RoBridge: A Hierarchical Architecture Bridging Cognition and Execution for General Robotic Manipulation

Kaidong Zhang[1] Rongtao Xu[2,5] Pengzhen Ren[3] Junfan Lin[3]
Hefeng Wu[1] Liang Lin[1,3,4†] Xiaodan Liang[1,2,3†]
[1]Sun Yat-sen University [2]MBZUAI [3]Peng Cheng Laboratory [4]X-Era.AI Inc. [5]Spatialtemporal AI

Abstract

Operating robots in open-ended scenarios with diverse tasks is a crucial research and application direction in robotics. While recent progress in natural language processing and large multimodal models has enhanced robots' ability to understand complex instructions, robot manipulation still faces the procedural skill dilemma and the declarative skill dilemma in open environments. Existing methods often compromise cognitive and executive capabilities. To address these challenges, in this paper, we propose RoBridge, a hierarchical intelligent architecture for general robotic manipulation. It consists of a high-level cognitive planner (HCP) based on a large-scale pre-trained vision-language model (VLM), an invariant operable representation (IOR) serving as a symbolic bridge, and a guided embodied agent (GEA). RoBridge maintains the declarative skill of VLM and unleashes the procedural skill of reinforcement learning, effectively bridging the gap between cognition and execution. RoBridge demonstrates significant performance improvements over existing baselines, achieving a 75% success rate on new tasks and an 83% average success rate in sim-to-real generalization using only five real-world data samples per task. This work represents a significant step towards integrating cognitive reasoning with physical execution in robotic systems, offering a new paradigm for general robotic manipulation.

1. Introduction

Operating robots in open-ended scenarios with diverse tasks is a significant research and application direction in the field of robotics. In such scenarios, robots are required to understand natural language instructions and accurately execute complex tasks while adapting to dynamic environmental changes and uncertainties [15, 23]. In recent years, the rapid development of natural language processing and large multimodal models [3, 37, 47] has led to substantial progress in robots' ability to understand and execute complex instructions. However, robot manipulation still faces two major challenges when dealing with complex instructions in open environments: the procedural skill dilemma and the declarative skill dilemma.

- **Procedural Skill Dilemma.** To acquire the ability to manipulate objects according to instructions, embodied models[5, 6, 33, 36] like RDT [33] and π_0 [5] typically employ data-driven trajectory fitting methods. However, these methods frequently suffer catastrophic performance degradation when confronted with environmental variations, including fluctuating illumination conditions, camera pose deviations, and contextual changes [42]. Reinforcement learning, though robust, has a trial-and-error nature and low learning efficiency, making it less applicable in real-world environments.
- **Declarative Skill Dilemma.** Recent endeavors to integrate vision-language models (VLM) into robotic systems [9, 24, 25, 32, 40] like ReKep [25] and OmniManip [40] use large multimodal models to generate operational instructions for open-domain tasks. While these models excel in understanding, they lack embodied experience and require constraining outputs to executable actions. This approach forces the language model to handle spatial-temporal reasoning without physical intuition, often resulting in implausible task planning. For instance, in the task "place block A atop block B", an insufficient spatial understanding (e.g., shape, height) often leads this kind of method to produce fatally flawed action sequences.

Unfortunately, in facing these dilemmas, most existing methods [7, 28] choose to restrict the capabilities of large multimodal models, forcing them to generate low-level execution commands to drive data-driven downstream control strategies to complete tasks. This enables a certain degree of task execution in open scenarios but essentially represents a compromise in cognitive and executive capabilities. Conversely, suppose we could leverage the strengths of large language/multimodal models in declarative tasks and the advantages of reinforcement learning in procedural tasks without mutual constraints. In that case, robots might

†Corresponding authors

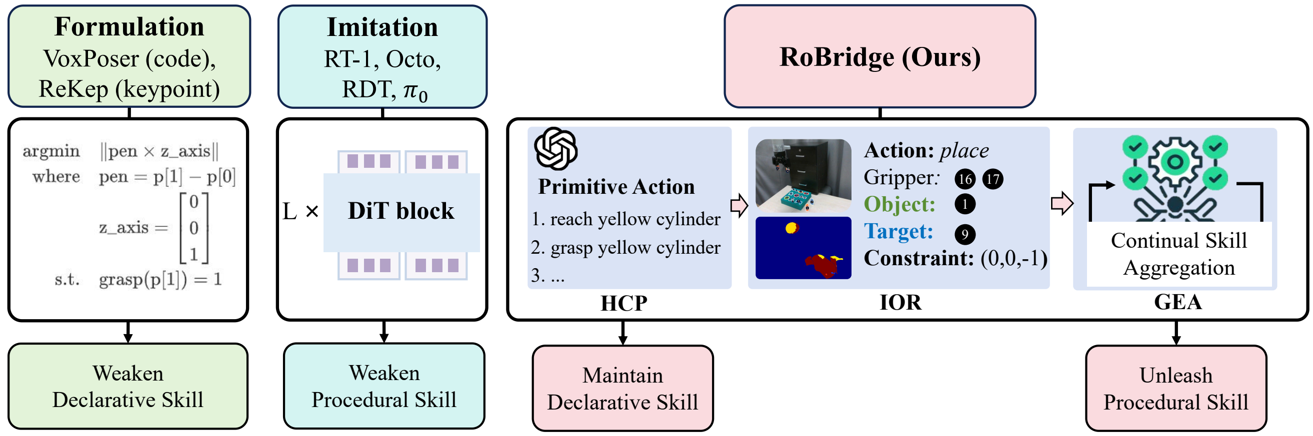

Figure 1. **Comparison of RoBridge and previous methods**. Declarative skill methods (left) directly generate specific control commands in a formulaic way, such as determining trajectories by minimizing cost. However, due to a lack of interaction experience with the physical world, the generated commands are often incorrect. Procedural skill methods (middle) forcibly transform a vision-language model (VLM) into a robotics model using a data-driven approach, but it is not effective in dealing with unseen situations. Our method, RoBridge(right), enables the VLM to generate physically intuitive representations as a symbolic bridge. This symbolic bridge is characterized by its invariance, allowing it to communicate with the underlying embodied agent in a universal manner. Meanwhile, the embodied agent continuously interacts with the physical world to gain continual skill aggregation, fully leveraging the strengths of both the VLM and reinforcement learning.

achieve significant breakthroughs in handling open-ended tasks. Therefore, the problem we would like to answer is: how to construct a novel framework that allows large multimodal models and reinforcement learning to utilize their respective strengths while supporting each other fully, ultimately maximizing the capability in open-ended scenarios.

In the theory of Central Pattern Generators (CPGs) in the cognitive domain [20, 35, 56], scientists have proposed that during the process of human beings transitioning from high-level reasoning to low-level control, the brain sends messages, such as walking, specific neurons will always be activated according to specific abstract concepts, regardless of the specific manifestations of the environment, such as appearance, color, size, etc. The activation of these specific neurons regulates the activity level of the motor regions [19]. Eventually, precise action behaviors emerge according to the manifestations of the environment, like determining how large a step to take with the left foot or the right foot. This theory provides an inspiring perspective, suggesting that the communication between high-level reasoning and low-level control might go through a generalized interaction representation with physical intuition and environmental invariance, serving as a symbolic bridge between abstract cognition and embodied execution, rather than taking the form of direct communication.

In this paper, we introduce *RoBridge*—a hierarchical intelligent architecture aimed at general robotic manipulation. As shown in Figure 1, our method maintains the declarative skill of VLM, and releases the procedural skill of reinforcement learning. This architecture innovatively constructs an intelligent system comprising three core elements: (1) a high-level cognitive planner (HCP) based on a large-scale pre-trained vision-language model; (2) an invariant operable representation (IOR) with physical intuition and environmental invariance, serving as a symbolic bridge between abstract cognition and embodied execution; (3) a guided embodied agent (GEA) that achieves precise action execution through interaction with the physical world. By generating IOR through the causal reasoning capabilities of the HCP and transforming abstract symbols into operations in physical space via the GEA, RoBridge effectively resolves the challenge of the disconnection between cognition and execution. While prior works have partially touched on individual ingredients, RoBridge is, to our knowledge, the first holistic system that synergistically unites the declarative power of large vision–language models with the procedural proficiency of reinforcement learning via the invariant operable representation (IOR) without mutual restriction. Our main contributions are as follows:

- We propose the first hierarchical intelligent architecture, RoBridge, for general robotic manipulation, breaking the paradigm dilemma of disconnection between cognitive abstraction and physical execution in traditional methods through a three-tier architecture of brain: HCP, symbolic bridge: IOR, and embodied agent: GEA.
- Guided Embodied Agent: We design a Guided Embodied Agent (GEA) capable of converting invariant operable representation (IOR) into concrete execution actions, maintaining superior performance under various interference conditions.

- We demonstrate the effectiveness of RoBridge through a series of experiments, showing its ability to generalize to unseen environments. We also find that RoBridge excels at handling new tasks and achieves a 75% success rate on five new tasks. And it excels in sim-to-real generalization, achieving an average success rate of 83% with only five real-world data samples per task.

2. Related Work

Robotic Manipulation Learning. Robotic manipulation has garnered extensive research attention in recent years. A straightforward approach utilizes imitation learning [41, 60], a form of supervised learning that maps observations to actions. Early methods achieved high performance by designing effective network architectures, constructing diverse training objectives, and utilizing suitable representations [8, 17, 36, 50]. To enable the deployment of strategies in diverse real-world scenarios, several large-scale robotic datasets have been introduced [11, 16, 27, 49]; however, the scale of these datasets pales in comparison to the expansive variability of real-world environments. Inspired by the success of large pre-trained models [3, 37, 47], researchers have begun fine-tuning vision-language models directly on robotic data to enhance the generalization capabilities of robotic models [5, 7, 28, 31]. Nonetheless, due to the domain gap between robotic and pre-training data, fine-tuned models often suffer from catastrophic forgetting, resulting in a significant decline in cognitive performance. An alternative approach involves reinforcement learning for robotic manipulation [12, 38, 51, 54], which can develop robust policies. However, these methods frequently struggle with complex tasks requiring language comprehension. In contrast, our approach transforms observations from the physical world into invariant, physically intuitive representations, integrating reinforcement learning and imitation learning to construct efficient and generalizable embodied agents capable of executing tasks across diverse environments.

Representations for Robotic Manipulation. Structured representations are designed to address the generalization challenges in robotic manipulation. In robotic manipulation, structured representation is a method of encoding complex environmental or object information into abstract forms with explicit geometric, semantic, or functional structures to support efficient reasoning, planning, and task execution. The core idea is to capture key information through simplified symbolic elements (such as keypoints [25, 32, 40], 6D poses [26, 30, 48], constraints [10, 22, 57, 64]) to reduce problem complexity and enhance generalization capabilities. Some methods [4, 14, 58] predict key frames of the motion process, utilizing them as primitive prompts for low-level control. Other approaches [26, 30] extract interaction trajectories from human videos to tap into diverse data sources, thereby enhancing the generalization to a wide array of previously unseen objects. This, however, often necessitates task-specific annotations. Additionally, some studies [25, 40] leverage the capabilities of vision-language models [37] and foundational models [29, 39, 53] to derive keypoints and 6D poses as representations, integrating these with motion planning for low-level control. Nonetheless, the absence of physical interaction data during pre-training often results in vision-language models generating plans that do not align with real-world physics, thereby limiting their effectiveness in practical applications. Our approach proposes a physically intuitive representation that does not involve specific execution details, leaving the execution to be handled by underlying embodied agents.

Generalizable Robot Manipulation. To enhance model generalization across different environments, particularly for transferring from simulated to real-world settings, existing methods frequently employ domain adaptation and domain randomization techniques. Domain adaptation aims to create additional synthetic images based on existing ones [21, 43, 63]. In contrast, domain randomization is more commonly used because it only requires randomizing material textures, object poses, camera parameters, and other factors [18, 45, 46]. Domain randomization is typically applied within simulated environments, with the trained policies subsequently transferred to real robots. Given its convenience, our approach adopts domain randomization to improve performance. We implement various domain randomization in the simulation environment, achieving high success rates in sim-to-real transfer with as few as five data samples.

3. Method

3.1. Framework

The framework of RoBridge is illustrated in Figure 2. It mainly contains three core components: a high-level cognitive planner, an invariant operation representation and a guided embodied agent. First, the high-level cognitive planner decomposes tasks into primitive actions based on observation and instruction. Then, for each action, the HCP integrates foundation models to generate an invariant operation representation. Finally, the guided embodied agent performs operations utilizing this representation, with the execution process being governed by closed-loop control. Detailed descriptions follow.

High-level Cognitive Planner (HCP). RoBridge's HCP consists of VLM (such as GPT-4o [37]) and some APIs (such as GroundingDINO [34], SAM [29] and Track-Anything [52] It is mainly responsible for high-level cognitive planning of tasks. Specifically, given the currently observed RGB Image $I^{h \times w \times 3}$ and instruction ℓ, we query VLM to decompose the task into several primitive actions,

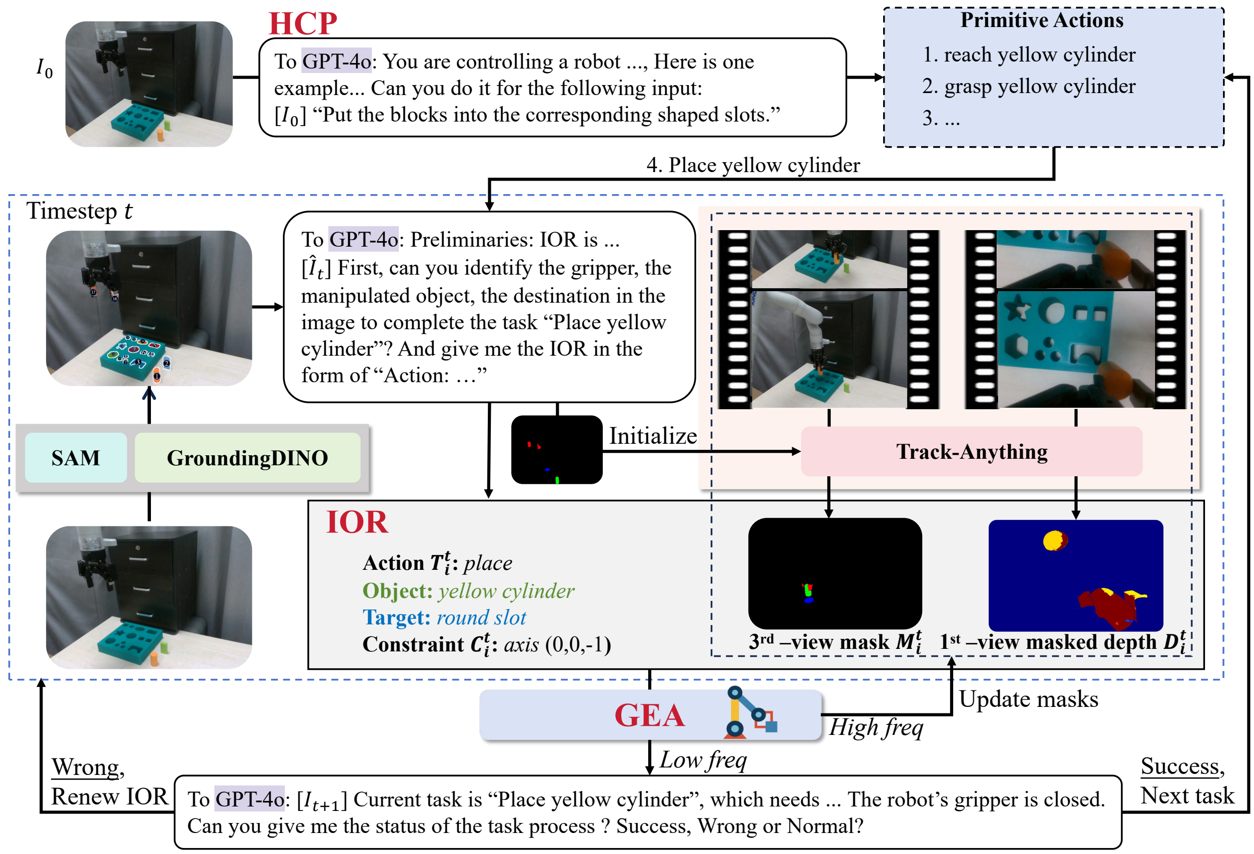

Figure 2. **RoBridge overview**. RoBridge adopts a three-layer architecture, consisting of a high-level cognitive planner (HCP), an invariant operable representation (IOR), and a guided embodied agent (GEA). For example, for the instruction "Put the blocks into the corresponding shaped slots", HCP will first plan and split the task into multiple primitive actions. Then, combined with the APIs composed of the foundation model, it will give IOR, which mainly includes the masked depth of the first perspective, the mask of the third perspective, the type of action, and the constraints. IOR is updated by HCP at a low frequency and track-anything updates the mask at a high frequency. IOR is used as the input of GEA, and GEA performs specific actions until the task is completed.

each primitive action $\mathcal{A}_i = \{T_i, obj_i, des_i\}$, with T_i denoting the type of action, obj_i indicating the name of the manipulated object and des_i indicating the destination or may not exist. The detailed definitions of primitive actions are shown in the Appendix. An example as shown in Figure 2, the task is decomposed into:
$\{\mathcal{A}_1 = \{reach, yellow\, cylinder, none\},$
$\mathcal{A}_2 = \{grasp, yellow\, cylinder, none\},$
$\mathcal{A}_3 = \{reach, round\, slot, none\},$
$\mathcal{A}_4 = \{place, yellow\, cylinder, round\, slot\}\}.$
This helps the model complete high-level cognitive planning of complex tasks.

Invariant Operable Representation (IOR). For each primitive action A_i and sensors data, we aim to transform them into an invariant operable representation $\mathcal{R}_i$, which consists of action type T_i, the 3^{rd}-view mask M_i, the 1^{st}-view masked depth D_i and constraints C_i. M_i consists of the 3^{rd}-view mask of the gripper M_i^g, the 3^{rd}-view mask of the manipulated object M_i^o, the 3^{rd}-view mask of destination M_i^d (if exists). Similarly, D_i consists of the 1^{st}-view masked depth of the gripper D_i^g, the 1^{st}-view masked depth of the manipulated object D_i^o, the 1^{st}-view masked depth of destination D_i^d (if exists). The constraints C_i include end-effector pose and direction of movement. The comprehensive representation for each action A_i is given by:

$$\mathcal{R}_i = \{T_i, M_i, D_i, C_i\}. \tag{1}$$

To obtain $\mathcal{R}_i$, we first use the APIs GroundingDINO [34] and SAM [29] in HCP to segment objects related to the primitive action. Subsequently, HCP's VLM determines the final selection of these objects like [53]. Simultaneously, for tasks with directional constraints, such as opening a drawer or turning a faucet, HCP provides corresponding direction $\mathbf{d} \in \mathbb{R}^3$, which is normalized and incorporated into the constraints C_i. Integrating this information with sensor data, we ultimately derive $\mathcal{R}_i$. IOR aims to help RoBridge gain better domain invariance and reduce the impact of environmental and task changes on the model.

Guided Embodied Agent (GEA). At each time step, a new $\mathcal{R}_i^t$ is generated. The specific update process will be described in the next closed-loop control section. For each $\mathcal{R}_i^t$, we need to map it into robotic movement $\mathbf{a}_t$ to proceed with the primitive action. In other words, we need to learn a policy $\pi\left(\mathcal{R}_i^t\right) \mapsto \mathbf{a}_t$. Following PSL [12], the primitive action "reach" typically involves moving the robot's end-effector to a well-defined target position. Unlike tasks such as grasping or placing, which involve complex object interactions and decision-making, "reach" can be efficiently addressed through motion planning. For other primitive actions, we integrate reinforcement learning and imitation learning to train a guided embodied agent capable of performing actions effectively, while ensuring robustness and consistent performance under varying input conditions. Detailed descriptions are shown in Section 3.2.

Closed-Loop Control. In order to maintain the accuracy of information and iteration of primitive actions in dynamic environments, we added closed-loop control. Since the speed of each part of the closed-loop control is different, we divide it into high-frequency control and low-frequency control. For high-frequency control, the update of $\mathcal{R}_i^t$ is performed as follows: first, new sensor data is acquired, and subsequently, Track-Anything [52] is employed to update M_i^t, D_i^t, thereby yielding the updated $\mathcal{R}_i^t$. For low-frequency control, we try to check and get the following three statuses: success, wrong and normal. We use GPT-4o combined with the gripper status to determine whether the task is successful, like [59]. We input the following information into GPT-4o for processing: an RGB image $\hat{I}_t$ annotated with task-related object tags, the status of the gripper (indicating whether it is open or closed), and the primitive action. Based on this input, GPT-4o generates a judgment regarding the success or failure of the current action. If the judgment indicates success, the system proceeds to the next action in the sequence or terminates the task, depending on whether all required actions have been completed. Conversely, if the judgment indicates failure, the system regenerates the input $\mathcal{R}_i$.

3.2. Guided Embodied Agent (GEA) Training

We aim to train an agent that achieves a high success rate and strong robustness, ensuring reliable and consistent performance across diverse scenarios, robot configurations, and tasks. To do this, we used multi-stage training, as shown in Figure 3.

RL Training. Initially, it is imperative to attain high performance. To this end, we employ reinforcement learning methodologies to train an expert agent π_e for each specific task, ensuring that each task is executed proficiently. Furthermore, to enhance the robustness of the expert agent, we introduce domain randomization variations in object shape, robotic arm placement, and camera orientation during the training process.

IL Training. Next, we start training a guided embodied agent π_g. We use expert agents to generate high-quality data for each task and extract generalized interaction representation $\mathcal{R}$ from the data as the input of π_g. In addition to the domain randomization employed during experts agent training, we further incorporated several domain randomization techniques, including depth distortion, dilation, random shifting, and the modification of masked information through addition and deletion. Such strategies are intended to improve the generalization capabilities and robustness of the trained agent across varying environmental conditions. To simulate the noisy depth sensors in the real world, we employ a series of augmentation techniques [13] when processing depth images. Specifically, we utilize depth warping, which applies Gaussian shifts to simulate variations in perspective and sensor noise. Additionally, we incorporate Gaussian blurring to mimic the effects of sensor blur and focus issues, which are common in real-world scenarios. Furthermore, we introduce random masking to create artificial holes in the depth maps, simulating occlusions and missing data that frequently occur in practical applications. Due to the potential inaccuracies and incomplete coverage inherent in the model-generated masks, we employ techniques such as random offsets and random cropping. These methods are applied to mitigate the risk of the agent becoming overly reliant on the masks.

Continual Skill Aggregation The agent often experiences compounding errors in imitation learning [61]. To address this, we introduce an iterative optimization strategy using DAgger [44]. Recognizing that online DAgger [1] can cause instability and offline DAgger updates slowly in multitask settings, we developed an adaptive sampling mechanism for offline DAgger. As detailed in Algorithm 1, this mechanism adjusts the sampling frequency based on task complexity: more samples for challenging ones. Initially, tasks are weighted equally. A function f maps rewards to values to adjust task weights. In each iteration, policy π_g is trained on current datasets, tasks are sampled by weight, and rewards are transformed by f to update weights. Failures are recorded, and experts π_e provide corrective data for the task dataset.

4. Experiments

We first introduce the experimental setup 4.1 along with the benchmarks and baselines 4.2. We present the main results 4.3 and demonstrate performance on unseen tasks 4.4. Additionally, we conduct ablation studies 4.5.

4.1. Experiment Settings

Architecture and Training. For each task, we employ a separate reinforcement learning (RL) policy, denoted as expert π_e. We train π_e using DRQ-v2 [54], a state-of-the-art RL algorithm, which takes as input both RGB images and the robot's state, alongside a task-specific one-hot vector

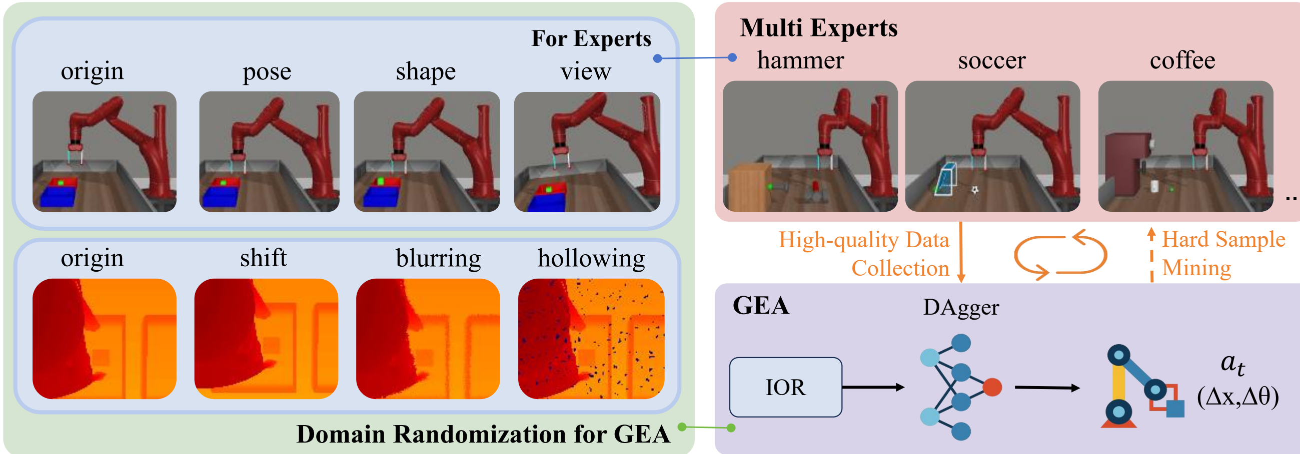

Figure 3. **Guided Embodied Agent(GEA) Training**. The left figure illustrates the domain randomization methods, with the first column showing the original images. The first row employs changes in robotic arm pose, object shape variations, and camera offsets, while the second row uses pixel offsets, depth distortion, hollowing, etc. The domain randomization in the first row is used during expert training, whereas all domain randomization methods are applied during GEA training. In the top right corner, it represents that we train an expert for each task, and the trained experts provide training data for GEA. The bottom right corner shows the training process of GEA, where it takes IOR as input and is trained using DAgger. Data from failed executions is corrected by the experts.

Table 1. Metaworld [55] Benchmark Results (%).

Model	MT50	Unseen Background	Unseen Light	Unseen Color	Unseen Camera Pose	Mean
DRQ-v2 [54]	24.0	24.6	17.4	22.2	19.8	21.60
SayCan [2]	77.8	26.6	27.8	12.8	10.4	31.08
PSL [12]	72.4	28.8	37.8	42.8	14.4	45.24
ManipGen [13]	73.2	75.4	70.4	72.4	62.8	70.84
ReKep [25]	31.4	32.8	31.8	32.2	32.6	32.16
RoBridge	**85.4**	**84.6**	**82.6**	**83.2**	**74.8**	**82.12**

representation, and outputs low-level actions. π_g employs a network architecture identical to that of DRQ-v2, with the exception of the absence of the critic network. This framework utilizes the Invariant Operable Representation (IOR) as input, where primitive actions are represented by one-hot vectors, and similarly outputs low-level actions.

Hardware Setup. In conducting real-world experiments, we utilize a Kinova Gen3 robotic arm. We also employ two Realsense D435i cameras: one mounted on the wrist to provide a first-person perspective, and the other one offers a third-person view.

4.2. Benchmark and Baselines

Simulation Benchmark. We conducted experiments in two simulation environments: Metaworld [55] and Robosuite [65]. Metaworld provides a diverse set of tasks, and we utilized it for both training and testing across 50 available tasks. In addition, we carefully selected 35 tasks for training and five distinct tasks for zero-shot testing, ensuring that there was no correlation between the training and testing task sets. The experiments on Robosuite are shown in the Appendix.

Real-world Evaluation. We conducted real-world experiments involving the following tasks: (1) "Pick up": accurately grasp and lift the specified object from the table. (2) "Sweep": move the object to a designated location. (3) "Press button": depress the button fully to achieve closure. (4) "Open Drawer": extend the drawer by a minimum distance of 10 cm. "Pick up" and "Sweep" tasks are tested with unseen objects to further evaluate generalization capability. We additionally designed a multi-stage task, which needs to pick up blocks and insert them into a correspondingly shaped groove. The task is designed to evaluate the model's capability and stability in handling long-horizon sequences. We set up four stages: "pick cylinder", "place cylinder", "pick cuboid", and "place cuboid", and measured the length of the execution process.

Baselines. DrQ-v2 [54] is a state-of-the-art reinforcement learning method. We adapted a multi-task DrQ-v2 using RGB inputs, robot proprioception, and one-hot encoded tasks. SayCan [2] uses a large language model (LLM) for skill planning. We used DrQ-v2 models trained on separate tasks as SayCan's skill library. PSL [12] organizes skills as actions, similar to SayCan. ManipGen [13] extends PSL

Table 2. Real-World Experimental Results (%).

Model	Pick Up		Sweep		Press Button	Open Drawer	Mean
	Seen	Unseen	Seen	Unseen			
ManipGen[13]	10	0	0	0	30	0	6.7
RDT [33]	20	0	0	0	25	0	7.5
π_0 [5]	40	10	30	0	10	20	18.3
π_0-fast [5]	35	10	25	0	30	10	18.3
RAM [30]	60	55	0	0	80	70	44.17
ReKep [25]	80	**85**	40	30	5	55	49.2
RoBridge	**85**	**85**	**70**	**65**	**95**	**100**	**83.3**

Algorithm 1 Adaptive Sampling DAgger

Initialize equal weights w_i for all tasks $\{T_i\}$
Define a piecewise function f : reward $\rightarrow$ value
Collect initial dataset D_i for each task T_i with equal size
while not converged **do**
  Train policy π_g using datasets $\{D_i\}$
  Sample n tasks according to weights w_i
  Test policy π_g with n tasks
  for each task T_i **do**
    for each reward in task T_i tests **do**
      $w_i' += f(\text{reward})$
    end for
    $w_i' = w_i'/$number of tests for T_i
    Record all failure data, let expert π_e generate correct data D_i'
    $D_i = D_i + D_i'$
  end for
  Update $w_i = w_i'$
end while

with DAgger and domain randomization. We expanded PSL and ManipGen's skill libraries for our tasks. RAM [30] is a retrieval-based, zero-shot robotic manipulation framework that transfers 2D affordances from diverse out-of-domain data to 3D executable actions. ReKep [25] employs keypoint representations for task planning. Due to its initial suboptimal performance, we enhanced prompts with additional task specifications and constraints. We also compared popular end-to-end methods, including RDT [33], π_0 [5], and its autoregressive version, π_0-fast. For real-world testing of RDT, π_0, and our RoBridge, we collected 5 demonstrations per task for fine-tuning.

4.3. Results Analysis

Simulation Results. We conducted tests on 50 tasks from Metaworld, and changed the background, lighting, object colors, and camera poses to evaluate generalization capabilities. The average success rate is presented in the last column. As shown in Table 1, our method achieved the best

Table 3. Real-World Long-horizon Experimental Results (%).

Model	1	2	3	4	Avg. Len.
ManipGen[13]	20	0	0	0	0.2
RDT [33]	30	0	0	0	0.3
π_0 [5]	45	5	0	0	0.5
π_0-fast [5]	30	10	0	0	0.4
ReKep [25]	**100**	40	25	5	1.7
RoBridge	**100**	**80**	**70**	**50**	**3.0**

performance across various tests, with an average success rate of 82.12%, which is 11.28% higher than the best baseline. This demonstrates the effectiveness and robustness of our approach.

Real-World Results. The real-world results are shown in Table 2 and Table 3. Our method achieved the best results, with an average success rate of 83.3% across four tasks and an average length of 3.0 in the long-horizon task. RoBridge's performance on long-horizon tasks demonstrates its combined planning and execution capabilities. We also show qualitative results, which are shown in Figure 4. The first row displays the tasks process completed by Rekep and π_0, respectively. The second row illustrates the process completed by RoBridge. In end-to-end models, π_0 achieved relatively good success rates. However, it exhibited instability when handling long-horizon tasks and is easy to failure. ReKep struggles with physics knowledge, such as not knowing that the gripper needed to be closed to press the button. Our model effectively solves the problems of these baselines. More visualizations are shown in the Appendix.

4.4. Zero-shot Unseen Tasks Generalization

We conducted tests on five novel tasks that are unrelated to those used during training. As shown in Table 4, RoBridge achieved an average success rate of 75% across these new tasks. This demonstrates that our method can effectively execute tasks beyond those seen during training, eliminating the need to collect data for each specific task, thereby reducing data collection costs.

4.5. Ablation Study

We conducted ablation experiments on the components of the IOR and the training of the GEA to understand their

Table 4. Experimental Results (%) on Unseen Tasks. '-' indicates that the method cannot handle this task due to its underlying principles.

Model	Bin Picking	Pick out	Handle press	Plate Slide	Sweep Into	Mean
DRQ-v2 [54]	-	-	-	-	-	-
SayCan [2]	-	-	-	-	-	-
PSL [12]	0	0	60	0	0	12
ManipGen [13]	20	0	60	10	0	18
ReKep [25]	0	**60**	50	0	30	28
RoBridge	**70**	45	**100**	**90**	**70**	**75**

Table 5. Ablations Study Results (%).

Model	MT50	Unseen Background	Unseen Light	Unseen Color	Unseen Camera Pose	Mean
RoBridge	**85.4**	**84.6**	**82.6**	**83.2**	**74.8**	**82.12**
RoBridge w/ original depth	87.2	80.3	84.2	84.2	60.8	79.74
RoBridge w/ DINOv2	62.2	57.2	47.6	54.2	30.6	50.36
RoBridge w/ keypoints	80.2	62.2	55.4	58.4	43.2	59.88
RoBridge w/ language-only	84.8	34.4	26.8	30.2	20.2	39.28
RoBridge w/ smaller VLM	65.4	62.8	60.4	66.2	64.4	63.84
RoBridge w/o DAgger	68.4	69.0	68.6	69.4	50.2	65.12
RoBridge w/o online learning	75.4	75.2	76.4	74.8	66.6	73.68
RoBridge w/o domain randomization	62.8	62.8	62.4	62.2	30.2	56.08

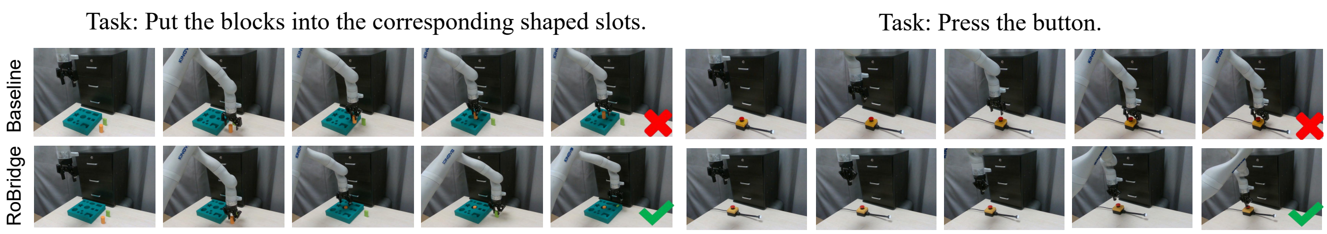

Figure 4. Demonstrations show the execution process of RoBridge (second row) and baselines π_0 (left in the first row), Rekep (right in the first row).

significance. [13] used the original depth instead of the 1st masked depth, and [62] used features obtained from DINOv2 [39] as inputs instead of the mask in 3rd view. We conducted ablations on our modules using these two methods. As shown in the second and third rows of Table 5, the original depth might be easier for the model to learn due to the presence of more details, resulting in a higher success rate on the MT50 tasks. However, the success rate drops sharply when there are changes in camera pose. The performance of DINOv2 was not satisfactory, which we think is due to its pre-trained features not being fully suitable for robotics. The sharp decline in performance for both methods under unseen camera poses indicates that these representations lack good invariance, making it difficult to achieve generalization. The comparison with using keypoints and language shows that RoBridge's success is not only due to richer supervision but also because the IOR design offers better invariance and intuitiveness. The degradation caused by using a smaller VLM indicates that the capability of VLM is necessary to effectively understand instructions and visual inputs. We also conducted ablation studies on the methods employed in GEA training, as shown in the last three rows of Table 5. Each method was found to be critically important in enhancing the training effectiveness of the GEA.

5. Conclusion

In this paper, we introduce RoBridge, a novel hierarchical intelligent architecture designed to enhance robotic manipulation by bridging the gap between high-level cognitive planning and low-level physical execution. The architecture integrates a high-level cognitive planner, an invariant operable representation, and a guided embodied agent, demonstrating significant advancements in task generalization and execution robustness. Through extensive experiments, RoBridge has shown superior performance and strong zero-shot generalization capabilities in unknown environments and novel tasks.

Acknowledgements

This work is supported by Scientific Research Innovation Capability Support Project for Young Faculty under Grant No.ZYGXQNJSKYCXNLZCXM-I28, National Natural Science Foundation of China (NSFC) under Grants No.62476293 and 62272494, Shenzhen Science and Technology Program under Grant No.GJHZ20220913142600001, Nansha Key R&D Program under Grant No.2022ZD014, General Embodied AI Center of Sun Yat-sen University, and China Postdoctoral Science Foundation under Grant No.2025M771522.

References

[1] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision, 2022. 5

[2] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022. 6, 8

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. 1, 3

[4] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models, 2023. 3

[5] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024. 1, 3, 7

[6] 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. 1

[7] 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, Alexander 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, 2023. 1, 3

[8] 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 J 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, 2023. 3

[9] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation, 2024. 1

[10] Shuo Cheng, Caelan Garrett, Ajay Mandlekar, and Danfei Xu. Nod-tamp: Generalizable long-horizon planning with neural object descriptors, 2024. 3

[11] 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 Frujeri, Freek Stulp, Gaoyue Zhou, Gau-

rav 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, Jingyun 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, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Baijal, 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 Haldar, 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 Matsushima, 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, Yansong 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, 2024. 3

[12] Murtaza Dalal, Tarun Chiruvolu, Devendra Chaplot, and Ruslan Salakhutdinov. Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks, 2024. 3, 5, 6, 8

[13] Murtaza Dalal, Min Liu, Walter Talbott, Chen Chen, Deepak Pathak, Jian Zhang, and Ruslan Salakhutdinov. Local policies enable zero-shot long-horizon manipulation, 2024. 5, 6, 7, 8

[14] Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation, 2023. 3

[15] Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied ai: From simulators to research tasks, 2022. 1

[16] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot, 2023. 3

[17] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation, 2024. 3

[18] Ricardo Garcia, Robin Strudel, Shizhe Chen, Etienne Arlaud, Ivan Laptev, and Cordelia Schmid. Robust visual sim-to-real transfer for robotic manipulation, 2023. 3

[19] Michael S Gazzaniga, Richard B Ivry, and GR Mangun. Cognitive neuroscience. the biology of the mind,(2014), 2006. 2

[20] Sten Grillner. Neurobiological bases of rhythmic motor acts in vertebrates. *Science*, 228 4696:143–9, 1985. 2

[21] Daniel Ho, Kanishka Rao, Zhuo Xu, Eric Jang, Mohi Khansari, and Yunfei Bai. Retinagan: An object-aware approach to sim-to-real transfer, 2021. 3

[22] Joy Hsu, Jiayuan Mao, Joshua B. Tenenbaum, and Jiajun Wu. What's left? concept grounding with logic-enhanced foundation models, 2023. 3

[23] Yafei Hu, Quanting Xie, Vidhi Jain, Jonathan Francis, Jay Patrikar, Nikhil Keetha, Seungchan Kim, Yaqi Xie, Tianyi Zhang, Hao-Shu Fang, Shibo Zhao, Shayegan Omidshafiei, Dong-Ki Kim, Ali akbar Agha-mohammadi, Katia Sycara, Matthew Johnson-Roberson, Dhruv Batra, Xiaolong Wang, Sebastian Scherer, Chen Wang, Zsolt Kira, Fei Xia, and Yonatan Bisk. Toward general-purpose robots via foundation models: A survey and meta-analysis, 2024. 1

[24] 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, 2023. 1

[25] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation, 2024. 1, 3, 6, 7, 8

[26] Yuanchen Ju, Kaizhe Hu, Guowei Zhang, Gu Zhang, Mingrun Jiang, and Huazhe Xu. Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation, 2024. 3

[27] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset, 2024. 3

[28] 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. 1, 3

[29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. 3, 4

[30] Yuxuan Kuang, Junjie Ye, Haoran Geng, Jiageng Mao, Congyue Deng, Leonidas Guibas, He Wang, and Yue Wang. Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation, 2024. 3, 7

[31] Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models, 2024. 3

[32] Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-world robotic manipulation through mark-based visual prompting, 2024. 1, 3

[33] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation, 2024. 1, 7

[34] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2024. 3, 4

[35] Josh Merel, Matthew Botvinick, and Greg Wayne. Hierarchical motor control in mammals and machines. *Nature communications*, 10(1):5489, 2019. 2

[36] 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, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. `https://octo-models.github.io`, 2023. 1, 3

[37] OpenAI. Gpt-4 technical report, 2024. 1, 3

[38] OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand, 2019. 3

[39] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. 3, 8

[40] Mingjie Pan, Jiyao Zhang, Tianshu Wu, Yinghao Zhao, Wenlong Gao, and Hao Dong. Omnimanip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints, 2025. 1, 3

[41] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988. 3

[42] Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. The colosseum: A benchmark for evaluating generalization for robotic manipulation, 2024. 1

[43] Kanishka Rao, Chris Harris, Alex Irpan, Sergey Levine, Julian Ibarz, and Mohi Khansari. Rl-cyclegan: Reinforcement learning aware simulation-to-real, 2020. 3

[44] Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011. 5

[45] Robin Strudel, Alexander Pashevich, Igor Kalevatykh, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Learning to combine primitive skills: A step towards versatile robotic manipulation, 2020. 3

[46] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world, 2017. 3

[47] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. 1, 3

[48] Yian Wang, Ruihai Wu, Kaichun Mo, Jiaqi Ke, Qingnan Fan, Leonidas Guibas, and Hao Dong. Adaafford: Learning to adapt manipulation affordance for 3d articulated objects via few-shot interactions, 2023. 3

[49] Zhiqiang Wang, Hao Zheng, Yunshuang Nie, Wenjun Xu, Qingwei Wang, Hua Ye, Zhe Li, Kaidong Zhang, Xuewen Cheng, Wanxi Dong, et al. All robots in one: A new standard and unified dataset for versatile, general-purpose embodied agents. *arXiv preprint arXiv:2408.10899*, 2024. 3

[50] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation, 2023. 3

[51] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. Daydreamer: World models for physical robot learning, 2022. 3

[52] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos, 2023. 3, 5

[53] 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. 3, 4

[54] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning, 2021. 3, 5, 6, 8

[55] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning, 2021. 6

[56] Kai Yuan, Noor Sajid, Karl Friston, and Zhibin Li. Hierarchical generative modelling for autonomous robots. *Nature Machine Intelligence*, 5(12):1402–1414, 2023. 2

[57] Wentao Yuan, Chris Paxton, Karthik Desingh, and Dieter Fox. Sornet: Spatial object-centric representations for sequential manipulation, 2022. 3

[58] Kaidong Zhang, Pengzhen Ren, Bingqian Lin, Junfan Lin, Shikui Ma, Hang Xu, and Xiaodan Liang. Pivot-r: Primitive-driven waypoint-aware world model for robotic manipulation. *arXiv preprint arXiv:2410.10394*, 2024. 3

[59] Kaidong Zhang, Pengzhen Ren, Bingqian Lin, Junfan Lin, Shikui Ma, Hang Xu, and Xiaodan Liang. Pivot-r: Primitive-driven waypoint-aware world model for robotic manipulation, 2024. 5

[60] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation, 2018. 3

[61] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. 5

[62] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning, 2025. 8

[63] Eric Zhu, Mara Levy, Matthew Gwilliam, and Abhinav Shrivastava. Nerf-aug: Data augmentation for robotics with neural radiance fields, 2024. 3

[64] Yifeng Zhu, Zhenyu Jiang, Peter Stone, and Yuke Zhu. Learning generalizable manipulation policies with object-centric 3d representations, 2023. 3

[65] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning, 2025. 6