

Robotic Visual Instruction

Yanbang Li^{1†} Ziyang Gong² Haoyang Li³ Xiaoqi Huang⁴ Haolan Kang⁵
 Guangping Bai¹ Xianzheng Ma¹

¹ Independent Researcher ² Shanghai AI Laboratory ³ UCSD ⁴ VIVO

⁵ South China University of Technology

[†] Corresponding Author

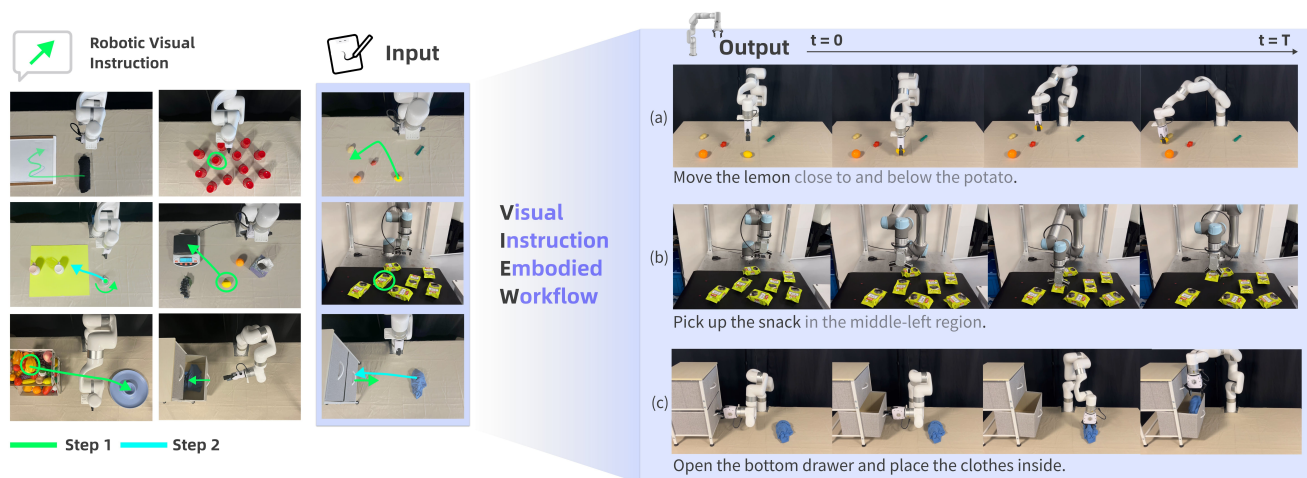


Figure 1. (Left) Robotic visual instruction is a hand-drawn approach for commanding robots, utilizing circles and arrows to convey task definition. In long-horizon tasks, green and blue sketches denote the first and second task steps, respectively. (Right) It illustrates the action sequences output via VIEW. Our method exhibits robust generalization to real-world manipulation tasks, including (a) trajectory-following tasks, (b) cluttered environments with disturbances, and (c) multi-step operations.

Abstract

Recently, natural language has been the primary medium for human-robot interaction. However, its inherent lack of spatial precision for robotic control introduces challenges such as ambiguity and verbosity. To address these limitations, we introduce the **Robotic Visual Instruction (RoVI)**, a novel paradigm to guide robotic tasks through an object-centric, hand-drawn symbolic representation. RoVI effectively encodes spatial-temporal information into human-interpretable visual instructions through 2D sketches, utilizing arrows, circles, colors, and numbers to direct 3D robotic manipulation. To enable robots to understand RoVI better and generate precise actions based on RoVI, we present **Visual Instruction Embodied Workflow (VIEW)**, a pipeline formulated for RoVI-conditioned policies. This approach leverages Vision-Language Models (VLMs) to interpret RoVI inputs, decode spatial and temporal constraints from 2D pixel space via keypoint extraction, and then trans-

form them into executable 3D action sequences. We additionally curate a specialized dataset of 15K instances to fine-tune small VLMs for edge deployment, enabling them to effectively learn RoVI capabilities. Our approach is rigorously validated across 11 novel tasks in both real and simulated environments, demonstrating significant generalization capability. Notably, VIEW achieves an 87.5% success rate in real-world scenarios involving unseen tasks that feature multi-step actions, with disturbances, and trajectory-following requirements. Code and Datasets in this paper will be released soon.

1. Introduction

Natural language, is not always the optimal medium between humans and robots. Alternatively, sketching visual instructions convey more precise spatiotemporal information. Traditionally, communication between humans and robots relies on natural language, leveraging the advances

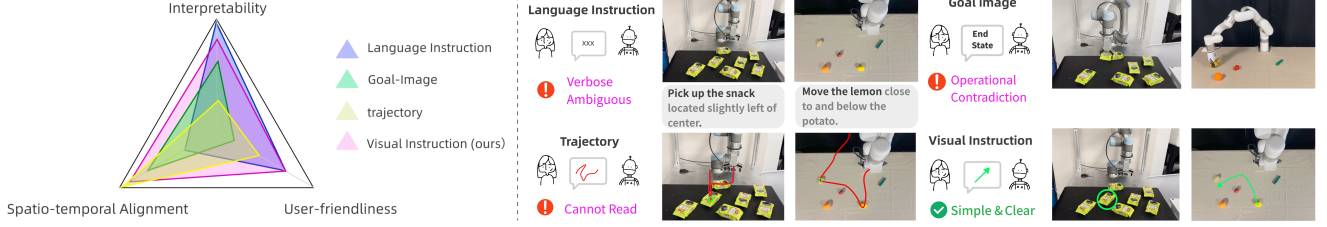


Figure 2. (Left) RoVI achieves an optimal balance of user-friendliness, interpretability, and spatiotemporal alignment. (Right) It shows examples and corresponding pros and cons of four types of human-robot interaction methods.

in large language models (LLMs) to convert verbal or textual language instructions into executable actions for robots [7, 9, 33, 39]. While natural language is an intuitive and convenient medium for Human-Robot Interaction (HRI), it presents certain challenges. Specifically, natural language has difficulty in describing spatial details such as the precise position, direction, or distance of objects [12, 23]. It is also prone to generating ambiguity and verbosity when expressing spatial requirements [6, 46] shown in Figure 2. Moreover, in certain public environments, such as libraries and hospitals, verbal communication may be inappropriate.

In contrast, visual modalities—such as goal images [14, 42, 45], trajectories [21, 48, 51], and subgoal images [32, 45]—offer a more direct and precise means of conveying spatio-temporal information. However, the practical application of such methods is not user-friendly shown in Figure 2. The goal image requires the input of the end state of the robotic arm and the scene upon task completion, which contradicts the user’s operational sequence. On the other hand, the trajectory represents the complete path of the end effector from the first to the last frame, posing challenges for users to imagine and draw the entire motion process of the robotic arm, which reduces the overall readability for users.

To address these limitations, we propose a novel communication paradigm: **Robotic Visual Instruction (RoVI)** shown in the left part of Figure 1, *which is a hand-drawn sketch instruction method, an object-centric representation that utilizes 2D symbolic language to command 3D embodiments*. This paradigm offers an intuitive, concise, and silent alternative to natural language instruction. Its basic primitives include arrows, circles, and various colors or numbers to represent different temporal sequences of actions. The arrows indicate the trajectory and direction, while the circles denote affordance location to identify target objects in a cluttered environment. Colors or numbers clearly convey the temporal sequence. By integrating these elements, RoVI compresses a temporal series of 3D coordinates into a human-understandable 2D visual language, thereby achieving an optimal balance of user-friendliness, interpretability, and spatiotemporal alignment, as shown in Figure 2 left.

In order to better understand RoVI and use it to guide robotic manipulation, we introduce **Visual Instruction**

Embodied Workflow (VIEW), a pipeline that transduces two-dimensional RoVI instructions into three-dimensional action sequences for robotic manipulation. VIEW facilitates the robotic interpretation of visual instructions, and translates them into hierarchical language responses and Python code functions via Vision-Language Models (VLMs). To decode temporal information from RoVI for high-level tasks, we decompose these tasks into multiple single-step subtasks based on color or numerical identifiers. Furthermore, we propose a keypoint module to extract keypoints from various RoVI components to serve as additional spatial and temporal constraints. Ultimately, our keypoint-conditioned policy directs the robot to execute the manipulation tasks, considering both spatial and temporal information from RoVI.

Except for the framework, we develop a dataset of 15K training instances to enable models to learn RoVI capabilities through Parameter-Efficient Fine-Tuning (PEFT) [20, 24]. Through the design outlined above, our approach performs well across diverse unseen tasks in both real-world and simulated environments, showing strong generalization and robustness. Compared to language-conditioned policies, our method achieves superior performance in cluttered settings, multi-step operations, and trajectory-following tasks (see the right part of Figure 1).

1. We propose a novel human-robot interaction paradigm: RoVI. It employs hand-drawn symbolic representations as robotic instructions, conveying more precise spatial-temporal information within task definition.
2. We design a pipeline, VIEW (Visual Instruction Embodied Workflow), to enable RoVI-conditioned manipulation tasks.
3. We develop an open-source dataset to enable models to learn RoVI capabilities. The lightweight model trained by this dataset demonstrates that VLMs are able to learn this capability with minimal computational resources and simple fine-tuning.

2. Related Work

Human Robot Interaction. Recent advancements in VLMs have made them a popular choice for language-

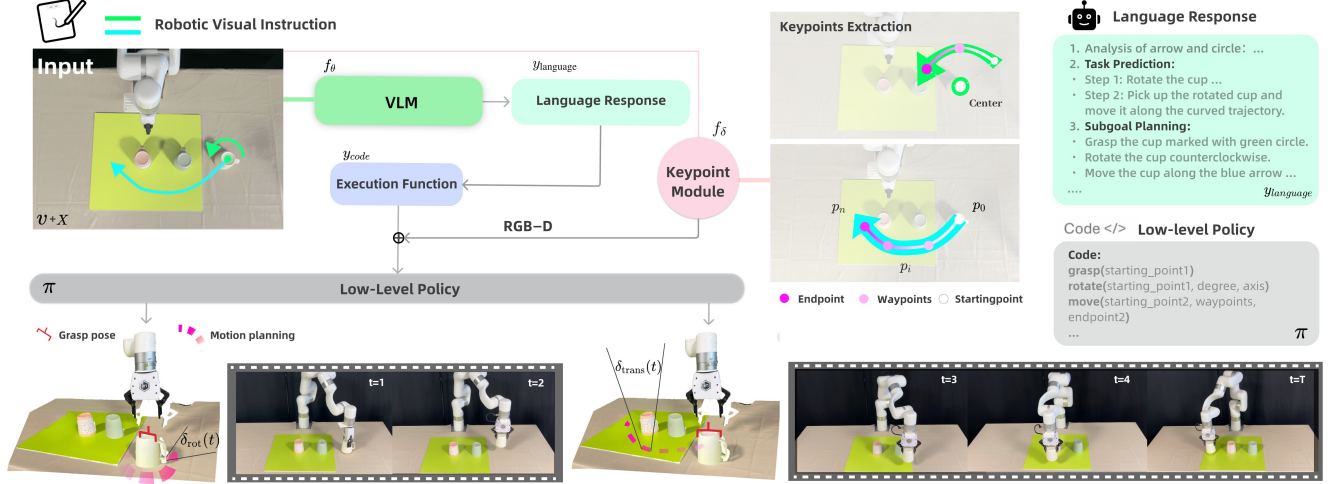


Figure 3. VIEW Architecture. This pipeline begins with a visual instruction drawn onto the initial observation. The VLM generates hierarchical language-to-action outputs, including task definition, detailed planning, and executable functions. The executable functions are then combined with keypoints extracted from the keypoint module and passed to a downstream low-level policy, which enables the robotic arm to execute each action step-by-step. This approach bridges hand-drawn visual instructions with precise robotic actions.

conditioned policies [3, 8, 10, 22, 27, 33]. Image-conditioned policies are also widely explored, such as goal-image policy [14, 32, 45], multimodal prompts [29], and trajectory-based inputs [21, 48, 51]. One common approach is goal-image conditioning, where a final goal image specifies the desired task’s end state. Trajectory-based policy utilizes the full 2D or 3D trajectory of the end-effector as input. However, these input methods present significant challenges for users, as it is often difficult for users to provide such inputs directly in real-world applications.

Visual Prompting for Robot. Recent studies have explored the use of visual prompts as user input for tasks like Visual Question Answering (VQA) [1, 11, 50]. These models use symbolic forms of language, such as arrows, sketches, and numbers, to assist natural language in providing more accurate VQA. However, these approaches have primarily focused on generalized image-based question-answering tasks, and the domain of visual prompts in robotic manipulation tasks remains largely unexplored. In the context of robot control, some methods use model-generated visual prompts to guide trajectory selection for manipulation [26, 34, 36]. Yet, they still rely on natural language as input. These methods do not resolve the issue with natural language instructions—specifically, the lack of spatial intent in task definitions provided by users.

Keypoint Constraints for Manipulation. Recent studies [15, 19, 28, 31, 44] have achieved significant advancements in manipulation by leveraging key points to formulate spatiotemporal constraints. However, unlike prior approaches that extract keypoints from environmental objects and then filter them through VLMs reasoning [28], our method directly extracts key points from RoVI symbols (ar-

rows and circles).

3. Robotic Visual Instruction Design

We present the paradigm design of RoVI, which consists of two visual primitives: an *arrow* and a *circle*. Tasks are decomposed into three object-centric motions: moving from A to B (represented by an arrow), rotating an object (a circle indicating affordance with an arrow for rotation degree), and picking up/selecting (represented by a circle).

Dissecting Arrow. We use 2D *arrows* to denote the trajectory and temporal sequence of robotic actions. An arrow is decomposed into three components: *Tail* (Starting Point p_0), *Shaft* (Waypoints $\{p_1, \dots, p_{n-1}\}$), and *Head* (Endpoint p_n). The starting point p_0 marks the grasp position on the object, and the endpoint p_n denotes the action’s goal. Intermediate waypoints capture the movement path, forming an ordered set:

$$\text{Arrow} = \{p_0, p_1, \dots, p_n\}, \quad p_i \in \mathbb{R}^2, \quad (1)$$

where p_i are 2D coordinates extracted by a keypoint module.

Dissecting Circle. The *circle* highlights key interaction areas on objects. The center point $p_0 \in \mathbb{R}^2$ represents the affordance center and is used for various tasks: as a grasping point, a pivot for rotation, or a pressure point for actions like pressing buttons.

Drawing Setting. RoVI is drawn directly using a tablet with a bright green color to ensure visibility across backgrounds, aiding both human and VLM recognition. We designed two drawing styles: *Loose Style* (casual, hand-drawn) and *Geometric Style* (structured with geometric

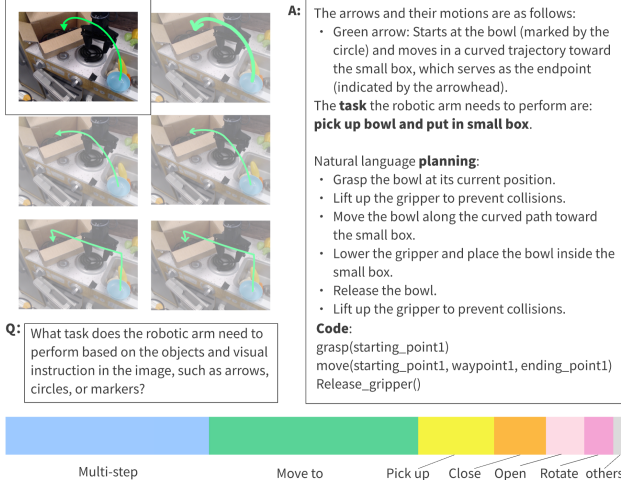


Figure 4. This is an example to demonstrate the RoVI Book dataset, adapted from the Open-X Embodiments dataset [13]. The bottom displays the proportion of each task type.

components for clearer interpretation by VLMs). We use a circle to signify affordances and replace the arrowhead with a standard triangle as depicted in Figure 10. A comparison of their effectiveness is in Section 6.4.

4. RoVI Book dataset

To enable VLMs to understand RoVI, we develop a dataset for RoVI-conditioned policy, termed **RoVI Book**. The dataset shown in Figure 7 comprises 15K image-text question-answer pairs. It includes (1) images of initial task observations annotated with RoVI, (2) simple queries serving as default prompts, and (3) answers generated by GPT-4o [1], covering RoVI analysis, task names, fine-grained planning steps, and Python functions. The original tasks and images were selected from the Open-X Embodiment dataset [13]. Our dataset covers 64% single-step tasks and 36% multi-step tasks, across five fundamental manipulation skills: move an object, rotate an object, pick up, open drawers/cabinets, and close drawers/cabinets. The visual instructions were primarily created using a stylus and drawing software on a tablet. The answers are initially generated using GPT-4o [1] and subsequently refined through semantic filtering based on human feedback. Each task retains its original semantic task name from the Open-X Embodiments [13], while we apply data augmentation to RoVI, introducing 3–8 visual variants, varying paths, drawing styles, and line thickness. Further details are provided in the appendix.

5. Visual Instruction Embodied Workflow

5.1. Overview of Workflow

The VIEW consists of three components: (1) A VLM f_θ for RoVI understanding and planning, (2) a keypoint module f_δ for generating spatiotemporal constraints [28], and (3) a low-level policy π for executing robot actions.

As shown in Figure 3, the pipeline begins with VLMs that take as input the hand-drawn RoVI $v \in \mathbb{R}^{H \times W \times 3}$, an initial observation image $X \in \mathbb{R}^{H \times W \times 3}$, and a system-provided default prompt (further details on the default prompt can be found in the appendix). The VLMs then produce language response y_{language} and the execution function y_{code} . Simultaneously, the keypoint module extracts keypoints from the RoVI to generate spatiotemporal constraints, including a starting point p_0 , multiple waypoints p_i , and an endpoint p_n . Finally, based on the input y_{code} and the keypoint coordinates, the low-level policy executes the corresponding actions.

5.2. VLMs for RoVI Understanding

Given the VLMs’ capabilities in visual perception, embedded world knowledge, and reasoning, we use the VLMs to interpret RoVI and translate it into a natural language response y_{language} . The language response acts as a universal interface for human feedback, enabling verification of VLMs’ comprehension and connecting it to downstream low-level policies. Compared with the end-to-end policies [7, 9] directly output parameters in SE(3) action space, y_{language} incorporates language-base action representations, which generalize more effectively across variable tasks and environments [5, 17, 27].

The language response is generated by VLMs with a Chain-of-Thought (CoT) reasoning process. It includes coarse-grained task predictions, providing high-level task descriptions, and fine-grained planning with sub-goal sequences, breaking tasks into smaller steps. Each sub-goal is subsequently converted into executable code functions y_{code} , which define the necessary actions or skills for the robotic arm, such as `move()` or `grasp()`. These functions, combined with keypoint constraints, form a low-level policy for action implementation. A comprehensive example of the model output is provided in the appendix.

$$y_{\text{language}}, y_{\text{code}} = f_\theta(v, X). \quad (2)$$

5.3. Keypoint Module

To decode spatiotemporal information from RoVI, $v \in \mathbb{R}^2$ in pixel space, we first decompose high-level tasks into single-step tasks based on color identifiers. Then, a trained keypoint module, f_δ , provides keypoint constraints, which include sequences of end-effector coordinates and keypoints’ semantic functionalities in manipulation such as

starting points $p_0 \in \mathbb{R}^2$, waypoints $p_i \in \mathbb{R}^2$, and endpoints $p_n \in \mathbb{R}^2$.

We employ YOLOv8 [30] as δ_δ and construct a dataset containing 2k images for its training (see details in the appendix). Compared to open-vocabulary object detection, our strategy simplifies the detection of all objects across different environments to identify components of the RoVI symbols, making it less susceptible to environmental variations or distractor objects (see Experiment Section 6.4).

5.4. Keypoint-Conditioned Low-Level Policy

We propose a keypoint-conditioned low-level policy that enables a robot to follow a sequence of target poses, defined as keypoints, for manipulation tasks. These keypoints $p_i \in \mathbb{R}^2$ are extracted from action arrows in an RGB image and mapped to 3D coordinates $p'_i \in \mathbb{R}^3$ using depth data from a RGB-D camera.

These N keypoints are then mapped to a sequence of desired end-effector poses in SE(3) space, which is represented as $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N\}$. The initial pose \mathbf{e}_0 is obtained using the grasp module [18] based on $p_0 \in \mathbb{R}^2$. The series of poses form the action to be executed. We categorize actions into two types: *translation* (e.g., move to, push, pull) and *rotation* (e.g., flip, knock-down, adjust knob). For translational tasks, we assume linear motion without rotation, ensuring stable and simplified planning for rigid object transport. At each time step t , the robot performs:

1. **State Observation:** Acquire the current end-effector pose $\mathbf{e}_t \in \text{SE}(3)$ and target keypoint $p'_i \in \mathbb{R}^3$ from the RGB-D camera.
2. **Cost Function Minimization:** $\mathcal{L}_i(t)$: Minimize the cost function by moving towards p'_i leveraging motion planning and interpolation.
3. **Keypoint Transition:** If $\mathcal{L}_i(t) \leq \epsilon$, mark p'_i as reached and proceed to p'_{i+1} . i accumulates until $i = N$, then end the current action step.

The goal at each time step t is to minimize $\mathcal{L}_i(t)$:

$$\arg \min \mathcal{L}_i(t), \quad (3)$$

$$\mathcal{L}_i(t) = \alpha_i \delta_{\text{trans}}(t) + (1 - \alpha_i) \delta_{\text{rot}}(t), \quad (4)$$

where α_i indicates the action type: $\alpha_i = 1$ for translation and $\alpha_i = 0$ for rotation.

Translational Cost: $\delta_{\text{trans}}(t) = \|\mathbf{e}_t - \mathbf{e}_i\|$, where \mathbf{e}_t is the current end-effector pose and \mathbf{e}_i is the target pose, with $\|\cdot\|$ denoting the Euclidean norm.

Rotational Cost: $\delta_{\text{rot}}(t) = |\theta_t - \theta_i|$, where θ_t is the current rotation angle, θ_i is computed as:

$$\theta_i = \arccos \left(\frac{(\mathbf{v}_i)^\top \mathbf{v}_{i+1}}{\|\mathbf{v}_i\| \|\mathbf{v}_{i+1}\|} \right), \quad (5)$$

with $\mathbf{v}_i = p'_i - c$ and $\mathbf{v}_{i+1} = p'_{i+1} - c$, where c is the rotation center.

6. Experiment

Our experiments aim to conduct in-depth research on the following questions:

1. How does RoVI perform in generalizing over unseen environments and tasks in the real world and simulation? (section 6.1 and 6.2)
2. How well do current VLMs understand RoVI? (section 6.3)
3. How do the components of RoVI and VIEW impact the overall performance of the whole pipeline? (section 6.4)

Model Training. We select GPT-4o [1] and LLaVA-13B [37] as the VLMs in VIEW to control the robotic manipulation tasks. We also fine-tune the LLaVA-7B and 13B models [37] using the LoRA [25] on our RoVI Book dataset, with one training epoch and a learning rate of $2e-4$. All experiments are conducted on an NVIDIA A40 GPU.

Drawing Setting. In our experiments, RoVI is generated directly using drawing software on tablets or PCs, with **green** (RGB: 0, 255, 94) marking the first step of the manipulation task, **blue** (RGB: 0, 255, 247) marking the second step, and **pink** (RGB: 255, 106, 138) marking the third step. For more steps, extra color can be assigned flexibly.

Implement Procedure. We train a YOLOv8 model [30] to extract starting points, waypoints, and endpoints from hand-drawn instructions, providing keypoint constraints. These constraints are used to filter the grasp poses generated by AnyGrasp [18] to obtain the closest one. The obtained 3D coordinates from RGB-D mapping and grasp poses are then input into VLM-generated Python functions for code-based low-level control.

Manipulation Tasks. We meticulously design 11 tasks: 8 in real environments and 3 in simulated settings shown in Figure 5 and 6. For our method, all tasks and environments are previously unseen, with new objects introduced. Our design includes 7 single-step tasks. Some involve cluttered environments with disturbances, such as ‘select a desired object’ or ‘move between objects’, requiring precise spatial alignment and trajectory-following abilities. Additionally, there are 4 multi-stage (Task 6-8 in real environment, Task 3 in simulation) tasks to test further reasoning ability for spatio-temporal dependency.

6.1. Generalization to In-the-Wild Manipulation

Real World Setting & Baselines. For real-world experiments, we use two robotic arms with two-finger grippers: UFACTORY X-Arm 6 and UR5. Two calibrated RealSense D435 cameras are positioned for top-down and third-person views. Both robotic arms operated at a 20 Hz control frequency with an end-effector delta control mode.

We compare our approach against two language-conditioned policy baselines, CoPa [26] and VoxPoser [27], both leveraging a GPT model for low-level policy control.

Robotic Baseline	Real World								Average	Robotic Baseline	Simulator			Average
	1	2	3	4	5	6	7	8			1	2	3	
Voxposer [27]	30	80	80	10	0	30	30	20	43.8	RT-1-X [7]	40	20	0	20
CoPa [26]	40	90	80	60	0	40	20	30	45	Octo-goal-image [43]	10	30	0	13.3
VIEW-GPT4o [1]	80	100	90	90	60	70	90	80	82.5	Octo-language [43]	10	0	0	3
VIEW-LLaVA-13B [37] (RoVI Book)	90	90	100	100	70	70	90	90	87.5	VIEW*	70	60	100	76.6

Table 1. Success Rate in unseen environments and unseen tasks. The numbers correspond to tasks in Figure 5 and Figure 6. VIEW* denotes both VIEW-GPT4o and VIEW-LLaVA 13B (RoVI Book), as their test results are identical. **Bold** score means the best result.

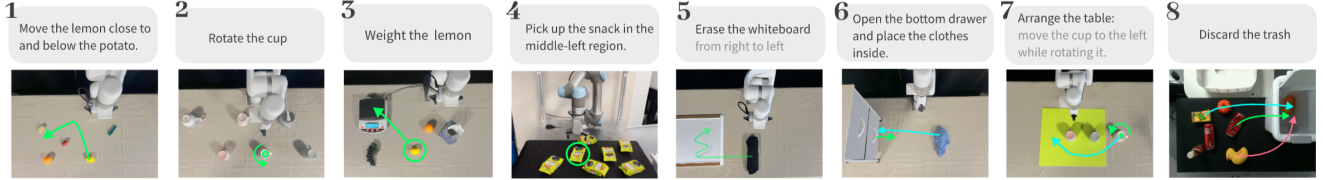


Figure 5. Robotic visual instruction is capable of generalizing to a variety of in-the-wild real-world situations, including multi-stage tasks that require precise spatial coordination, reasoning, and spatio-temporal dependencies, even in cluttered environments with disturbances.

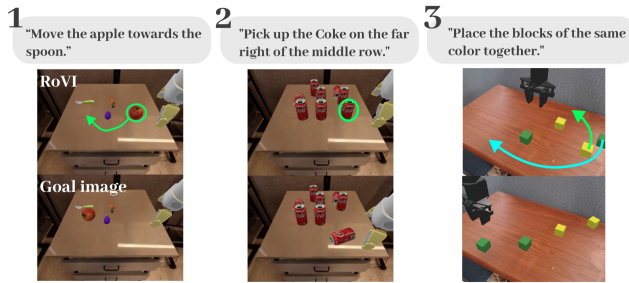


Figure 6. Experiment tasks in the SIMPLER [35] environment for the comparative study of language instruction, goal image, and visual instruction.

CoPa [26] additionally utilizes Set-of-Mark (SoM) [49] for object tagging as a visual prompt. To ensure a fair comparison, all methods used GPT-4o [1] as the VLM.

Evaluation Metrics for Action. We report two metrics for assessing manipulation execution: *action success rate*, measuring the percentage of tasks that meet defined goals, and *spatiotemporal alignment*, evaluating the consistency of movement trajectories and the alignment of an object’s final spatial state with semantic goals. A 6-point Likert scale is used for assessment (details in the appendix). Each task is evaluated over 10 trials.

Results. Table 1 shows that Voxposer [27] and CoPa [26] struggle with spatial precision tasks, such as ‘move the lemon close to and below the potato’ and the ‘choose a snack’ task with similar object disturbances. Both of these two methods also failed in Task 5, indicating the difficulty of the trajectory following. This is due to the inherent ambiguity of language-based instructions, which provide only object-level information, whereas RoVI en-

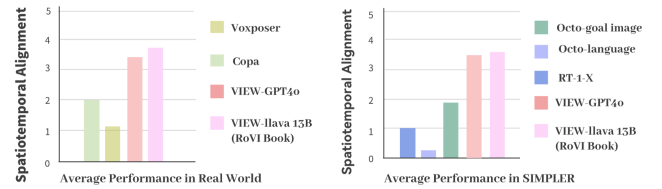


Figure 7. Performance comparison of average spatiotemporal alignment across all methods. See supplementary materials for detailed statistics.

ables pixel-level precision. In contrast, VIEW performs well on these tasks, as its keypoint module provides spatial constraints and waypoints. Unlike VoxPoser [27] and CoPa [26], which use an open-vocabulary object detector, VIEW’s keypoint module focuses on RoVI symbol parts, making it less susceptible to environmental variation or distractors. This enables VIEW’s strong generalization and robustness in real-world manipulation tasks. Compared to other approaches that employ VLMs for temporal sequence reasoning in embodied planning, our method also achieves superior performance on long-horizon tasks (Task 6-8). By decomposing multi-step tasks into individual steps guided by color cues, we effectively reduce the complexity of temporal reasoning.

6.2. Comparative Study in Simulation

Simulation Setting & Baselines. This section compares the manipulation performance of three instruction methods—language instruction, goal-image, and RoVI—in a simulated environment. We use SAPIEN [47] as the simulator and SIMPLER [35] as the base environment.

For the simulated experiments, we evaluate our approach

VLMs w/ RoVI	Real World										Simulator			
	1	2	3	4	5	6	7	8	Average		1	2	3	Average
Small Models	0	0	0	0	0	0	0	0	0		0	0	0	0
Claude 3.5-Sonnet [4]	100	95	0	100	90	55	50	67	70		30	100	50	60
Gemini-1.5 Pro [2]	10	100	100	100	20	95	60	57	68		0	100	0	33
GPT-4o [1]	100	100	100	40	60	90	100	55	81		100	100	90	97
LLaVA-13B [37] (RoVI Book)	9	45	0	82	0	75	14	82	38		36	82	73	64

Table 2. Task and Planning evaluation in language response. It showcases the capability of existing VLMs to comprehend RoVI. The numbers correspond to tasks in Figure 5 and Figure 6.

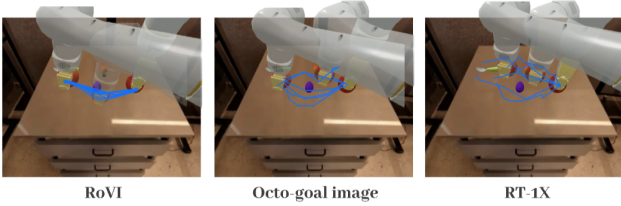


Figure 8. Visual comparison of trajectory between RoVI, natural language, and goal image policies. For each example, we sample six successful action trajectories from 50 trials and find that only RoVI’s end state and path are more convergent and controllable.

against RT-1-X [7] and Octo [43], both of which are end-to-end, language-conditioned Vision-Language-Action (VLA) models trained on the Open X-Embodiment dataset [13]. Octo [43] additionally supports goal-image input modalities. In our setup, we use the same robotic arms and background settings as in their training set and include new tasks in cluttered environments to test generalization.

Quantitative Analysis. These three tasks are performed in cluttered environments, where both language and goal-image inputs face significant challenges. Long-horizon tasks, in particular, are nearly impossible to accomplish under such conditions. However, our approach performs exceptionally well. These results indicate that end-to-end vision-language-action (VLA) models struggle with generalization to new tasks, while our method demonstrates robust generalization, with performance in simulation closely aligning with real-world outcomes.

Qualitative Study. To study the potential capability of RoVI, we delve into further qualitative comparison with nature language and goal-image conditioned policies. As shown in Figure 8, RoVI is the only instruction format that effectively conveys both path information and the end state. In contrast, the goal image policy performs well in terms of the end state but falls short in describing movement paths. For methods like RT-X [7] and Octo [43], the generated paths and end states lack consistency and exhibit limited spatial precision. In the evaluated examples, RoVI demonstrates a clear advantage in spatiotemporal alignment.

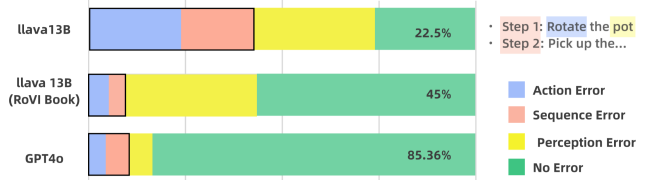


Figure 9. Error breakdown of language responses. Training with the RoVI book significantly reduces errors in action decisions and temporal sequences (highlighted in the black box).

6.3. RoVI Comprehension by Modern VLMs.

We evaluate the capability of VLMs to extract semantic meaning from RoVI in novel tasks and environments, employing in-context learning and a zero-shot approach (see details in supplementary in-context learning).

Metrics. We evaluate ‘*Task and Planning*’ success rates by assessing the accuracy of language responses using human feedback. This evaluation has two components: ‘*task*’, measuring the VLMs’ comprehension of task definitions based on RoVI and observations (e.g. ‘Open the bottom drawer, then place the clothes inside’); and ‘*planning*’, evaluating the reasoning capability of VLMs to decompose complex RoVI tasks into sequential sub-goals. Each task is evaluated over 10 trials. We compare our trained model with diverse VLMs, including large-scale models: GPT-4o [1], Gemini-1.5 Pro [2], Claude 3.5-Sonnet [4], as well as smaller models: InternLM-XComposer2-VL-7B [16], LLaVA-HF/LLaVA-v1.6-Mistral-7B [37], MiniGPT-4 [52], and VIP-LLaVA 7B [11].

Results. The Table 2 demonstrates that advanced large models (Gemini [2], GPT-4o [1], Claude [4]) exhibit a strong ability to understand RoVI-conditioned manipulation tasks through in-context learning, even without being trained on expert datasets. In contrast, models with fewer than 13 billion parameters fail to comprehend RoVI effectively. Combining both simulation and real-world performance, GPT-4o [1] exhibits the best overall results. Furthermore, advanced large models generalize better in terms of RoVI comprehension compared to smaller models trained

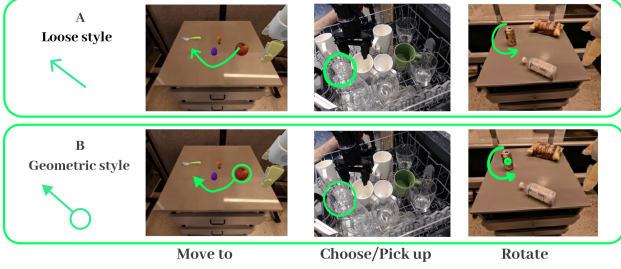


Figure 10. Showcase of two drawing styles in modified Open X-Embodiment dataset [13].

Model	Task Prediction + Subgoal Planning							
	Move		Pick up / Choose		Rotate		Average	
	L	G	L	G	L	G	L	G
GPT-4o [1]	0.6	1.0	0.9	0.9	0.2	1.0	0.57	0.97
Gemini 1.5 pro [2]	0.1	0.0	1.0	1.0	1.0	0.6	0.7	0.53
Claude 3.5 sonnet [4]	1.0	0.8	1.0	1.0	0.9	0.9	0.97	0.9
Total Average	0.57	0.6	0.97	0.97	0.7	0.83	0.74	0.8

Table 3. Comparison of drawing styles in modified Open X-Embodiment. ‘L’ and ‘G’ denote Loose style and Geometric style respectively. On average, the more structured geometric style offers VLMs better task comprehension ability.

on the RoVI Book dataset, such as LLaVA-13B [37]. However, as the number of steps in the task increases, the large models’ comprehension accuracy decreases. In contrast, LLaVA-13B [37], trained on the RoVI Book dataset, performs well on long-sequence task 8, indicating that the RoVI Book dataset is effective for learning multi-step tasks under RoVI conditions.

Error Breakdown. It is worth noting that LLaVA-13B [37] (trained on the RoVI Book) shows a low success rate in task and planning predictions but performs exceptionally well in action execution. In conjunction with Figure 9, we can conclude that the execution function maps action and sequence errors, making it unaffected by perception errors. After training on the RoVI Book, errors related to the execution function were significantly reduced.

6.4. Ablations Study

Drawing. Analogous to how language prompts often require ‘prompt engineering’, free-form drawing can exhibit significant variability. And hand-drawn instruction raises another question: how can we optimize the drawing style to enhance model comprehension? In this section, we classify the drawing styles into two distinct categories for comparison to investigate their impact on VLMs’ reasoning performance. The corresponding visualization and experiments are shown in Figure 10 and Table 3. Our findings indicate that the more structured geometric style yielded superior comprehension. Further experimental details are attached in the supplementary material.

Task	Metric	GDINO [38]	OWL-ViT [40]	OWL-V2 [41]	YOLOv8 [30]
1	MD	482.71 \pm 0.00	N/A	114.18 \pm 92.65	6.83 \pm 0.00
	mAP	0.00 \pm 0.00	N/A	0.33 \pm 0.47	1.00 \pm 0.00
2	MD	507.35 \pm 183.27	N/A	52.47 \pm 61.44	19.45 \pm 8.92
	mAP	0.00 \pm 0.00	N/A	0.57 \pm 0.49	1.00 \pm 0.00
3	MD	510.33 \pm 183.25	153.92 \pm 0.00	131.03 \pm 33.43	13.27 \pm 5.81
	mAP	0.00 \pm 0.00	0.00 \pm 0.00	0.00 \pm 0.00	1.00 \pm 0.00
4	MD	751.44 \pm 196.85	57.51 \pm 89.85	63.28 \pm 80.47	11.64 \pm 4.73
	mAP	0.00 \pm 0.00	0.67 \pm 0.47	0.64 \pm 0.48	1.00 \pm 0.00

Table 4. Ablation of the proposed keypoint module. The tested tasks and RoVI are shown in the supplementary material. MD represents mean distance.

Keypoint Module. We evaluate the proposed keypoint module, a trained YOLOv8 model [30], for spatial constraint generation across four different RoVI tasks. We compare it against three popular open-vocabulary detection models [38, 40, 41], using two strategies: (1) manually inputting the target’s semantic information as the text prompt, and (2) identifying and localizing arrow components (arrowhead and tail). Two primary metrics are used for evaluation: Euclidean distance error (measured in pixels) to assess precision, and Mean Average Precision (mAP) at a 50-pixel threshold to measure accuracy. Results in Table 4 indicate that, despite its smaller parameter size, the keypoint module achieves more efficient task-relevant keypoint extraction directly from pixel space compared to transformer-based open-vocabulary detection models. Additional limitations and details can be found in the supplementary material.

7. Conclusion and Future works

In this paper, we propose **Robotic Visual Instruction (RoVI)**, a user-friendly and spatially precise alternative to natural language for guiding robotic tasks. To implement RoVI, we develop a pipeline, **Visual Instruction Embodied Workflow (VIEW)**, which demonstrates strong generalization and robustness across cluttered environments and long-horizon tasks. Additionally, we meticulously create a dataset to fine-tune VLMs for a better understanding of RoVI and potential future edge device deployment. Ablation studies also reveal the factors influencing the performance of RoVI-conditioned policies, including RoVI comprehension, drawing strategies, and grounding methods.

Future works. Future research will focus on scaling up the RoVI Book dataset and collecting a wider variety of free-form drawn instruction. This expansion aims to equip the model with a broader understanding of the general principles by which humans employ visual symbols to convey dynamic movements. On the other hand, by scaling up the dataset, we can more efficiently train a smaller model like 7B, further enhancing its ability to interpret RoVI instructions and output code. This will facilitate the deployment of edge devices within our robotic system.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3, 4, 5, 6, 7, 8
- [2] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 1, 2023. 7, 8
- [3] Anonymous. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *Under Review*, 2023. 3
- [4] AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024. 7, 8
- [5] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidda Dwivedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. In <https://arxiv.org/abs/2403.01823>, 2024. 4
- [6] Andrea Bonarini. Communication in human-robot interaction. *Current Robotics Reports*, 1(4):279–285, 2020. 2
- [7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022. 2, 4, 6, 7
- [8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 3
- [9] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023. 2, 4
- [10] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 3
- [11] Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee. Vip-llava: Making large multimodal models understand arbitrary visual prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12914–12923, 2024. 3, 7
- [12] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024. 2
- [13] Open X-Embodiment Collaboration, Abby O’Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frueger, Freck Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyu Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-

- Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minh Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Sunderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Panag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundareshan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Bajjal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Hal-dar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Mat-sushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yan-song Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. <https://arxiv.org/abs/2310.08864>, 2023. 4, 7, 8
- [14] Michael Danielczuk, Andrey Kurenkov, Ashwin Balakrishna, Matthew Matl, David Wang, Roberto Martin-Martin, Animesh Garg, Silvio Savarese, and Ken Goldberg. Mechanical search: Multi-step retrieval of a target object occluded by clutter. In *2019 International Conference on Robotics and Automation (ICRA)*, page 1614–1621. IEEE, 2019. 2, 3
- [15] Norman Di Palo and Edward Johns. Keypoint action tokens enable in-context imitation learning in robotics. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024. 3
- [16] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Ji-qi Wang. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*, 2024. 7
- [17] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model. In *arXiv preprint arXiv:2303.03378*, 2023. 4
- [18] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023. 5
- [19] Jianfeng Gao, Zhi Tao, Noémie Jaquier, and Tamim Asfour. K-vil: Keypoints-based visual imitation learning. *IEEE Transactions on Robotics*, 39(5):3888–3908, 2023. 3
- [20] Ziyang Gong, Fuhao Li, Yupeng Deng, Deblina Bhattacharjee, Xianzheng Ma, Xiangwei Zhu, and Zhenming Ji. Coda: Instructive chain-of-domain adaptation with severity-aware visual prompt tuning. In *European Conference on Computer Vision*, pages 130–148. Springer, 2024. 2
- [21] Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. *arXiv preprint arXiv:2311.01977*, 2023. 2, 3
- [22] Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Proceedings of the 2023 Conference on Robot Learning*, 2023. 3
- [23] Kaveh Hassani and Won-Sook Lee. Visualizing natural language descriptions: A survey. *ACM Computing Surveys (CSUR)*, 49(1):1–34, 2016. 2
- [24] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019. 2
- [25] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 5
- [26] Haoxu Huang, Fanqi Lin, Yingdong Hu, Shengjie Wang, and Yang Gao. Copa: General robotic manipulation through spatial constraints of parts with foundation models. *arXiv preprint arXiv:2403.08248*, 2024. 3, 5, 6
- [27] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 3, 4, 5, 6
- [28] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024. 3, 4
- [29] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manip-

- ulation with multimodal prompts. In *Fortieth International Conference on Machine Learning*, 2023. 3
- [30] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO, 2023. 5, 8
- [31] Ananth Jonnavittula, Sagar Parekh, and Dylan P. Losey. View: Visual imitation learning with waypoints. *arXiv preprint arXiv:2404.17906*, 2024. 3
- [32] Xuhui Kang and Yen-Ling Kuo. Incorporating task progress knowledge for subgoal generation in robotic manipulation through image edits, 2024. 2, 3
- [33] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024. 2, 3
- [34] Olivia Y Lee, Annie Xie, Kuan Fang, Karl Pertsch, and Chelsea Finn. Affordance-guided reinforcement learning via visual prompting. *arXiv preprint arXiv:2407.10341*, 2024. 3
- [35] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024. 6
- [36] Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-vocabulary robotic manipulation through mark-based visual prompting. *arXiv preprint arXiv:2403.03174*, 2024. 3
- [37] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 5, 6, 7, 8
- [38] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 8
- [39] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time, 2022. 2
- [40] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pages 728–755. Springer, 2022. 8
- [41] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36, 2024. 8
- [42] Fei Ni, Jianye Hao, Shiguang Wu, Longxin Kou, Jiashun Liu, Yan Zheng, Wang Xian Bin, and Yuzheng Zhuang. Generate subgoal images before act: Unlocking the chain-of-thought reasoning in diffusion model for robot manipulation with multimodal prompts. pages 13991–14000, 2024. 2
- [43] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024. 6, 7
- [44] Lucy Xiaoyang Shi, Archit Sharma, Tony Z. Zhao, and Chelsea Finn. Waypoint-based imitation learning for robotic manipulation, 2023. 3
- [45] Priya Sundareshan, Quan Vuong, Jiayuan Gu, Peng Xu, Ted Xiao, Sean Kirmani, Tianhe Yu, Michael Stark, Ajinkya Jain, Karol Hausman, et al. Rt-sketch: Goal-conditioned imitation learning from hand-drawn sketches. 2024. 2, 3
- [46] Jiaqi Wang, Zihao Wu, Yiwei Li, Hanqi Jiang, Peng Shu, Enze Shi, Huawen Hu, Chong Ma, Yiheng Liu, Xuhui Wang, et al. Large language models for robotics: Opportunities, challenges, and perspectives. *arXiv preprint arXiv:2401.04334*, 2024. 2
- [47] 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. 6
- [48] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. In *8th Annual Conference on Robot Learning*, 2024. 2, 3
- [49] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v, 2023. 6
- [50] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of Imms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1, 2023. 3
- [51] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024. 2, 3
- [52] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 7