

FoAM: Foresight-Augmented Multi-Task Imitation Policy for Robotic Manipulation

Litao Liu¹⁵, Wentao Wang², Yifan Han³, Zhuoli Xie¹⁶, Pengfei Yi³, Junyan Li³, Wenzhao Lian^{4*}

¹Corenetic AI

²University of Southern California

³Institute of Automation, Chinese Academy of Sciences

⁴School of Artificial Intelligence, Shanghai Jiao Tong University

⁵Rutgers University-New Brunswick

⁶University of Minnesota Twin Cities

litao.liu@rutgers.edu, lianwenzhao@sjtu.edu.cn

Abstract

Multi-task imitation learning (MTIL) has shown significant potential in robotic manipulation by enabling agents to perform various tasks using a single policy. It simplifies the policy deployment and enhances the agent’s adaptability across different scenarios. However, key challenges remain, such as maintaining action reliability (e.g., avoiding abnormal action sequences that deviate from nominal task trajectories) and generalizing to unseen tasks with a few expert demonstrations. To address these challenges, we introduce the **Foresight-Augmented Manipulation (FoAM)** policy, a novel MTIL policy that pioneers the use of multi-modal goal conditions as input and introduces a foresight augmentation in addition to the general action reconstruction. FoAM enables the agent to reason about its actions’ visual consequences (foresight) and to be guided by these more expressive representations during task execution. Extensive experiments on over 100 tasks in simulation and real-world settings demonstrate that FoAM significantly enhances MTIL policy performance, outperforming state-of-the-art baselines by up to 41% in success rate. We released our simulation suites that include over 80 challenging tasks across more than 10 scenarios designed for manipulation policy training and evaluation.

Homepage — <https://projfoam.github.io/>

Introduction

One of the main goals of robot learning is to develop a general agent capable of performing various tasks based on user commands. Multi-task imitation learning (MTIL) is a key approach that enables agents to learn multiple skills from expert demonstrations and trains an efficient policy, eliminating the need for complex, hardcoded solutions or reward functions. However, suboptimal policies and how to obtain generalization with limited training data are key issues that need to be addressed in MTIL policy development. This requires the new MTIL framework to enable the agent to learn

*This work was completed by Liu Litao during his internship at Corenetic.ai. He is currently a Ph.D. student at Rutgers University. Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

task-independent skills and more expressive features from demonstrations, capturing task-specific details to ensure reliable execution of individual tasks and generalizing to unseen tasks and scenarios (Kroemer, Niekum, and Konidaris 2020; Zhu and Hu 2018; Rivero-Moreno et al. 2023).

Previous research has shown that task adaptation in MTIL can be achieved by incorporating goal conditions into multi-task policy training (Haldar, Peng, and Pinto 2024; Kim et al. 2024; Zhen et al. 2024; Brohan et al. 2022, 2023; Ding et al. 2019). However, the unreliable execution of multi-task policies remains a challenge. Existing MTIL policies that align robotic actions with expert actions based on goal conditions often fail to reason about these distribution differences in the demonstrations of various tasks, severely impacting the agents’ performance on individual tasks. Meanwhile, most MTIL policies adopt uni-modal goal conditions (Haldar, Peng, and Pinto 2024; Sundaresan et al. 2024; Rivera et al. 2022; Ni et al. 2024; Jiang et al. 2022; Yokoyama et al. 2024; Nasiriany et al. 2019), which results in some inherent limitations. For example, with limited demonstrations and without data augmentation, policies conditioned on task instructions have difficulty generalizing to unseen tasks (discuss in Section **Experiment Results**). While policies conditioned on goal images offer fine-grained guidance and even demonstrate striking zero-shot capabilities (Rivera et al. 2022; Pertsch et al. 2020), they encounter ambiguities during task execution. Ambiguity means that the same goal image may exist for different tasks in the same scenario. For instance, when the robot is tasked with placing an object into a multi-layer locker, the resulting goal image has ambiguity due to closedness. The manipulated object in the initial image disappears in the goal image, making it impossible to determine which specific layer the object is placed in just based on the goal image (**Figure 4** real-world Scenario IV).

In this paper, we introduce the **Foresight-Augmented Manipulation (FoAM)** policy, a novel MTIL policy designed to enhance the task performance of agents while addressing the limitations of uni-modal goal-conditioned policies. This approach is inspired by the perception ability of

humans when performing tasks. When humans receive task instructions, they can easily foresee the state when completing the task in mind, and guide task execution based on this foresight and real-time actions until the expected results are achieved (Dezfouli and Balleine 2013). Similarly, FoAM cleverly uses foresight to guide task execution. In addition, it addresses the limitations of uni-modal goal-conditioned MTIL policies by utilizing a multi-modal goal condition (i.e., goal image and task instruction). It also innovatively attempts to leverage a fine-tuned vision-language model (VLM) (Brooks, Holynski, and Efros 2023) to autonomously generate goal images. During training, we apply an action loss to refine the policy’s behavior and novelly introduce a foresight loss to control the consequences of its actions. This allows the agent to reason about its action across diverse tasks, and handle the ambiguities and variations in expert demonstration data, leading to more forward-looking and precise action during inference. FoAM demonstrates significant effectiveness with evaluations across more than 100 tasks in both simulation and the real world. It outperforms state-of-the-art baselines, achieving an increase in success rate by up to 41% in success rate. Our main contributions are summarized as follows:

- We employ a multi-modal goal condition to address the limitations of unimodal goal-conditioned policies. FoAM demonstrates the capability to generalize to unseen tasks with limited expert data while ensuring effective execution in ambiguous scenarios. Additionally, we incorporate a Vision-Language Model (VLM) into FoAM, which enables the agent to acquire goal images autonomously.
- We propose foresight augmentation, a novel method designed to enable agents to learn more expressive representations and enhance task performance by aligning task instructions with the consequences of their actions.
- We have open-sourced a dual-arm system with rich tasks (over 80). This system replicates the UR3e robot in MuJoCo (Todorov, Erez, and Tassa 2012). It serves as a valuable simulation tool for developing robotic manipulation policies, such as MTIL and Sim2Real reinforcement learning policies (Lum et al. 2024).

Related Work

Goal-conditioned Learning for Robotic Manipulation. In recent years, significant progress has been made in single-task learning policies (Zhao et al. 2023; Chi et al. 2023; Buamane et al. 2024; Mishra et al. 2023; Fu, Zhao, and Finn 2024). However, to enable a wider adaptability, intelligent robots must be equipped with the ability to conduct diverse tasks and complete them effectively. Among current MTIL approaches, language-conditioned policies utilize large-scale datasets to achieve task generalization, or apply data augmentation techniques, such as vision generation models, to modify backgrounds and manipulated objects, enabling generalization across more tasks and scenarios with limited training data (Brohan et al. 2022; Padalkar et al. 2023; Brohan et al. 2023; Bharadhwaj et al. 2024; Hal-dar, Peng, and Pinto 2024; Ha, Florence, and Song 2023;

Reuss et al. 2023; Kim et al. 2024; Wang et al. 2024). Despite the promising initial success, we found that language-conditioned policies often struggle with unseen tasks without additional data augmentation. In parallel, some policies have introduced the goal image as a task condition (Sundaresan et al. 2024; Hal-dar, Peng, and Pinto 2024; Rivera et al. 2022; Mandlekar et al. 2023; Fang et al. 2024; Black et al. 2023). Compared to language inputs, images could provide more expressive representation at the pixel level, enabling stronger generalization capabilities, and even allowing agents to perform zero-shot tasks (Rivera et al. 2022). However, goal images are susceptible to scenario ambiguity, where visually identical images can be produced by different tasks, causing actions inconsistent with intention.

Collecting goal images requires human involvement, such as hand-drawing sketch (Sundaresan et al. 2024), which reduces the autonomy of the agent. Recent work has explored multi-modal goal conditions to enhance agents’ task performance (Jiang et al. 2022; Ni et al. 2024; Reed et al. 2022; Alayrac et al. 2022; Black et al. 2023). These policies extract workspaces and manipulated objects from both task instructions and initial observation, then compose a multi-modal embedding based on predefined templates. In contrast, FoAM leverages a vision-language model to generate a goal image with semantic information. The generated goal image and task instruction are directly processed to infer actions and predict their consequences. By doing so, our method addresses the limitations of uni-modal goal-conditioned policies that rely solely on language instructions, goal images, or predefined templates.

Agents with Vision Language Models. In recent years, Vision-Language Models (VLMs) have been introduced to robotics (Huang et al. 2022, 2023; Alayrac et al. 2022; Yang et al. 2023; Zhi et al. 2024), enabling more complex visual reasoning and multi-modal applications. Meanwhile, in the community of image editing, VLMs also demonstrate the ability to understand language instructions, edit real-world images, and produce highly realistic visual effects (Zhang et al. 2024; Fang et al. 2024). Works (Bharadhwaj, Gupta, and Tulsiani 2023; Sridhar et al. 2024) further validate that edited images can be interpreted by robotic agents, where the generated goal images are directly used as a goal condition. In contrast, we integrate VLMs seamlessly into our framework not only during the inference stage, but also in policy training. The generated goal images serve as the “labels” to compute the reconstruction loss for the policy (Section **Foresight Augmentation**), coupling the action learning and action result prediction.

Method

We seek to develop an MTIL policy that can reliably generate actions consistent with semantics while effectively generalizing to unseen tasks with a small amount of training data, enabling agents to complete various tasks more efficiently and accurately. In the following four sections, we will provide an overview of the FoAM, detail the fine-tuning of a cutting-edge visual-language model to generate goal images for FoAM, propose the policy foresight augmentation method, and introduce the implementation details of FoAM.

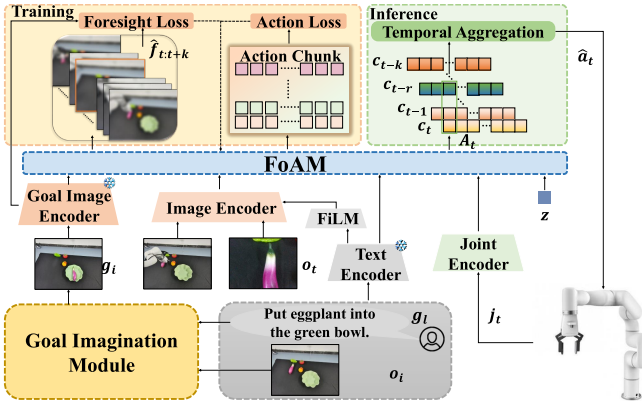


Figure 1: Training and inference pipelines of FoAM. The input terms remain consistent throughout both training and inference. During training, actions and their corresponding foresight are predicted, with the foresight and actions being aligned with the input goal image and expert actions, respectively, to update the parameters of FoAM. During inference, the trained policy is used solely to predict the action \hat{a}_t .

Pipeline Overview

The pipeline of FoAM is illustrated in **Figure 1**. FoAM is a transformer-based (Vaswani 2017) policy that inherits the architecture of the prior work (Zhao et al. 2023) and is trained as a conditional variational autoencoder (CVAE) (Kingma 2013; Sohn, Lee, and Yan 2015). The process begins with the user providing a task instruction g_t to the agent, which is then fed into FoAM as a language goal condition through a pre-trained text encoder (Gadre et al. 2023). Concurrently, the task instruction, such as *Put eggplant into the green bowl*, is input into the Goal Imagination Module along with the initial observation o_i . The generated goal image or human-collected real goal image g_i serves as image goal condition. By integrating the agent’s proprioception j_t , visual observations o_t , and the latent style variable z , the multi-modal goal-conditioned policy $\pi_\theta(\hat{a}_{t:t+k}|o_t, j_t, g_i, g_t, z)$ is reconstructed by the coupling of foresight and action loss. During inference, action chunks are predicted and smooth actions are produced through temporal aggregation (Zhao et al. 2023).

Fine-tuned Goal Imagination Module

The acquisition of goal images often requires human participation, such as hand-drawing sketch (Sundaresan et al. 2024) or collecting real goal images (Halder, Peng, and Pinto 2024) before inference. In order to improve the efficiency of the agent’s task execution, we embed a goal imagination module in FoAM to generate goal images. In this work, we chose InstructPix2Pix (Ip2p) (Brooks, Holynski, and Efros 2023) as the goal imagination module, leveraging a dataset of over 20,000 training pairs. Of these, 16,000 pairs were obtained from robot expert demonstrations provided by RT-1 (Padalkar et al. 2023; Brohan et al. 2022, 2023). The first and last frames of these demonstrations were used as the original and edited images, respectively, and the correspond-



Figure 2: Inference demonstrations of the fine-tuned goal imagination module. The leftmost image illustrates the initial observation, while the following four images represent the edited goal images generated according to the task instruction. Please visit the homepage for more examples.

ing task name served as the instruction. Since RT-1 demonstrations exhibited perturbations in the final frames due to robot arm movements, we implemented a data-cleaning procedure to remove noise and ensure high-quality training data. Additionally, we incorporated over 4,000 data pairs from our own simulation and real-world datasets.

We fine-tuned Ip2p for 500 epochs on a single NVIDIA H100 GPU, a process that required approximately 3 days. During the inference stage, with the model weights pre-loaded, generating one goal image of size $480 \times 640 \times 3$ took about 4 seconds. **Figure 2** presents some inference demonstrations captured during **VLM-FoAM Joint Inference** experiments. They highlight the model’s ability to generate realistic and semantically consistent visuals based on the given initial observations and task instructions.

Foresight Augmentation

Humans possess exceptional perception abilities for understanding and interacting with external events. When performing a task, people can easily foresee the goal state of a scenario before execution in mind and use it as a guide. Inspired by this capability, we developed a key module called Foresight Augmentation (FA) to equip agents with a similar perception mechanism. FA enables the agent to simultaneously comprehend both its actions and the subsequent consequences these actions will produce. By integrating FA into the agent’s decision-making process, we hope to enhance its overall performance in tasks execution.

We train FoAM as a CVAE, utilizing an encoder similar to those in (Zhao et al. 2023; Bharadhwaj et al. 2024; Shi et al. 2024; Buamane et al. 2024) to generate the latent variable $z \sim q_\phi(z|a_{t:t+k}, j_t)$. The decoder is defined as policy $\pi_\theta(\hat{a}_{t:t+k}, \hat{f}_{t:t+k}|o_t, j_t, g_i, g_t, z)$, which predicts a $k \times n$ -dimensional action chunk $\hat{a}_{t:t+k}$ and foresight sequence $\hat{f}_{t:t+k}$ based on real-time observation and conditions, where k represents the hyperparameter chunk size during training and n denotes the dimension of the agent’s action space. To enhance the agent’s ability to interpret and respond to the dynamic work scenario, we strategically increase the value of k , thereby expanding the agent’s foresight horizon. For example, since each episode in the multi-task expert demonstration dataset \mathcal{D} often has different time steps, we define the maximum time step of all episodes as T and set k to a value close to T . During training, while FoAM predicts the action chunk $\hat{a}_{t:t+k}$, FA generates k potential foresight $\hat{f}_{t:t+k}$ and selects the frame $\hat{g}_i = \hat{f}_{t:t+k}[k-t]$ that is temporally consistent with the goal image g_i . This

Algorithm 1: FoAM Training

Require: Expert demo \mathcal{D} , maximum episode time step T , chunk size k ($k \approx T$), and loss weights α, β, γ

- 1: Each episode includes a_t, j_t, o_t, g_t and g_i , representing the action, agent proprioception, visual observation at time t , task prompt, and goal image, respectively.
- 2: Init CVAE encoder $q_\phi(z|a_{t:t+k}, j_t)$
- 3: Init CVAE decoder $\pi_\theta(\hat{a}_{t:t+k}, \hat{f}_{t:t+k}|o_t, j_t, g_i, g_t, z)$
- 4: **for** each batch $i = 1, 2, \dots$ **do**
- 5: Random sample $a_{t:t+k}, j_t, o_t$ from \mathcal{D}
- 6: Encode latent variable z from $q_\phi(z|a_{t:t+k}, j_t)$
- 7: Predict actions $\hat{a}_{t:t+k}$ and foresight $\hat{f}_{t:t+k}$ using decoder $\pi_\theta(\hat{a}_{t:t+k}, \hat{f}_{t:t+k}|o_t, j_t, g_i, g_t, z)$
- 8: $\mathcal{L}_{\text{action}} = L_1(\hat{a}_{t:t+k}, a_{t:t+k})$
- 9: $\mathcal{L}_{\text{foresight}} = \text{Huber}(\hat{g}_i, g_i)$, where $\hat{g}_i = \hat{f}_{t:t+k}[k-t]$
- 10: $\mathcal{L}_{\text{reg}} = D_{\text{KL}}(q_\phi(z|a_{t:t+k}, j_t) \parallel \mathcal{N}(0, I))$
- 11: Update CVAE parameters θ and ϕ using ADAM optimizer with total loss $\mathcal{L} = \alpha\mathcal{L}_{\text{action}} + \beta\mathcal{L}_{\text{foresight}} + \gamma\mathcal{L}_{\text{reg}}$
- 12: **end for**

selected frame \hat{g}_i is then aligned with the goal image g_i using the foresight loss $\mathcal{L}_{\text{foresight}}$ calculated by the Huber Loss, which is coupled with the action loss $\mathcal{L}_{\text{action}}$ as the reconstruction loss $\mathcal{L}_{\text{recon}}$ of the policy to play a role in updating the policy’s parameters (**Algorithm 1**). This process simulates the strong perception abilities humans exhibit when performing tasks. We demonstrated in experiments that the FA significantly enhances the agent’s task performance.

FoAM Policy Implementation

FoAM is designed as a transformer-based policy with sufficient capacity to predict specific sequences by effectively integrating sequence information from the inputs. FoAM is implemented using an ACT-like architecture (Zhao et al. 2023) with the CVAE framework. The language-conditioned embedding is obtained by the pre-trained language encoder (Gadre et al. 2023) to produce a 384-dimensional feature, which is subsequently projected to 512 dimensions through an MLP. ResNet18 (He et al. 2016) with FiLM conditional layer (Perez et al. 2018) is used to encode visual observations of size $480 \times 640 \times 3$ and embed task instructions into images, ensuring robust task performance in multiple scenarios (Bharadhwaj et al. 2024). The visual observations are finally transformed into a $(300 \times n) \times 512$ feature sequence, where n denotes the number of used viewpoints. The goal image g_i is encoded by the pre-trained ResNet18, producing a 300×512 feature, and remains fixed during training without parameter updates. The latent variable z is obtained with a 4-layer transformer encoder and projected to 512 dimensions. Proprioceptive input j_t is projected to 512 dimensions through an MLP. The CVAE decoder consists of a 4-layer transformer encoder and a 7-layer transformer decoder. The input feature dimensions for the transformer encoder are $(303 + 300 \times n) \times 512$. The encoder fuses features from different modalities, and the decoder predicts the action chunk $\hat{a}_{t:t+k}$, while generating k foresight $\hat{f}_{t:t+k}$ (each 300×512)

Algorithm 2: FoAM Inference

Require: trained policy $\pi_\theta(\hat{a}_{t:t+k}|o_t, j_t, g_i, g_t, z)$, where $z = 0$, maximum inference time step L , chunk size k , temporal aggregation range r and weight coefficient λ .

- 1: Init an action buffer $B[L, L+k, *]$, where $B[t]$ stores action chunk $\hat{a}_{t:t+k}$.
- 2: **for** time step $t = \text{range}(L)$ **do**
- 3: Predict $\hat{a}_{t:t+k}$ with π_θ
- 4: Add $\hat{a}_{t:t+k}$ to buffer $B[t, t:t+k]$
- 5: Extract temporal aggregation array $A_t = B[-r :, t]$
- 6: Get $\hat{a}_t = \sum_i w_i A_t[i] / \sum_i w_i$, with $w_i = \exp(-\lambda * i)$
- 7: **end for**

through a second-dimensional full connection layer.

The FoAM training process, which incorporates the FA, is outlined in **Algorithm 1**. During training, we use L1 Loss and Huber Loss to compute the action loss $\mathcal{L}_{\text{action}}$ and foresight loss $\mathcal{L}_{\text{foresight}}$ respectively, along with a KL divergence term \mathcal{L}_{reg} regularizing the CVAE encoder. These losses are weighted by α, β , and γ . In our experiments, the weight values are set to 1, 2, and 10, respectively.

During inference, the FA module is discarded. The implemented policy is represented by $\pi_\theta(\hat{a}_{t:t+k}|o_t, j_t, g_i, g_t, z)$. Based on the current observations and goals, the action chunk $c_t = \hat{a}_{t:t+k}$ is predicted. Following prior action chunk-based policies (Bharadhwaj et al. 2024; Zhao et al. 2023; Haldar, Peng, and Pinto 2024), we apply exponential temporal aggregation to produce smooth action trajectories. **Unlike previous work**, we introduce the hyperparameter temporal aggregation range r as the actual chunk size during inference, which eliminates the equality constraint on chunk size k during training and inference. This is particularly crucial for deploying FoAM, as it allows flexibly adjusting the aggregation range according to the user’s intentions and the characteristics of different tasks, and optimizing task performance. During experiments, we observed that when r is large (> 100), the agent can perform tasks more efficient, and when r is relatively small ($50 \sim 80$), the agent has better reactivity and ability to resist external disturbance (**Section Robustness Analysis**). The inference code is shown in **Algorithm 2**. There are approximately $160M$ parameters in the training and around $80M$ in the inference.

Experiments

Our experiments will answer the following questions:

- (a) How does multi-modal goal condition perform?
- (b) How does foresight augmentation perform?
- (c) Can FoAM generalize to unseen tasks with a few expert episodes and without data augmentation?
- (d) How does FoAM perform differently when guided by a real or generated goal image?
- (e) How well does FoAM respond to external disturbance?

Data Collection

FoAM Benchmark. We developed a dual-arm system in MuJoCo (Todorov, Erez, and Tassa 2012), a popular physics

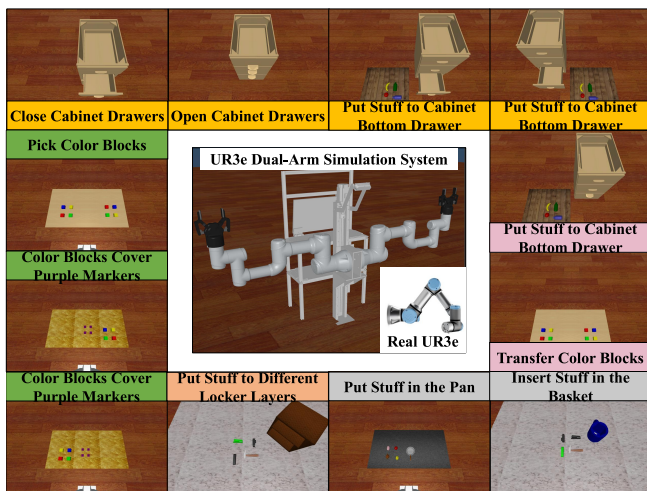


Figure 3: Snapshots of each scenario in the FoAM benchmark. The middle snapshot provides an overview of the simulated dual-arm system we developed in MuJoCo (Todorov, Erez, and Tassa 2012). The tasks in the benchmark are divided into five categories for evaluation. The objects in these scenarios are sourced from (Dasari, Gupta, and Kumar 2023; Xiang et al. 2020; Mo et al. 2019; Chang et al. 2015). The FoAM benchmark offers high-degree-of-freedom simulation suites. Tutorials for creating custom environments are available on the homepage.

simulation engine, with 6 degrees of freedom (DoF) each arm and a 1-DOF parallel-jaw gripper. This system is the first open-source simulated dual-arm system with rich tasks, manipulation scenarios, and the same physics as the real UR3e robot. It will serve as a valuable tool for developing zero-shot Sim2Real robotic reinforcement learning policies, such as DextrAH-G (Lum et al. 2024). We have designed 10 distinct multi-task suites in the system. A total of 86 simulation tasks are involved, encompassing a broad range of practical skills, such as picking, moving, pushing, placing, sliding, inserting, opening, closing, and transferring. Given the community’s growing interest in goal image-based task execution, we design an interface for each suite that enables quick acquisition of real goal images before inference begins. Such functionality is not available in previous simulation benchmarks (James et al. 2020; Liu et al. 2023).

Figure 3 provides an overview of snapshots from each multi-task scenario along with their corresponding names. Each scenario includes varying numbers of subtasks. For example, the Open Cabinet Drawer scenario consists of three subtasks, with a general task instruction “Open the cabinet bottom drawer”, where “bottom” can be replaced with “middle” or “top”. The subtasks in the scenarios Transfer Color Blocks and Put Stuff to the Cabinet Bottom Drawer are dual-arm tasks, while the remaining suites involve single-arm tasks. The FoAM benchmark is a high-degree-of-freedom simulation data generator, enabling users to customize textures, colors, and even trajectories. This tool facilitates the rapid generation of high-quality simulation data tailored to

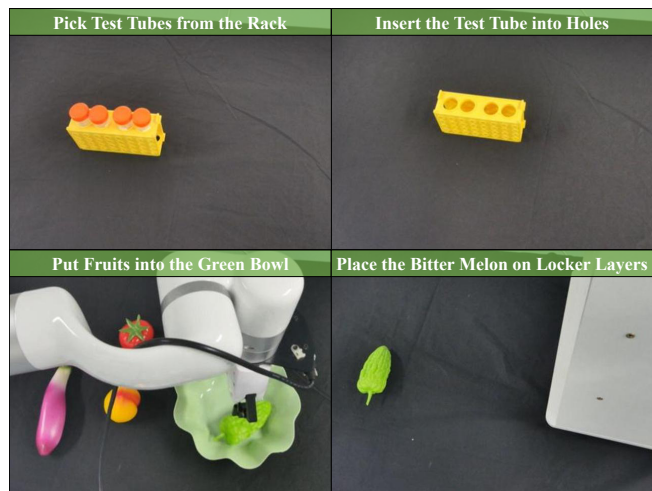


Figure 4: Snapshots of the real-world multi-task environment are captured from a static externally mounted Orbbec Femto Bolt camera. The tasks include Pick test tubes from the rack, Put fruits into the green bowl, Insert the test tube into holes (four subtasks each), and Place the bitter melon on locker layers (two subtasks). Please see the homepage for more details.

user requirements by running scripts. The FoAM benchmark will enrich the existing simulation benchmark libraries, such as RL Bench (James et al. 2020) and LIBERO (Liu et al. 2023). We expect to contribute to the development of multi-task policies in complex scenarios.

Simulation Dataset. We generated 50 episodes for each task. Before recording each demonstration, objects in the scene were randomly initialized within a specified range. The dataset was recorded at a frequency of 50 Hz, capturing the robot’s proprioceptive data, action sequences, and visual observations. The visual observations were captured solely through a head-mounted camera with a resolution of 480×640 pixels. For the single-arm tasks, even though one of the robotic arms remained inactive, we still recorded its action and proprioceptive data. As a result, the controlled action space n of the dataset was unified to 14, allowing us to accommodate all tasks within a **single MTIL policy**.

Real World Dataset. Our real-world robot system was composed of a UFACTORY xArm 7 robotic arm, a parallel-jaw gripper, a static externally mounted Orbbec Femto Bolt camera, and a wrist-mounted Intel RealSense D435 camera. To evaluate the performance of FoAM in the real world, we designed 14 tasks across four multi-task scenarios in the real world. The snapshots of four scenarios were illustrated in **Figure 4**. The dataset was collected using a custom-built, low-cost teleoperation platform inspired by Gello (Wu et al. 2024). We collected 50 episodes for each task, with the objects randomly placed on the table before data collection. The randomization was constrained within a rectangular area measuring approximately 50×60 cm. The final dataset comprised RGB data from cameras with a resolution of 480×640 each, along with joint states from both the lead-

ing and following arms, recorded at a frequency of 30 Hz.

Experiment Results

We compared FoAM against state-of-the-art open-source MTIL policies, including *Multi-task Action Chunking Transformer (MT-ACT)* (Bharadhwaj et al. 2024) and *BAKU with a deterministic policy head* (Halder, Peng, and Pinto 2024), both of which utilize only task instruction as the goal condition. We evaluated *ACT with goal images (Gimg-ACT)* as a baseline guided solely by the goal image. The baseline of *ACT with both task instruction and goal image* served as the multi-modal goal-conditioned policy, which could also be an **ablation experiment** to assess the effectiveness of the foresight augmentation (*FoMA w/o FA*). All policies were trained on the same server equipped with 8×NVIDIA RTX 4090D GPUs (24 GB each). Their inferences were deployed on the same laptop with an NVIDIA RTX 3060 GPU.

All scenarios in the FoAM benchmark were categorized into five distinct task categories and four unseen tasks. We conducted 50 test trials for each task, and the average success rates of the different policies across these categories were presented in **Table 1**. Compared to all the policies evaluated, FoAM achieved the highest success rate across all task categories. Notably, in dual-arm tasks, FoAM outperformed the second-best policy (Gimg-ACT) by 41% in success rate, with varying degrees of improvement observed in the other task categories as well.

Advantages of multi-modal goal condition. The multi-modal goal condition enables the agent to perform unseen tasks with limited expert demonstrations and without data augmentation while solving the scenario ambiguity involved in the goal image-based uni-modal policy. To evaluate the generalization capabilities of interested policies, we designed four unseen tasks by modifying the Scenario *Pick Color Blocks*: green blocks were changed to purple, and blue to black. The language-based uni-modal policies (MT-ACT, BAKU) could not complete these unseen tasks, while policies involving goal image (Gimg-ACT, FoAM, FoAM w/o FA) demonstrated varying levels of generalization. We attribute this to the language-based policies that rely on text embeddings to conduct conditional responses, executing tasks based on the sub-tasks level. These policies leave the agent struggling when faced with unseen task instructions. In contrast, image-based uni-modal policies take actions based on pixel information, aligning goal information with a single episode, which would provide a more fine-grained representation on the episode level. This enables the policy to focus on precise pixel differences between visual features and goals, allowing it to tackle unseen tasks.

The goal-image-based uni-modal policy suffers from scenario ambiguity, making it difficult to align tasks with intentions. For example, the scenarios in Block-based category, such as *Transfer Color Blocks* and *Pick Color Blocks*, have the same initial observation and goal images. The tasks, such as *Transfer the right yellow block* and *Pick the right yellow block*, have the same initial observation and goal image, which is the visual observation after the right yellow block disappears in the field of view. Therefore, Gimg-ACT has the worst performance in Block-based tasks compared

to other policies. The multi-modal goal condition of FoAM provides reliable execution conditions for each task while retaining the generalization ability offered by goal images.

Advantages of Foresight Augmentation. To discuss the benefits brought by FA, we designed the ablation item FoAM w/o FA. A comparative analysis of success rates across all five task categories reveals a significant performance enhancement in task execution after integrating the FA. We attribute this improvement to the policy’s ability to learn more expressive representations by aligning actions with their consequences, facilitating more accurate task execution. According to **Table 1**, our experiment results indicate that the ability to perform unseen tasks follows the order FoAM>Gimg-ACT>FoAM w/o FA, while language-based goal-conditioned policies (MT-ACT and BAKU) fail to generalize to unseen tasks. These results suggest that incorporating task instructions into the multi-modal goal condition influences the generalization capability conferred by the goal image. However, the integration of FA mitigates the negative impact of task instructions on generalization while enhancing the acquisition of more expressive representations, ultimately improving the policy’s generalization performance.

Real-world experiment. Based on the performance of the policies that were of interest in the FoAM benchmark, we strategically selected MT-ACT, Gimg-ACT, FoAM, and FoAM w/o FA for real-world deployment. In the real-world experiments, for each scenario, we randomly initialized ten different locations and sequentially executed the tasks associated with each scenario. The performance of these MTIL policies in real-world scenarios was summarized in **Table 2**.

Each policy experiences a notable performance decline when deployed in real-world environments. We attribute this to the inherent randomness of expert demonstrations, the distribution differences in multiple tasks, and the variability present in the dynamic real world, all of which complicate the processes of learning and inference. Furthermore, we utilize only two cameras, head and wrist, and the manipulated objects are randomly initialized within a large workspace, increasing the challenge for the agent to learn skills. Scenarios I and II demand higher accuracy from the actions, as the test tubes and racket holes are closely located, making them sensitive to robot misoperations. In Scenario III, the four fruits and the green bowl are placed randomly, resulting in complex visual observations. In contrast, Scenario IV is relatively simple for robot execution, involving just one large manipulated object and a well-defined goal space. However, it introduces ambiguity, making Gimg-ACT unable to reliably execute tasks consistent with its intention.

Consistent with the simulation results, FoAM demonstrates superior performance across nearly all four real world scenarios. To evaluate policy performance on unseen tasks, we replaced the eggplant in Scenario III with a carambola. MT-ACT still struggled to achieve any success in the new tasks. The other three policies exhibited varying degrees of generalization, and FoAM achieved the highest success rate.

VLM-FoAM Joint Inference

To improve the agent’s autonomy in acquiring the goal image, we conducted a joint inference experiment in Scenario

Policy	Model size	Dual-Arm (12 tasks)	Block-based (40 tasks)	Cabinet-based (14 tasks)	Locker-based (8 tasks)	Others (8 tasks)	Unseen Tasks (4 tasks)
BAKU	11M	31%	47%	52%	25%	17%	0
MT-ACT	86M	33%	71%	50%	32%	50%	0
Gimg-ACT	84M	45%	39%	52%	23%	28%	45%
FoAM w/o FA	86M	36%	81%	55%	51%	49%	11%
FoAM (Ours)	86M	86%	91%	75%	85%	71%	66%

Table 1: Performance of interested policies in FoAM benchmark. In the table, the first column lists the names of the policies included in the evaluation, and the second column provides the model size of each policy. The subsequent six columns report the average success rates of the respective policies across five task categories and four unseen tasks.

Policy	Scenario I	Scenario II	Scenario III	Scenario IV	Unseen Task
	MT-ACT	6/40	10/40	10/40	10/20
Gimg-ACT	7/40	8/40	11/40	-	1/10
FoAM w/o FA	7/40	13/40	13/40	12/20	1/10
FoAM (Ours)	11/40	11/40	17/40	14/20	3/10

Table 2: Performance of interested policies in real-world scenarios. Scenario I: Pick the first (second, third, fourth) test tube from the rack. II: Insert the test tube into the first (second, third, forth) hole. III: Put the eggplant (bitter melon, peach, tomato) in the green bowl. IV: Place the bitter melon on the bottom (middle) locker layer.

Policy	Bitter Melon	Eggplant	Peach	Tomato
FoAM	5/10	7/10	2/10	2/10
VLM-FoAM	7/10	7/10	3/10	3/10

Table 3: Performance comparison in success rates of the FoAM and VLM-FoAM policies in Scenario III.

III. Two policies were trained using data exclusively from Scenario III: FoAM, trained with the last frame of demonstration and evaluated with real goal images, and VLM-FoAM, trained and evaluated with goal images generated by VLM. The experiment results are shown in **Table 3**.

Due to their shapes, Peach and Tomato are prone to rolling, which is difficult for the robot to grasp, leading to task failure. In contrast, Bitter Melon and Eggplant are more easily grasped. Although the experimental results have a small gap, we observed that VLM-FoAM exhibits more robust performance during the experiments. We attribute this to the deep semantic information retained in the images generated by VLM, which helps prevent the model from overfitting when working with small datasets. Furthermore, the goal images generated by VLM maintain a consistent overall style. This style uniformity ensures that goal images generated at different times share similar features, enhancing the robot’s ability to adapt to the dynamic real world, thereby improving task execution reliability. Additionally, with the introduction of VLM, the agent can autonomously and efficiently acquire the goal image, with a 480×640 pixel goal image being obtained in an average of 4 seconds.

Although VLM brings benefits to robotic manipulation,

we found some limitations during the experiment. When executing tasks in scenarios with complex workspaces and small manipulated objects, such as Scenarios I and II, the generated goal images exhibited instability in semantic alignment. For example, when given the instruction *Insert the test tube into the second hole*, the generated image sometimes corresponded to an incorrect but visually similar placement, such as *Inserting the test tube into the first/third hole*. We attribute these discrepancies to the limited dataset and the limitations of current VLMs in processing fine-grained pixel-level details. Specifically, the test tube’s features constitute only a small portion of the overall visual input, posing a significant challenge for the model in accurately distinguishing such fine-grained spatial relationships.

Robustness Analysis

We conducted an in-depth exploration of FoAM, focusing on two key aspects: external disturbance, and reactivity. Relevant videos can be viewed on the homepage.

External Disturbance. Despite the introduction of additional objects to disrupt the operation process, the robot was able to complete the task without difficulties.

Reactivity. During the task execution, we forcibly removed the object from the gripper. In response, the robot exhibited the ability to attempt re-grasping the object and ultimately complete the task.

Conclusion and Future Work

In this work, we introduced FoAM, a novel multi-modal goal-conditioned policy designed to enhance the performance of multi-task policies and address the limitations of uni-modal ones. Inspired by human behavior perception, FoAM improves agent performance through foresight guidance, mimicking expert actions, and coupling predicted actions with their visual consequences. In our benchmark and various real-world scenarios, FoAM achieves improvements of up to 41% in success rate compared with previous policies. Meanwhile, FoAM exhibits certain limitations in real-world Scenarios I and II, which involve high precision requirements. To address this, we will explore refining long-horizon tasks by generating fine-grained intermediate goal images to serve as consequence controllers. By leveraging them, we seek to reduce cumulative errors during manipulations and improve the agent’s execution reliability.

References

- Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736.
- Bharadhwaj, H.; Gupta, A.; and Tulsiani, S. 2023. Visual affordance prediction for guiding robot exploration. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 3029–3036. IEEE.
- Bharadhwaj, H.; Vakil, J.; Sharma, M.; Gupta, A.; Tulsiani, S.; and Kumar, V. 2024. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 4788–4795. IEEE.
- Black, K.; Nakamoto, M.; Atreya, P.; Walke, H.; Finn, C.; Kumar, A.; and Levine, S. 2023. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*.
- Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Chen, X.; Choromanski, K.; Ding, T.; Driess, D.; Dubey, A.; Finn, C.; et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.
- Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; et al. 2022. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*.
- Brooks, T.; Holynski, A.; and Efros, A. A. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18392–18402.
- Buamane, T.; Kobayashi, M.; Uranishi, Y.; and Takemura, H. 2024. Bi-ACT: Bilateral Control-Based Imitation Learning via Action Chunking with Transformer. *arXiv preprint arXiv:2401.17698*.
- Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.
- Chi, C.; Feng, S.; Du, Y.; Xu, Z.; Cousineau, E.; Burchfiel, B.; and Song, S. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*.
- Dasari, S.; Gupta, A.; and Kumar, V. 2023. Learning Dexterous Manipulation from Exemplar Object Trajectories and Pre-Grasps. In *IEEE International Conference on Robotics and Automation 2023*.
- Dezfouli, A.; and Balleine, B. W. 2013. Actions, action sequences and habits: evidence that goal-directed and habitual action control are hierarchically organized. *PLoS computational biology*, 9(12): e1003364.
- Ding, Y.; Florensa, C.; Abbeel, P.; and Phielipp, M. 2019. Goal-conditioned imitation learning. *Advances in neural information processing systems*, 32.
- Fang, Z.; Yang, M.; Zeng, W.; Li, B.; Yue, J.; Ding, Z.; Li, X.; and Lu, Z. 2024. Egocentric Vision Language Planning. *arXiv preprint arXiv:2408.05802*.
- Fu, Z.; Zhao, T. Z.; and Finn, C. 2024. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*.
- Gadre, S. Y.; Wortsman, M.; Ilharco, G.; Schmidt, L.; and Song, S. 2023. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23171–23181.
- Ha, H.; Florence, P.; and Song, S. 2023. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, 3766–3777. PMLR.
- Haldar, S.; Peng, Z.; and Pinto, L. 2024. Baku: An efficient transformer for multi-task policy learning. *Advances in Neural Information Processing Systems*, 37: 141208–141239.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Huang, W.; Abbeel, P.; Pathak, D.; and Mordatch, I. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, 9118–9147. PMLR.
- Huang, W.; Wang, C.; Zhang, R.; Li, Y.; Wu, J.; and Fei-Fei, L. 2023. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*.
- James, S.; Ma, Z.; Arrojo, D. R.; and Davison, A. J. 2020. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2): 3019–3026.
- Jiang, Y.; Gupta, A.; Zhang, Z.; Wang, G.; Dou, Y.; Chen, Y.; Fei-Fei, L.; Anandkumar, A.; Zhu, Y.; and Fan, L. 2022. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094*, 2(3): 6.
- Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Sanke, P.; et al. 2024. OpenVLA: An Open-Source Vision-Language-Action Model. *arXiv preprint arXiv:2406.09246*.
- Kingma, D. P. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Kroemer, O.; Niekum, S.; and Konidaris, G. 2020. A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms. *arXiv:1907.03146*.
- Liu, B.; Zhu, Y.; Gao, C.; Feng, Y.; Liu, Q.; Zhu, Y.; and Stone, P. 2023. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36: 44776–44791.
- Lum, T. G. W.; Matak, M.; Makoviychuk, V.; Handa, A.; Allshire, A.; Hermans, T.; Ratliff, N. D.; and Van Wyk, K. 2024. Dextrah-g: Pixels-to-action dexterous arm-hand grasping with geometric fabrics. *arXiv preprint arXiv:2407.02274*.

- Mandlekar, A.; Nasiriany, S.; Wen, B.; Akinola, I.; Narang, Y.; Fan, L.; Zhu, Y.; and Fox, D. 2023. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*.
- Mishra, U. A.; Xue, S.; Chen, Y.; and Xu, D. 2023. Generative skill chaining: Long-horizon skill planning with diffusion models. In *Conference on Robot Learning*, 2905–2925. PMLR.
- Mo, K.; Zhu, S.; Chang, A. X.; Yi, L.; Tripathi, S.; Guibas, L. J.; and Su, H. 2019. PartNet: A Large-Scale Benchmark for Fine-Grained and Hierarchical Part-Level 3D Object Understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Nasiriany, S.; Pong, V.; Lin, S.; and Levine, S. 2019. Planning with goal-conditioned policies. *Advances in neural information processing systems*, 32.
- Ni, F.; Hao, J.; Wu, S.; Kou, L.; Liu, J.; Zheng, Y.; Wang, B.; and Zhuang, Y. 2024. Generate Subgoal Images before Act: Unlocking the Chain-of-Thought Reasoning in Diffusion Model for Robot Manipulation with Multimodal Prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13991–14000.
- Padalkar, A.; Pooley, A.; Jain, A.; Bewley, A.; Herzog, A.; Irpan, A.; Khazatsky, A.; Rai, A.; Singh, A.; Brohan, A.; et al. 2023. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*.
- Perez, E.; Strub, F.; De Vries, H.; Dumoulin, V.; and Courville, A. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.
- Pertsch, K.; Rybkin, O.; Ebert, F.; Zhou, S.; Jayaraman, D.; Finn, C.; and Levine, S. 2020. Long-horizon visual planning with goal-conditioned hierarchical predictors. *Advances in Neural Information Processing Systems*, 33: 17321–17333.
- Reed, S.; Zolna, K.; Parisotto, E.; Colmenarejo, S. G.; Novikov, A.; Barth-Maron, G.; Gimenez, M.; Sulsky, Y.; Kay, J.; Springenberg, J. T.; et al. 2022. A generalist agent. *arXiv preprint arXiv:2205.06175*.
- Reuss, M.; Li, M.; Jia, X.; and Lioutikov, R. 2023. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*.
- Rivera, C. G.; Handelman, D. A.; Ratto, C. R.; Patrone, D.; and Paulhamus, B. L. 2022. Visual goal-directed meta-imitation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3767–3773.
- Rivero-Moreno, Y.; Echevarria, S.; Vidal-Valderrama, C.; Pianetti, L.; Cordova-Guilarte, J.; Navarro-Gonzalez, J.; Acevedo-Rodríguez, J.; Dorado-Avila, G.; Osorio-Romero, L.; Chavez-Campos, C.; et al. 2023. Robotic surgery: a comprehensive review of the literature and current trends. *Cureus*, 15(7).
- Shi, L. X.; Hu, Z.; Zhao, T. Z.; Sharma, A.; Pertsch, K.; Luo, J.; Levine, S.; and Finn, C. 2024. Yell at your robot: Improving on-the-fly from language corrections. *arXiv preprint arXiv:2403.12910*.
- Sohn, K.; Lee, H.; and Yan, X. 2015. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28.
- Sridhar, A.; Shah, D.; Glossop, C.; and Levine, S. 2024. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 63–70. IEEE.
- Sundaresan, P.; Vuong, Q.; Gu, J.; Xu, P.; Xiao, T.; Kirmani, S.; Yu, T.; Stark, M.; Jain, A.; Hausman, K.; Sadigh, D.; Bohg, J.; and Schaal, S. 2024. RT-Sketch: Goal-Conditioned Imitation Learning from Hand-Drawn Sketches. *arXiv:2403.02709*.
- Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5026–5033. IEEE.
- Vaswani, A. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.
- Wang, C.; Fang, H.; Fang, H.-S.; and Lu, C. 2024. RISE: 3D Perception Makes Real-World Robot Imitation Simple and Effective. *arXiv preprint arXiv:2404.12281*.
- Wu, P.; Shentu, Y.; Yi, Z.; Lin, X.; and Abbeel, P. 2024. GELLO: A General, Low-Cost, and Intuitive Teleoperation Framework for Robot Manipulators. *arXiv:2309.13037*.
- Xiang, F.; Qin, Y.; Mo, K.; Xia, Y.; Zhu, H.; Liu, F.; Liu, M.; Jiang, H.; Yuan, Y.; Wang, H.; Yi, L.; Chang, A. X.; Guibas, L. J.; and Su, H. 2020. SAPIEN: A SimulATED Part-based Interactive ENvironment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yang, Z.; Li, L.; Lin, K.; Wang, J.; Lin, C.-C.; Liu, Z.; and Wang, L. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1): 1.
- Yokoyama, N.; Ha, S.; Batra, D.; Wang, J.; and Bucher, B. 2024. Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 42–48. IEEE.
- Zhang, K.; Mo, L.; Chen, W.; Sun, H.; and Su, Y. 2024. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36.
- Zhao, T. Z.; Kumar, V.; Levine, S.; and Finn, C. 2023. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*.
- Zhen, H.; Qiu, X.; Chen, P.; Yang, J.; Yan, X.; Du, Y.; Hong, Y.; and Gan, C. 2024. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*.
- Zhi, P.; Zhang, Z.; Han, M.; Zhang, Z.; Li, Z.; Jiao, Z.; Jia, B.; and Huang, S. 2024. Closed-loop open-vocabulary mobile manipulation with gpt-4v. *arXiv preprint arXiv:2404.10220*.
- Zhu, Z.; and Hu, H. 2018. Robot learning from demonstration in robotic assembly: A survey. *Robotics*, 7(2): 17.