Text-Guided 6D Object Pose Rearrangement via Closed-Loop VLM Agents

Sangwon Baik[1], Gunhee Kim[1], Mingi Choi[1], and Hanbyul Joo[1,2]

[1] Seoul National University, Seoul, Republic of Korea
[2] RLWRLD, Seoul, Republic of Korea
https://tlb-miss.github.io/vlmpose

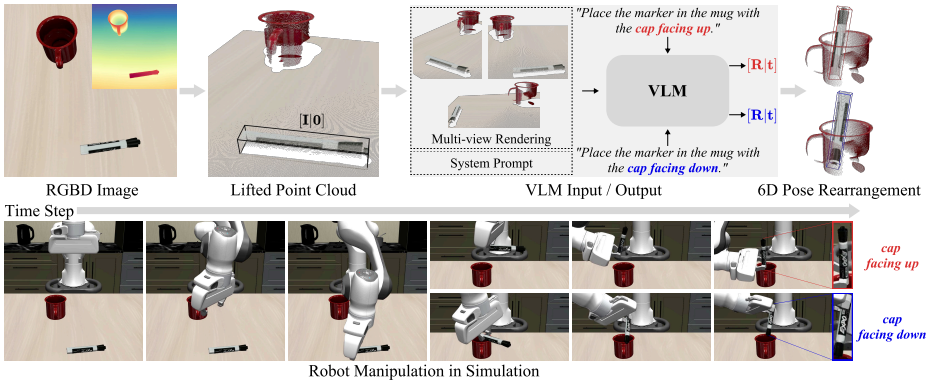


Fig. 1: Given a 3D scene and a text instruction specifying the desired goal state of the scene, our method uses a VLM to iteratively refine the target object's 6D pose in a closed loop until it reaches the goal.

Abstract. Vision-Language Models (VLMs) exhibit strong visual reasoning capabilities, yet they still struggle with 3D understanding. In particular, VLMs often fail to infer a text-consistent goal 6D pose of a target object in a 3D scene. However, we find that with some inference-time techniques and iterative reasoning, VLMs can achieve dramatic performance gains. Concretely, given a 3D scene represented by an RGB-D image (or a compositional scene of 3D meshes) and a text instruction specifying a desired state change, we repeat the following loop: observe the current scene; evaluate whether it is faithful to the instruction; propose a pose update for the target object; apply the update; and render the updated scene. Through this closed-loop interaction, the VLM effectively acts as an agent. We further introduce three inference-time techniques that are essential to this closed-loop process: (i) multi-view reasoning with supporting view selection, (ii) object-centered coordinate system visualization, and (iii) single-axis rotation prediction. Without any additional fine-tuning or new modules, our approach surpasses prior methods at predicting the text-guided goal 6D pose of the target object.

It works consistently across both closed-source and open-source VLMs. Moreover, when combining our 6D pose prediction with simple robot motion planning, it enables more successful robot manipulation than recent Vision-Language-Action models (VLAs). Finally, we conduct an ablation study to demonstrate the necessity of each proposed technique.

Keywords: Vision-Language Models · Object 6D Pose Rearrangement · Closed-Loop Refinement

1 Introduction

Modern Vision-Language Models (VLMs) are brilliant observers that can accurately describe the detailed patterns of a key, yet they remain limited in spatial reasoning when tasked with fitting that key into a lock. While these models show human-level fluency in interpreting complex scenes, they still struggle to translate a language instruction such as *"Place the marker in the mug with the cap facing up."* into a precise goal 6D pose. This spatial gap becomes more severe when VLMs infer in a feed-forward manner from a single image. These one-shot systems lack the visual feedback necessary to resolve depth ambiguities, occlusions, and complex geometric constraints.

In this work, we address this limitation by moving from passive one-shot prediction to active closed-loop refinement. We introduce a training-free framework for text-guided 6D pose rearrangement, which we formulate as a sequential decision problem over rendered observations of a 3D scene. At each iteration, the VLM alternates between an evaluator that judges whether the current scene is faithful to the instruction and a proposer that predicts an incremental 6D pose update for the target object. In the closed-loop view of our framework, this pose update serves as the agent's action. This allows the model to leverage direct visual feedback from a 3D renderer to iteratively improve alignment and correct spatial errors over multiple steps. Unlike prior methods that rely on text-based abstractions, our approach keeps the VLM in a tighter visual loop and refines the object pose directly from rendered feedback.

To make this closed-loop process stable and precise, we introduce three inference-time techniques that provide stronger 3D grounding for the VLM. First, we use **multi-view reasoning with supporting view selection** to reduce the depth ambiguity of single-view observations and handle occlusions. Second, we propose **object-centered coordinate system visualization**, which renders explicit 3D axes on the target object and provides the VLM with clear cues about direction and relative scale. Third, we use **single-axis rotation prediction** to simplify the SO(3) search space by breaking rotational reasoning into axis-aligned steps. We further maintain a context memory of prior evaluations and predicted pose updates to reduce oscillating predictions and support stable convergence toward the goal state.

The contributions of this paper are as follows:

- We introduce a closed-loop agentic framework for text-guided 6D pose rearrangement that operates entirely training-free, using pre-trained VLMs in alternating evaluator and proposer roles.
- We propose a set of inference-time 3D grounding techniques, including multi-view reasoning with supporting view selection, object-centered coordinate system visualization, and single-axis rotation prediction, which significantly improve the 3D spatial reasoning of VLMs. We further demonstrate the effectiveness of these techniques through ablation studies.
- Through extensive evaluation on the Open6DOR V2 [31] and SIMPLER [20] benchmarks, we show that our method achieves substantial performance gains, particularly in orientation-sensitive tasks, and leads to higher success rates in zero-shot robotic manipulation.

2 Related Work

2.1 Vision-Language Models as Agents

Recent work has increasingly used VLMs as interactive agents that observe visual inputs, take actions, and update their behavior based on feedback. In GUI settings, CogAgent [13] studies screenshot-based GUI understanding and navigation. ShowUI [22] formulates GUI interaction as a vision-language-action problem for visual GUI agents. UI-TARS [32] and UI-TARS-2 [37] further treat screenshot-based GUI interaction as a native agent problem with multi-step decision-making and iterative interaction. In web environments, SeeAct [43] studies grounded action generation for generalist web agents, while WebVoyager [12] demonstrates end-to-end multimodal web interaction on real websites. Beyond software interfaces, AVA [26] applies a VLM-based agent to a game environment, and VIGA [41] uses an iterative loop of generation, rendering, and verification in graphics and 3D settings. Under this broader view of VLMs as agents, we study text-guided object 6D pose rearrangement, where a VLM iteratively refines the target pose through closed-loop interaction with the rendered scene.

2.2 3D Spatial Reasoning in Vision-Language Models

Recent work improves VLM 3D reasoning via spatial supervision, geometric priors, and reconstruction. SpatialVLM [5] and SpatialRGPT [6] utilize large-scale supervision and depth-aware modeling. Meanwhile, SpatialPIN [25] enables zero-shot reasoning via 3D foundation model priors, while Spatial-MLLM [39] leverages geometry-aware features from 2D observations. Others, like VLM-3R [9] and G^2VLM [14], unify spatial reasoning with 3D reconstruction. For manipulation, VoxPoser [15] grounds language into 3D value maps, and RoboBrain [16] integrates high-level planning with affordance perception.

In robotics, generalist policies like RT-1 [3] and Octo [36] have evolved into vision-language-action (VLA) models such as RT-2 [45] and OpenVLA [18]. Recent advances further incorporate 3D representations [33, 42] and emerging foundation models [1, 2].

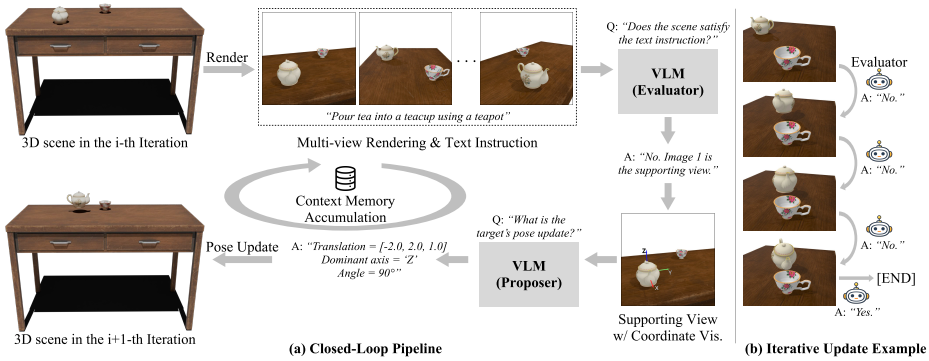


Fig. 2: Our Method Overview. (a) We first ask the evaluator to assess how well the rendered multi-view images satisfy the input text instruction. If the current scene is judged inconsistent with the instruction, we provide the proposer with the supporting view, augmented with coordinate system visualization, that best explains the evaluator's judgment. Based on this input, the proposer predicts an incremental 6D pose update. The predicted update is then applied to the target object, and this process is repeated iteratively. Throughout all iterations, both roles receive the accumulated context memory. (b) Example of iterative pose updates produced by our method for a scene containing a teacup and a teapot on a table, given the text instruction *"Pour tea into a teacup using a teapot."*

Specific to language-conditioned rearrangement, Dream2Real [17] employs VLM-based scoring over rendered candidates, Open6DOR-GPT [8] utilizes simulation assistance, and SoFar [31] relies on structured 6-DoF scene graphs. Unlike these methods that depend on candidate scoring or structured/simulator-assisted search, our approach keeps the VLM in a tighter visual loop, iteratively refining the target 6D pose directly from rendered visual feedback via inference-time techniques.

3 Method

Given a 3D scene represented either by an RGB-D image with camera intrinsics or by a composition of meshes, our goal is to rearrange the pose of a target object so that the scene satisfies a given text instruction $\mathbf{c}$. To this end, we propose a method that can be applied to pre-trained Vision-Language Models (VLMs) without introducing specialized modules or requiring additional fine-tuning. In our framework, a single VLM is used iteratively in two roles: an evaluator that measures how well the current scene aligns with the text instruction, and a proposer that predicts an incremental 6D pose update for the target object. This closed-loop process is supported by three inference-time techniques that improve the VLM's 3D reasoning performance. Fig. 2 summarizes the overall pipeline. For visual clarity, we show the mesh setting, although the same overall procedure is applied to the RGB-D setting after preprocessing.

3.1 Closed-Loop Formulation

Let $\boldsymbol{\mathcal{S}}_i$ denote the 3D scene state at i-th iteration, represented either by object meshes or by point clouds lifted from an RGB-D observation. Given an iteration-dependent camera set $\mathit{\Pi}_i = \{\pi_i^{(k)}\}_{k=1}^K$, the renderer $\mathcal{R}$ produces the multi-view images

$$\boldsymbol{\mathcal{I}}_i = \mathcal{R}(\boldsymbol{\mathcal{S}}_i, \mathit{\Pi}_i) = \{\boldsymbol{I}_i^{(k)}\}_{k=1}^K, \tag{1}$$

where $\boldsymbol{I}_i^{(k)} = \mathcal{R}(\boldsymbol{\mathcal{S}}_i, \pi_i^{(k)})$ is the rendered image of the k-th camera at iteration i. For each iteration, the evaluator $\mathit{\Phi}_{\text{eval}}$ predicts

$$(y_i, k_i, r_i^{\text{eval}}) = \mathit{\Phi}_{\text{eval}}(\boldsymbol{\mathcal{I}}_i, \mathbf{c}), \tag{2}$$

where $y_i \in \{0, 1\}$ indicates whether the current scene is faithful to the instruction $\mathbf{c}$, $k_i \in \{1, ..., K\}$ is the supporting-view index identified as the best view among rendered views, and r_i^{eval} is the evaluator's rationale for selecting $\boldsymbol{I}_i^{(k_i)}$ as the supporting view. If $y_i = 0$, meaning that the current scene does not satisfy the instruction, the proposer $\mathit{\Phi}_{\text{prop}}$ predicts an incremental 6D pose update for the target object:

$$(\hat{\mathbf{t}}_i, \omega_i, \theta_i, r_i^{\text{prop}}) = \mathit{\Phi}_{\text{prop}}(\tilde{\boldsymbol{I}}_i^{(k_i)}, \mathbf{c}), \tag{3}$$

where $\tilde{\boldsymbol{I}}_i^{(k_i)}$ denotes $\boldsymbol{I}_i^{(k_i)}$ augmented with the object-centered coordinate system visualization, $\hat{\mathbf{t}}_i \in \mathbb{R}^3$ is the translation in normalized axis units, $\omega_i \in \{x, y, z\}$ is the dominant rotation axis, $\theta_i \in \mathbb{R}$ is the rotation angle, and r_i^{prop} is the proposer's rationale for the target pose update. In the closed-loop view of our framework, $(\hat{\mathbf{t}}_i, \omega_i, \theta_i)$ is the agent's action. Applying this predicted pose update to the target object yields the next scene state $\boldsymbol{\mathcal{S}}_{i+1}$. For simplicity, context memory is omitted from Eqs. 2 and 3.

3.2 Overall Procedure with Inference-Time Techniques

To focus only on the main objects relevant to the instruction $\mathbf{c}$, we first perform target and related object selection on the initial scene $\boldsymbol{\mathcal{S}}_0$ before starting the loop. To do this, we compute an axis-aligned bounding box (AABB) that encloses all objects in the $\boldsymbol{\mathcal{S}}_0$, place cameras on a circular trajectory around its center, and render multi-view images together with per-object masks. From these images, we ask the VLM to select the view in which the objects are most clearly visible and distinguishable. Using the masks in the selected view, we annotate each object with a bounding box and label, and ask the VLM to identify the target object and the related objects mentioned in the text instruction. Additional prompt details for this step are provided in the supplementary material. Now, we write the scene state $\boldsymbol{\mathcal{S}}_i$ more explicitly as follow:

$$\boldsymbol{\mathcal{S}}_i = (\mathbf{O}^{tgt}, \mathbf{O}^{rel}, \mathbf{O}^{oth}, \mathbf{R}_i^{tgt}, \mathbf{t}_i^{tgt}), \tag{4}$$

where $\mathbf{O}^{tgt} \in \mathbb{R}^{N_{tgt} \times 3}$ is vertices of the target object, $\mathbf{O}^{rel} \in \mathbb{R}^{N_{rel} \times 3}$ represents vertices of the related objects, $\mathbf{O}^{oth} \in \mathbb{R}^{N_{oth} \times 3}$ represents vertices of the other

objects in the scene, $\mathbf{R}_i^{tgt} \in SO(3)$ is the rotation matrix of the $\mathbf{O}^{tgt}$ in the world coordinate system, and $\mathbf{t}_i^{tgt} \in \mathbb{R}^3$ is the translation vector of the $\mathbf{O}^{tgt}$ in the world coordinate system. Note that $\mathbf{O}^{tgt}$ is defined in canonical coordinates, while $\mathbf{O}^{rel}$ and $\mathbf{O}^{oth}$ are defined in world coordinates. This notation setting is chosen because the 6D pose of $\mathbf{O}^{tgt}$ changes in each iteration, but the 6D poses of the other objects do not change. $\mathbf{O}_i^{tgt} = \mathbf{R}_i^{tgt}\mathbf{O}^{tgt} + \mathbf{t}_i^{tgt} \in \mathbb{R}^{N_{tgt}\times 3}$ is the target object defined in world coordinates.

After target and related object selection, our VLM agent enters the closed-loop inference phase. We introduce the following three techniques to improve the VLM's performance in the loop: (1) multi-view reasoning with supporting view selection; (2) object-centered coordinate system visualization; and (3) single-axis rotation prediction. Specifically, technique (1) is applied to the evaluator, technique (2) is always applied to the proposer and optionally to the evaluator when directional disambiguation is needed, and technique (3) is applied to the proposer. Fig. 5 shows the necessity of each technique.

There are two challenges when inferring 6D pose rearrangement from a single-view image. First, due to depth ambiguity, it is difficult to judge whether the object is placed correctly along the forward (or backward) direction. Second, if one of the interacting objects becomes fully occluded in that view during inference, the required spatial reasoning can no longer be performed from that view alone. To address these issues, we introduce multi-view reasoning. At iteration i, Π_i consists of K equally spaced cameras placed on a circular trajectory such that the center of the AABB enclosing $\mathbf{O}_i^{tgt}$ and $\mathbf{O}^{rel}$ is the center of the frame, and all of these objects remain within the image frame. Now, multi-view images are rendered as in Eq. 1, and the evaluator Φ_{eval} reasons as in Eq. 2. Concretely, Φ_{eval} checks whether the target and related objects are present, whether the instruction-specified spatial relations are satisfied, and whether any physically implausible interpenetration is present. If the evaluator judges the current scene faithful, the loop terminates.

Otherwise, the proposer Φ_{prop} predicts the pose update through Eq. 3. The proposer takes as input the supporting view augmented with object-centered coordinate system visualization, denoted by $\tilde{I}_i^{(k_i)}$. Coordinate system visualization provides explicit visual cues for direction and scale. This is particularly important for reasoning about rotations. We apply rotations according to the right-hand rule. However, if the coordinate system is described only in text without explicit axis visualization, even humans find it difficult to distinguish clockwise from counterclockwise rotation about a given axis. It also disambiguates directional concepts such as left, right, forward, and backward across views. For instance, a direction that appears as left in the front view corresponds to right in the back view, which can easily lead the VLM to incorrect judgments. By visually associating each axis with a direction (e.g., +x is front, +y is right, and +z is top), the VLM can reason about directions consistently across views. For such direction-sensitive tasks, Φ_{eval} can also be given these augmented multi-view images. In addition, by assuming that each axis has unit length, the VLM can infer translations independently of the absolute scale of the scene. The coordinate system

is visualized in an object-centered manner. At the i-th iteration, the system's origin is set at the center of the $\mathbf{O}_i^{tgt}$'s AABB, and the frame shares its axis directions with the world coordinate system. Each axis is drawn outward from the center of the corresponding AABB face with a prescribed length. In most cases, it is sufficient to choose this axis length L such that:

$$L = \begin{cases} \min(B_x, B_y, B_z), & 2\min(B_x, B_y, B_z) \leq \max(B_x, B_y, B_z), \\ \min(B_x, B_y, B_z)/2, & \text{otherwise} \end{cases}, \tag{5}$$

where B_x, B_y, and B_z are the edge lengths of the $\mathbf{O}^{tgt}$'s (not $\mathbf{O}_i^{tgt}$'s) AABB, respectively. Note that the center of the coordinate system changes with each iteration, but the length of the axis is fixed. The VLM is instructed to assume a unit axis length, regardless of the actual axis length. Thus, the translational component of the proposer's output is converted to world coordinates as $\mathbf{t}_i = L\hat{\mathbf{t}}_i$.

Inferring an object's full 3D rotation from an image is a challenging problem even for specialized models [21, 24, 27, 29, 38] for object 6D pose estimation. In our setting, the ambiguity is even greater, since the model needs to infer the pose of the goal state rather than the current state. Therefore, directly inferring the full 3D rotation is nearly impossible for current pre-trained VLMs. However, as shown in Fig. 2, we observe that restricting rotation inference to a single axis allows VLMs to perform reasonably well in combination with iterative reasoning. Because the coordinate system is object-centered but shares its axis directions with the world coordinate system, the predicted pose update is interpreted in the same axis convention. Thus, the rotational component of the Φ_{prop}'s output is interpreted as $\mathbf{R}_i = R_{\omega_i}(\theta_i)$ in world coordinates. Now, integrating all of the above, the scene state at the $(i+1)$-th iteration is updated as follows:

$$\begin{aligned} \mathbf{R}_{i+1}^{tgt} &= \mathbf{R}_i \mathbf{R}_i^{tgt}, \;\; \mathbf{R}_i = R_{\omega_i}(\theta_i), \\ \mathbf{t}_{i+1}^{tgt} &= \mathbf{R}_i(\mathbf{t}_i^{tgt} - b_i) + b_i + \mathbf{t}_i, \;\; \mathbf{t}_i = L\hat{\mathbf{t}}_i, \\ \mathbf{O}_{i+1}^{tgt} &= \mathbf{R}_i(\mathbf{O}_i^{tgt} - b_i) + b_i + \mathbf{t}_i, \end{aligned} \tag{6}$$

where b_i is the center of the $\mathbf{O}_i^{tgt}$'s AABB.

Throughout the loop, the context memory accumulates prior evaluator judgments, previously predicted pose updates, and rationales. This memory is provided to both roles in subsequent iterations. This design improves overall loop performance, and its effect is analyzed through an ablation study in Sec. 4.4. The loop terminates when either the evaluator declares the scene faithful or the maximum number of iterations is reached. We provide an analysis of the maximum number of iterations in the supplementary material.

In the RGB-D setting, the only change is the preprocessing stage: we first lift the RGB-D input to a point cloud using the camera intrinsics, obtain object masks with off-the-shelf segmentation models [4, 19, 34, 40], and remove outlier points from each segmented object point cloud.

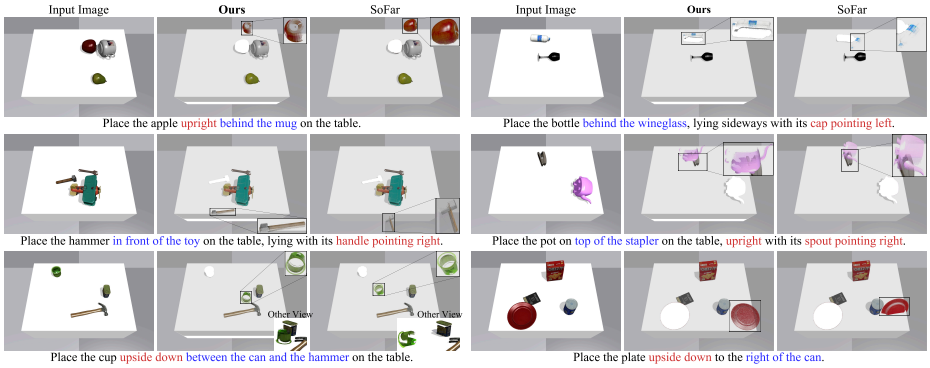


Fig. 3: Qualitative Comparison in the 6-DoF Rearrangement Task of the Open6DOR V2 Benchmark. SoFar predicts positions reasonably well but often fails on rotation, while our method predicts both position and orientation more accurately. In the prompt, the target object's goal orientation is highlighted in red, and its goal position is highlighted in blue.

Method		Position Track			Rotation Track				6-DoF Track		
		Level 0	Level 1	Overall	Level 0	Level 1	Level 2	Overall	Position	Rotation	Overall
Open6DOR-GPT [8]		56.7	78.6	61.0	N/A	N/A	N/A	N/A	N/A	N/A	N/A
SoFar [31]		96.5	**82.0**	93.7	37.2	22.3	42.4	31.3	**94.7**	28.0	26.3
Ours	GPT-5.2 [28]	**97.5**	81.6	**94.4**	**63.3**	**50.1**	**58.3**	**56.6**	91.0	**51.1**	**46.4**
	Gemini-2.5-Flash Lite [7]	95.6	78.8	92.3	45.4	16.1	45.8	32.3	83.7	32.4	26.7
	InternVL3-14B [44]	83.0	72.9	81.0	21.8	20.4	41.7	24.4	80.2	28.2	22.8

Table 1: Quantitative Comparison in the 6-DoF Rearrangement Task of the Open6DOR V2 Benchmark. We report success rates (%) on the position, rotation, and 6-DoF tracks. Our method achieves comparable position accuracy to baselines while substantially improving rotation prediction and overall 6-DoF rearrangement performance. Open6DOR-GPT is reported only on the position track because its released code supports only that track.

4 Experiments

We evaluate our framework from three perspectives. First, we measure how accurately it predicts text-guided 6D object goal poses on Open6DOR V2 [8, 31]. Second, we test whether the predicted poses are useful for downstream robotic manipulation on Open6DOR V2 and SIMPLER [20]. Finally, we analyze the contribution of each component through ablation studies. For additional experiments on the effects of iterative loops, failure cases, and other analyses, see the supplementary material.

4.1 Object 6-DoF Rearrangement Evaluation on Open6DOR V2

Dataset. Following SoFar [31], we evaluate object 6-DoF rearrangement on the Open6DOR V2 benchmark [8, 31]. It is the first benchmark for the task of inferring the goal 6D pose of a target object from natural language instructions

that specify object interactions within a scene. Open6DOR V2 consists of three tracks: the position track, the rotation track, and the 6D pose track. Each track covers a wide range of settings, from simple to complex, with higher levels corresponding to more challenging tasks. For example, in the position track, level 1 includes simple directional relations such as left, right, front, and back, whereas level 2 includes more challenging tasks such as between and center. In total, the benchmark contains 4,389 tasks. After excluding invalid scenes in which preprocessing failed, we conduct experiments on the remaining 4,370 tasks.

Metrics. Following SoFar, we report success rate as the evaluation metric. For the position and rotation tracks, a prediction is considered successful if it falls within a predefined tolerance. For the 6-DoF track, a task is counted as successful only when both the position and rotation predictions are successful.

Baselines. We compare our method with Open6DOR-GPT [8] and SoFar [31]. Like our method, both Open6DOR-GPT and SoFar use VLMs to infer the goal 6D pose of the target object. However, they rely more heavily on structured textual scene information, such as descriptions of coordinate axes and object bounding boxes, rather than directly refining poses through iterative visual feedback. Furthermore, to demonstrate the generality of our method, we apply it to multiple VLMs, including GPT-5.2 [28], Gemini-2.5-Flash Lite [7], and InternVL3 [44]. These models also span different reasoning capabilities, including a reasoning model (GPT-5.2) and weaker- or non-reasoning models (Gemini-2.5-Flash Lite and InternVL3). For fair comparison, we use GPT-5.2 as the underlying VLM for both SoFar and Open6DOR-GPT.

Results. Table. 1 shows the quantitative results of our method and the baselines on the Open6DOR V2 benchmark. Our method outperforms the baselines on almost all tasks. While SoFar performs comparably to our method on position prediction, it fails on rotation prediction in most cases. For Open6DOR-GPT, only the code for the position track was released, so we report results only for that track. Gemini-2.5-Flash Lite achieves performance comparable to that of SoFar, and InternVL3, despite being a non-reasoning VLM, outperforms Open6DOR-GPT. Fig. 3 clearly highlights the performance gap between SoFar and our method. Because it relies primarily on a text scene graph describing the target and related objects, it tends to underutilize global visual cues from the scene. As a result, it may place objects outside the table or cause the target object to collide with objects that are not included in the scene graph, leading to physically invalid configurations.

4.2 Robot Manipulation Evaluation on Open6DOR V2

Dataset and Metrics. We use the same task set and success metric as in Sec. 4.1, and evaluate them under the LIBERO [23] robot execution protocol with a Franka arm. Since Open6DOR V2 contains tasks in which the initial scene already matches the ground truth, we add an extra rule to prevent false positives: if the robot fails to grasp the target object and the scene remains unchanged, the trial is counted as a failure. We report results on 528 randomly sampled tasks.

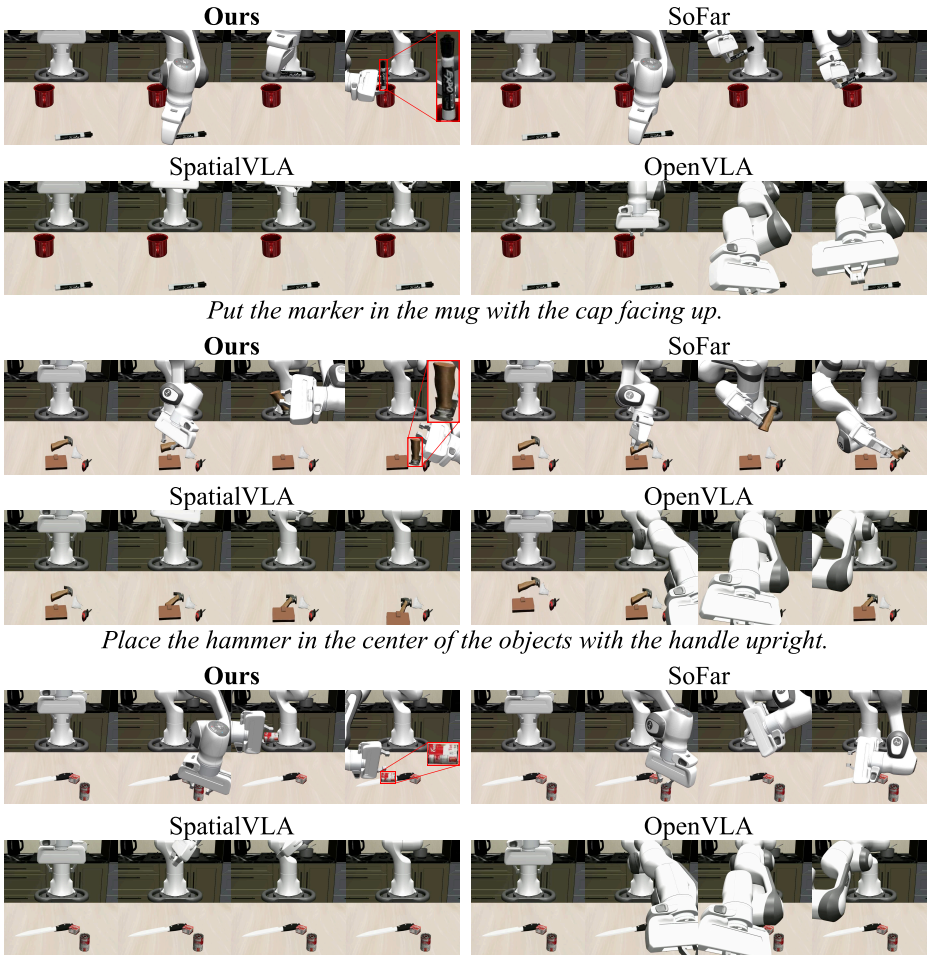


Fig. 4: Qualitative Comparison on Robot Manipulation Using Open6DOR V2 Objects and Scenes. Our method predicts both position and orientation more accurately than the baselines.

Baselines. We compare our method with SoFar and VLA methods, such as OpenVLA [18] and SpatialVLA [33]. We use Open X-Embodiment [30] pre-trained checkpoints for OpenVLA and SpatialVLA, rather than task-specific fine-tuned variants. Following SoFar, we adopt GraspNet [10, 11] for robot grasping and OMPL [35] for robot motion planning. We additionally introduce two simple modifications in the LIBERO setting to improve grasping and motion planning. In the default setup, the robot wrist is aligned with the grasp point predicted by GraspNet. We instead align the gripper center with the predicted grasp point to obtain more robust grasps. We also modify the motion trajectory from the

Method	Position Track			Rotation Track				6-DoF Track		
	Level 0	Level 1	Overall	Level 0	Level 1	Level 2	Overall	Position	Rotation	Overall
OpenVLA [18]	4.9	6.9	5.5	0.0	0.0	0.0	0.0	4.1	4.2	4.1
SpatialVLA [33]	1.4	3.4	2.0	0.0	0.0	0.0	0.0	2.5	0.6	0.6
SoFar [31]	73.9	69.0	72.5	15.6	30.6	36.4	25.3	**62.3**	17.1	12.9
Ours	**74.6**	**70.7**	**73.5**	**26.6**	**41.7**	**40.9**	**35.4**	56.5	**22.9**	**14.1**

Table 2: Quantitative Comparison in the Robot Manipulation Task of the Open6DOR V2 Benchmark. Our method demonstrates better performance than the baselines in most tasks.

Policy	Training Data	Put Spoon on Towel		Put Carrot on Plate		Stack Green Block on Yellow Block		Put Eggplant in Yellow Basket	
		Grasp Spoon	Success	Grasp Carrot	Success	Grasp Green Block	Success	Grasp Eggplant	Success
OpenVLA [18]	OXE [30]	4.2	0.0	16.7	0.0	12.5	4.2	16.7	0.0
SpatialVLA [33]	OXE [30]	25.0	12.5	45.8	25.0	75.0	33.3	**79.2**	62.5
SoFar [31]	OrienText300K [31]	62.5	58.3	79.2	**70.8**	50.0	33.3	62.5	8.3
Ours	Training-free	**62.5**	**58.3**	**83.3**	45.8	**50.0**	**37.5**	70.8	**66.7**

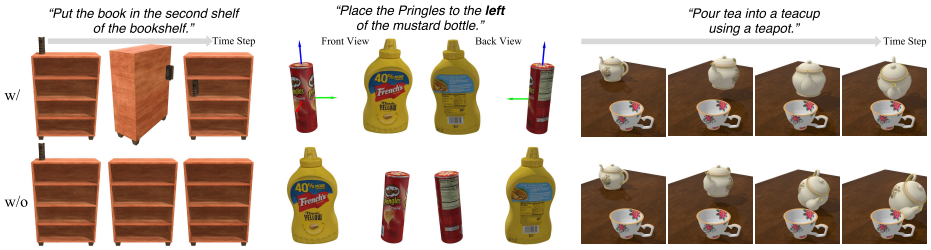
Table 3: Quantitative Comparison in the Robot Manipulation Task of the SIMPLER Benchmark. For each task, we report the final task success rate ("Success") together with a partial success metric indicating whether the target object was successfully grasped (e.g., "Grasp Spoon").

default trapezoidal lift–translate–lower path to a rectangular one: the robot first lifts the object, then moves in the xy-plane while adjusting the pose, and finally lowers it vertically at the goal. This reduces inter-object collisions and improves the success rate. We apply these modifications to both SoFar and our method for fair comparison.

Results. Tab. 2 reports the quantitative results on robot manipulation in the Open6DOR V2 benchmark. VLA models rarely work without task-specific fine-tuning. In contrast, our method works effectively without any task-specific tuning. The pose prediction quality is reflected in downstream manipulation performance. SoFar often fails when accurate orientation is required, whereas our method more reliably supports successful execution. Fig. 4 shows the effectiveness of our method. The VLA baselines fail even to grasp the target object. SoFar fails to predict the accurate orientation, causing the marker to get stuck before entering the mug or failing to place the hammer handle upright. In contrast, our method accurately predicts both position and orientation, satisfying the instruction and successfully completing the manipulation. Additional quantitative comparisons with more recent VLAs are provided in the supplementary material.

4.3 Robot Manipulation Evaluation on SIMPLER

Dataset. We evaluate on the SIMPLER [20] benchmark under the WidowX + Bridge setup, a simulated manipulation setting designed to reflect common real-



(a) Multi-view Reasoning (b) Coordinate System Visualization (c) Single-axis Rotation Prediction

Fig. 5: Qualitative Ablation Study of Inference-time Techniques. (a) Without multi-view reasoning, errors caused by occlusion cannot be corrected. (b) Without coordinate system visualization, the VLM struggles to reason consistently about directions across views. (c) Inferring the full SO(3) at once is very difficult for current VLMs.

world robot configurations. In this setup, we consider four language-conditioned manipulation tasks: *Put Spoon on Towel*, *Put Carrot on Plate*, *Stack Green Block on Yellow Block*, and *Put Eggplant in Yellow Basket*.

Metrics. We report both intermediate grasp success rate and final task success rate for each task. The former measures whether the robot successfully grasps the target object during execution, while the latter measures whether the full instructed goal state is achieved by the end of the rollout. Reporting both metrics allows us to separately evaluate object acquisition and full task completion, which is particularly informative in manipulation settings.

Baselines. We use the same baselines as in Sec. 4.2, including SoFar, OpenVLA, and SpatialVLA. We follow the evaluation protocol used in SoFar.

Results. Tab. 3 summarizes the quantitative results on SIMPLER. Despite being entirely training-free, our approach achieves competitive performance across the benchmark. In particular, our method achieves the best task success on *Stack Green Block on Yellow Block*, *Put Eggplant in Yellow Basket*, and *Put Spoon on Towel*. Fig. 6 further shows the effectiveness of our method through qualitative examples. These results suggest that iterative visual feedback enables robust downstream robot manipulation even in a fully training-free setting. Additional quantitative comparisons with more recent VLAs are provided in the supplementary material.

4.4 Ablation Study

Dataset and Metrics. We use Open6DOR V2 as in Sec. 4.1. From each track (position, rotation, and 6-DoF), we randomly sample 180 subtasks and evaluate on a total of 540 subtasks. We report success rate as the evaluation metric, following Sec. 4.1.

Ablation Settings. We ablate the context memory accumulation and the three inference-time techniques described in Sec. 3.2. Specifically, we evaluate the performance drop caused by removing each of these four components individually.

Method				Position Track	Rotation Track	6-DoF Track
MV Reasoning	Coord Vis	Single-Axis Rot	Context Memory	Overall	Overall	Overall
✓	✓	✓		77.8	52.8	28.9
✓	✓		✓	80.6	57.6	36.7
✓		✓	✓	71.7	56.9	31.1
	✓	✓	✓	81.7	60.4	38.9
✓	✓	✓	✓	**85.6**	**61.1**	**40.6**

Table 4: Quantitative ablation results on Open6DOR V2 with different combinations of (i) multi-view reasoning, (ii) coordinate system visualization, (iii) single-axis rotation prediction, and (iv) context memory accumulation.

(1) Without multi-view reasoning with supporting view selection, the evaluator is forced to reason from a single input image. (2) Without coordinate system visualization, the coordinate system is described only in text, without any explicit visual overlay. (3) Without single-axis rotation prediction, the VLM is asked to infer 3D Euler angles, which are then applied in x–y–z order. (4) Without context memory accumulation, the VLM does not receive context memory as an additional input.

Results. Tab. 4 shows the evaluation results on a subset of Open6DOR V2. Applying all four components yields the best performance across all tasks. We observe that coordinate system visualization and context memory accumulation are particularly important for 6D pose prediction. Coordinate system visualization has a substantial impact on the position track, likely because Open6DOR V2 contains many directional instructions such as front, back, left, and right, and visual cues for the coordinate system play a crucial role in helping the VLM perceive these directions. The effect of removing single-axis rotation prediction is relatively small, likely because most rotation tasks in the Open6DOR V2 benchmark involve only simple target orientations such as upright or lying. In practice, even when instructed to infer 3D Euler angles, the VLM often behaved as though it were performing single-axis rotation prediction, providing a nonzero angle for only one axis while setting the others to zero in most cases. Finally, the effect of multi-view reasoning is relatively small on Open6DOR V2, likely because the benchmark mainly involves simple tabletop placement and the input view already provides sufficient spatial information as shown in Fig. 3.

To further show the importance of each technique, we present qualitative results in Fig. 5. In the (bookshelf, book) example, the VLM struggles to recover from severe occlusion without multi-view reasoning. In the (pringles, mustard bottle) example, it becomes confused when directional concepts are specified only in text without coordinate system visualization. In the (teacup, teapot) example, it fails on nontrivial orientations such as *pour* when asked to infer the full SO(3) at once.

5 Discussion

In this paper, we proposed a framework that uses Vision-Language Models as active closed-loop agents for text-guided 6D object pose rearrangement. By integrating inference-time techniques such as multi-view reasoning with supporting view selection, object-centered coordinate system visualization, and single-axis rotation prediction, our method effectively mitigates depth ambiguity and the difficulty of full 3D rotation reasoning that plague traditional open-loop, single-view approaches. Extensive evaluations across multiple benchmarks demonstrate that our framework significantly improves goal 6D pose prediction capabilities and leads to higher success rates in zero-shot robotic manipulation tasks.

Despite these strong results, our approach has a few limitations. The primary limitation is inference speed. Operating as a closed-loop system that repeatedly queries a VLM and renders visual feedback at each step incurs a higher computational cost and latency compared to one-shot models. This makes the current framework less suitable for real-time robotic control scenarios that require low-latency responses. Additionally, the performance of our system remains inherently limited by the visual understanding ability of the underlying model. Reducing the computational cost of closed-loop inference and improving the robustness of the underlying VLM are important next steps toward deploying this framework in real-time robotic manipulation.

Acknowledgements

This work was supported by RLWRLD, Institute of Engineering Research at Seoul National University, NRF grant funded by the Korean government (MSIT) [No. RS-2022-NR070498, No. RS-2025-25396144, No. RS-2026-25540546, and No. PJT-25-122310], and IITP grant funded by the Korea government (MSIT) [No. RS-2024-00439854, No. RS-2025-25442338, and No. RS-2021-II211343]. H. Joo is the corresponding author.

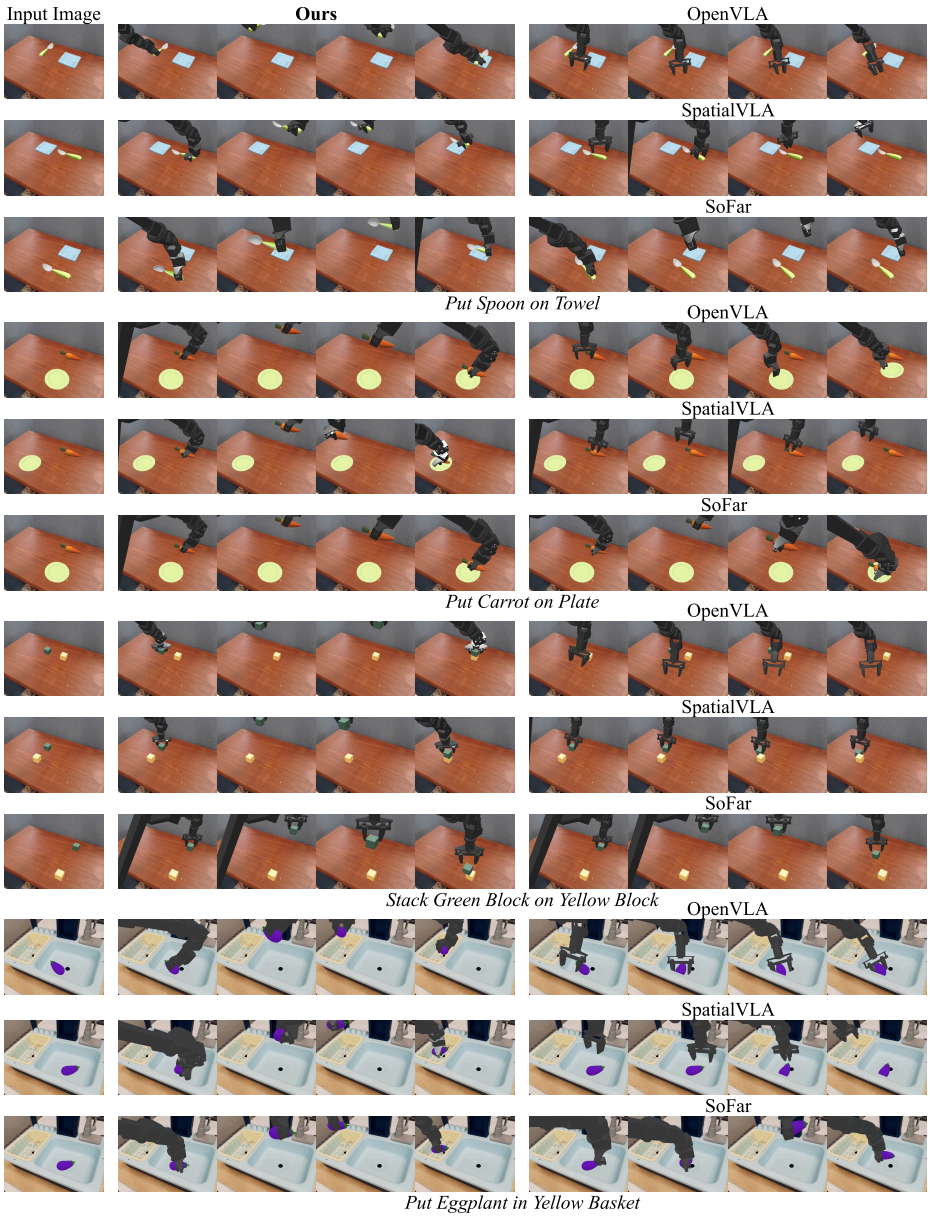


Fig. 6: Qualitative Comparison in the Robot Manipulation Task of the SIMPLER Benchmark under the WidowX + Bridge Setup.

References

1. Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., et al.: Gr00t n1: An open foundation model for generalist humanoid robots. arXiv:2503.14734 (2025)
2. Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al.: π_0 : A vision-language-action flow model for general robot control. arXiv:2410.24164 (2024)
3. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: Rt-1: Robotics transformer for real-world control at scale. arXiv:2212.06817 (2022)
4. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., Lei, J., Ma, T., Guo, B., Kalla, A., Marks, M., Greer, J., Wang, M., Sun, P., Rädle, R., Afouras, T., Mavroudi, E., Xu, K., Wu, T.H., Zhou, Y., Momeni, L., Hazra, R., Ding, S., Vaze, S., Porcher, F., Li, F., Li, S., Kamath, A., Cheng, H.K., Dollár, P., Ravi, N., Saenko, K., Zhang, P., Feichtenhofer, C.: Sam 3: Segment anything with concepts. arXiv:2511.16719 (2025)
5. Chen, B., Xu, Z., Kirmani, S., Ichter, B., Sadigh, D., Guibas, L., Xia, F.: Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In: CVPR (2024)
6. Cheng, A.C., Yin, H., Fu, Y., Guo, Q., Yang, R., Kautz, J., Wang, X., Liu, S.: Spatialrgpt: Grounded spatial reasoning in vision-language models. In: NeurIPS (2024)
7. Comanici, G., Bieber, E., Schaekermann, M., et al.: Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. arXiv:2507.06261 (2025)
8. Ding, Y., Geng, H., Xu, C., Fang, X., Zhang, J., Wei, S., Dai, Q., Zhang, Z., Wang, H.: Open6dor: Benchmarking open-instruction 6-dof object rearrangement and a vlm-based approach. In: IROS (2024)
9. Fan, Z., Zhang, J., Li, R., Zhang, J., Chen, R., Hu, H., Wang, K., Qu, H., Wang, D., Yan, Z., et al.: Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. arXiv:2505.20279 (2025)
10. Fang, H.S., Gou, M., Wang, C., Lu, C.: Robust grasping across diverse sensor qualities: The graspnet-1billion dataset. IJRR (2023)
11. Fang, H.S., Wang, C., Gou, M., Lu, C.: Graspnet-1billion: A large-scale benchmark for general object grasping. In: CVPR (2020)
12. He, H., Yao, W., Ma, K., Yu, W., Dai, Y., Zhang, H., Lan, Z., Yu, D.: Webvoyager: Building an end-to-end web agent with large multimodal models. In: ACL (2024)
13. Hong, W., Wang, W., Lv, Q., Xu, J., Yu, W., Ji, J., Wang, Y., Wang, Z., Dong, Y., Ding, M., Tang, J.: Cogagent: A visual language model for gui agents. In: CVPR (2024)
14. Hu, W., Lin, J., Long, Y., Ran, Y., Jiang, L., Wang, Y., Zhu, C., Xu, R., Wang, T., Pang, J.: G^2vlm: Geometry grounded vision language model with unified 3d reconstruction and spatial reasoning. arXiv:2511.21688 (2025)
15. Huang, W., Wang, C., Zhang, R., Li, Y., Wu, J., Fei-Fei, L.: Voxposer: Composable 3d value maps for robotic manipulation with language models. In: CoRL (2023)
16. Ji, Y., Tan, H., Shi, J., Hao, X., Zhang, Y., Zhang, H., Wang, P., Zhao, M., Mu, Y., An, P., et al.: Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In: CVPR (2025)

17. Kapelyukh, I., Ren, Y., Alzugaray, I., Johns, E.: Dream2real: Zero-shot 3d object rearrangement with vision-language models. In: ICRA (2024)
18. Kim, M., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., Finn, C.: Openvla: An open-source vision-language-action model. In: CoRL (2024)
19. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. arXiv:2304.02643 (2023)
20. Li, X., Hsu, K., Gu, J., Pertsch, K., Mees, O., Walke, H.R., Fu, C., Lunawat, I., Sieh, I., Kirmani, S., Levine, S., Wu, J., Finn, C., Su, H., Vuong, Q., Xiao, T.: Evaluating real-world robot manipulation policies in simulation. In: CoRL (2024)
21. Lin, J., Liu, L., Lu, D., Jia, K.: Sam-6d: Segment anything model meets zero-shot 6d object pose estimation. In: CVPR (2024)
22. Lin, K.Q., Li, L., Gao, D., Yang, Z., Wu, S., Bai, Z., Lei, S.W., Wang, L., Shou, M.Z.: Showui: One vision-language-action model for gui visual agent. In: CVPR (2025)
23. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. In: NeurIPS (2023)
24. Liu, L., Lin, J., Liu, Z., Jia, K.: Picopose: Progressive pixel-to-pixel correspondence learning for novel object pose estimation. In: CoRL (2025)
25. Ma, C., Lu, K., Cheng, T.Y., Trigoni, N., Markham, A.: Spatialpin: Enhancing spatial reasoning capabilities of vision-language models through prompting and interacting 3d priors. In: NeurIPS (2024)
26. Ma, W., Fu, Y., Zhang, Z., Ghanem, B., Li, G.: Ava: Attentive vlm agent for mastering starcraft ii. arXiv:2503.05383 (2025)
27. Nguyen, V.N., Groueix, T., Salzmann, M., Lepetit, V.: Gigapose: Fast and robust novel object pose estimation via one correspondence. In: CVPR (2024)
28. OpenAI: Chatgpt: Optimizing language models for dialogue (2023), `https://openai.com/blog/chatgpt`
29. Örnek, E.P., Labbé, Y., Tekin, B., Ma, L., Keskin, C., Forster, C., Hodan, T.: Foundpose: Unseen object pose estimation with foundation features. In: ECCV (2024)
30. O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., Jain, A., et al.: Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In: ICRA (2024)
31. Qi, Z., Zhang, W., Ding, Y., Dong, R., Yu, X., Li, J., Xu, L., Li, B., He, X., Fan, G., Zhang, J., He, J., Gu, J., Jin, X., Ma, K., Zhang, Z., Wang, H., Yi, L.: Sofar: Language-grounded orientation bridges spatial reasoning and object manipulation. In: NeurIPS (2025)
32. Qin, Y., Ye, Y., Fang, J., Wang, H., Liang, S., Tian, S., Zhang, J., Li, J., Li, Y., Huang, S., et al.: Ui-tars: Pioneering automated gui interaction with native agents. arXiv:2501.12326 (2025)
33. Qu, D., Song, H., Chen, Q., Yao, Y., Ye, X., Ding, Y., Wang, Z., Gu, J., Zhao, B., Wang, D., et al.: Spatialvla: Exploring spatial representations for visual-language-action model. In: RSS (2025)
34. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: Sam 2: Segment anything in images and videos. arXiv:2408.00714 (2024)

35. Sucan, I.A., Moll, M., Kavraki, L.E.: The open motion planning library. IEEE Robotics & Automation Magazine **19**(4), 72–82 (2012)
36. Team, O.M., Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Kreiman, T., Xu, C., et al.: Octo: An open-source generalist robot policy. In: RSS (2024)
37. Wang, H., Zou, H., Song, H., Feng, J., Fang, J., Lu, J., Liu, L., Luo, Q., Liang, S., Huang, S., et al.: Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. arXiv:2509.02544 (2025)
38. Wen, B., Yang, W., Kautz, J., Birchfield, S.: Foundationpose: Unified 6d pose estimation and tracking of novel objects. In: CVPR (2024)
39. Wu, D., Liu, F., Hung, Y.H., Duan, Y.: Spatial-mllm: Boosting mllm capabilities in visual-based spatial intelligence. In: NeurIPS (2025)
40. Xiao, B., Wu, H., Xu, W., Dai, X., Hu, H., Lu, Y., Zeng, M., Liu, C., Yuan, L.: Florence-2: Advancing a unified representation for a variety of vision tasks. In: CVPR (2024)
41. Yin, S., Ge, J., Wang, Z.Z., Li, X., Black, M.J., Darrell, T., Kanazawa, A., Feng, H.: Vision-as-inverse-graphics agent via interleaved multimodal reasoning. arXiv:2601.11109 (2026)
42. Zhen, H., Qiu, X., Chen, P., Yang, J., Yan, X., Du, Y., Hong, Y., Gan, C.: 3d-vla: A 3d vision-language-action generative world model. In: ICML (2024)
43. Zheng, B., Gou, B., Kil, J., Sun, H., Su, Y.: Gpt-4v (ision) is a generalist web agent, if grounded. arXiv:2401.01614 (2024)
44. Zhu, J., Wang, W., Chen, Z., Liu, Z., Ye, S., Gu, L., Tian, H., Duan, Y., Su, W., Shao, J., et al.: Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. arXiv:2504.10479 (2025)
45. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., Vuong, Q., Vanhoucke, V., Tran, H., Soricut, R., Singh, A., Singh, J., Sermanet, P., Sanketi, P.R., Salazar, G., Ryoo, M.S., Reymann, K., Rao, K., Pertsch, K., Mordatch, I., Michalewski, H., Lu, Y., Levine, S., Lee, L., Lee, T.W.E., Leal, I., Kuang, Y., Kalashnikov, D., Julian, R., Joshi, N.J., Irpan, A., brian ichter, Hsu, J., Herzog, A., Hausman, K., Gopalakrishnan, K., Fu, C., Florence, P., Finn, C., Dubey, K.A., Driess, D., Ding, T., Choromanski, K.M., Chen, X., Chebotar, Y., Carbajal, J., Brown, N., Brohan, A., Arenas, M.G., Han, K.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. In: CoRL (2023)