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

Litao Liu[1], Wentao Wang[2], Yifan Han[3], Zhuoli Xie[1], Pengfei Yi[3], Junyan Li[3], Yi Qin[1], Wenzhao Lian[3*]

Abstract— **Multi-task imitation learning (MTIL) has shown significant potential in robotic manipulation by enabling agents to perform various tasks using a unified policy. This simplifies the policy deployment and enhances the agent's adaptability across different contexts. However, key challenges remain, such as maintaining action reliability (e.g., avoiding abnormal action sequences that deviate from nominal task trajectories), distinguishing between similar tasks, and generalizing to unseen scenarios. To address these challenges, we introduce the Foresight-Augmented Manipulation Policy (FoAM), an innovative MTIL framework. FoAM not only learns to mimic expert actions but also predicts the visual outcomes of those actions to enhance decision-making. Additionally, it integrates multi-modal goal inputs, such as visual and language prompts, overcoming the limitations of single-conditioned policies. We evaluated FoAM across over 100 tasks in both simulation and real-world settings, demonstrating that it significantly improves IL policy performance, outperforming current state-of-the-art IL baselines by up to 41% in success rate. Furthermore, we released a simulation benchmark for robotic manipulation, featuring 10 task suites and over 80 challenging tasks designed for multi-task policy training and evaluation. See project homepage https://projFoAM.github.io/ for project details.**

I. INTRODUCTION

One of the main goals in robot learning is to develop a general-purpose agent capable of performing various tasks based on user commands. For manipulation tasks, multi-task imitation learning (MTIL) serves as a key approach that enables agents to learn different tasks from expert demonstrations, eliminating the need for complex, hard-coded solutions or reward functions and leading to efficient and generalizable policies. However, in practical settings, achieving both generalization and reliability in MTIL remains a key challenge [1]. The agent must develop task-agnostic skills to generalize to new tasks and environments, while also capturing task-specific details to ensure reliable execution for individual tasks [2], [3].

Previous research has shown that task adaptation in MTIL can be achieved by incorporating goal conditions into multi-task policy training [4]–[11]. However, the low reliability of multi-task policies remains a persistent challenge. Existing MTIL policies that align robotic actions with expert actions based on goal conditions often fail to reason about these ambiguities and variations in expert demonstration data, severely impacting the agents' performance on individual tasks. Meanwhile, single-goal conditioned policies [9], [10],

[12]–[17] come with its own limitations. For instance, policies conditioned on language instructions struggle to generalize to unseen tasks without data augmentation. While policies conditioned on goal images offer fine-grained guidance, they frequently encounter ambiguities in task activation and necessitate human intervention to accurately acquire and interpret the goal images. For instance, when the robot was tasked with placing an object into a multi-compartment locker, the resulting goal image (an object that was in the initial image disappeared in the goal image) was ambiguous due to occlusion, making it impossible to determine which specific compartment the object was placed in (see Figure 4).

In this paper, we introduce the Foresight-Augmented Manipulation Policy (FoAM), a novel multi-task imitation learning policy designed to generate fine-grained actions while considering their consequences. This approach is inspired by how humans perform tasks, refining actions by comparing expected outcomes with the intended goal until the desired result is achieved [18]. Similarly, FoAM processes observation inputs, task prompts, and goal images to generate robotic actions and predict embeddings that represent the outcomes of those actions. During training, we apply an action loss to refine the policy's behavior and introduce a foresight loss to ensure the policy accounts for the consequences of its actions. This foresight 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. Additionally, FoAM leverages a fine-tuned vision-language model (VLM) [19] to autonomously generate goal images, enhancing the agent's autonomy and improving its ability to generalize to unseen tasks and scenarios. Our approach demonstrates significant effectiveness, with evaluations across more than 100 tasks in both simulation and real world. FoAM 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:

1) We introduce the Foresight-Augmented Manipulation Policy (FoAM), a novel multi-task IL policy with the ability to reason about the consequences of its actions.
2) We integrate a Vision-Language Model (VLM) as the Goal Imagination Module for FoAM, fusing language prompts with goal images, enhancing its ability to generalize to unseen tasks and scenarios.
3) We opensource a simulation benchmark with 10 task suites and over 80 tasks, alongside a real-world dataset comprising 14 tasks. We demonstrate that FoAM achieves state-of-the-art performance on these tasks.

II. RELATED WORK

Goal-Guided Learning for Robotic Manipulation. In recent years, significant progress has been made in single-task learning policies [20]–[24]. However, to enable a wider adoption, intelligent robots must be equipped with the ability to adapt to diverse tasks and complete them effectively. Among current MTIL approaches, language-guided manipulation 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 data [6]–[10], [25]–[31]. Despite the promising initial success, we found that language-conditioned policies often struggle with unseen tasks without sufficient data or extensive additional data augmentation. In parallel, other approaches have introduced goal images as task conditions [10], [12], [13], [32]–[34]. Compared to language inputs, images encapsulate richer information, enabling stronger generalization capabilities, and even allowing agents to perform zero-shot tasks [13]. However, goal images are susceptible to task ambiguity, where visually identical outcomes can be produced by different tasks, causing incorrect task activation.

Moreover, the need to collect goal images from humans reduces agent autonomy. To leverage the strengths of both images and language, recent work has explored multi-modal prompts [14], [15], [34]–[36]. These methods extract entities from both language and image prompts, then compose a multi-modal prompt embedding based on predefined templates. In contrast, our approach leverages vision-language models to directly generate a goal image with high-dimensional semantic information. This generated goal image, along with the task prompt, is processed to further infer actions and predict future states. By doing so, our method addresses the limitations of goal-conditioned policies that rely solely on language prompts, goal images, or predefined templates.

Agents with Vision Language Models. In recent years, Vision-Language Models (VLMs) have been introduced to robotics [30], [36]–[43], enabling more complex visual reasoning and multimodal tasks. Meanwhile, in the community of image editing, VLMs also demonstrated the ability to understand language prompts, edit real-world images and produce highly realistic visual effects [33], [44]. [14], [33], [45]–[47] further validates that edited images generated via VLMs can be interpreted by robotic agents, where the generated goal images are directly used as a task activation condition. In contrast, we integrate VLMs seamlessly into our framework not only during the inference stage, but also in model training. Specifically, the VLM-generated goal images serve as the "labels" to compute the reconstruction loss for the foresight augmentation module (Section III-C), coupling the action policy learning and action result prediction.

III. METHOD

We seek to develop an imitation learning policy with strong generalization and robustness, empowering the agent to effectively complete a variety of tasks. To achieve this, we introduce FoAM, a novel multi-modal goal-conditioned policy learning framework. In the following sections, we will provide an overview of the FoAM in III-A, detail the fine-tuning of a cutting-edge visual-language model to generate goal images for FoAM in III-B, propose the foresight-augmented module in III-C, and introduce the implementation details of FoAM in III-D.
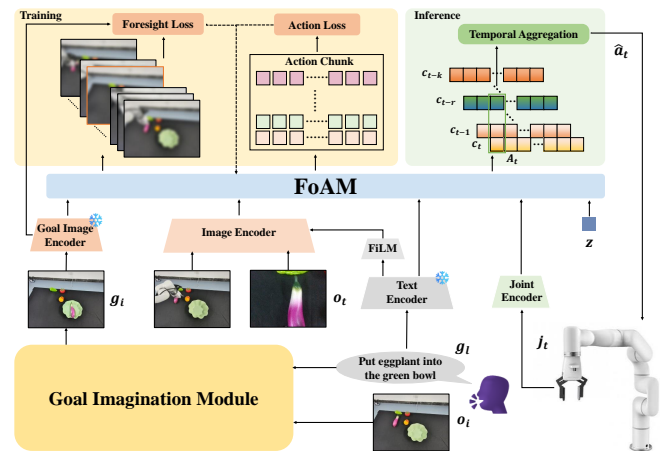

Fig. 1: *Training and inference pipelines of FoAM*. The inputs remain the same during both training and inference. During training, actions and their consequences are predicted and compared with ground truth to update FoAM's parameters. During inference, the trained policy predicts the action $\hat{a}_t$.

A. Pipeline Overview

The pipeline of FoAM is illustrated in Figure 1. FoAM is a transformer-based [48] policy that inherits the architecture of prior works [20] and is trained as a conditional variational autoencoder (CVAE) [49], [50]. The process begins with the user providing a task prompt g_l to the agent based on the current observation, which is then fed into FoAM as a language goal condition through a pre-trained text encoder [51]. Concurrently, the task prompt, such as *Put eggplant into the green bowl*, is input into the Goal Imagination Module along with the initial observation o_i. The generated imaginary goal image or human collected real goal image g_i serves as an additional condition. By integrating the agent's proprioception j_t, visual observations o_t, and the latent style variable z encoded from the action and joint data, the multi-goal conditional policy $\pi_\theta(\hat{a}_{t:t+k}|o_t, j_t, g_i, g_l, z)$ is learned through gradient descent during training. During inference, action chunks are predicted and robust actions are produced through temporal aggregation [20].

B. Fine-tuned Goal Imagination Module

We chose InstructPix2Pix (Ip2p) [19] as our goal imagination module, utilizing approximately 20,000 pairs of training data. Of these, 16,000 pairs were derived from the cleaning robot expert demonstrations provided by RT-1 [6], [7], [25]. In this process, the first and last frames of the demonstrations were used as the original and edited images, respectively, with the corresponding task name serving as the instruction. Given that many of the demonstration datasets

contained perturbations in the final frames caused by robot arm movements, we undertook a detailed data cleaning procedure to remove noise and ensure the training data quality. Additionally, we incorporated over 4,000 data pairs sourced from our own simulation and real-world datasets (see Section IV-A for details). We fine-tuned the model for 500 epochs on a single NVIDIA H100 GPU, a process that required approximately 3 days. During the image generation stage, with the model weights pre-loaded, processing each initial observation of size $480 \times 640 \times 3$ took about 4 seconds. Figure 2 presents a selection of initial observations alongside their corresponding imagined goal images, captured during VLM-FoAM joint inference experiments (see Section IV-C for details) in real-world. These results highlight the model's ability to generate realistic and contextually visual effects based on the given initial observations and task prompts. We anticipate that these imagined visual scenes will help address the limitations of single-goal conditional policies, and allow the robot autonomously determine goal conditions, improving the robot's generalization performance.

Fig. 2: *Inference demonstration of the fine-tuned goal imagination module.* The leftmost image illustrates the initial observation, while the next four images represent the imagined goal images generated based on the initial observation and the task prompt provided at the top. Please visit the project homepage for more examples.

C. Foresight-augmented Module

Humans possess exceptional cognitive abilities when it comes to understanding and interacting with external objects and events. This cognitive prowess enables individuals to not only perceive the current environment but also to anticipate future outcomes based on their actions. When tasked with a specific objective, humans often mentally envision the goal image even before initiating the task. Inspired by this human capability, we developed a foresight-augmented (FA) module to endow agents with a similar foresight mechanism. The FA enables the agent to concurrently comprehend both the actions it takes and the subsequent outcomes these actions will produce. By integrating this foresight into the agent's decision-making process, we significantly improve the agent's overall manipulation task performance.

We train FoAM as a CVAE, utilizing an encoder similar to those in [9], [20], [22], [52] to generate the latent variable $z \sim q_\phi(z|a_{t:t+k}, j_t)$. The decoder is defined as policy $\pi_\theta(\hat{a}_{t:t+k}, \hat{f}_{t:t+k}|o_t, j_t, g_i, g_l, z)$, which predicts a $k \times n$-dimensional action chunk $\hat{a}_{t:t+k}$ and foresight sequence $\hat{f}_{t:t+k}$ based on current observation and conditions, where k represents the hyperparameter chunk size and n denotes the dimension of the agent's controlled action space. To enhance the agent's ability to interpret and react to extended temporal contexts, we strategically increase the value of k, thereby broadening the agent's vision of foresight. This adjustment is particularly important in the context of the multi-task dataset,

where the length of each expert demonstration varies. For shorter tasks, we extend the data by replicating the final frame up to the maximum time step T of all demonstrations, setting k to be close to T. As the action chunk is being predicted, the FA module envisions k potential foresight scenes, and selects the frame $\hat{g}_i = \hat{f}_{t:t+k}[k-t]$ that is temporally aligned with the goal image. This selected frame is then compared against the goal image g_i to compute the foresight loss $L_{\text{foresight}}$. This process successfully emulates the strong cognitive abilities humans exhibit when approaching tasks, and we demonstrated in experiments that the FA module significantly enhances the agent's task performance.

D. 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 input. FoAM is implemented using an ACT-like architecture [20] with the CVAE framework. The language conditional embedding is obtained by pre-training the language encoder [51] to produce a 384-dimensional feature, which is subsequently projected to 512 dimensions through an MLP. Visual observations of size $480 \times 640 \times 3$ are encoded using ResNet18 [53], with a FiLM conditional layer [54] applied to each view encoding, ensuring robust task activation performance in multi-task scenarios [9]. The visual observations are finally transformed into a $(300 \times n) \times 512$ feature sequence, where n denotes the number of views used. 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 transformer encoder fuses features from different modalities, and the decoder outputs the predicted action chunk $\hat{a}_{t:t+k}$ and k envisioned foresight scenes $\hat{f}_{t:t+k}$.

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

During inference, the FA module is discarded, and the policy at this stage is represented by $\pi_\theta(\hat{a}_{t:t+k}|o_t, j_t, g_i, g_l, z)$. 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 [9], [10], [20], we apply exponential temporal aggregation to produce smoother motion trajectories. Unlike previous work, we introduce the hyperparameter r to eliminate the equality constraint on chunk size k and the temporal aggregation range r. This is particularly crucial for deploying FoAM in real-world scenarios, as it allows flexibly adjusting the smoothing range according to the characteristics of dif-

ferent tasks, and optimizing task performance. The inference code is shown in Algorithm 2. There are approximately $160M$ parameters in the training and around $80M$ in the inference.

Algorithm 1 FoAM Training

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

1: Each episode includes a_t, j_t, o_t, g_l 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 $\pi_\theta(\hat{a}_{t:t+k}, \hat{f}_{t:t+k}|o_t, j_t, g_i, g_l, z)$
4: **for** each batch $i = 1, 2, \ldots$ **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 $\pi_\theta(\hat{a}_{t:t+k}, \hat{f}_{t:t+k}|o_t, j_t, g_i, g_l, z)$
8:     $\mathcal{L}_{\text{action}} = L_1(\hat{a}_{t:t+k}, a_{t:t+k})$
9:     $\mathcal{L}_{\text{foresight}} = Huber(\hat{g}_i, g_i)$, 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**

Algorithm 2 FoAM Inference

Require: trained policy $\pi_\theta(\hat{a}_{t:t+k}|o_t, j_t, g_i, g_l, z)$, 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 = 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**

IV. EXPERIMENTS

Our experiments focus on investigating the following questions:

(a) How does FoAM perform, and how does it outperform the baseline across multiple tasks?

(b) What advantages does FoAM provide compared to a single goal-conditioned policy?

(c) How effectively does FoAM handle unseen tasks without any data augmentation?

(d) When FoAM and the VLM are used jointly during inference, how does FoAM perform differently when guided by an imagined goal image versus a real one?

(e) A robust imitation learning policy is essential for achieving sustainable scalability. How well does FoAM respond to external disturbance?

A. Data Collection

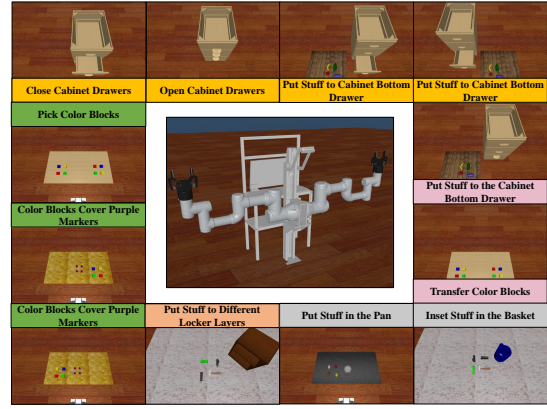**FoAM Benchmark**: We developed a simulated dual-

Fig. 3: *Snapshots of each scenario in the FoAM benchmark.* The middle snapshot provides an overview of the dual-arm robot we developed in MuJoCo [55]. The tasks in the benchmark are categorized into five groups for performance analysis: pink for dual-arm tasks, yellow for cabinet-based tasks, green for color-block-based tasks, orange for locker-based tasks, and gray for other tasks. The objects in these scenarios are sourced from [56]–[59]. The FoAM benchmark offers high-degree-of-freedom simulation suites. Tutorials for creating custom environments are available on the project homepage.
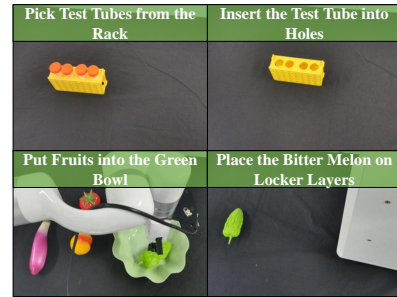

Fig. 4: *Snapshots of the real-world multi-task environment are captured from static externally mounted Orbbec Femto Bolt camera.* We use a UFACTORY xArm 7 robot, a parallel-jaw gripper, a static externally mounted Orbbec Femto Bolt camera, and a wrist mounted Intel Realsense D435 camera, to evaluate multi-task policies across 4 real scenarios, comprising a total of 14 tasks. The tasks include: picking test tubes from the rack, placing fruits into the green bowl, inserting a test tube into holes (each with four subtasks), and placing the bitter melon on locker layers (with two subtasks). For videos and more details, please refer to the project homepage.

arm robotic system, with each arm possessing 6 degrees of freedom (DoF) and a 1-dof parallel-jaw gripper, closely replicating a commonly used UR3e robot. This system was implemented with the MuJoCo physics simulation engine [55], and we have open-sourced 10 distinct multi-task suites involving this robot. A total of 86 simulation tasks were designed, encompassing a broad range of practical skills, such as picking, moving, pushing, placing, sliding, inserting, opening, closing, and transferring [25]. Figure 3 provides an overview of snapshots from each multi-task environment along with their corresponding scenario names. Each multi-task simulation environment includes varying numbers of subtasks. For example, the *Open Cabinet Drawer* scenario consists of three subtasks, with a general task prompt "Open the cabinet *bottom* drawer", where "*bottom*" can be replaced with *middle* or *top*. The subtasks in the scenarios *Transfer*

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 [10]	11M	31%	47%	52%	25%	17%	0
MT-ACT [9]	86M	33%	71%	50%	32%	50%	0
Gimg-ACT	84M	45%	39%	52%	23%	28%	45%
FoAM (Ours)	86M	**86%**	**91%**	**75%**	**85%**	**71%**	**66%**
FoAM w/o FA	86M	36%	81%	55%	51%	49%	11%

TABLE I: *Performance of multi-task policies in our 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 (See project homepage for the evaluation metrics of each task) of the respective policies across different task categories: dual-arm tasks, color block-based tasks, cabinet-based tasks, locker-based tasks, other tasks (including "Put something in the pan" and "Insert something in the basket"), and unseen tasks.

Policy	Scenario I	Scenario II	Scenario III	Scenario IV	Unseen Task
MT-ACT	6/40	10/40	10/40	10/20	0/10
Gimg-ACT	7/40	8/40	11/40	-	1/10
FoAM (Ours)	**11/40**	11/40	**17/40**	**14/20**	**3/10**
FoAM w/o FA	7/40	**13/40**	13/40	12/20	1/10

TABLE II: *The success rate of multi-task policies in real-world scenarios.* Scenario I corresponds to the task: "Pick the *first (second, third, forth)* test tube from the rack." Scenario II corresponds to: "Insert the test tube into the *first (second, third, forth)* hole." Scenario III involves: "Put the *eggplant (bitter melon, peach, tomato)* in the green bowl." Scenario IV refers to the task: "Place the bitter melon on the *bottom (middle)* layer of the locker."

Color Blocks and Dual Arm: *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 action trajectories for tasks. This tool facilitates the rapid generation of high-quality simulation data tailored to user requirements. We expect that FoAM contributes to the development of multi-task policies in complex environments.

Simulation Dataset: We generate 50 expert demonstrations for each task to evaluate the performance of FoAM versus other multi-task policies. Before recording each demonstration, objects in the scene are randomly initialized within a specified range. The dataset is recorded at a frequency of 50 Hz, capturing the robot's proprioceptive data, action sequences, and visual observations. To simulate a real-world scenario where the agent interacts with users, visual observations are 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 is unified to 14, allowing us to accommodate all tasks within a single IL policy.

Real World Dataset: Our robot system is 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 real-world applications, we designed 14 tasks across four multi-task environments. The 4 real-world scenarios are illustrated in Figure 4. The dataset was collected using a custom-built, low-cost teleoperation platform inspired by Gello [60]. The leading arm, which a teleoperator directly controls, includes eight Dynamixel XL330-M288-T servos and custom 3D-printed connectors (see project homepage for details). For each task, we collected 50 expert demonstrations, 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 comprises RGB data from two cameras with a resolution of 480×640, along with joint states from both the leading and following arms, recorded at a frequency of 30 Hz.

B. Experiment Results

We compare our approach against state-of-the-art open-source multi-task policies, including *Multi-task Action Chunking Transformer (MT-ACT)* [9] and *BAKU with a deterministic policy head* [10], both of which utilize only language prompts. Additionally, we use ACT with goal images (Gimg-ACT) as a baseline guided solely by the goal image. For multi-modal prompting, ACT with both language prompts and goal images serves as a baseline policy, which can also function as an ablation policy to assess the effectiveness of the FA module (FoMA w/o FA).

To facilitate statistical analysis, all scenarios in the FoAM benchmark were categorized into five distinct task groups. We conducted 50 test trials for each subtask, and the average success rates of the different strategies across these task categories are presented in Table I.

Compared to all the policies evaluated, FoAM achieved the highest success rate across all five task categories. Notably, in the dual-arm task, FoAM outperformed the second-best policy by 41% in success rate, with varying degrees of improvement observed in the other task categories as well. To further assess the generalization capabilities of these policies, we designed 4 unseen tasks by modifying the Scenario *Pick Color Blocks*: green blocks were changed to purple, and blue blocks to black. The language-based MT policies were unable to complete these unseen tasks, while policies incorporating goal images demonstrated varying levels of generalization. We think the reason is language-based policies rely on text embeddings to activate conditional responses, classifying tasks based on broad categories. This approach leaves the agent struggling when faced with unseen task prompts, as it lacks the flexibility to adapt to unseen

tasks. In contrast, policies guided by goal images employ a more fine-grained classification approach, capturing detailed information from individual demonstrations. This enables the policy to focus more precisely on the differences between visual observations and goals, enhancing its ability to tackle unseen tasks. However, using a single goal image also presents limitations. In block-based scenarios, such as *Transfer Color Blocks* and *Pick Color Blocks*, where the initial and goal images are identical, this similarity can trigger the incorrect activation of the conditional policy. As a result, the Gimg-ACT performs significantly worse in these tasks compared to other policies. FoAM, by fusing both language and image inputs, provides stable activation conditions for each task while retaining the generalization advantages offered by goal images. When enhanced with FA, FoAM's task performance is significantly improved.

Based on the performance of MT policies in 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 multi-task policies in real-world scenarios are summarized in Table II.

Each policy experienced a notable performance decline when deployed in real-world environments. We attribute this to the inherent randomness of human demonstrations and the variability present in real-world conditions, both of which complicate the processes of learning and inference. Furthermore, unlike the experimental environments used in prior works [9], [20], our scenarios closely mirror real-world robotic applications. Specifically, we utilized only two perspectives—head and wrist—and the manipulated objects were randomly initialized within a large workspace, increasing the challenge for the agent to complete tasks effectively. Scenarios I and II demand higher precision from the policy, as the test tubes and racket holes are closely located, making it sensitive to robot misoperations. In Scenario III, the 4 fruits and the green bowl were placed randomly, resulting in complex visual observations. In contrast, Scenario IV was relatively simpler for robot execution, involving larger manipulated objects and a well-defined goal space. However, this scenario also introduced goal image ambiguity, leading it difficult for Gimg-ACT to reliably activate specific tasks. Notably, FoAM demonstrated the best performance across all 4 scenarios. To assess policy performance on unseen tasks, we replaced the eggplant in Scenario III with a carambola. MT-ACT struggled achieving any success in the new tasks without data augmentation; in contrast, the other three goal-conditioned policies exhibited varying degrees of generalization, and FoAM achieves the highest success rate.

C. VLM-FoAM Joint Inference

To improve the agent's autonomy in acquiring the goal image, we conducted a joint inference experiment in Scenario III. Two policies were employed using data exclusively from Scenario III: FoAM, trained with the last frame of demon-

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 III: *Performance comparison of the FoAM and VLM-FoAM policies in Scenario III.* The first column lists the evaluated policies, while the last four columns present the success rates for operating each corresponding object in Scenario III.

stration and evaluated with real goal images, and VLM-FoAM, trained and evaluated with goal images generated by VLM. The results of this experiment are shown in Table III.

Due to their shapes, Peach and Tomato are difficult for the robot to grasp and are prone to rolling, often leading to task failure. In contrast, Bitter Melon and Eggplant are more easily grasped. Our experiments showed that VLM-FoAM demonstrated more robust performance. 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 dynamic real-world conditions, thereby improving task activation 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.

D. Robustness Analysis

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

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

Reactiveness. 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.

V. CONCLUSION AND FUTURE WORK

In this work, we introduced FoAM, a novel multimodal goal-conditioned policy designed to enhance the performance of multi-task policies and address the limitations of single goal-conditioned approaches. Inspired by human behavior, FoAM improves agent performance by imitating expert actions while simultaneously considering the visual outcomes of those actions. In our published FoAM-benchmark and across real-world scenarios, FoAM achieved improvements of up to 41% in success rate compared with previous methods. However, FoAM exhibited certain limitations in real-world Scenarios I and II, which involve high precision requirements. To address this, we will explore to refine long-horizon tasks by generating fine-grained intermediate goal images to serve as guidance. By leveraging these intermediate visual states, we seek to reduce cumulative errors during operations and improve the agent's execution accuracy.

REFERENCES

[1] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," 2020. [Online]. Available: https://arxiv.org/abs/1907.03146

[2] Z. Zhu and H. Hu, "Robot learning from demonstration in robotic assembly: A survey," *Robotics*, vol. 7, no. 2, p. 17, 2018.

[3] Y. Rivero-Moreno, S. Echevarria, C. Vidal-Valderrama, L. Pianetti, J. Cordova-Guilarte, J. Navarro-Gonzalez, J. Acevedo-Rodríguez, G. Dorado-Avila, L. Osorio-Romero, C. Chavez-Campos *et al.*, "Robotic surgery: a comprehensive review of the literature and current trends," *Cureus*, vol. 15, no. 7, 2023.

[4] B. Fang, S. Jia, D. Guo, M. Xu, S. Wen, and F. Sun, "Survey of imitation learning for robotic manipulation," *International Journal of Intelligent Robotics and Applications*, vol. 3, pp. 362–369, 2019.

[5] H. Zhen, X. Qiu, P. Chen, J. Yang, X. Yan, Y. Du, Y. Hong, and C. Gan, "3d-vla: A 3d vision-language-action generative world model," *arXiv preprint arXiv:2403.09631*, 2024.

[6] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[7] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[8] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "Open-vla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[9] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar, "Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 4788–4795.

[10] S. Haldar, Z. Peng, and L. Pinto, "Baku: An efficient transformer for multi-task policy learning," 2024. [Online]. Available: https://arxiv.org/abs/2406.07539

[11] Y. Ding, C. Florensa, P. Abbeel, and M. Phielipp, "Goal-conditioned imitation learning," *Advances in neural information processing systems*, vol. 32, 2019.

[12] P. Sundaresan, Q. Vuong, J. Gu, P. Xu, T. Xiao, S. Kirmani, T. Yu, M. Stark, A. Jain, K. Hausman, D. Sadigh, J. Bohg, and S. Schaal, "Rt-sketch: Goal-conditioned imitation learning from hand-drawn sketches," 2024. [Online]. Available: https://arxiv.org/abs/2403.02709

[13] C. G. Rivera, D. A. Handelman, C. R. Ratto, D. Patrone, and B. L. Paulhamus, "Visual goal-directed meta-imitation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3767–3773.

[14] F. Ni, J. Hao, S. Wu, L. Kou, J. Liu, Y. Zheng, B. Wang, and Y. Zhuang, "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*, 2024, pp. 13 991–14 000.

[15] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "Vima: General robot manipulation with multimodal prompts," *arXiv preprint arXiv:2210.03094*, vol. 2, no. 3, p. 6, 2022.

[16] N. Yokoyama, S. Ha, D. Batra, J. Wang, and B. Bucher, "Vlfm: Vision-language frontier maps for zero-shot semantic navigation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 42–48.

[17] S. Nasiriany, V. Pong, S. Lin, and S. Levine, "Planning with goal-conditioned policies," *Advances in neural information processing systems*, vol. 32, 2019.

[18] A. Dezfouli and B. W. Balleine, "Actions, action sequences and habits: evidence that goal-directed and habitual action control are hierarchically organized," *PLoS computational biology*, vol. 9, no. 12, p. e1003364, 2013.

[19] T. Brooks, A. Holynski, and A. A. Efros, "Instructpix2pix: Learning to follow image editing instructions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 392–18 402.

[20] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.

[21] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *arXiv preprint arXiv:2303.04137*, 2023.

[22] T. Buamanee, M. Kobayashi, Y. Uranishi, and H. Takemura, "Bi-act: Bilateral control-based imitation learning via action chunking with transformer," *arXiv preprint arXiv:2401.17698*, 2024.

[23] U. A. Mishra, S. Xue, Y. Chen, and D. Xu, "Generative skill chaining: Long-horizon skill planning with diffusion models," in *Conference on Robot Learning*. PMLR, 2023, pp. 2905–2925.

[24] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," *arXiv preprint arXiv:2401.02117*, 2024.

[25] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, 2023.

[26] H. Ha, P. Florence, and S. Song, "Scaling up and distilling down: Language-guided robot skill acquisition," in *Conference on Robot Learning*. PMLR, 2023, pp. 3766–3777.

[27] M. Reuss, M. Li, X. Jia, and R. Lioutikov, "Goal-conditioned imitation learning using score-based diffusion policies," *arXiv preprint arXiv:2304.02532*, 2023.

[28] K. Hausman, Y. Chebotar, S. Schaal, G. Sukhatme, and J. J. Lim, "Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets," *Advances in neural information processing systems*, vol. 30, 2017.

[29] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on robot learning*. PMLR, 2022, pp. 894–906.

[30] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on robot learning*. PMLR, 2023, pp. 287–318.

[31] C. Wang, H. Fang, H.-S. Fang, and C. Lu, "Rise: 3d perception makes real-world robot imitation simple and effective," *arXiv preprint arXiv:2404.12281*, 2024.

[32] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, "Mimicgen: A data generation system for scalable robot learning using human demonstrations," *arXiv preprint arXiv:2310.17596*, 2023.

[33] Z. Fang, M. Yang, W. Zeng, B. Li, J. Yue, Z. Ding, X. Li, and Z. Lu, "Egocentric vision language planning," *arXiv preprint arXiv:2408.05802*, 2024.

[34] K. Black, M. Nakamoto, P. Atreya, H. Walke, C. Finn, A. Kumar, and S. Levine, "Zero-shot robotic manipulation with pretrained image-editing diffusion models," *arXiv preprint arXiv:2310.10639*, 2023.

[35] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg *et al.*, "A generalist agent," *arXiv preprint arXiv:2205.06175*, 2022.

[36] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in neural information processing systems*, vol. 35, pp. 23 716–23 736, 2022.

[37] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *International conference on machine learning*. PMLR, 2022, pp. 9118–9147.

[38] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 343–352.

[39] J. Pang, M. A. Lodhi, and D. Tian, "Grasp-net: Geometric residual analysis and synthesis for point cloud compression," in *Proceedings of the 1st International Workshop on Advances in Point Cloud Compression, Processing and Analysis*, 2022, pp. 11–19.

[40] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[41] W. Qi, R. T. Mullapudi, S. Gupta, and D. Ramanan, "Learning to move with affordance maps," *arXiv preprint arXiv:2001.02364*, 2020.

[42] Z. Yang, L. Li, K. Lin, J. Wang, C.-C. Lin, Z. Liu, and L. Wang, "The dawn of lmms: Preliminary explorations with gpt-4v (ision)," *arXiv preprint arXiv:2309.17421*, vol. 9, no. 1, p. 1, 2023.

[43] P. Zhi, Z. Zhang, M. Han, Z. Zhang, Z. Li, Z. Jiao, B. Jia, and

S. Huang, "Closed-loop open-vocabulary mobile manipulation with gpt-4v," *arXiv preprint arXiv:2404.10220*, 2024.

[44] K. Zhang, L. Mo, W. Chen, H. Sun, and Y. Su, "Magicbrush: A manually annotated dataset for instruction-guided image editing," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[45] H. Bharadhwaj, A. Gupta, and S. Tulsiani, "Visual affordance prediction for guiding robot exploration," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3029–3036.

[46] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8798–8807.

[47] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 63–70.

[48] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[49] D. P. Kingma, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[50] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, 2015.

[51] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "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*, 2023, pp. 23 171–23 181.

[52] L. X. Shi, Z. Hu, T. Z. Zhao, A. Sharma, K. Pertsch, J. Luo, S. Levine, and C. Finn, "Yell at your robot: Improving on-the-fly from language corrections," *arXiv preprint arXiv:2403.12910*, 2024.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[54] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[55] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[56] S. Dasari, A. Gupta, and V. Kumar, "Learning dexterous manipulation from exemplar object trajectories and pre-grasps," in *IEEE International Conference on Robotics and Automation 2023*, 2023.

[57] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su, "SAPIEN: A simulated part-based interactive environment," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[58] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "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)*, June 2019.

[59] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[60] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel, "Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators," 2024. [Online]. Available: https://arxiv.org/abs/2309.13037