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

- Appendix A Related Works.
- Appendix B Dataset Details.
- Appendix C Implementation Details.
- Appendix D Full Results.
- Appendix E More Qualitative Examples.

A Related Works

VLMs for Robotics. Vision-language models (VLMs) have emerged as pivotal tools in robotics, enabling systems to interpret and act upon complex visual and textual information. By integrating visual perception with language understanding, VLMs facilitate more intuitive human-robot interactions and enhance autonomous decision-making capabilities. Recent advancements have demonstrated the potential of VLMs in various robotic applications. For instance, vision-language-action models (VLAs) Kim et al. (2024); Brohan et al. (2023); Octo Model Team et al. (2024) enable robots to interpret and execute complex instructions and output executable robot actions. Additionally, VLMs like GPT-4v OpenAI et al. (2024) have been utilized for high-level task planning Wake et al. (2023), allowing robots to generate detailed action sequences from natural language instructions. Furthermore, VLMs have been used for keypoint/mask prediction Huang et al. (2024c); Wi et al. (2023); Nasiriany et al. (2024), error analysis Duan et al. (2024), grasp pose prediction Huang et al. (2024a). Despite these advancements, integrating VLMs Cai et al. (2024); Cheng et al. (2024); Yuan et al. (2024) into robotic systems presents challenges. One significant hurdle is the need for precise spatial reasoning to navigate and manipulate objects effectively. While VLMs excel in understanding and generating language, their ability to comprehend and reason about spatial relationships in dynamic environments remains limited Yamada et al. (2024); Xu et al. (2024); Wang et al. (2024a). Therefore, RoboSpatial aims to tackle this gap by presenting a large scale pretraining and evaluation setup for teaching spatial understanding to VLM for robotics.

Spatial Understanding with VLMs. Spatial understanding has been implicitly and explicitly part of various vision and question answering tasks Fu et al. (2024); Azuma et al. (2022); Jia et al. (2024); Suhr & Artzi (2019); Salewski et al. (2022); Krishna et al. (2017); Johnson et al. (2017); Hudson & Manning (2019). While many benchmarks and methods have been proposed, they often come with limitations: some focus exclusively on simulations Szymanska et al. (2024) or generic images Liu et al. (2023a); Rajabi & Kosecka (2024); Cheng et al. (2024); Chen et al. (2024); Fu et al. (2024); Kamath et al. (2023); Shiri et al. (2024); Ranasinghe et al. (2024), others are difficult to evaluate Szymanska et al. (2024); Du et al. (2024); Linghu et al. (2024), rely on complete 3D scans Zhang et al. (2024); Man et al. (2024); Ma et al. (2023); Linghu et al. (2024), or do not consider reference frames Zhang et al. (2024); Man et al. (2024); Ma et al. (2023); Linghu et al. (2024); Chen et al. (2024); Cheng et al. (2024); Fu et al. (2024); Ranasinghe et al. (2024). Furthermore, they often fail to address actionable, robotics-relevant spatial relationships such as spatial compatibility and context Du et al. (2024); Fu et al. (2024); Wang et al. (2024b); Shiri et al. (2024); Kamath et al. (2023); Linghu et al. (2024); Ranasinghe et al. (2024).

Inspired by prior works on spatial reasoning Liu et al. (2023a); Kamath et al. (2023)—where the impact of reference frames and spatial configurations was explored in generic images Lin et al. (2015); Hudson & Manning (2019)—we extend spatial understanding to a robotics-specific context with actionable spatial relationships such as spatial compatibility and spatial context. Our aim is to enable direct application to robotic workflows, such as task planning and verification.

To achieve this, we have developed and are planning to open-source a large-scale 2D/3D ready pretraining dataset, an automated data annotation pipeline, and trained models. We further show how our dataset can be used to teach spatial reasoning to a suite of vision-language models (VLMs) in in-domain and out-of-domain spatial reasoning datasets. We hope these resources lower the barrier to entry for exploring spatial understanding tailored to robotics.

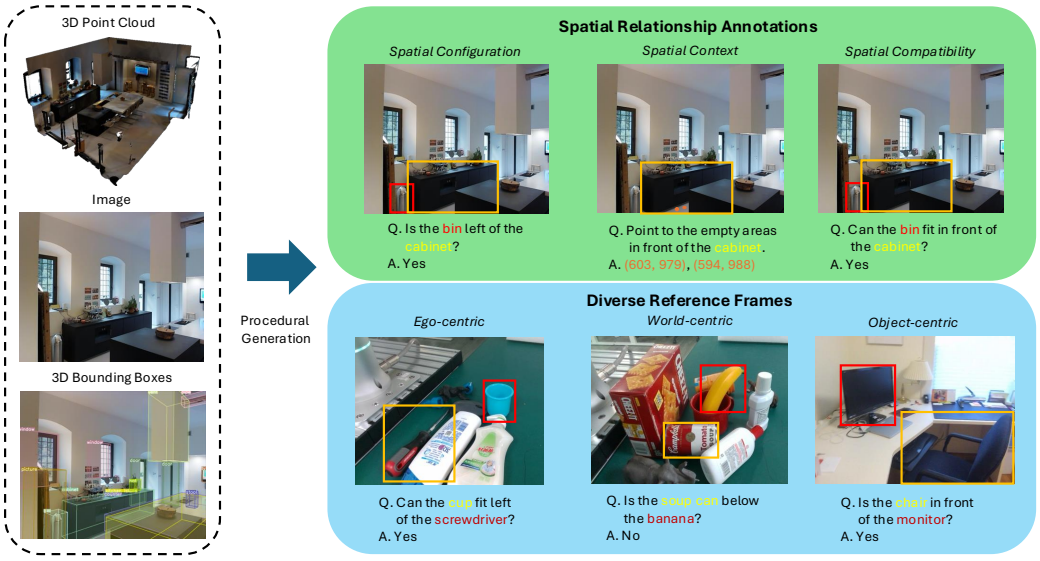


Figure 3: Overview of the RoboSpatial dataset. We automatically generate spatial relationship annotations from existing datasets with 3D point clouds, egocentric images, and 3D bounding box annotations. We create question/answer pairs covering three classes of spatial relationships, three spatial reference 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.

B Dataset Details

B.1 Dataset Statistics

We provide the full dataset statistics in Table 3. For all training, we use only 900,000 spatial relationships, equally divided across all datasets, due to computational constraints. We further experiment on the effect of data scaling on Table 7 and explain the results. Notably, HOPE Tyree et al. (2022) and GraspNet-1B Fang et al. (2020) contain similar tabletop images captured from different perspectives, resulting in lower dataset diversity for the tabletop environment. We plan to enhance the diversity of our dataset by incorporating additional tabletop datasets.

B.2 Dataset Generation Details

Frame Generation. We explain how answers are generated for each frame Figure 3. Each question type has three answers from ego-centric, object-centric, and world-centric perspectives. Each frame may share the same answer since not all frames lead to unique answers. For our answer generation, we used the following logic:

- **Ego-centric:** The default perspective from the ego-centric view.
- **Object-centric:** Using the oriented bounding box directions, we determine the front of the object. With this information, we assign front, behind, left, and right. Above and below remain the same as in the ego-centric perspective.
- **World-centric:** Using the z-coordinate of the oriented bounding box, we modify the above and below relationships to reflect whether an object is above or below another object with respect to elevation.

Compatibility Generation. For compatibility, we construct a top-down map as shown in Figure 4. Using the top-down map and the top-down 2D bounding box of the object to be placed, we determine if there exists an empty space within a threshold distance (i.e., 1% of the object's longest width or length). If the object meets this condition, it is deemed compatible to be fitted.

Table 3: Full dataset statistics for indoor and tabletop datasets.

Category	Dataset	Split	Scans	Images	Configuration Q	Context Q	Compatibility Q
Indoor	Matterport3D Chang et al. (2017)	Train	1859 scans	236243	298439	298439	298439
		Validation	5 scans	100	100	100	100
		Test	5 scans	100	100	100	100
	ScanNet Dai et al. (2017)	Train	1512 scans	278402	298439	298439	298439
		Validation	12 scans	1200	500	500	500
		Test	12 scans	1200	500	500	500
	3RScan Wald et al. (2019)	Train	1543 scans	365355	298439	298439	298439
		Validation	9 scans	900	400	400	400
		Test	9 scans	900	400	400	400
Tabletop	HOPE Tyree et al. (2022)	Train	50 scenes	50000	36317	36317	36317
		Validation	10 scenes	50	500	500	500
		Test	47 scenes	235	500	500	500
	GraspNet-1B Fang et al. (2020)	Train	100 scenes	25500	36317	36317	36317
		Validation	30 scenes	120	500	500	500
		Test	30 scenes	120	500	500	500

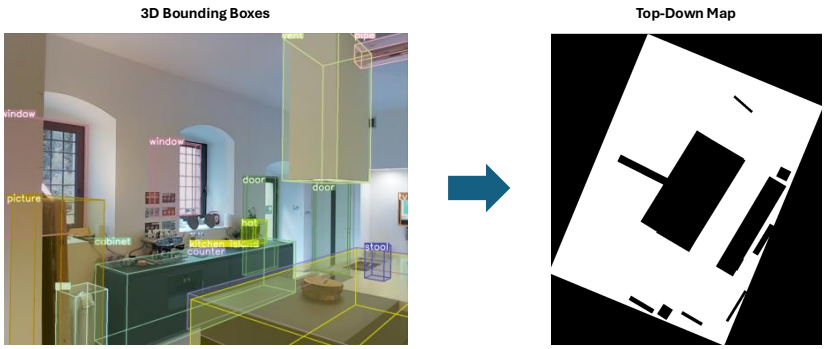


Figure 4: An example of generated top-down map of the image from 3D bounding boxes.

C Implementation Details

C.1 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.

VILA Lin et al. (2023) We initialize our 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 our model using the default hyperparameters.
LLaVA-NeXT Liu et al. (2023b) We initialize our 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 Chen et al. (2024) As official code and weights for SpatialVLM Chen et al. (2024) is not released, we use a community implementation which is endorsed by SpatialVLM Chen et al. (2024) authors. We initialize our model from remyxai/SpaceLLaVA from Hugging Face. We use LLaVA-1.5 finetuning script from LLaVa Liu et al. (2023c) repository using the default hyperparameters.
RoboPoint Yuan et al. (2024) We initialize our model from wentao-yuan/robopoint-v1-vicuna-v1.5-13b on Hugging Face. We use the fine-tuning script provided in the RoboPoint Yuan et al. (2024) GitHub repository to train our model using the default hyperparameters.
3D-LLM Hong et al. (2023) We initialize our 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 our environments. Because the model does not accept image input, we append the following text in front of the

Table 4: Results of existing 2D/3D VLMs on a held-out test split of images and scans. All methods, for all tasks, perform better (↑) when fine-tuned on our RoboSpatial dataset. The best result for each column is bolded.

Model	Indoor			Tabletop			Average		
	Configuration	Context	Compatibility	Configuration	Context	Compatibility	Indoor	Tabletop	Total
Open-source VLMs									
2D VLMs									
VILA Lin et al. (2023)	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 Liu et al. (2023b)	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 Chen et al. (2024)	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 Yuan et al. (2024)	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 Hong et al. (2023)	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 Huang et al. (2024b)	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 Deitke et al. (2024)	40.6	48.2	60.0	61.5	35.8	54.6	49.6	50.6	50.1
GPT-4o OpenAI et al. (2024)	63.5	25.1	59.4	62.3	27.9	66.8	49.3	52.3	50.8

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 Huang et al. (2024b) We initialize our 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 Deitke et al. (2024) from allenai/Molmo-7B-D-0924 or GPT-4o OpenAI et al. (2024) 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.

C.2 Robot Setup

For picking, we find which object the point maps to using SAM 2 Ravi et al. (2024) and execute our 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 Sundaralingam et al. (2023) 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 Sundaralingam et al. (2023). 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.

C.3 Omitted Results in the Main Text

We show the full results in held-out test split in Table 4 and out-of-domain splits in Table 5.

Table 5: 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.

Model	RoboSpatial-Home			BLINK-Spatial
	Localization	Affordance	Compatibility	Accuracy
Open-source VLMs				
2D VLMs				
VILA Lin et al. (2023)	53.3	12.0	52.0	72.7
+RoboSpatial	62.0 ↑	32.0 ↑	58.0 ↑	79.7 ↑
LLaVA-NeXT Liu et al. (2023b)	48.0	9.3	37.3	71.3
+RoboSpatial	58.0 ↑	24.0 ↑	44.0 ↑	79.0 ↑
SpaceLLaVA Chen et al. (2024)	60.0	16.0	49.3	76.2
+RoboSpatial	**68.7** ↑	38.0 ↑	56.0 ↑	**81.8** ↑
RoboPoint Yuan et al. (2024)	43.3	41.3	36.0	63.6
+RoboSpatial	50.0 ↑	**54.0** ↑	48.0 ↑	70.6 ↑
3D VLMs				
3D-LLM Hong et al. (2023)	40.0	8.0	46.0	N/A
+RoboSpatial	48.0 ↑	36.0 ↑	52.7 ↑	N/A
LEO Huang et al. (2024b)	50.7	10.0	48.0	N/A
+RoboSpatial	64.0 ↑	40.0 ↑	**60.0** ↑	N/A
Not available for fine-tuning				
Molmo Deitke et al. (2024)	44.7	38.0	58.0	67.1
GPT-4o OpenAI et al. (2024)	64.0	25.3	56.7	76.2

Table 6: Average accuracy for dataset generalization when training on indoor scenes and testing on tabletop scenes (indoor→tabletop), and vice versa (tabletop→indoor).

	Indoor → Tabletop	Tabletop → Indoor
RoboPoint Yuan et al. (2024)	38.7	38.2
+RoboSpatial	48.9 ↑	51.3 ↑
LEO Huang et al. (2024b)	41.9	43.7
+RoboSpatial	47.2 ↑	54.5 ↑

C.4 Cross-Dataset Generalization

We evaluate the generalization capability of our method by testing it across different scene types—specifically, both indoor and tabletop scenes—to control for any bias in the annotations of the underlying datasets that make up our benchmark. We train on data derived from subsets of the datasets corresponding to one scene type (either indoor or tabletop) and test on held-out datasets from the other scene type, representing unseen environments. We expect that even when training on a subset of datasets, the performance on unseen scene types will improve if our method generalizes well. The results of this cross-dataset evaluation are shown in Table 6.

C.5 Data Scaling

In Table 7, 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 our data generation pipeline to a diverse set of indoor and tabletop environments to further improve the performance of our models.

C.6 Accuracy Per Frame of Reference

We show the results per frame in Table 8 for our out-of-domain test set. 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-

Table 7: Results of scaling experiment on LLaVa-Next Liu et al. (2023b) with varied spatial relationship annotations. Average accuracy on held-out test set is reported.

Annotation Size	100K	300K	900k (Default)	1.8M	3M (Full)
LLaVa-Next Liu et al. (2023b)	38.1	46.7	60.5	65.8	72.4

Table 8: Results of per frame accuracy of existing 2D/3D VLMs on a held-out test split of images and scans. All methods, for all tasks, perform better (↑) when fine-tuned on our RoboSpatial dataset. 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 Lin et al. (2023)	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 Liu et al. (2023b)	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 Chen et al. (2024)	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 Yuan et al. (2024)	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 Hong et al. (2023)	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 Huang et al. (2024b)	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 Deitke et al. (2024)	50.4	50.8	47.6	64.4	33.6	53.8	49.6	50.6	50.1
GPT-4o OpenAI et al. (2024)	52.9	38.7	56.3	62.5	30.7	63.7	49.3	52.3	50.8

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

D Full Results

D.1 Robot Experiments

We present additional results from our robot experiments in Figure 5 and Figure 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 Yuan et al. (2024). 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.

E More Qualitative Examples

Figure 7 and Figure 8 present additional qualitative comparisons between models trained on RoboSpatial. Our findings demonstrate that models trained on RoboSpatial consistently exhibit spatial understanding in the challenging RoboSpatial-Home dataset, even outperforming closed

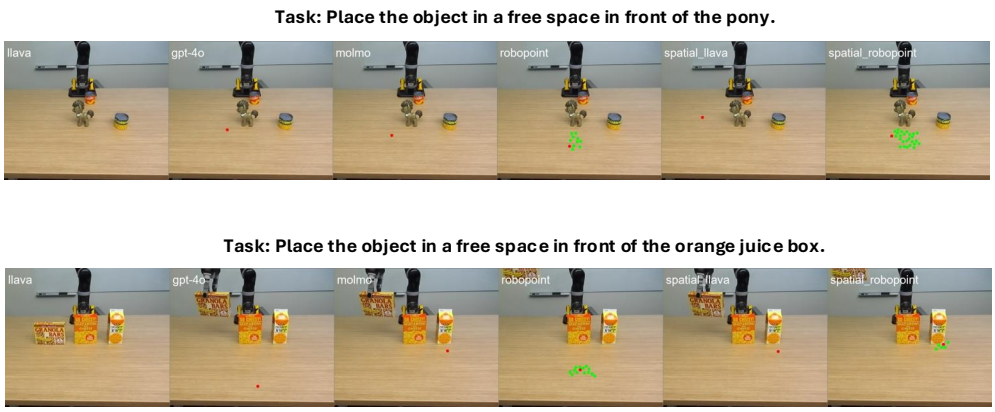


Figure 5: Robotics experiments: the red dot shows the model output (if not present, the model failed to provide a valid point in the image); green dots are used to show when a model outputs multiple points. The robot motion generator, cuRobo Sundaralingam et al. (2023), is used to grasp the item referenced by the generated point. The *spatial-* prefix indicates model trained with RoboSpatial.

models like GPT-4o OpenAI et al. (2024). 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, in Figure 8, we show that the RoboSpatial-trained model successfully generalizes to unseen spatial relationships in Blink-Spatial Fu et al. (2024), including those involving distance, such as "touching."

Question: pick lone object

GPT-4o OpenAI et al. (2024)	×
Molmo Deitke et al. (2024)	✓
LLaVa-Next Liu et al. (2023b)	×
S-LLaVa-Next Liu et al. (2023b)	✓
RoboPoint Yuan et al. (2024)	×
S-RoboPoint Yuan et al. (2024)	✓

Question: Is there room to slot the pancake mix in the middle of the row of boxes

GPT-4o OpenAI et al. (2024)	✓
Molmo Deitke et al. (2024)	✓
LLaVa-Next Liu et al. (2023b)	✓
S-LLaVa-Next Liu et al. (2023b)	✓
RoboPoint Yuan et al. (2024)	×
S-RoboPoint Yuan et al. (2024)	✓

Question: Is there space in the white container for the orange juice box

LLaVa-Next Liu et al. (2023b)	×
S-LLaVa-Next Liu et al. (2023b)	✓
RoboPoint Yuan et al. (2024)	×
S-RoboPoint Yuan et al. (2024)	×
Molmo Deitke et al. (2024)	×
GPT-4o OpenAI et al. (2024)	✓

Question: alphabet soup fit in the purple box

LLaVa-Next Liu et al. (2023b)	✓
S-LLaVa-Next Liu et al. (2023b)	×
RoboPoint Yuan et al. (2024)	✓
S-RoboPoint Yuan et al. (2024)	✓
Molmo Deitke et al. (2024)	×
GPT-4o OpenAI et al. (2024)	✓

Question: pick object behind the middle container

LLaVa-Next Liu et al. (2023b)	×
S-LLaVa-Next Liu et al. (2023b)	✓
RoboPoint Yuan et al. (2024)	✓
S-RoboPoint Yuan et al. (2024)	×
Molmo Deitke et al. (2024)	×
GPT-4o OpenAI et al. (2024)	×

Question: pick shortest object

LLaVa-Next Liu et al. (2023b)	×
S-LLaVa-Next Liu et al. (2023b)	✓
RoboPoint Yuan et al. (2024)	✓
S-RoboPoint Yuan et al. (2024)	✓
Molmo Deitke et al. (2024)	✓
GPT-4o OpenAI et al. (2024)	✓

Question: place object in container behind popcorn

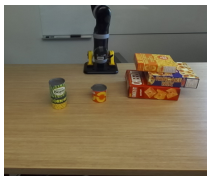
LLaVa-Next Liu et al. (2023b)	×
S-LLaVa-Next Liu et al. (2023b)	✓
RoboPoint Yuan et al. (2024)	✓
S-RoboPoint Yuan et al. (2024)	✓
Molmo Deitke et al. (2024)	×
GPT-4o OpenAI et al. (2024)	×

Question: place the object inside the smallest box

LLaVa-Next Liu et al. (2023b)	×
S-LLaVa-Next Liu et al. (2023b)	✓
RoboPoint Yuan et al. (2024)	✓
S-RoboPoint Yuan et al. (2024)	✓
Molmo Deitke et al. (2024)	✓
GPT-4o OpenAI et al. (2024)	×

Question: can the robot directly pick the red orange peaches can without disturbing other objects?

LLaVa-Next Liu et al. (2023b)	✓
S-LLaVa-Next Liu et al. (2023b)	✓
RoboPoint Yuan et al. (2024)	×
S-RoboPoint Yuan et al. (2024)	×
Molmo Deitke et al. (2024)	✓
GPT-4o OpenAI et al. (2024)	✓



Question: is there an object that is not in a stack?

GPT-4o OpenAI et al. (2024)	✓
Molmo Deitke et al. (2024)	✓
LLaVa-Next Liu et al. (2023b)	✓
S-LLaVa-Next Liu et al. (2023b)	✓
RoboPoint Yuan et al. (2024)	✓
S-RoboPoint Yuan et al. (2024)	✓

Question: can the macaroni and cheese be placed on top of cheez-it without touching other objects?

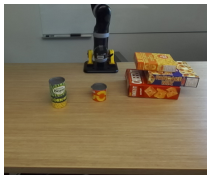
LLaVa-Next Liu et al. (2023b)	×
S-LLaVa-Next Liu et al. (2023b)	×
RoboPoint Yuan et al. (2024)	✓
S-RoboPoint Yuan et al. (2024)	✓
Molmo Deitke et al. (2024)	×
GPT-4o OpenAI et al. (2024)	✓

Question: is there space to place one of the cans on the cheez-it box?

GPT-4o OpenAI et al. (2024)	×
Molmo Deitke et al. (2024)	×
LLaVa-Next Liu et al. (2023b)	×
S-LLaVa-Next Liu et al. (2023b)	×
RoboPoint Yuan et al. (2024)	×
S-RoboPoint Yuan et al. (2024)	×

Question: place on the object to the left of macaroni and cheese

GPT-4o OpenAI et al. (2024)	×
Molmo Deitke et al. (2024)	✓
LLaVa-Next Liu et al. (2023b)	×
S-LLaVa-Next Liu et al. (2023b)	✓
RoboPoint Yuan et al. (2024)	✓
S-RoboPoint Yuan et al. (2024)	✓



Question: pick the highest object on the stack of two objects

GPT-4o OpenAI et al. (2024)	×
Molmo Deitke et al. (2024)	×
LLaVa-Next Liu et al. (2023b)	×
S-LLaVa-Next Liu et al. (2023b)	×
RoboPoint Yuan et al. (2024)	×
S-RoboPoint Yuan et al. (2024)	×

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

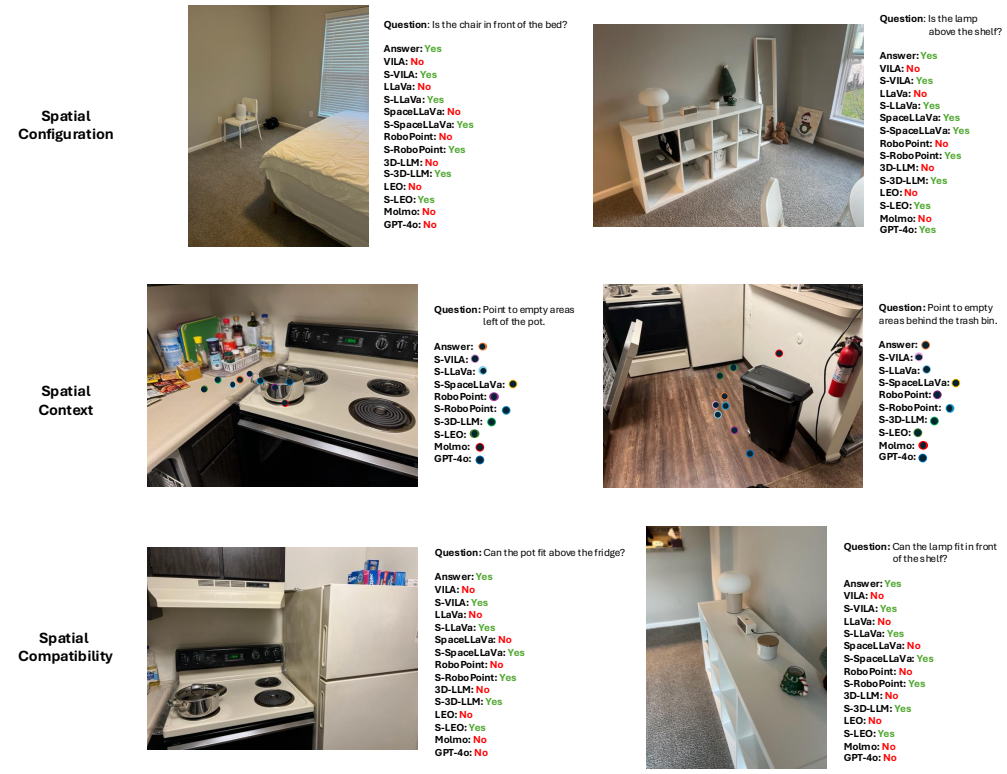


Figure 7: Qualitative results on RoboSpatial-Home. For spatial context questions, we omit the baseline models which fail to generate a point inside the image. Furthermore, the first predict point for each model is shown in the image.

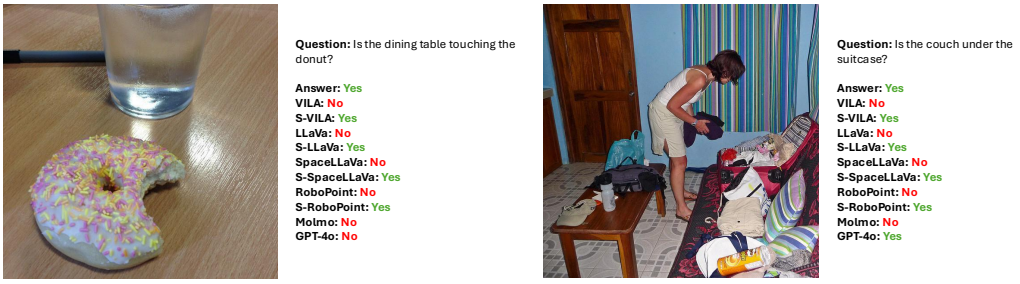


Figure 8: Qualitative results on Blink-Spatial Fu et al. (2024). RoboSpatial-trained model can generalize to unseen spatial relationships.