

ReaAct: Bridging Robotic Reasoning and Action Generation Toward Real-World Spatial Generalization

Supplementary Material

6. The Importance of Each Reasoning Step

To further validate the necessity of each reasoning stage in our hierarchical reasoning design, we conduct an additional ablation experiment by selectively dropping a single reasoning step during inference only, without retraining the model. Specifically, for a given test trajectory, we force the model to skip one intermediate reasoning stage and continue autoregressive generation using the remaining stages.

This setting exposes how each reasoning stage influences not only the final action prediction, but also the downstream reasoning outputs that depend on upstream stages. The quantitative results are provided in Table 4, and the key observations are summarized below.

Model	Acc \uparrow	DTW \uparrow	mIoU \uparrow	ADE \downarrow	AFE \downarrow
ReaAct	0.769	0.533	0.742	0.120	0.261
w/o Plan	—	0.000	0.513	0.210	0.519
w/o Coarse Trajectory	—	—	0.770	0.159	0.340
w/o Spatial Affordance	—	—	—	0.132	0.289
w/o Fine-grained Waypoint	—	—	—	0.162	0.386
ReaAct-Lite	—	—	—	0.167	0.411

Table 4. Supplementary ablation on individual reasoning steps. “—” indicates that the metric remains unchanged from ReaAct.

Table 4 shows that the *Plan* stage is the most critical component: dropping it completely collapses the global trajectory and severely decreases action precision, as the model loses the high-level semantic grounding needed to identify manipulated objects. Removing the *Spatial Affordance* step produces a noticeably smaller drop in action performance compared to removing *Coarse Trajectory*. This aligns with intuition: Spatial Affordance primarily refines local manipulation precision, whereas the Coarse Trajectory stage governs the global motion layout. As a result, dropping Spatial Affordance affects fine-grained control but has limited influence on the overall action-following behavior. Finally, removing *Fine-grained Waypoint* primarily increases low-level control errors, indicating that this stage refines motion execution.

7. Details of Reasoning Data Generation

This section provides additional details of our data generation pipeline, covering both action-conditioned reasoning annotations and the construction of large-scale in-domain VQA datasets. The goal is to derive structured spatial reasoning supervision directly from human demonstrations,

and to further enrich them with synthetic VQA pairs that introduce new spatial concepts not present in the original robot trajectories.

7.1. VQA Reasoning Data Generation

7.1.1. Image Sampling from Demonstrations

Since all VQA samples are generated from in-domain robot scenes, the visual input must originate from real demonstrations rather than external data. We use the head-view camera as the sole observation source. For each demonstration video, we ensure that the first frames corresponding to pick and place phases are preserved, and uniformly sample additional frames to form a 10-frame image set per trajectory.

7.1.2. Object Detection and Instance Segmentation

To enable spatial reasoning, each sampled frame must include reliable object-level grounding. We initialize object annotations on the first frame of every trajectory using a hybrid labeling strategy: a subset of frames is manually annotated to establish high-quality references, while the remaining ones are labeled with DINO-X to efficiently assign categories and bounding boxes. These first-frame annotations are then used to train a YOLO-v11 detector tailored to all manipulated objects in the dataset. Once detection is performed on the first frame of each sequence, we apply SAM2 to propagate the instance masks throughout the subsequent frames, ensuring consistent segmentation under heavy occlusion and motion-induced blur.

In Figure 6, we found that alternative pipelines[41] such as RAM, GroundingDINO produce incomplete object categories and unstable recall under heavy arm occlusion, whereas our YOLO + SAM2 combination yields robust tracking throughout the manipulation sequence.

7.1.3. Object-Level Captioning

Object category labels alone provide limited semantic richness. To enrich language context, we use Qwen-VL 2.5 (3B) to generate short captions for each object mask, as shown in Figure 10. A moderate decoding temperature (0.7) produces diverse but reliable descriptions, significantly improving the linguistic variety used in downstream VQA generation.

7.1.4. Recovering 3D Scene Geometry

Accurate 3D relationships are essential for creating spatial reasoning questions. We use VGGT[33] to estimate per-pixel depth. Instance masks are used to extract object-level point clouds. We recover the dominant planar surface as



Figure 5. Visualization of sampling and object detection, a):put the yellow octopus in big blue plate, b):put the Minnie Mouse in big blue plate, c):put the cow in big blue plate.



Figure 6. Visualization of depth map (above) and 3D bounding bboxes (below).

the table plane and project each object’s point cloud to this plane to obtain an oriented 3D bounding rectangle, which provides size, shape, and footprint information. All coordinates are represented in the head-camera coordinate frame to maintain consistency across trajectories.

7.1.5. Choice VQA Generation

Using object captions and recovered 3D geometry, we employ a rule-based template engine to construct four categories of spatial reasoning QA pairs: *Size*, *Position*, *Distance*, and *Vacant Region*. Each category contains multiple question formats, including single-choice, multiple-choice, and binary classification. We further design both absolute (camera-centric) and relative (object-centric) reference frames. Representative templates are shown in Table 5.

7.1.6. Relational VQA Generation

To increase linguistic diversity beyond rule-based templates, we adopt a two-stage LLM pipeline: Qwen-VL 7B generates global scene captions and detailed object descriptions. QwQ-32B produces free-form spatial reasoning questions and answers based on object captions, scene summaries, and a small set of seed Choice-VQA examples. This produces more natural, diverse, and challenging spatial reasoning data, as shown in Figure 10.

7.1.7. CoT Reasoning VQA

To align VQA reasoning with the model’s action conditioning interface, we also create CoT Reasoning VQA samples that follow the same multi-stage reasoning structure used in action annotations. Since VQA samples do not have robot trajectories, we approximate the coarse motion plan using four canonical keypoints: start, grasp, place, end. These provide a plausible long-horizon trajectory for reasoning supervision while remaining independent of robot-specific motion. Spatial affordances are represented using the same format as action data:object 2D bounding boxes for grasping and point targets or free-space indicators for placing

7.1.8. CoT Item VQA

To further improve the accuracy and stability of the model’s reasoning process, we generate CoT-Item VQA pairs by decomposing each reasoning sequence into small, independently supervised subtasks. Instead of providing an en-

Category	Example Templates
Size	Absolute: Which object is tallest , [A] or [B]? Relative: Find all objects taller than [X]: [A], [B], [C]?
Position	Absolute: Which object is leftmost , [A] or [B]? Relative: Is [A] to the left of [B], True or False?
Distance	Which object is closest to [X], [A] or [B]?
Vacant Region	Which points represent the vacant region to the left of [X], [A] or [B]?

Table 5. Representative templates used to generate choice VQA.

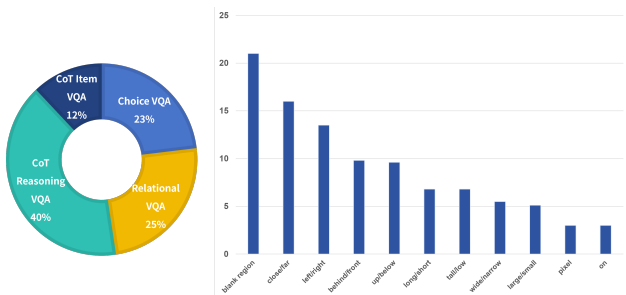


Figure 7. Details of all VQA reasoning data.

tire chain-of-thought as a single answer, each intermediate component is converted into a standalone question–answer pair. These items cover: identifying the high-level task implied by the coarse trajectory, interpreting step-by-step task procedures, determining which subtask the robot initiates with, inferring the current subtask from short-horizon waypoints, associating waypoint sequences with the overall task plan, and describing short-term waypoint patterns. This fine-grained formulation increases the density of reasoning supervision and encourages the model to master each reasoning operation individually, thereby enhancing robustness and reducing error propagation in the full CoT sequence.

7.2. Details of VQA data

Our in-domain VQA dataset contains a total of 946k automatically generated question–answer pairs derived from real robot demonstrations. As shown in Figure 7, the dataset is composed of four complementary components: 218k Choice VQA (23%), 232k Relational VQA (25%), 382k CoT Reasoning VQA (40%), and 114k CoT Item VQA (12%). Together, these formats span a spectrum from simple spatial comparisons to multi-step procedural reasoning, enabling the model to progressively acquire semantic, geometric, and task-level spatial understanding.

As summarized in Figure 7, this distribution reflects the spatial complexity present in real-world demonstrations and

provides rich supervisory signals for grounding hierarchical CoT reasoning.

To ensure annotation reliability, we perform both model-based and manual checking. A subset of QA pairs is verified by human annotators, while the others are checked through multiple VLM. Empirically, these steps significantly reduce noisy labels and help maintain high-quality spatial supervision.

8. Details of Open-loop Evaluation

To assess the model’s reasoning and action-generation quality without executing actions in the real world, we construct an open-loop evaluation set derived entirely from the in-domain robot demonstrations. Open-loop evaluation allows us to measure how accurately the model predicts *both* the reasoning steps and the actions given a single observation, without requiring the next observation that would result from physically executing the predicted action. This provides two benefits: (1) during training, it enables fine-grained monitoring of model overfitting or underfitting; (2) before real-world deployment, it offers a safety check to evaluate whether the model can reliably understand scenes, reducing the risk of hardware faults.

8.1. Construction of the Open-loop Dataset

The open-loop dataset is generated from the same in-domain trajectories used for training, following the VQA reasoning and annotation procedure described in Section 7.1. During data collection, all objects in the scene are first placed into their target containers, after which each object’s pick and place sequence is extracted as an individual demonstration. Because object occlusion is minimal and object positions remain fixed throughout recording, the action labels from different objects can be directly concatenated to form a clean and consistent supervision signal.

Following this procedure, we construct approximately 3k open-loop evaluation samples, as shown in Table 6. Each sample contains the observation image, the instruction, the multi-stage reasoning steps, and the corresponding action.

Type	Size (Abs.)	Size (Rel.)	Pos. (Abs.)	Pos. (Rel.)	Distance	All
Num	1038	584	736	337	452	3147

Table 6. Statistics of open-loop evaluation tasks types.

8.2. Example of Open-loop Evaluation

Figure 8 provides an example of our open-loop evaluation format. Given a single observation, the model must generate (1) the structured CoT reasoning and (2) the tokens.

Both outputs are compared directly with the ground-truth annotations, without executing the physical actions.

This evaluation paradigm provides a fast, safe, and interpretable way to diagnose reasoning failures and action inconsistencies before deploying the model on physical hardware.

9. Details of Real-world Deployment

9.1. In-domain Evaluation

Our in-domain evaluation focuses on testing the model using object categories and container configurations that also appear in the training set, as shown in Figure 9. To more clearly isolate the contribution of VQA-based reasoning, we evaluate under two settings:

- **ID Layout:** A spatial arrangement consistent with the layouts seen in demonstration videos. Since the VQA-CoT data is collected entirely in-domain, this setting directly reflects whether the VLA model has successfully learned transferrable spatial behaviors from the reasoning data rather than merely fitting demonstration trajectories.
- **OOD Layout:** A novel spatial arrangement that preserves object types but changes their positions. This setting examines whether the model is capable of re-computing spatial relations (e.g., leftmost, closest) under unseen visual observations, instead of memorizing reasoning outputs from the VQA dataset.

Across both layouts, we evaluate the same manipulation tasks to disentangle memorization from genuine reasoning-driven behavior transfer.

9.2. OOD-domain Evaluation

To further assess the reasoning capability of each model, we design a fully out-of-distribution (OOD) benchmark covering four families of spatial relations: *absolute size comparison*, *absolute position comparison*, *distance comparison*, and *vacant region placement*. As shown in Figure 3, each scenario is executed 10 times and all tasks are listed in Table 7.

(1) Absolute size comparison (2 tasks). Largest / smallest object selection among strawberries.

(2) Absolute position comparison (8 tasks). Leftmost / rightmost selection among lemons of different colors (yellow / green / normal).

(3) Distance comparison (6 tasks). Closest / farthest selection among candidate pairs with respect to various reference objects (strawberry, donut, rubber duck).

(4) Vacant region placement (14 tasks). Placing an object in free space located (left / right / between) relative to two other objects, including cases such as strawberry, lemon, banana, sandwich, etc.

10. Limitations and Future Work

Although extensive experiments have demonstrated that our proposed method ReaAct shows strong improvements in spatial reasoning and real-world generalization, there remain some limitations that warrant further exploration in future work.

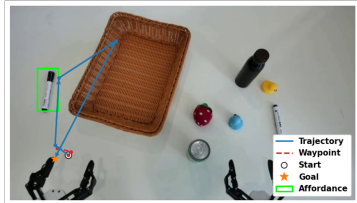
For scope of behaviors, our study focuses on spatial variants of pick-and-place. Although this setting is representative, it covers a limited range of manipulation primitives. Future extensions may incorporate more diverse behaviors such as tool use, multi-stage rearrangement, or tasks with non-prehensile actions. Integrating ReaAct with simulation-based skill discovery or leveraging modular skill libraries could broaden the applicability of the reasoning-to-action framework.

For data generation quality, the reasoning data are produced through a multi-stage pipeline involving detection, mask propagation, 3D reconstruction, and LLM-generated VQA. Although effective in practice, inevitable noise may affect supervision quality. Future work could incorporate noise-robust training objectives or confidence-weighted reasoning supervision to further enhance robustness.

For reasoning and action coupling, hierarchical CoT significantly improves interpretability, but the model may still be sensitive to upstream reasoning errors under challenging layouts. Developing confidence-aware CoT steps, or fallback strategies that replan when reasoning conflicts with visual evidence may reduce error propagation and improve reliability in OOD-domain deployments.

Overall, ReaAct offers a step toward unifying spatial reasoning and action generation, and we believe that advancing these directions can further expand its capability in complex, diverse real-world environments.

Among:blue-cappedmarker; yellow duck; black-cappedmarker, put the object which is to the left of sprite can in woven basket

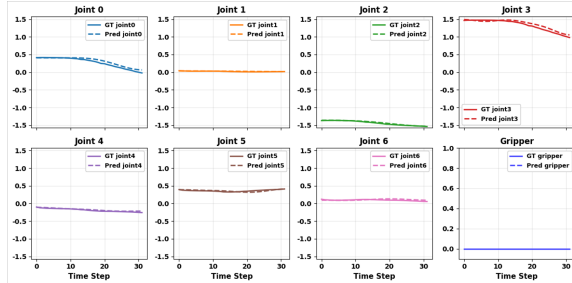


Plan: 1:the left arm pick up the black-capped marker; 2:the left arm put down the black-capped marker in woven basket; 3:back home

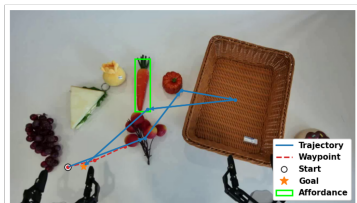
Trajectory: <loc0786><loc0172>;<loc0735><loc0134>;<loc0381><loc0145>;<loc0186><loc0315>;<loc0806><loc0132>

Affordance: the left arm pick up <loc0328><loc0080><loc0553><loc0142>black-capped whiteboard marker

Waypoint:<loc0786><loc0172>;<loc0768><loc0181>;<loc0750><loc0140>



Among:lychee; plastic carrot, put the object which is closest to pumpkin in woven basket

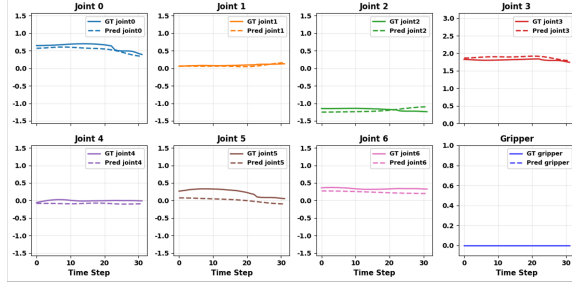


Plan: 1:the left arm pick up the plastic carrot; 2:the left arm put down the plastic carrot in woven basket; 3:back home

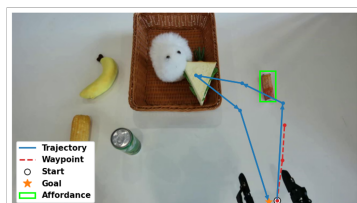
Trajectory: <loc0829><loc0171>;<loc0675><loc0400>;<loc0423><loc0508>;<loc0473><loc0668>;<loc0523><loc0408>;<loc0826><loc0217>

Affordance: the left arm pick up <loc0256><loc0369><loc0533><loc0416>plastic carrot

Waypoint:<loc0829><loc0171>;<loc0793><loc0251>;<loc0719><loc0340>



Put the smallest length object between push banana and vitamin c effervescent tablets in woven basket

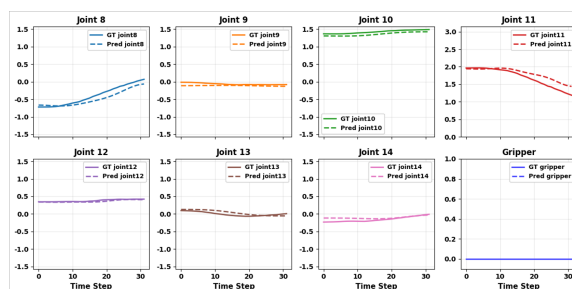


Plan: 1:the right arm pick up the vitamin c effervescent tablets; 2:the right arm put down the vitamin c effervescent tablets in woven basket; 3:back home

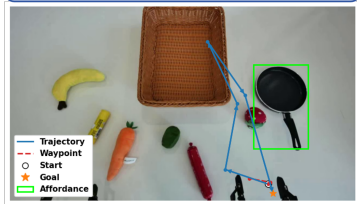
Trajectory: <loc0991><loc0786>;<loc0477><loc0803>;<loc0372><loc0681>;<loc0331><loc0544>;<loc0514><loc0676>;<loc1039><loc0785>

Affordance: the right arm pick up <loc0308><loc0723><loc0469><loc0782>vitamin c effervescent tablets

Waypoint:<loc0991><loc0786>;<loc0779><loc0802>;<loc0593><loc0808>



Among:push banana; ham sausage; black handle frying pan; green lemon, put the object which is to the right of strawberry in woven basket

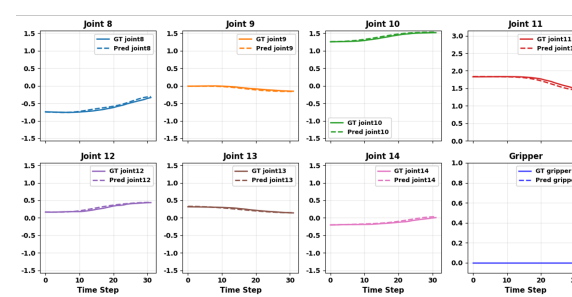


Plan: 1:the right arm pick up the black handle frying pan; 2:the right arm put down the black handle frying pan in woven basket; 3:back home

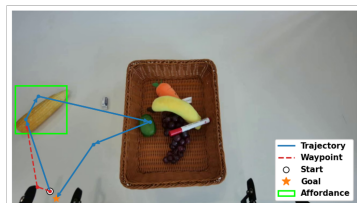
Trajectory: <loc0937><loc0766>;<loc0860><loc0639>;<loc0512><loc0674>;<loc0184><loc0586>;<loc0467><loc0697>;<loc0987><loc0780>

Affordance: the right arm pick up <loc0308><loc0723><loc0749><loc0884>black handle frying pan

Waypoint:<loc0937><loc0766>;<loc0927><loc0772>;<loc0907><loc0710>



Between eraser and corn, put the object which is farthest to grape in woven basket



Plan: 1:the left arm pick up the corn; 2:the left arm put down the corn in woven basket; 3:back home

Trajectory: <loc0952><loc0112>;<loc0574><loc0041>;<loc0452><loc0077>;<loc0587><loc0409>;<loc0704><loc0241>;<loc0992><loc0131>

Affordance: the left arm pick up <loc0396><loc0009><loc0647><loc0161>corn

Waypoint:<loc0952><loc0112>;<loc0930><loc0075>;<loc0614><loc0044>

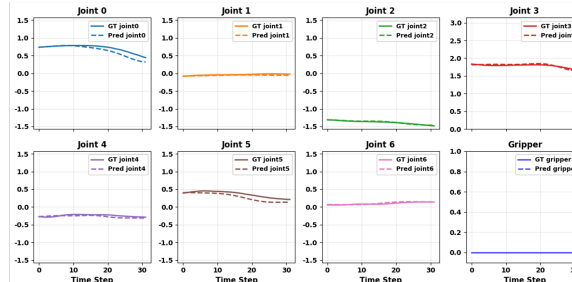
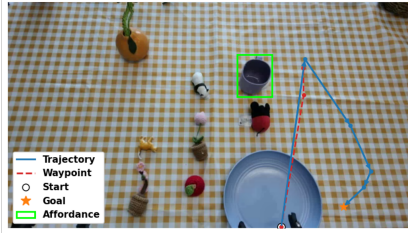
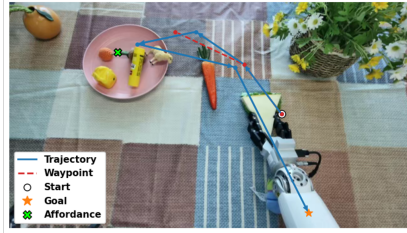


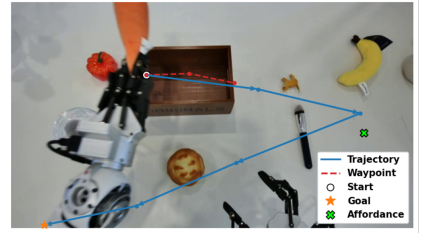
Figure 8. Open-loop evaluation examples across diverse spatial reasoning tasks. Joints 0-7 represent the left arm, and joints 8-15 represent the right arm.



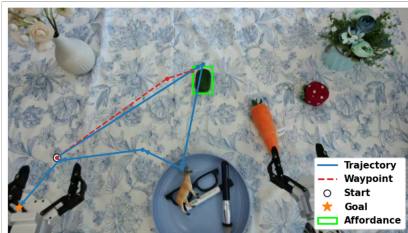
Instruction: Put the **backmost** object between woven apple and purple mug in big blue plate
Plan: 1:the right arm pick up the purple mug; 2:the right arm put down the purple mug in big blue plate; 3:back home
Affordance: the right arm pick up <loc0237><loc0584><loc0430><loc0673> purple mug



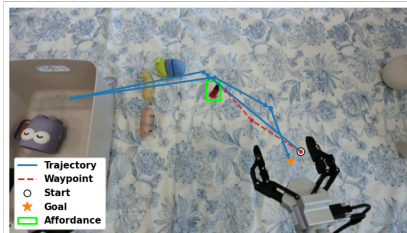
Instruction: Put the **rightmost** object between sandwich and plastic carrot in big pink plate
Plan: 1:the right arm put down the sandwich in big pink plate; 2:back home
Affordance: the right arm put down the sandwich in <loc0227><loc0275> big pink plate



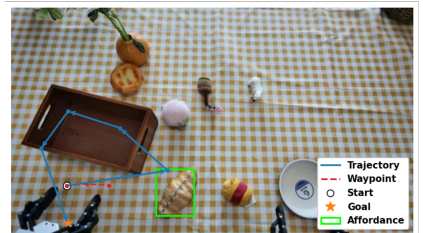
Instruction: put the plush carrot in **vacant region** in front of banana
Plan: 1:the left arm put down the plush carrot in vacant region; 2:back home
Affordance: the left arm put down the plush carrot in <loc0590><loc0898> vacant region



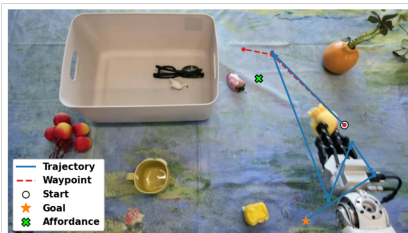
Instruction: Between green lemon and plush strawberry, put the object which is to the **left** of carrot in big blue plate
Plan: 1:the left arm pick up the green lemon; 2:the left arm put down the green lemon in big blue plate; 3:back home
Affordance: the left arm pick up <loc0267><loc0471><loc0388><loc0520> green lemon



Instruction: put the **smallest** object between pig and chicken in white storage box
Plan: 1:the right arm pick up the chicken; 2:the right arm put down the chicken in white storage box; 3:back home
Affordance: the right arm pick up <loc0332><loc0501><loc0418><loc0536> chicken



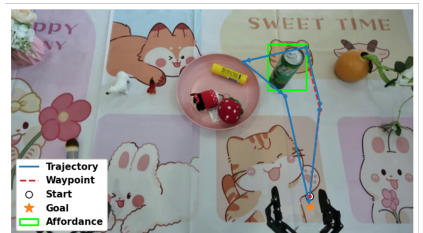
Instruction: Among:woven plant; egg tart; croissant, put the **longest** object in wooden box
Plan: 1:the left arm pick up the croissant; 2:the left arm put down the croissant in wooden box; 3:back home
Affordance: the left arm pick up <loc0731><loc0369><loc0941><loc0466> croissant



Instruction: put the yellow octopus in **vacant region** to the **right** of white storage box
Plan: 1:the right arm put down the yellow octopus in vacant region; 2:back home
Affordance: the right arm put down the yellow octopus in <loc0312><loc0637> vacant region



Instruction: Between tiger and plush carrot, put the object which is **in front of** the blue-capped whiteboard marker in large wooden tray
Plan: 1:the right arm pick up the tiger; 2:the right arm put down the tiger in large wooden tray; 3:back home
Affordance: the right arm pick up <loc0661><loc0688><loc0893><loc0732> tiger



Instruction: put the object which is **closest** to strawberry in big pink plate
Plan: 1:the right arm pick up the sprite can; 2:the right arm put down the sprite can in big pink plate; 3:back home
Affordance: the right arm pick up <loc0159><loc0653><loc0365><loc0752> sprite can

Figure 9. More qualitative examples of successful CoT reasonings.

Object Caption



- > Prompt for Qwen-VL 3B
 - Please describe the {class_name} marked by red box with this format: [Color][Material/Texture]{class_name} at [Position]
 - Each can be one word or multiple words. You may randomly include or omit them.
- > Output
transparent glass black coffee at right

Image Caption



- > Prompt for Qwen-VL 7B
 - Please carefully examine the provided images and generate a detailed description.
 - Include all visible elements such as objects, scenery, colors, shapes, textures, and lighting.
 - Ensure that your description is thorough, accurate, and complete.
- > Output
The images present a top-down view of various objects arranged on a plain white surface, which appears to be a table or desk. The lighting is bright and even, suggesting an indoor setting with good illumination. There is a rectangular woven basket with a natural brown color and a slightly textured surface, indicating a wicker or similar material. Its shape is slightly rounded at the edges, giving it a practical and decorative appearance. To the right of the basket, several small items are scattered. These include: A black cylindrical object resembling a water bottle with a dark cap. A small yellow rubber duck toy. Two markers, one white with a black cap and the other blue with a white cap. A red and white polka-dotted object that looks like a small plush toy or decoration...

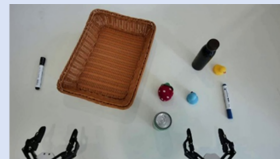
Relational VQA

- > Prompt for QwQ
You are a helpful assistant that generates robot spatial reasoning questions and answers from detailed scene descriptions.

Rules:

- We have three types of input information:
 - [Scene]: A detailed description of the full scene, including background, table/clutter layout, and all objects present.
 - [Objects]: A list containing one or more object labels separated by "|".
 - [Objects Description]: Spatial, positional, size, or coordinate information among some or all of the objects above. It may include:
 - Direct coordinates (e.g., "located at point (x, y)")
 - Size comparison (e.g., "bigger than...")
 - Placement relations (e.g., "on top of..." , "behind...")
 - Vacant regions (e.g., "These points represent the vacant region behind...")
- When crafting a Question :
 - The question should describe a robot operation scenario (picking, placing, sorting, stacking, clearing, etc.).
 - Mention at least one object from '[Objects]' .
 - Critical: Only objects mentioned in '[Objects Description]' are included in spatial comparisons or selection logic in the question.
 - Do not reveal exact spatial details (coordinates, points) in the question.
 - The question should read from an observer's perspective.
 - Make the question natural and task-oriented.
- When crafting an Answer:
 - The answer must be inferred by the robot from observation. Do not say "according to the description" .
 - Include short reasoning showing how the robot identifies the object or area.
 - If coordinates or region points exist in '[Objects Description]' , include them explicitly.
 - End with a clear robot action.
 - If only two objects are involved, explain the relative observation.
- What you need to Return:
 - Return ONLY in JSON format
 - The JSON keys must be "Question" and "Answer"
 - Example output:

```
{
  "Question": ...
  "Answer": ...
}
```



- > Output
Q: The robot needs to place the taller object between blue rubber duck and coffee into the basket. Which object should it select?
A: Observing both objects, the black coffee has greater height compared to the blue rubber duck. The robot should grasp the taller black coffee and place it into the basket.

Figure 10. Prompt for object caption, image caption, relational VQA.

ID	Category	Task Description
1	Absolute size	Put the largest-volume strawberry into the woven basket.
2	Absolute size	Put the smallest-volume strawberry into the woven basket.
3	Absolute position	Put the leftmost lemon into the woven basket.
4	Absolute position	Put the rightmost lemon into the woven basket.
5	Absolute position	Put the leftmost green lemon into the woven basket.
6	Absolute position	Put the rightmost green lemon into the woven basket.
7	Absolute position	Put the leftmost yellow lemon into the woven basket.
8	Absolute position	Put the rightmost yellow lemon into the woven basket.
9	Distance	Among (small plush strawberry, donut), place the object closest to the yellow rubber duck into the woven basket.
10	Distance	Among (small plush strawberry, donut), place the object farthest from the yellow rubber duck into the woven basket.
11	Distance	Among (yellow rubber duck, donut), place the object closest to the small plush strawberry into the woven basket.
12	Distance	Among (yellow rubber duck, donut), place the object farthest from the small plush strawberry into the woven basket.
13	Distance	Among (small plush strawberry, yellow rubber duck), place the object closest to the donut into the woven basket.
14	Distance	Among (small plush strawberry, yellow rubber duck), place the object farthest from the donut into the woven basket.
15	Vacant region	Place the green lemon in the vacant region to the right of the plush strawberry.
16	Vacant region	Place the green lemon in the vacant region to the right of the plush banana.
17	Vacant region	Place the plush banana in the vacant region to the left of the sandwich.
18	Vacant region	Place the plush banana in the vacant region to the left of the green lemon.
19	Vacant region	Place the plush strawberry in the vacant region to the left of the green lemon.
20	Vacant region	Place the plush strawberry in the vacant region to the left of the sandwich.
21	Vacant region	Place the sandwich in the vacant region to the right of the plush strawberry.
22	Vacant region	Place the sandwich in the vacant region to the right of the plush banana.
23	Vacant region	Place the plush strawberry in the vacant region between the sandwich and the plush banana.
24	Vacant region	Place the green lemon in the vacant region between the sandwich and the plush banana.

Table 7. Complete list of all real-world spatial reasoning tasks used in OOD-domain evaluation.