Object-Centric Prompt-Driven Vision-Language-Action Model for Robotic Manipulation

Supplementary Material

1. Input Details

We outline the complete input language and ground-truth supervision in Section3.3.2:

Input(a): "Predict the contact point and orientation for manipulating the object. The hints in the image include the contact point with a blue dot. Specifically, the contact point is at a_0^p."

Input(b): "Predict the contact point and orientation for manipulating the object. The hints in the image include a blue dot for the contact point and a red line for the gripper z-axis 2D direction. Specifically, the contact point is at a_0^p, and the gripper z-axis 2D direction is a_0^z."

Input(c): "Predict the contact point and orientation for manipulating the object. The hints in the image include a blue dot for the contact point, a red line for the gripper z-axis 2D direction, and a green line for the gripper y-axis 2D direction. Specifically, the contact point is at a_0^p, the gripper z-axis 2D direction is a_0^z, and the gripper y-axis 2D direction is a_0^y."

Input(d): "Predict the contact point and orientation for manipulating the object. The hints in the image include a blue dot for the contact point, a red line for the gripper z-axis 2D direction, a green line for the gripper y-axis 2D direction, and a yellow line for the moving 2D direction. Specifically, the contact point is at a_0^p, the gripper z-axis 2D direction is a_0^z, the gripper y-axis 2D direction is a_0^y, and the gripper moving 2D direction is a_0^m."

Ground truth: "The contact point is at a_0^p, the gripper z-axis 3D direction is $a_0^{Z'}$, the gripper y-axis 3D direction is $a_0^{Y'}$, and the moving 3D direction is $a_0^{M'}$."

2. Data Collection Details

The size of the training dataset is around 10,000. Regarding the variation between training and testing data, the specific variations can be divided into two aspects: 1) asset variation and 2) camera view variation.

Asset Variation: We use 20 categories from PartNet-mobility [1] for seen objects and reserve the remaining 10 categories for unseen objects to analyze whether Crayon-Robo can generalize to novel categories. Specifically, we further divide the seen objects into 1,037 training shapes and 489 testing shapes, using only the training shapes to construct the training data. Thus, the shapes of the seen objects encountered during training and testing are different. For unseen categories, there are a total of 274 shapes, which are used exclusively in the testing data.

Camera View Variation: We observe the object in the scene from an RGB-D camera with known intrinsics, mounted 4.5 to 5.5 units away from the object, facing its center. The camera is located in the upper hemisphere of the object with a random azimuth between 45 and -45, and a random altitude between 30 and 60. We initialize the starting pose for each articulated part randomly between its rest joint state (fully closed) and any position up to half of its joint state (half-opened). In both the training and testing phases, the object is placed and captured randomly within the aforementioned scope.

3. Details of RT-Trajectory Replication

Since the code for RT-Trajectory is not publicly available, we replicate its method based on the paper's description. During data collection in the simulator, we record the 3D position of the end-effector and project it onto the camera frame to create the corresponding 2D trajectory. Given that the tasks are atomic, consisting of a single step (e.g., opening a door), color grading is unnecessary. Instead, we mark the start and end positions, as well as the gripper state, by drawing blue and green circles, respectively.

We use the same backbone as in our model, the LLaMA-adapter, and fine-tune it to process both the trajectory image and the current object image. This allows the model to output the 6DoF poses required to complete the tasks. The same training and testing splits are applied, resulting in an average success rate of 0.57 on seen categories and 0.52 on unseen categories for RT-Trajectory, while our model achieves 0.74 and 0.72, respectively.

Further investigation reveals that in our replication of RT-Trajectory, while the method accurately captures the end-effector's trajectory position, the rotation estimation is not precise enough for interacting with articulated objects. Unlike tasks such as pick-and-place, where the end-effector's rotation is relatively uniform, interactions with articulated objects demand more diverse and complex rotational adjustments, making it challenging for RT-Trajectory to learn effectively. This also highlights the need to provide directional prompts for the model to interpret.

4. Visual Demonstration of Typical Atomic Tasks

In Figure. 1, we demonstrate how the key-frame tasks discussed in Section 3.4.1 are executed. We focus on the "rotation button" task, which is unique because it does not in-

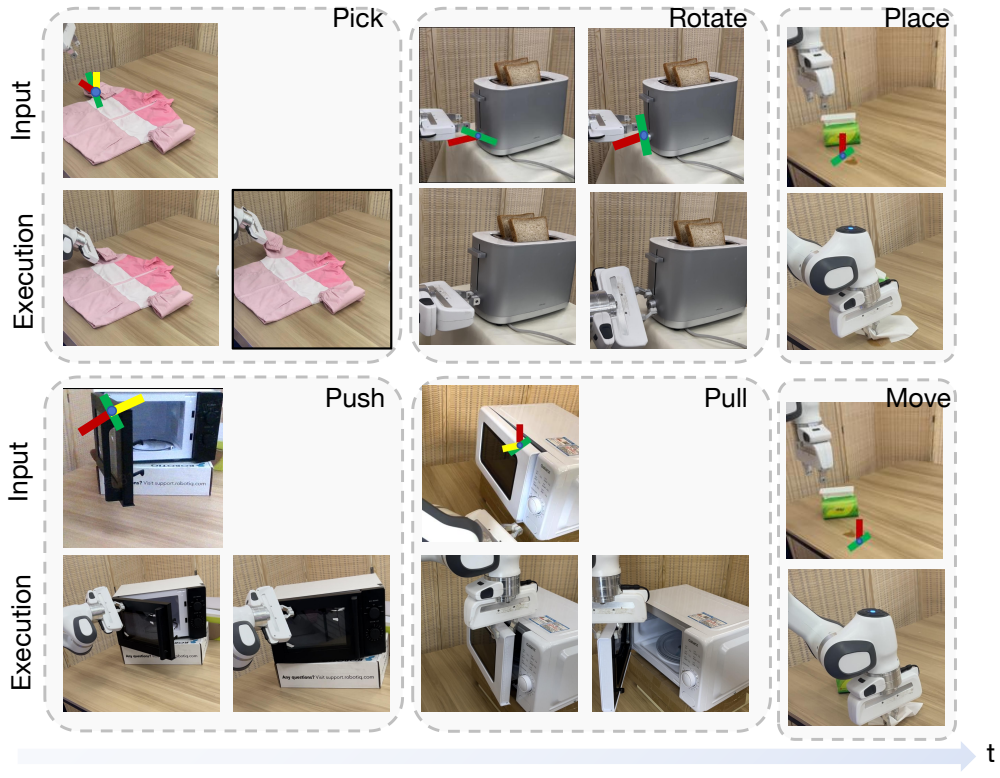

Figure 1. The completion of tyical key-frame tasks.

volve the translation of the end effector but rather the rotation of the last joint. Using visual prompts from two key frames, we predict two poses. By analyzing the predicted poses, we observe that the only difference between them is the orientation of the end effector along the y-axis. We then pass these predictions to the ROS package for inverse kinematics (IK) computation, which allows us to achieve the desired rotation effect at the last joint.

5. The Effectiveness of Each Loss.

	$\mathcal{L}_T$	$\mathcal{L}_O$	$\mathcal{L}_P$	Seen	Unseen
Ex1	✓	-	-	0.68	0.57
Ex2	✓	✓	-	0.71	0.70
Ours	✓	✓	✓	0.74	0.72

Table 1. The effectiveness of each loss

In Table. 1, we progressively introduce each loss objective during training: Ex1 involves training solely with $\mathcal{L}_T$; Ex2 combines $\mathcal{L}_T$ with $\mathcal{L}_O$; and Ours integrates $\mathcal{L}_T$ with both $\mathcal{L}_O$ and $\mathcal{L}_P$. Comparing Ex2 and Ex1, we observe that incorporating $\mathcal{L}_O$ enhances accuracy by explicitly enforcing the orthogonality constraints between the z-axis and y-axis directions. Furthermore, the addition of $\mathcal{L}_P$ in Ours results in a further accuracy improvement compared to Ex2, showing its effectiveness in capturing the correlation between 2D prompts and 3D directions.

6. Failure Analysis

We analyze the failure cases in real-world: for the push button step in the open microwave task, the excessive reactive force during button pressing prevented the robotic arm from completing the push successfully. For the slide lever step in the heat toaster task, the gripper fingers we use are too short, which sometimes prevent them from firmly contacting the lever during movement.

References

[1] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1