. URL https://arxiv.org/abs/2308.09778.

[46] K. Ranasinghe, S. N. Shukla, O. Poursaeed, M. S. Ryoo, and T.-Y. Lin. Learning to localize objects improves spatial reasoning in visual-llms. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12977–12987, 2024. doi:10.1109/CVPR52733.2024.01233.

[47] X. Linghu, J. Huang, X. Niu, X. Ma, B. Jia, and S. Huang. Multi-modal situated reasoning in 3d scenes. In *Advances in Neural Information Processing Systems*, 2024. NeurIPS.

[48] Y. Zhang, Z. Xu, Y. Shen, P. Kordjamshidi, and L. Huang. SPARTUN3d: Situated spatial understanding of 3d world in large language model. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=FGMkSL8NR0.

[49] Y. Man, L.-Y. Gui, and Y.-X. Wang. Situational awareness matters in 3d vision language reasoning. In *CVPR*, 2024.

[50] X. Ma, S. Yong, Z. Zheng, Q. Li, Y. Liang, S.-C. Zhu, and S. Huang. Sqa3d: Situated question answering in 3d scenes. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=IDJx97BC38.

[51] T. Wang, X. Mao, C. Zhu, R. Xu, R. Lyu, P. Li, X. Chen, W. Zhang, K. Chen, T. Xue, X. Liu, C. Lu, D. Lin, and J. Pang. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[52] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[53] Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan. 3d-llm: Injecting the 3d world into large language models. In *Advances in Neural Information Processing Systems*, 2023. NeurIPS.

[54] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang. An embodied generalist agent in 3d world. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[55] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[56] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[57] J. Wald, A. Avetisyan, N. Navab, F. Tombari, and M. Niessner. Rio: 3d object instance re-localization in changing indoor environments. In *Proceedings IEEE International Conference on Computer Vision (ICCV)*, 2019.

[58] S. Tyree, J. Tremblay, T. To, J. Cheng, T. Mosier, J. Smith, and S. Birchfield. 6-dof pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark. In *International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[59] H.-S. Fang, C. Wang, M. Gou, and C. Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2020.

[60] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, et al. cuRobo: Parallelized collision-free minimum-jerk robot motion generation. *arXiv preprint arXiv:2310.17274*, 2023.

[61] A. Campeau-Lecours, H. Lamontagne, S. Latour, P. Fauteux, V. Maheu, F. Boucher, C. Deguire, and L.-J. C. L'Ecuyer. Kinova modular robot arms for service robotics applications. *Int. J. Robot. Appl. Technol.*, 5(2):49–71, July 2017. ISSN 2166-7195. doi:10.4018/IJRAT.2017070104. URL https://doi.org/10.4018/IJRAT.2017070104.

[62] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

[63] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun, C. Wei, B. Yu, R. Yuan, R. Sun, M. Yin, B. Zheng, Z. Yang, Y. Liu, W. Huang, H. Sun, Y. Su, and W. Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[64] C. Fu, P. Chen, Y. Shen, Y. Qin, M. Zhang, X. Lin, J. Yang, X. Zheng, K. Li, X. Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[65] Y. Liu, H. Duan, Y. Zhang, B. Li, S. Zhang, W. Zhao, Y. Yuan, J. Wang, C. He, Z. Liu, K. Chen, and D. Lin. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision (ECCV)*, 2024.

[66] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollar, and C. Feichtenhofer. SAM 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Ha6RTeWMd0.

Appendices

In this supplementary material, we present additional details and clarifications that are omitted in the main text due to space constraints.

A Limitations

While RoboSpatial significantly improves spatial reasoning capabilities in VLMs, certain design choices naturally introduce trade-offs and areas for future exploration.

First, the dataset relies on a top-down occupancy map to identify and annotate empty regions for spatial context and compatibility tasks. This approach simplifies reasoning about object placement on horizontal surfaces and enables efficient data generation, but it currently does not support spatial questions involving containment—such as whether an object can fit inside or under another object—which would require more detailed volumetric modeling.

Second, although the models are deployed on a real robot using a modular approach, we do not yet explore tighter forms of integration such as training it jointly with robot trajectories [25]. Investigating these alternatives could enhance downstream policy learning and enable more seamless end-to-end systems.

Finally, RoboSpatial focuses on indoor and tabletop scenes containing objects commonly encountered in household environments, and does not include humans or animals. This reflects the nature of source datasets and our emphasis on robot object manipulation. While this limits coverage of social or dynamic interaction scenarios, trained models still generalizes well to out-of-distribution benchmarks like BLINK, which include humans and animals—suggesting that the learned spatial representations are broadly transferable.

B Dataset Details

B.1 Dataset Statistics

We provide the full dataset statistics in table 5. For all training, we use only 900,000 spatial relationships, sampled equally across all datasets, due to computational constraints. We further experiment on the effect of data scaling on table 8 and explain the results. Notably, HOPE [58] and GraspNet-1B [59] contain similar tabletop images captured from different perspectives, resulting in lower dataset diversity for the tabletop environment. We plan to enhance the diversity of RoboSpatial by incorporating additional tabletop datasets.

B.2 Out-of-Domain Benchmarks

RoboSpatial-Home: 350 manually authored spatial questions over diverse real-world RGB-D scenes captured with an iPhone depth sensor. Questions omit explicit frame-of-reference labels to test implicit reasoning.

BLINK (spatial subset) [23]: Binary questions covering diverse prepositions (e.g., "under," "next to," "touching"). We evaluate only the spatial subset aligned with our task formulation.

SpatialBench (position category) [24]: Tests fine-grained spatial localization and placement reasoning using RGB-only inputs.

B.3 Choice of Spatial Relationships

In designing the dataset, we focused on spatial relationships that directly impact robotic perception, planning, and interaction: context, compatibility, and configuration. These were selected to reflect the core spatial reasoning challenges that robots encounter when operating in complex, real-world environments.

We intentionally excluded tasks such as object counting, as we consider them to fall outside the scope of spatial understanding. While counting is an important visual reasoning skill, it does not require reasoning about spatial relations between objects or between objects and their environment. For example, determining that "three cups are on the table" is a perceptual task rather than a spatial reasoning one. As such, counting may complement but does not substitute for the types of relational reasoning we target. We leave the integration of counting tasks into spatial benchmarks as future work.

Similarly, we exclude tasks that rely solely on distance measurements. Although distance is a fundamental spatial quantity, it is difficult to define consistently across different environments, object scales, and robot embodiments. Absolute distances can vary significantly between indoor and outdoor scenes, small and large objects, or different robot perspectives, making them hard to normalize or interpret in a general way. Moreover, distance alone often lacks the relational semantics required for higher-level reasoning—for example, understanding that an object is behind, above, or in front of others. RoboSpatial instead focuses on spatial relationships that are more invariant, interpretable, and transferable across diverse robotic scenarios.

That said, the data generation pipeline is general and could readily support auxiliary tasks involving object counting or distance estimation if desired. These metrics may serve as useful complements in future extensions of the benchmark or as auxiliary supervision signals in model training.

B.4 Object Grounding Dataset

To support accurate spatial understanding, we generate an auxiliary dataset for object grounding. Many spatial reasoning tasks assume that the model can correctly identify which object is being referred to in the scene. However, in practice, this can be a major source of error—especially in cluttered environments or when multiple instances of the similar object type are present.

The grounding dataset provides direct supervision to help models learn to associate text descriptions with specific objects in the image. For each image, we include a set of object descriptions (e.g., "the keyboard" or "the chair") paired with the corresponding 2D bounding box of the object in the image. These 2D boxes are projected from the annotated 3D bounding boxes using camera intrinsics and extrinsics.

A total of 100k grounding QA pairs are generated and used during training to reduce reference ambiguity and improve object identification accuracy in spatial tasks. While not part of the main spatial reasoning taxonomy, grounding accuracy is a prerequisite for answering spatial questions correctly, and we find that including this data helps reduce errors caused by incorrect object identification.

B.5 Dataset Generation Details

The dataset generation pipeline is detailed in the main text (subsection 3.2), which introduces a two-stage process for computing 3D spatial relationships and projecting them into 2D image space. Here, we expand on implementation details not covered in the main paper and provide clarification on the reasoning logic used in spatial annotation.

Reference Frame Annotation. For each spatial configuration question, we label relationships from three perspectives: ego-centric (camera view), object-centric (based on object heading), and world-centric (aligned with the dataset's global frame). To compute object-centric directions, we use the heading vector of each oriented 3D bounding box to define the "front" of the object. Left, right,

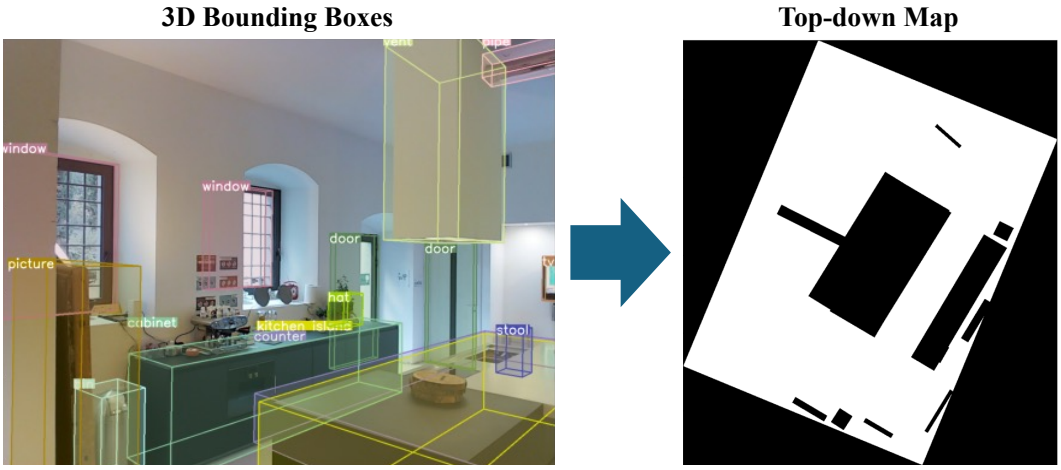


Figure 5: An example of generated top-down map of the image from 3D bounding boxes.

Category	Dataset	Split	Scans	Images	Configuration Q	Context Q	Compatibility Q
Indoor	Matterport3D [56]	Train	1859 scans	236243	298439	298439	298439
		Validation	10 scans	200	200	200	200
	ScanNet [55]	Train	1514 scans	280402	299039	299039	299039
		Validation	12 scans	400	400	400	400
	3RScan [57]	Train	1543 scans	366755	298839	298839	298839
		Validation	18 scans	400	400	400	400
Tabletop	HOPE [58]	Train	60 scenes	50050	36817	36817	36817
		Validation	47 scenes	235	500	500	500
	GraspNet-1B [59]	Train	130 scenes	25620	36817	36817	36817
		Validation	30 scenes	120	500	500	500

Table 5: Full dataset statistics for indoor and tabletop datasets.

behind, and front relations are then assigned accordingly. World-centric annotations modify vertical relationships (above/below) using global z-coordinates to reflect elevation.

Surface Detection and Free Space Sampling. To identify support surfaces such as tables, counters, or floors, we use GPT-4o to select candidate objects that are likely to support placement. A top-down occupancy map is constructed from bounding boxes in the scene fig. 5. We sample 3D points in unoccupied regions and project them into the image plane for spatial context tasks. Points are filtered via occlusion checks using raycasting, ensuring sampled points are visible and unobstructed.

Compatibility Check and Object Placement. For spatial compatibility, we simulate placing a virtual object bounding box at candidate locations. The placement must fit without intersecting other objects and must allow a clearance of at least 10 cm in all axes. We allow in-plane rotation and translation to test flexible placement. This provides a binary label (True/False) indicating whether the object can be compatibly placed in the region.

Output Format. Though RoboSpatial uses point prediction for ease of integration with robot setups, the pipeline also supports mask-based outputs and can be extended in future work.

C Implementation Details

C.1 Evaluation Metrics

For RoboSpatial-Val and RoboSpatial-Home, each of the three spatial reasoning categories (configuration, compatibility, context) contains 2,000 questions. Binary questions are scored by accuracy. Coordinate-based context tasks require predicting one or more points in free space; pre-

Model	Indoor			Tabletop			Average		
	Configuration	Context	Compatibility	Configuration	Context	Compatibility	Indoor	Tabletop	Total
Open-source VLMs									
2D VLMs									
VILA [12]	54.7	18.3	56.3	45.1	13.2	53.8	43.1	37.4	40.2
+RoboSpatial	71.4 ↑	45.9 ↑	77.2 ↑	71.8 ↑	43.7 ↑	73.3 ↑	64.8 ↑	62.9 ↑	63.9 ↑
LLaVA-NeXT [11]	48.9	12.5	32.7	48.3	8.4	30.9	31.4	29.2	30.3
+RoboSpatial	69.3 ↑	41.3 ↑	70.5 ↑	70.7 ↑	44.8 ↑	66.1 ↑	60.4 ↑	60.5 ↑	60.5 ↑
SpaceLLaVA [17]	52.6	15.3	49.0	66.5	12.2	60.1	38.9	46.2	43.6
+RoboSpatial	76.0 ↑	50.7 ↑	76.6 ↑	74.9 ↑	46.4 ↑	70.5 ↑	67.8 ↑	63.6 ↑	65.7 ↑
RoboPoint [19]	39.0	41.4	38.3	37.9	31.6	45.2	39.6	38.2	38.9
+RoboSpatial	72.2 ↑	**68.9** ↑	72.1 ↑	70.3 ↑	**61.7** ↑	78.4 ↑	71.0 ↑	70.1 ↑	70.6 ↑
3D VLMs									
3D-LLM [53]	54.5	8.1	53.6	59.2	10.6	57.4	37.6	42.4	40.0
+RoboSpatial	76.3 ↑	35.4 ↑	77.5 ↑	76.2 ↑	46.8 ↑	75.0 ↑	63.1 ↑	66.0 ↑	64.6 ↑
LEO [54]	56.1	11.3	58.3	60.8	11.1	59.3	41.9	43.7	42.8
+RoboSpatial	**80.2** ↑	56.7 ↑	**82.5** ↑	**78.1** ↑	55.2 ↑	**78.9** ↑	**73.1** ↑	**70.7** ↑	**71.9** ↑
Not available for fine-tuning									
2D VLMs									
Molmo [20]	40.6	48.2	60.0	61.5	35.8	54.6	49.6	50.6	50.1
GPT-4o [10]	63.5	25.1	59.4	62.3	27.9	66.8	49.3	52.3	50.8

Table 6: Results of existing 2D/3D VLMs on a held-out validation split (RoboSpatial-Val) of images and scans. All methods, for all tasks, perform better (↑) when fine-tuned on RoboSpatial. The best result for each column is bolded.

Model	Indoor			Tabletop			Average		
	Ego-centric	Object-centric	World-centric	Ego-centric	Object-centric	World-centric	Indoor	Tabletop	Total
Open-source VLMs									
2D VLMs									
VILA [12]	55.9	40.5	32.9	43.6	39.7	28.9	43.1	37.4	40.2
+RoboSpatial	74.3↑	57.8 ↑	62.3 ↑	70.3 ↑	58.1 ↑	60.3 ↑	64.8 ↑	62.9 ↑	63.9 ↑
LLaVA-Next [11]	35.2	24.3	34.7	36.4	28.5	22.7	31.4	29.2	30.3
+RoboSpatial	75.4 ↑	54.1 ↑	68.8 ↑	67.9 ↑	54.7 ↑	58.9 ↑	60.4 ↑	60.5 ↑	60.5 ↑
SpaceLLaVA [17]	40.6	36.0	30.1	52.3	32.8	53.5	38.9	46.2	43.6
+RoboSpatial	**78.5** ↑	**60.6** ↑	64.3 ↑	73.0 ↑	49.5 ↑	68.3 ↑	67.8 ↑	63.6 ↑	65.7 ↑
RoboPoint [19]	41.9	36.2	40.7	46.2	30.5	37.9	39.6	38.2	38.9
+RoboSpatial	76.4 ↑	58.3 ↑	78.3 ↑	**76.7** ↑	62.6 ↑	71.0 ↑	71.0 ↑	70.1 ↑	70.6 ↑
3D VLMs									
3D-LLM [53]	28.9	38.3	45.6	38.9	35.7	52.6	37.6	42.4	40.0
+RoboSpatial	60.7 ↑	52.1 ↑	76.5 ↑	57.9 ↑	**62.8** ↑	77.3 ↑	63.1 ↑	66.0 ↑	64.6 ↑
LEO [54]	46.9	30.6	48.2	41.4	34.3	55.4	41.9	43.7	42.8
+RoboSpatial	68.1 ↑	71.6 ↑	**79.6** ↑	71.4 ↑	60.2 ↑	**80.5** ↑	**73.1** ↑	**70.7** ↑	**71.9** ↑
Not available for fine-tuning									
2D VLMs									
Molmo [20]	50.4	50.8	47.6	64.4	33.6	53.8	49.6	50.6	50.1
GPT-4o [10]	52.9	38.7	56.3	62.5	30.7	63.7	49.3	52.3	50.8

Table 7: Results of per frame accuracy of existing 2D/3D VLMs on a RoboSpatial-Val. All methods, for all tasks, perform better (↑) when fine-tuned on RoboSpatial. The best result for each column is bolded.

dictions are deemed correct if their 3D locations fall within the convex hull of the ground-truth set derived from scene geometry. This criterion is intentionally strict—predictions close but outside the hull are marked incorrect—making reported scores a conservative estimate.

C.2 Model Training

We further explain the training details for all 2D and 3D VLMs trained on RoboSpatial. For all models, we perform instruction tuning using the model weights from public repositories. All training is done using 8 Nvidia H100 GPUs, with the training time between 20 and 40 hours.

C.3 Model Setup

VILA [12] We initialize the model from Efficient-Large-Model/Llama-3-VILA1.5-8B on Hugging Face. We use the fine-tuning script from the VILA GitHub repository to train the model using the

	100K	**300K**	**900k** (Default)	**1.8M**	**3M** (Full)
LLaVa-Next [11]	38.1	46.7	60.5	65.8	72.4

Table 8: Results of scaling experiment on LLaVa-Next [11] with varied number of spatial relationship annotations. Average accuracy on RoboSpatial-Val is reported.

	MMMU$_{val}$	**MME**$_p$	**MME**$_c$	**MMBench**$_{dev}$
LLaVA-NeXT	39.4	1561.8	**305.4**	**71.6**
+RoboSpatial	**39.8**	**1604.5**	293.2	**71.6**

Table 9: Evaluation on general-purpose multimodal benchmarks (MMMU, MME, MMBench) to assess whether training on RoboSpatial affects commonsense and factual reasoning.

default hyperparameters.

LLaVA-NeXT [11] We initialize the model from lmms-lab/llama3-llava-next-8b on Hugging Face. We use the LLaVA-Next fine-tuning script from the LLaVA-Next repository using the default hyperparameters.

SpaceLLaVA [17] As official code and weights for SpatialVLM [17] is not released, we use a community implementation which is endorsed by SpatialVLM [17] authors. We initialize the model from remyxai/SpaceLLaVA from Hugging Face. We use LLaVA-1.5 finetuning script from LLaVa [62] repository using the default hyperparameters.

RoboPoint [19] We initialize the model from wentao-yuan/robopoint-v1-vicuna-v1.5-13b on Hugging Face. We use the fine-tuning script provided in the RoboPoint [19] GitHub repository to train the model using the default hyperparameters.

3D-LLM [53] We initialize the model using the pretrain_blip2_sam_flant5xl_v2.pth checkpoint downloaded from the official GitHub repository. Since the model requires preprocessing of multiview images, we follow the author's pipeline to process multiview images from the environments. Because the model does not accept image input, we append the following text in front of the question to ensure the model understands the perspective from which the question is being asked: "I am facing ANCHOR OBJECT." We use the default hyperparameters and train the model for 20 epochs per the author's guidelines. We choose the best model based on validation accuracy.

LEO [54] We initialize the model from the sft_noact.pth checkpoint downloaded from the official GitHub repository.

Since LEO supports dual image and 3D point cloud input, we input both of them and modify the question as in 3D-LLM. We use the default hyperparameters and train the model for 10 epochs per the author's guidelines, and choose the best model based on validation accuracy.

We could not fine-tune Molmo [20] from allenai/Molmo-7B-D-0924 or GPT-4o [10] from the gpt-4o-2024-08-06 API due to the unavailability of the fine-tuning script at the time of this work, thus we use them as a zero-shot baselines.

D More Results

D.1 Accuracy Per Reference Frame

We show the results per frame in table 7 for RoboSpatial-Val. From the results, we can see a distinct difference between 2D and 3D VLMs in understanding the world-centric frame before training with RoboSpatial. Baseline 2D VLMs have trouble understanding the world-centric frame, which involves understanding elevation, while 3D VLMs comparatively excel at it. Furthermore, we can see that since baseline 3D VLMs are trained on point clouds without information of perspective, their accuracy in ego-centric and object-centric frames is lower. However, with RoboSpatial training, we were able to teach the 3D VLMs to think in a certain frame, thus considerably improving their performance on ego-centric and object-centric frames. However, we hypothesize that, due to their design—specifically, the lack of a means to visually inject perspective information since they require

	Base	Auxiliary	RoboSpatial	Both
LLaVA-NeXT	30.3	32.4	51.8	60.5

Table 10: Ablation study evaluating the impact of the auxiliary grounding dataset on RoboSpatial-Val.

complete 3D point clouds—3D VLMs still lag behind 2D VLMs on ego-centric and object-centric frames.

D.2 Data Scaling

In table 8, we experiment with scaling the number of annotations while keeping images fixed. We found that even though the number of images stays consistent, increasing the number of annotations can improve performance. For future work, we plan to apply the data generation pipeline to a diverse set of indoor and tabletop environments to further improve the performance of the models.

D.3 Commonsense Knowledge Retention

To ensure that training on RoboSpatial does not degrade a model's general reasoning or commonsense capabilities, we evaluate the RoboSpatial-trained model on a suite of standard multimodal benchmarks: MMMU [63], MME [64], and MMBench [65]. As shown in Table 9, the RoboSpatial-trained model maintains or slightly improves performance across all benchmarks, suggesting that spatial fine-tuning preserves broader knowledge capabilities.

D.4 Ablation of the Auxiliary Grounding Dataset

As shown in Table 10, training on the auxiliary dataset alone yields a small improvement over the base model (+2.1), but it falls far short of the gains achieved with RoboSpatial, which is explicitly designed to teach spatial reasoning. This confirms that grounding supervision alone is insufficient for spatial understanding. However, combining both datasets leads to the best performance, suggesting that improving object localization can complement spatial supervision when jointly trained.

D.5 Robot Experiments Details

D.5.1 Robot Setup

For picking, we find which object the point maps to using SAM 2 [66] and execute the picking behavior on that object. For placing, we simply compute the 3D coordinate based on the depth value at that pixel and place the object at that coordinate. There were no failures due to cuRobo [60] failing. The experiments were purposely designed to consist of behaviors that our robot system can handle in order to avoid introducing irrelevant factors. The picking behavior consists of computing a top-down grasp pose and reaching it with cuRobo [60]. To compute the grasp pose:

1. We estimate the major axis of the object's point cloud in top-down view using PCA.
2. The grasp orientation is orthogonal to the major axis.
3. The grasp height is based on the highest point in the object's point cloud minus an offset of 3cm. This heuristic ensures the system can grip long objects.

The placing behavior is the same as picking, except that an area within 5cm of the placement coordinate is used as the point cloud for estimating orientation and height, and a vertical height offset is added to account for the height at which the object was picked.

D.5.2 Additional Results

We present additional results from the robot experiments in fig. 6. We observe that models trained with RoboSpatial consistently outperform baseline models in most cases, even though the prompt

is not optimized for RoboSpatial-trained models. This demonstrates that the power of VLMs enables templated language to generalize to language unseen during training while maintaining spatial understanding capabilities. However, even with RoboSpatial training, the models struggle with understanding stacked items, indicating a need for further data augmentation with diverse layouts. In a few cases, RoboSpatial training adversely affects performance, especially with RoboPoint [19]. We hypothesize that mixing the dataset with RoboPoint training data and RoboSpatial training data may lead to unforeseen side effects, particularly in grounding objects. Nevertheless, we demonstrate that RoboSpatial training enhances VLM's spatial understanding in real-life robotics experiments, even with freeform language.

D.6 More Qualitative Examples

fig. 7 present additional qualitative comparisons between models trained on RoboSpatial. The findings demonstrate that models trained on RoboSpatial consistently exhibit spatial understanding in the challenging RoboSpatial-Home dataset, even outperforming closed models like GPT-4o [10]. However, we observed that object grounding is a crucial prerequisite for spatial understanding; the improvement is often hindered by the model's inability to ground objects in cluttered scenes, where GPT-4o performs more effectively. Additionally, we show that the RoboSpatial-trained model successfully generalizes to unseen spatial relationships in BLINK-Spatial [23], including those involving distance, such as "touching."

Question: pick lone object	
LLaVa-Next [11]	✗
LLaVa-Next-FT [11]	✓
RoboPoint [19]	✗
RoboPoint-FT [19]	✓
Molmo [20]	✓
GPT-4o [10]	✗

Question: Is there room to slot the pancake mix in the middle of the row of boxes	
LLaVa-Next [11]	✓
LLaVa-Next-FT [11]	✓
RoboPoint [19]	✗
RoboPoint-FT [19]	✓
Molmo [20]	✓
GPT-4o [10]	✓

Question: Is there space in the white container for the orange juice box	
LLaVa-Next [11]	✗
LLaVa-Next-FT [11]	✓
RoboPoint [19]	✗
RoboPoint-FT [19]	✗
Molmo [20]	✗
GPT-4o [10]	✓

Question: alphabet soup fit in the purple box	
LLaVa-Next [11]	✓
LLaVa-Next-FT [11]	✗
RoboPoint [19]	✓
RoboPoint-FT [19]	✓
Molmo [20]	✗
GPT-4o [10]	✓

Question: pick object behind the middle container	
LLaVa-Next [11]	✗
LLaVa-Next-FT [11]	✓
RoboPoint [19]	✓
RoboPoint-FT [19]	✗
Molmo [20]	✗
GPT-4o [10]	✗

Question: pick shortest object	
LLaVa-Next [11]	✗
LLaVa-Next-FT [11]	✓
RoboPoint [19]	✓
RoboPoint-FT [19]	✓
Molmo [20]	✓
GPT-4o [10]	✓

Question: place object in container behind popcorn	
LLaVa-Next [11]	✗
LLaVa-Next-FT [11]	✓
RoboPoint [19]	✓
RoboPoint-FT [19]	✓
Molmo [20]	✗
GPT-4o [10]	✗

Question: place the object inside the smallest box	
LLaVa-Next [11]	✗
LLaVa-Next-FT [11]	✓
RoboPoint [19]	✓
RoboPoint-FT [19]	✓
Molmo [20]	✓
GPT-4o [10]	✗

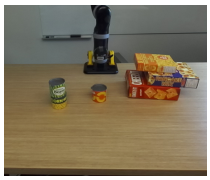
Question: can the robot directly pick the red orange peaches can without disturbing other objects?	
LLaVa-Next [11]	✓
LLaVa-Next-FT [11]	✓
RoboPoint [19]	✗
RoboPoint-FT [19]	✗
Molmo [20]	✓
GPT-4o [10]	✓

Question: is there an object that is not in a stack?	
LLaVa-Next [11]	✓
LLaVa-Next-FT [11]	✓
RoboPoint [19]	✓
RoboPoint-FT [19]	✓
Molmo [20]	✓
GPT-4o [10]	✓

Question: can the macaroni and cheese be placed on top of cheez-it without touching other objects?	
LLaVa-Next [11]	✗
LLaVa-Next-FT [11]	✗
RoboPoint [19]	✓
RoboPoint-FT [19]	✓
Molmo [20]	✗
GPT-4o [10]	✓

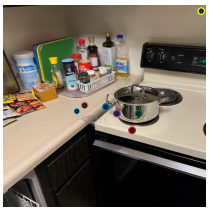
Question: is there space to place one of the cans on the cheez-it box?	
LLaVa-Next [11]	✗
LLaVa-Next-FT [11]	✗
RoboPoint [19]	✗
RoboPoint-FT [19]	✗
Molmo [20]	✗
GPT-4o [10]	✗

Question: place on the object to the left of macaroni and cheese	
LLaVa-Next [11]	✗
LLaVa-Next-FT [11]	✓
RoboPoint [19]	✓
RoboPoint-FT [19]	✓
Molmo [20]	✓
GPT-4o [10]	✗



Question: pick the highest object on the stack of two objects	
LLaVa-Next [11]	✗
LLaVa-Next-FT [11]	✗
RoboPoint [19]	✗
RoboPoint-FT [19]	✗
Molmo [20]	✗
GPT-4o [10]	✗

Figure 6: Additional robot experiments. A green check mark indicates that the model answered correctly. The -FT suffix denotes a model trained with RoboSpatial. The questions are purposely not cleaned to reflect realistic language inputs.

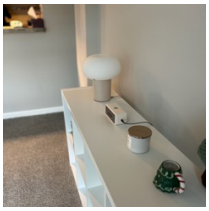


Question: Pinpoint several points within the vacant space situated to the left of the pot.

Answer	
LLaVa-Next [11]	
LLaVa-Next-FT [11]	
RoboPoint [19]	
RoboPoint-FT [19]	
Molmo [20]	
GPT-4o [10]	

Question: Pinpoint several points within the vacant space situated behind the trash bin.

Answer	
LLaVa-Next [11]	
LLaVa-Next-FT [11]	
RoboPoint [19]	
RoboPoint-FT [19]	
Molmo [20]	
GPT-4o [10]	



Question: Can the lamp fit in front of the shelf?

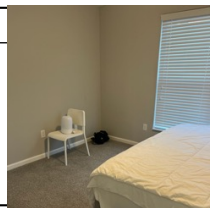
Answer	Yes
LLaVa-Next [11]	×
LLaVa-Next-FT [11]	✓
RoboPoint [19]	×
RoboPoint-FT [19]	✓
Molmo [20]	×
GPT-4o [10]	×

Question: Can the pot fit above the fridge?

Answer	Yes
LLaVa-Next [11]	×
LLaVa-Next-FT [11]	✓
RoboPoint [19]	×
RoboPoint-FT [19]	✓
Molmo [20]	×
GPT-4o [10]	×

Question: Is the lamp above the shelf?

Answer	Yes
LLaVa-Next [11]	×
LLaVa-Next-FT [11]	✓
RoboPoint [19]	×
RoboPoint-FT [19]	✓
Molmo [20]	×
GPT-4o [10]	✓

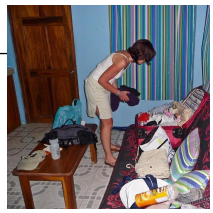


Question: Is the chair behind the bed?

Answer	Yes
LLaVa-Next [11]	×
LLaVa-Next-FT [11]	✓
RoboPoint [19]	×
RoboPoint-FT [19]	✓
Molmo [20]	×
GPT-4o [10]	×

Question: Is the dining table touching the donut?

Answer	Yes
LLaVa-Next [11]	×
LLaVa-Next-FT [11]	✓
RoboPoint [19]	×
RoboPoint-FT [19]	✓
Molmo [20]	×
GPT-4o [10]	×



Question: Is the couch under the suitcase?

Answer	Yes
LLaVa-Next [11]	×
LLaVa-Next-FT [11]	✓
RoboPoint [19]	×
RoboPoint-FT [19]	✓
Molmo [20]	×
GPT-4o [10]	×

Figure 7: Qualitative results on spatial reasoning benchmarks. The -FT suffix denotes a model trained with RoboSpatial. The first three rows show examples from RoboSpatial-Home, covering spatial context, spatial compatibility, and spatial configuration. For spatial context questions, only the first predicted point from each model is shown. The fourth row shows generalization to unseen spatial relationships on the Blink-Spatial [23] dataset, demonstrating that the RoboSpatial-trained model can transfer to unseen relationships.