IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Diffusion Trajectory-guided Policy for Long-horizon Robot Manipulation

Shichao Fan[1], Quantao Yang[4], Yajie Liu[2], Kun Wu[3], Zhengping Che[3], Qingjie Liu[2*], Min Wan[1]

***Abstract*—Recently, Vision-Language-Action models (VLA) have advanced robot imitation learning, but high data collection costs and limited demonstrations hinder generalization and current imitation learning methods struggle in out-of-distribution scenarios, especially for long-horizon tasks. A key challenge is how to mitigate compounding errors in imitation learning, which lead to cascading failures over extended trajectories. To address these challenges, we propose the Diffusion Trajectory-guided Policy (DTP) framework, which generates 2D trajectories through a diffusion model to guide policy learning for long-horizon tasks. By leveraging task-relevant trajectories, DTP provides trajectory-level guidance to reduce error accumulation. Our two-stage approach first trains a generative vision-language model to create diffusion-based trajectories, then refines the imitation policy using them. Experiments on the CALVIN benchmark show that DTP outperforms state-of-the-art baselines by 25% in success rate, starting from scratch without external pretraining. Moreover, DTP significantly improves real-world robot performance. Our project is at diffusion-trajectory-guided-policy.github.io/.**

***Index Terms*—Imitation Learning, Learning from Demonstration, Deep Learning in Grasping and Manipulation**

I. Introduction

Imitation Learning (IL) demonstrates significant potential in addressing manipulation tasks within real robotic systems, this is evidenced by its ability to acquire diverse behaviors such as preparing coffee [1] and flipping mugs [2] through learning from expert demonstrations. However, these demonstrations are often limited in coverage [3], failing to encompass every possible robot pose and environmental variation throughout long-horizon manipulation tasks (Fig. 1a)). This limitation leads to a key challenge in IL—compounding errors over extended trajectories, where small deviations from the expert trajectory accumulate, ultimately causing task failures. Additionally, robot data is often scarce compared to computer vision tasks because it requires costly and time-consuming human demonstrations. Therefore, improving the generalization capabilities of imitation learning methods using extremely limited and scarce data, given the constraints and high costs of expert demonstrations, becomes a significant challenge.

Recent research has introduced Vision-Language Action (VLA) models [4], [5], [6] that map multi-modal inputs to robot actions using Transformer structures [7]. Some approaches, like Susie [8] and others [9], [10], integrate vision and language to generate goal images or future videos, pretrained on large-scale internet datasets. RT-Trajectory [11] uses coarse trajectory sketches instead of language, while RT-H [12] breaks down complex instructions into simpler, hierarchical commands. For instance, "Close the pistachio jar" is decomposed into steps like "rotate arm right" and "move the arm forward," facilitating robot action generation. These methods transform complex instructions into goal images, replace language with trajectory sketches, or simplify instructions into directional commands, mitigating compounding errors in imitation policies, especially in long-horizon tasks. However, they often depend on manually provided trajectories or goal images, limiting flexibility in diverse or unstructured environments.

In this paper, we present a novel diffusion-based paradigm aimed at bridging the feature gap between vision-language inputs and action spaces. By generating task-relevant 2D trajectories from vision-language inputs and mapping them to the action space, our approach improves performance in long-horizon robotic manipulation tasks. Unlike robots that rely on precise instructions, humans use high-level visualizations, like imagined trajectories, to intuitively guide actions, adapting to changes and refining movements in real-time. Similarly, instructing a robot with language should allow envisioning a trajectory to guide future actions based on current observations. To achieve this, we introduce the Diffusion Trajectory-guided Policy (DTP), consisting of two stages: the Diffusion Trajectory Model (DTM) learning stage and the vision-language action policy learning stage, as depicted in Fig. 1c). The first stage generates a task-relevant trajectory using a diffusion model, which then guides the robot's manipulation policy learning in the second stage, enhancing data efficiency and generalization. The two-stage design allows the first stage to set conditions for the second. We designed the first stage as an independent helper module, easily integrated into any Transformer-based baseline as an additional input, making our approach adaptable across different models. We validated our method through extensive experiments on the CALVIN simulation benchmark [13], achieving a 25% higher average success rate than state-of-the-art baselines across various set-

Manuscript received: July 7, 2025; Accepted September 20, 2025.

This paper was recommended for publication by Editor Vincze.Markus upon evaluation of the Associate Editor and Reviewers' comments. This work was partially supported by the Beijing Innovation Center of Humanoid Robotics, China, where part of the work was conducted during Shichao Fan's internship.

[1] Shichao Fan and Min Wan are with School of Mechanical Engineering and Automation, BeiHang University, China. shichaofan@buaa.edu.cn.

[2] Yajie Liu and Qingjie Liu are with School of Computer Science and Engineering, BeiHang University, China. *Corresponding Author: qingjie.liu@buaa.edu.cn.

[3] Kun Wu and Zhengping Che are with Beijing Innovation Center of Humanoid Robotics, China.{gongda.wu, z.che}@x-humanoid.com

[4] Quantao Yang is with Division of Robotics, Perception and Learning (RPL), KTH Royal Institute of Technology, Sweden.

Digital Object Identifier (DOI): see top of this page.

©2026 IEEE

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

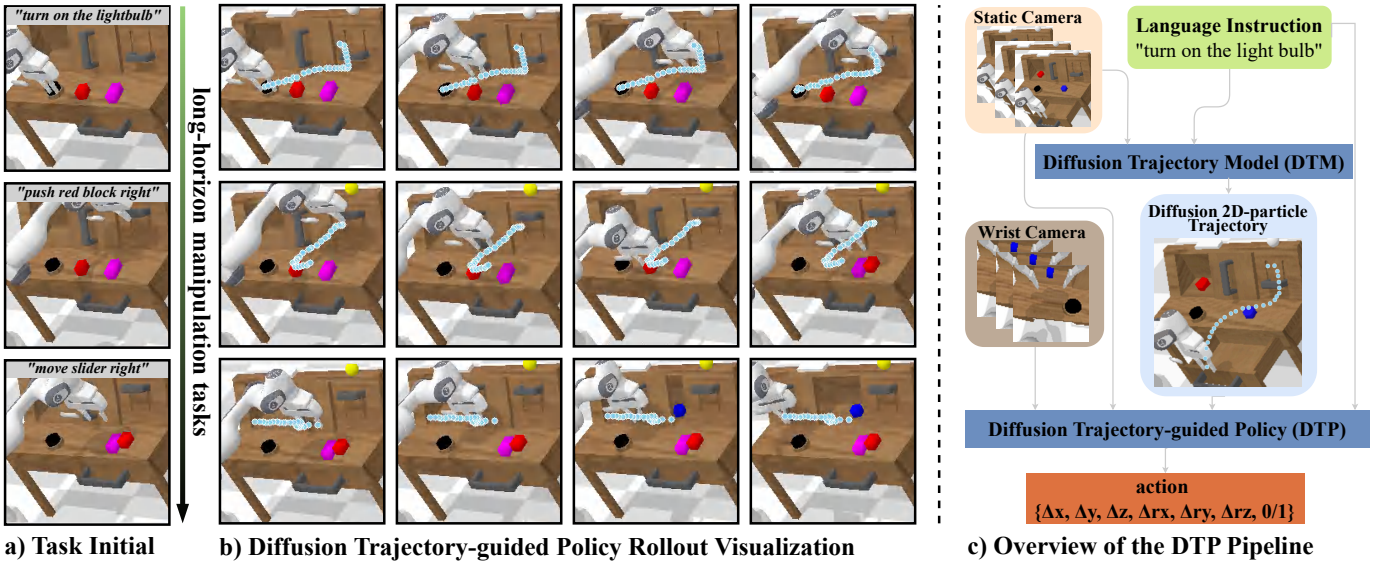

Fig. 1: **System overview.** a) and b) present a task instruction with the initial task observation, allowing our Diffusion Trajectory Model to predict the complete future 2D-particle trajectories; c) illustrates the Diffusion Trajectory-guided pipeline, showcasing how these predicted trajectories guide the manipulation policy.

tings. Additionally, our approach is computationally efficient, requiring only consumer-grade GPUs for training. The main contributions of the paper include:

1) We propose the Diffusion Trajectory-guided Policy (DTP), a novel imitation learning framework that utilizes a diffusion trajectory model to guide policy learning for long-horizon robot manipulation tasks.
2) We leverage robot video data to pretrain a generative vision-language diffusion model, which enhances imitation policy training efficiency by fully utilizing available robot data. Furthermore, our method can be combined with large-scale pretraining methods, serving as a simple and effective plugin to enhance performance.
3) We conducted extensive experiments in both simulated and real-world environments to evaluate the performance of DTP across diverse settings.

II. Related Work

Language-conditioned Visual Manipulation Policy Control. Language-conditioned visual manipulation has made significant progress due to advancements in large language models (LLMs) and vision-language models (VLMs). By using task planners like GPT-4 [14] or PaLM-E [15], it is possible to break down complex embodied tasks into simpler, naturally articulated instructions. Recently, several innovative methods have been developed in this domain. RT-1 [4] pioneered the end-to-end generation of actions for robotic tasks. RT-2 [5] explores the capabilities of LLMs for Vision-Language-Action (VLA) tasks by leveraging large-scale internet data. RoboFlamingo [16] follows a similar motivation as RT-2, focusing on the utilization of extensive datasets. RT-X [17] prioritizes the accumulation of additional robotic demonstration data to refine training and establish scaling laws in robotic tasks. The Diffusion Policy [2] addresses the prediction of robot actions using a denoising model. Lastly, Octo [18] serves as a framework for integrating the aforementioned contributions into a unified system, further advancing the field of language-conditioned visual manipulation.

Policy Conditioning Representations. Leveraging the high-dimensional semantic information in language, video prediction as a pre-training method [10], [19] yields reasonable results by generating future subgoals for the policy to achieve. Similarly, the goal image generation method [8] uses subgoal images instead of full video sequences. However, both approaches often result in hallucinations and unrealistic movements, and they require substantial computational resources, especially during inference. Methods such as those represented by MimicPlay [20] involve learning a latent planner [21]. These approaches require an additional training phase, and the latent planner's ability to learn useful features can only be indirectly visualized through a decoder, which is not very intuitive. RT-Trajectory [11] and ATM [22] provide innovative methods for generating coarse or particle trajectories. Unlike RT-Trajectory, which uses coarse trajectories with significant noise, we use particle trajectories for generation precision and flexibility. In contrast to ATM, which tracking any sampled points, we use a single key point to illustrate the task process regarding the end-effector's position in RGB space. This key point is readily derived using the camera's intrinsic and extrinsic parameters. Motion Tracks [23] and Im2Flow2Act [24] stem from ATM methods. Our method diverges from Motion Tracks by generating language-related long-horizon 2D points, while Motion Tracks only generates short-horizon 2D points without language input. Im2Flow2Act directly uses 2D flow for policy conditions, losing depth information. In contrast, our framework merges the benefits of each method, using 2D points to boost policy output. To standardize the notion of 2D points or waypoints in the RGB space, we label the key point sequences throughout a task as 2D-particle trajectories. Our method functions similarly to video prediction, serving as a

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

plugin to enhance policy learning.

Diffusion Model for Generation. Diffusion models in robotics are primarily utilized in two areas. Firstly, as previously discussed [8], [9], [10], they are used for generating future imagery in both video and goal image generation tasks. Secondly, diffusion models are applied to visuomotor policy development, as detailed in recent studies [2], [25], [18]. These applications highlight the versatility of diffusion models in enhancing robotic functionalities. Unlike these methods, our approach uses diffusion models not to directly generate the final policy but to create a 2D-particle trajectory for future end-gripper movement planning in the RGB domain.

III. METHOD

Our goal is to create a policy that enables robots to handle long-horizon manipulation tasks by interpreting vision and language inputs. We simplify the VLA task using two distinct phases (Fig. 2b)c)): a DTM learning phase and a DTP learning phase. First, we generate diffusion-based 2D particle trajectories for the task. Subsequently, in the second stage, these trajectories are used to guide the learning of the manipulation policy.

A. Problem Formulation

Multi-Task Visual Robot Manipulation. We consider the problem of learning a language-conditioned policy π_θ that takes advantage of language instruction l, observation $\boldsymbol{o}_t$, robot states $\boldsymbol{s}_t$ and diffusion trajectory $\boldsymbol{p}_{t:T}$ to generate a robot action $\boldsymbol{a}_t$:

$$\pi_\theta(l, \boldsymbol{o}_t, \boldsymbol{s}_t, \boldsymbol{p}_{t:T}) \rightarrow \boldsymbol{a}_t \tag{1}$$

The robot receives language instructions detailing its objectives, such as "turn on the light bulb". The observation sequence, $\boldsymbol{o}_{t-h:t}$, captures the environment's data from the previous h time steps. The state sequence, $\boldsymbol{s}_{t-h:t}$, records the robot's configurations, including the pose of the end-effector and the status of the gripper. The diffusion trajectory, $\boldsymbol{p}_{t:T}$, predicts the future movement of the end-gripper from time t to the task's completion at time T. Our dataset, $\mathbb{D}$, comprises n expert trajectories across m different tasks, denoted as $\mathbb{D}_m = \{\tau_i\}_{i=1}^n$. Each expert trajectory τ includes a language instruction along with a sequence of observation images, robot states, and actions: $\tau = \{\{l, \boldsymbol{o}_1, \boldsymbol{s}_1, \boldsymbol{a}_1\} ..., \{l, \boldsymbol{o}_T, \boldsymbol{s}_T, \boldsymbol{a}_T\}\}$.

B. Framework

We introduce the Diffusion Trajectory-guided Policy, as illustrated in Fig. 2. DTP operates within a two-stage framework. In the first stage, our primary focus is on generating the diffusion trajectory $\boldsymbol{p}_{t:T}$ which outlines the motion trends essential for completing the task, as observed from a static perspective camera (Fig. 2b)). This 2D-particle trajectory serves as the guidance for subsequent policy learning. We take a causal Transformer as the backbone network which is designed to handle diverse modalities, processing inputs to predict future images and robotic actions with learnable observation and action query tokens respectively. It integrates CLIP [26] as the language encoder for processing language instructions l and employs a MAE [27] as the vision encoder for $\boldsymbol{o}_{t-h:t}$, both of which are with frozen parameters. The vision tokens are then processed with a perceiver resampler [28] to reduce their number. Additionally, it incorporates the robot's state $\boldsymbol{s}_{t-h:t}$ in world coordinates, as part of its input. All input modalities are shown in Fig. 2a). Our approach is divided into two main sections. Initially, we detail the process of learning a diffusion trajectory model from the dataset $\mathbb{D}$ in Section III-C. Subsequently, in Section III-D, we illustrate how diffusion trajectories can be used to guide policy learning for long-horizon robot tasks.

C. Diffusion Trajectory Model

In the first stage (Fig. 2b)), we focus on generating diffusion trajectory that maps out the motion trends required for task completion, as viewed from a static perspective camera. To achieve this, we employ a model $\boldsymbol{M}_d$ to transform language instructions l and initial visual observations $\boldsymbol{o}_t$ into a sequence of diffusion 2D-particle trajectories $\boldsymbol{p}_{t:T}$. These points indicate the anticipated movements for the remainder of the task:

$$\boldsymbol{M}_d(l, \boldsymbol{o}_t) \rightarrow \boldsymbol{p}_{t:T} \tag{2}$$

1) Data Preparation: According to Eq. 2, our input consists of observations $\boldsymbol{o}_t$ and language instruction l. For outputs, our aim is to determine the future 2D-particle trajectory $\boldsymbol{p}_{t:T}$ of the end effector gripper for finishing the task. Recent advancements in video tracking work make it easy to monitor the end effector gripper [29]. For enhanced convenience and precision, we achieve this by mapping the world coordinates (x_w, y_w, z_w) to pixel-level positions (x_c, y_c) according to camera's intrinsic and extrinsic parameters in the static camera frame, as shown in (Fig. 2b)) right part. In the first stage, our data format is structured as $\mathbb{D}_{\text{trajectory}} = \{l, \boldsymbol{o}_t, \boldsymbol{p}_{t:T}\}$, facilitating straightforward acquisition of the sequence $\boldsymbol{p}_{t:T}$, thereby simplifying the process of training our model to accurately predict end effector positions.

2) Training Objective: Denoising Diffusion Probabilistic Models (DDPMs) [30] constitute a class of generative models that predict and subsequently remove noise during the generation process. In our approach, we utilize a causal diffusion decoding structure [2] to generate diffusion 2D-particle trajectories $\boldsymbol{p}_{t:T}$. Specifically, we initiate the generation process by sampling a Gaussian noise vector $x^K \sim \mathcal{N}(0, I)$ and proceed through K denoising steps using a learned denoising network $\epsilon_\theta(x^k, k)$ where x^k represents the diffusion trajectory noised over K steps. This network iteratively predicts and removes noise K times, ultimately resulting in the output x^0, which denotes the complete removal of noise. The process is described in the equation below, where α, γ, and σ are parameters that define the denoising schedule:

$$x^{k-1} = \alpha(x^k - \gamma\epsilon_\theta(x^k, k)) + \mathcal{N}(0, \sigma^2 I) \tag{3}$$

Eq. 3, illustrates the functioning of the basic diffusion model. For our application, we adapt this model to generate diffusion trajectories $\boldsymbol{p}_{t:T}$ based on the observation $\boldsymbol{o}_t$ and language instruction l:

$$\boldsymbol{p}_{t:T}^{k-1} = \alpha(\boldsymbol{p}_{t:T}^k - \gamma\epsilon_\theta(\boldsymbol{o}_t, l, \boldsymbol{p}_{t:T}^k, k)) + \mathcal{N}(0, \sigma^2 I) \tag{4}$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

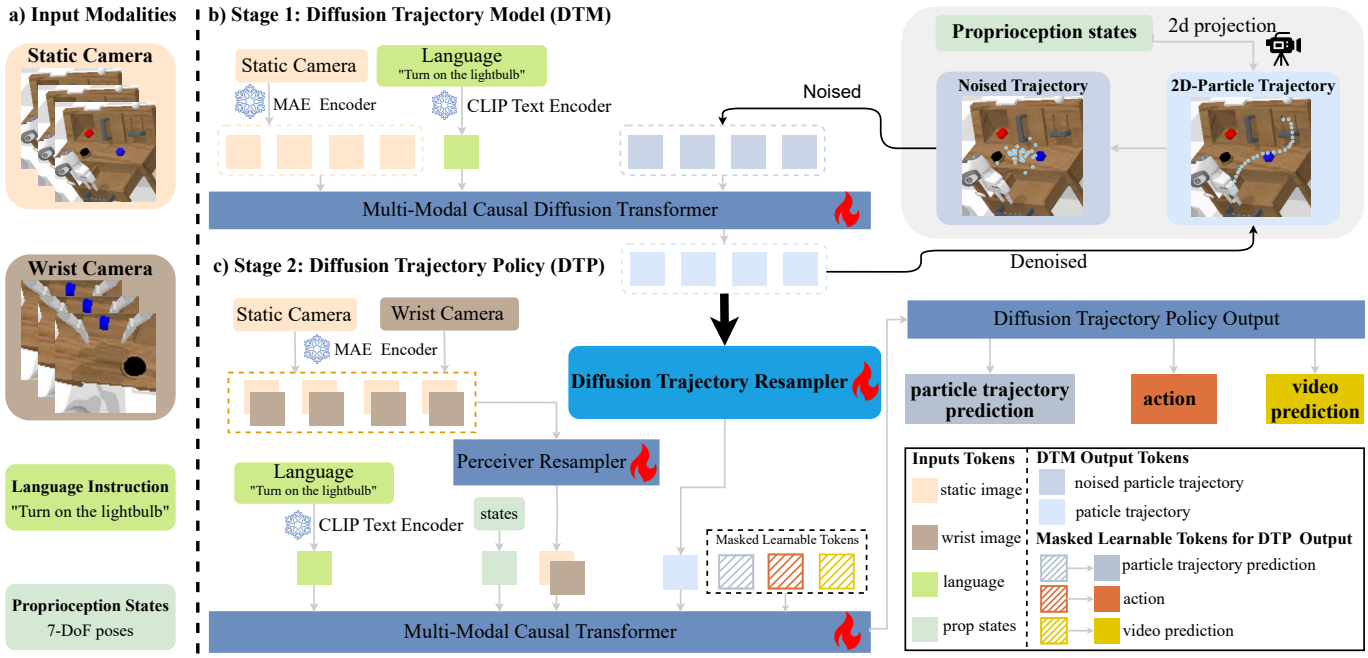

Fig. 2: **System architecture** for learning language-conditioned policies. a) shows the input modalities, including vision, language, and proprioception. b) describes the Diffusion Trajectory Model, detailing how vision and language inputs generate diffusion particle trajectories. c) explains how these trajectories guide the training of robot policies, focusing on the learning of the Diffusion Trajectory Policy. Masked learnable tokens represent the particle trajectory prediction token, action token, and video prediction token, respectively.

During the training process, the loss is calculated as Mean Square Error (MSE), where ϵ_k represents Gaussian noise sampled randomly for step k:

$$\mathcal{L}_{DTM} = \text{MSE}(\epsilon_k, \epsilon_\theta(\boldsymbol{o}_t, l, \boldsymbol{p}_{t:T} + \epsilon_k, k)) \tag{5}$$

This transformation integrates our specific inputs into the diffusion process, enabling the tailored generation of diffusion trajectory in alignment with both the observed data and the provided language instruction. This training loss ensures that diffusion 2D-particle trajectories are accurately generated by systematically reducing noise, thereby enhancing the precision of the final trajectory predictions.

D. Diffusion Trajectory-guided Policy

In the second stage, we focus on illustrating how the diffusion trajectory guides the robot manipulation policy (Fig. 2c)). As previously outlined in our problem formulation, we define our task as a language-conditioned visual robot manipulation task. We base our Diffusion Trajectory-guided Policy on the GR-1 [31] model and incorporate our diffusion trajectory $\boldsymbol{p}_{t:T}$ as an additional input, as specified in Eq. 1.

Policy Input. This consists of language and image inputs, as detailed in the Sec. III-B and shown in the left side of Fig. 2c). To clearly demonstrate our method's performance, we maintain the same configuration as GR-1. Importantly, for the diffusion trajectory, we do not rely on the inference results from the first training stage. Instead, we use the labeled data from this stage as the diffusion trajectory. This approach enhances precision in training and conserves computational resources by using the labels directly. The simplest training approach is to inject the diffusion particle trajectory directly into the causal baseline. However, our fixed set of 2D particle trajectories $\boldsymbol{p}_{t:T}$ can lead to computational intensity during training due to the high number of tokens. Inspired by the perceiver resampler [28], we designed a diffusion trajectory resampler module to reduce the number of trajectory tokens, as shown in Fig. 2b) and c).

Diffusion Trajectory-guided Policy Training. During the policy learning phase (Fig. 2c)), we generate future particle trajectories to supervise the diffusion trajectory resampler module with $\mathcal{L}_{\text{trajectory}}$. Our policy framework employs a causal Transformer architecture, where future particle trajectory tokens are generated prior to action tokens with $\mathcal{L}_{\text{action}}$. This ensures that the particle trajectory tokens effectively guide the formation of action tokens, optimizing the action prediction process in a contextually relevant manner. Additionally, we retain the output of video prediction with $\mathcal{L}_{\text{video}}$, maintaining the same setting as GR-1. This consistency in output makes it easier to conduct ablation studies, as we can directly compare our approach to the original GR-1 model. The optimal DTP objective can be expressed as the following equation:

$$\mathcal{L}_{DTP} = \mathcal{L}_{trajectory} + \mathcal{L}_{action} + \mathcal{L}_{video} \tag{6}$$

Furthermore, to demonstrate the effectiveness and superiority of our method in the ablation study, we split the GR-1 baseline into two versions: one that is fully pretrained on the video dataset and another that only uses the GR-1 structure without any pretraining. We will discuss these two baseline configurations in Sec. IV.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

IV. EXPERIMENT

In this section, we evaluate the performance of Diffusion Trajectory Policy in both simulation and real-world robot experiments.

A. CALVIN Benchmark and Baselines

CALVIN [13] is a comprehensive simulated benchmark designed for evaluating language-conditioned policies in long-horizon robot manipulation tasks. It comprises four distinct yet similar environments (A,B,C, and D) which vary in desk shades and item layouts, as shown in Fig. 3. This benchmark includes 34 manipulation tasks with unconstrained language instructions. Each environment features a Franka Emika Panda robot equipped with a parallel-jaw gripper, and a desk that includes a sliding door, a drawable drawer, color-varied blocks, an LED, and a light bulb, all of which can be interacted with or manipulated.

Experiment Setup. We train DTP to predict relative actions in xyz positions and Euler angles for arm movements, along with binary actions for the gripper. Our simulation setup aligns with the base model [31], where both the number of generated and executed actions are set to 1. The training dataset includes over 20,000 expert trajectories from four scenes, each paired with language instruction labels. Our DTP method is evaluated using the long-horizon benchmark, which features 1,000 unique sequences of instruction chains articulated in natural language. Each sequence requires the robot to sequentially complete five tasks.

Baselines. We compare our proposed policy against the following state-of-the-art language-conditioned multi-task policies on CALVIN: **MT-ACT** [32]: A multitask Transformer-based policy predicting action chunks. **HULC** [33]: A hierarchical approach predicting latent sub-goal features from language and observations. **RT-1** [4]: Utilizes convolutional layers and Transformers for end-to-end action generation from language and observations. **RoboFlamingo** [34]: A fine-tuned Vision-Language Foundation model with 3 billion parameters. **GR-1**: Pretrained on the Ego4D dataset, featuring large-scale human-object interactions. **3D Diffuser Actor** [35]: Integrates 3D scene representations with diffusion objectives to learn policies from demonstrations.

B. Comparisons with State-of-the-Art Methods

Results in Seen Environments. In the D→D setting, using about 5,000 expert demonstrations, training on 8*3090 GPUs took 1.5 days. As shown in Table I, DTP outperforms all baselines in long-horizon tasks, increasing Task 5's success rate from 0.400 to 0.509 and the average sequence length from 3.30 to **3.55**. Compared to GR-1, DTP boosts performance across all metrics, with a 33.9% increase in average sequence length, demonstrating superior performance as task length increases.

Results in Unseen Environments. In the challenging ABC→D setting, models are trained on environments A, B, and C, and tested in the unseen environment D. Training took about 5 days on 8*3090 GPUs. As shown in Table I, DTP demonstrates strong generalization, increasing the average task completion length from 3.06 to **3.43** and achieving a Task 5 success rate of 0.497, the highest recorded. Notably, without using depth modality, DTP outperformed the 3D Diffuser Actor, highlighting its effectiveness in guiding policy learning for long-horizon tasks in unseen settings.

Data Efficiency. In the ABCD→D setting, we evaluated data efficiency by training with only 10% of the dataset, using around 2,000 expert demonstrations. Training took about 1 day on 8*3090 GPUs. As shown in Table I, while all methods perform worse with less data, the best baseline, GR-1, achieves a success rate of 0.778 with an average length of 2.00. DTP excels in long-horizon tasks, with a success rate increase and an average length of **2.55**, outperforming others. These results demonstrate the data efficiency of DTP. By leveraging diffusion-based trajectories, the policy effectively captures positional preferences that are critical for long-horizon tasks. Furthermore, these 2D-particle trajectories provide guidance to the robot arm, enabling it to acquire skills even with a limited number of demonstrations.

TABLE I: Summary of Experiments

Method	Experiment	Tasks completed in a row					Avg. Len.
		1	2	3	4	5	
HULC	D→D	0.827	0.649	0.504	0.385	0.283	2.64
GR-1	D→D	0.822	0.653	0.491	0.386	0.294	2.65
MT-ACT	D→D	0.884	0.722	0.572	0.449	0.353	3.03
HULC++	D→D	0.930	0.790	0.640	0.520	0.400	3.30
DTP (ours)	D→D	0.924	0.819	0.702	0.603	0.509	3.55
HULC	ABC→D	0.418	0.165	0.057	0.019	0.011	0.67
RT-1	ABC→D	0.533	0.222	0.094	0.038	0.013	0.90
RoboFlamingo	ABC→D	0.824	0.619	0.466	0.380	0.260	2.69
GR-1	ABC→D	0.854	0.712	0.596	0.497	0.401	3.06
3D Diffuser Actor	ABC→D	0.922	0.787	0.639	0.512	0.412	3.27
DTP (ours)	ABC→D	0.890	0.773	0.679	0.592	0.497	3.43
RT-1	10% ABCD→D	0.249	0.069	0.015	0.006	0.000	0.34
HULC	10% ABCD→D	0.668	0.295	0.103	0.032	0.013	1.11
GR-1	10% ABCD→D	0.778	0.533	0.332	0.218	0.139	2.00
DTP (ours)	10% ABCD→D	0.813	0.623	0.477	0.364	0.275	2.55

This table details the performance of all baseline methods in sequentially completing 1, 2, 3, 4, and 5 tasks in a row. The average length, shown in the last column and calculated by averaging the number of completed tasks in a series of 5 across all evaluated sequences, illustrates the models' long-horizon capabilities. 10% ABCD→D indicates that only 10% of the training data is used.

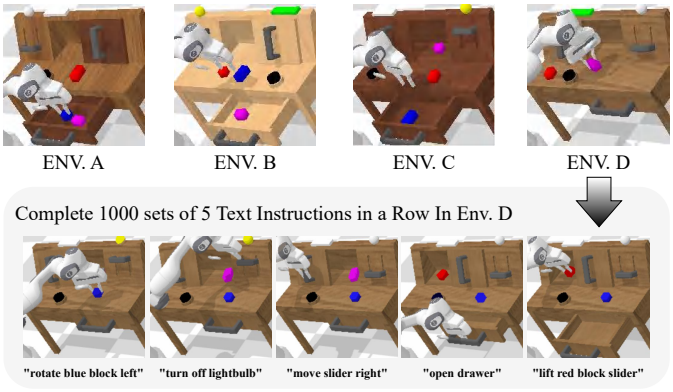

Fig. 3: The upper four environments correspond to the CALVIN ABCD settings. The bottom section shows a sequence of five long-horizon tasks, each guided by a specific instruction.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE II: Ablation Studies

Pre-Training	DTP (Ours)	Data	1	2	3	4	5	Avg. Len.
×	×	ABC→D	0.815	0.651	0.498	0.392	0.297	2.65
×	✓	ABC→D	0.869	0.751	0.636	0.549	0.465	3.27
×	×	10% ABCD→D	0.698	0.415	0.223	0.133	0.052	1.52
×	✓	10% ABCD→D	0.742	0.511	0.372	0.269	0.188	2.08
✓	×	ABC→D	0.854	0.712	0.596	0.497	0.401	3.06
✓	✓	ABC→D	0.890	0.773	0.679	0.592	0.497	3.43
✓	×	10% ABCD→D	0.778	0.533	0.332	0.218	0.139	2.00
✓	✓	10% ABCD→D	0.813	0.623	0.477	0.364	0.275	2.55
✓	100%✓	10% ABCD→D	0.822	0.643	0.526	0.416	0.302	2.71

Pre-Training indicates whether we use only the baseline model structure or the baseline pre-trained on the Ego4D dataset. DTP (ours) indicates whether the generated 2D particle trajectory is input into the policy. In our ablation studies, we established these two baselines to evaluate the effectiveness and compatibility of our DTM method with other approaches. 10% ABCD→D indicates that only 10% of the training data is used. 100%✓indicates DTM trained on full ABCD→D.

C. Ablation Studies

In this section, we conduct ablation studies to assess how diffusion trajectories enhance policy learning in visual robot manipulation tasks. This key contribution significantly improves imitation policy training efficiency by fully utilizing robot data. Integrated with large-scale pretraining baselines, it offers a straightforward performance boost. We compare our method against two baselines: one using the GR-1 framework (Sec. III-B) without video pretraining, and another with large-scale video pretraining using the Ego4D [36] dataset, both based on GR-1. These baselines verify our method's efficacy and compatibility.

Diffusion Trajectory Policy from Scratch. We evaluate our method in the ABC→D and 10% ABCD→D settings (see Table II). The diffusion trajectory method significantly enhances performance without pretraining, excelling in sequential tasks and increasing the average task completion length by 23.4%. The success rate for Task 5, indicative of long-horizon success, rises by 56.6%.

Diffusion Trajectory Policy with Video Pretrain. As shown in the bottom part of Table II, our diffusion trajectory variants enhance baseline model performance to state-of-the-art levels. Evaluated in ABC→D and 10% ABCD→D settings, our method consistently outperforms traditional scratch training, significantly boosting baseline performance. Success rates increase notably, from 4.2% in the first task to 23.9% in the fifth, validating DTP's effectiveness in long-horizon manipulation tasks.

Diffusion Trajectory Model Scaling Law. The last row highlights the initial training stage of our Diffusion Trajectory Model. Increasing training data improves point accuracy, enhancing DTP. Even with limited demonstration data, scaling diffusion trajectory training boosts success rates and task completion length. This suggests a potential direction: while robot demonstration data is costly, DTM data is easier to annotate, requiring only a coarse trajectory sketch on an RGB image with language instructions.

D. Real Robot Experiment

Experiment Setup. Our robotic system features a Franka Emika Panda robot with three Intel RealSense D435i cameras and a Robotiq gripper. We collected 1,512 demonstrations using a teleoperation system [37], with 290, 258, 100, 184, and 254 demonstrations for the tasks *PickBread*, *PickStrawberry*, *OpenTrash*, *CloseSideDrawer*, and *OpenSideDrawer*, respectively, covering object transportation and manipulation (see Fig. 4a). Additionally, we collected 426 demonstrations for a long-horizon task comprising three subtasks(A-B-C) (see Fig. 4b). In real robot setup, the number of generated actions is set to 25. This adjustment addresses sim-to-real discrepancies and latency, as setting both values to 1 causes the robot to remain almost completely still, with only minor vibrations. The number of executed actions remains at 1. We also incorporate an action ensemble mechanism, similar to ALOHA [38]. The training process spanned 20 epochs and took approximately one day on four RTX 3090 GPUs.

Results. The performance of DTP and baseline methods in five individual tasks is summarized in Table III upper part. Each task was evaluated over 10 trials, with success rates calculated for comparison. Overall, DTP achieved the highest aggregate success rate across tasks. However, in the *PickStrawberry* task, DTP underperformed compared to ACT. We attribute this to the small size of the target object, as DTP uses an image input resolution of 224x224, while ACT operates at a higher resolution of 480x640, which likely impacts performance. In long-horizon tasks, the robot arm's initial pose is determined by the completion of the previous task, resulting in random starting configurations. To evaluate DTP's robustness in such scenarios, we tested it on the *OpenSideDrawer* task with randomized initial arm poses. DTP achieved a success rate twice as high as the second-best method. Additionally, in the *OpenTrash* task, which requires precise alignment to a specific area to open the trash bin, DTP demonstrated superior guidance capabilities. While other baseline methods positioned the arm near the target, they often failed to locate the precise opening mechanism, leading to task failure.

In Table III (bottom), we evaluate variations of the training scenario. While training followed the A–B–C sequence, testing extended to longer sequences of up to five subtasks, such as A-B-C-A-C. This design probes long-horizon challenges: for example, repeating Task A after A-B-C occurs in a new context because the bread is already in the drawer, creating situations unseen during training. We created five such long-sequence variants for evaluation. Each value in Table III indicates the number of successful subtasks (out of five) for our method and the baseline, with "Ave. Len." reporting the average sequence length achieved. In A-B-C-A-C, the baseline fails at the fourth subtask due to missing bread, while our method continues. In A-C-A-B-C, the direct A→C transition introduces both an unseen scenario and initial pose, limiting the baseline to the first task, whereas our method progresses further. Overall, our approach reaches an average length of 4.6, compared to 2.0 for the baseline. This evaluation highlights the impact of ***accumulated compounding errors*** in long-horizon tasks: as sequences lengthen and subtask orders vary, accumulated deviations—such as altered object states or unseen robot poses—lead the baseline to fail, while our method remains more robust.

Visualization of Diffusion Trajectory Model. As shown in Fig. 5, we present the overall visualization of the diffusion

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

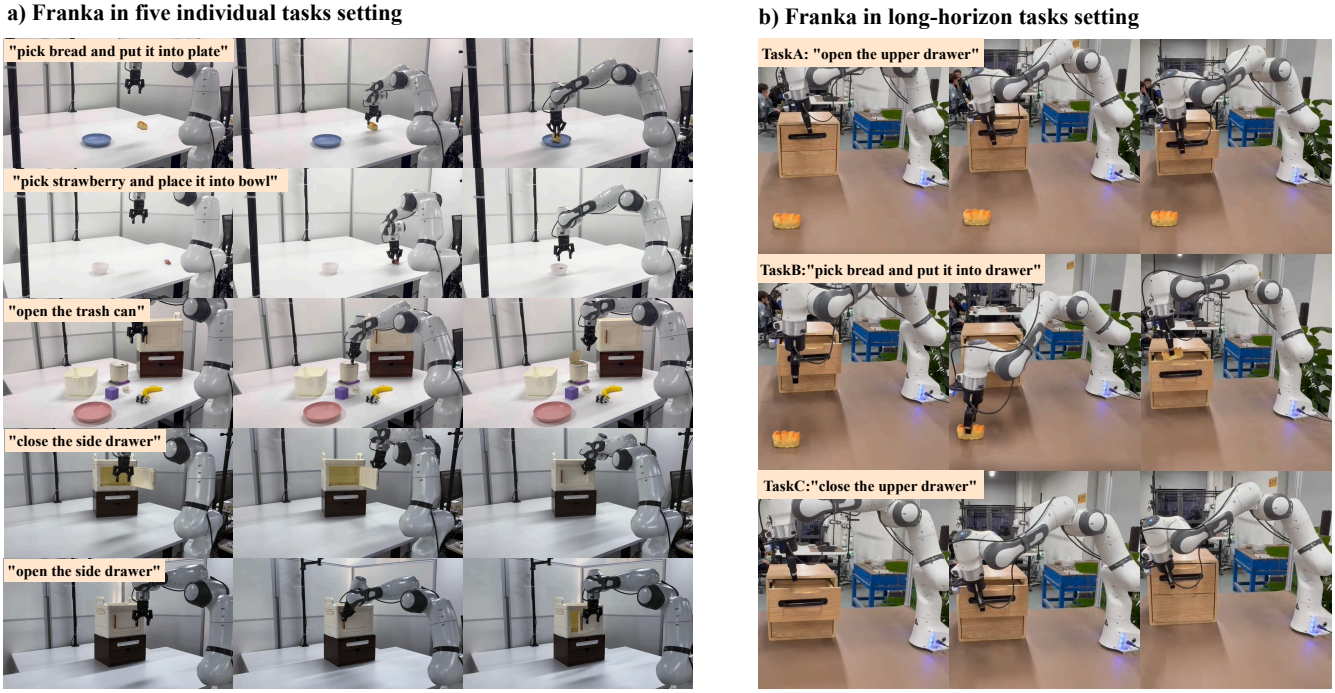

Fig. 4: Real-robot experiments: a) Franka performing five distinct manipulation tasks. b) Franka performing one long-horizon task composed of subtasks (A–B–C).

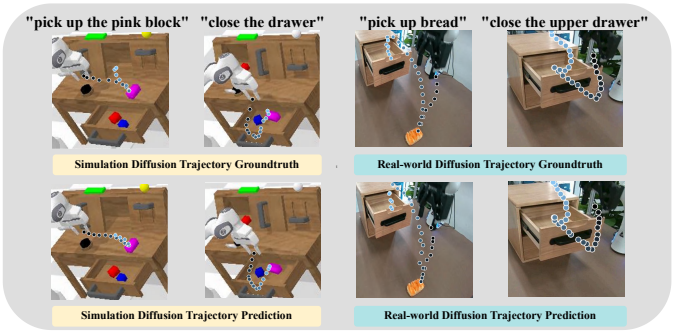

Fig. 5: **Diffusion Trajectory Visualization.** The left half part illustrates diffusion trajectory generation in the CALVIN environment, while the right half part show trajectory generation in a real-world robotic scenario.

TABLE III: Summary of Real Robot Experiments

Tasks / Method	Pick Bread	Pick Strawberry	Open Trash	CloseSide Drawer	OpenSide Drawer*	Ave. Suc.
ACT [38]	0.7	0.9	0.3	0.3	0.4	0.52
BAKU [39]	0.0	0.5	0.2	0.2	0.3	0.24
GR1 [31]	0.7	0.7	0.2	0.4	0.4	0.48
DTP (ours)	0.8	0.8	0.9	0.9	0.8	**0.84**

Tasks / Method	ABCAC	ACABC	CABCA	CACAB	BCACA	Ave. Len.
GR1	3/5	1/5	4/5	2/5	0/5	2.0
DTP (ours)	5/5	5/5	5/5	3/5	5/5	**4.6**

The upper table shows real-robot success rates for five tasks, where OpenSideDrawer* starts from a random robot pose. The lower table extends the long-sequence task A-B-C into longer forms (e.g., A-B-C-A-C) and rearranges them into new variants. Reported values indicate how many of the five subtasks were completed in each variant. "Ave. len" denotes the average sequence length for our method and the baseline GR1.

trajectory generation phase, tested in both the CALVIN environment and real-world scenarios. The visualizations demonstrate that the trajectories generated by our diffusion trajectory prediction closely align with the ground truth. Even when minor deviations occur, the generated trajectories remain consistent with the robotic arm paths dictated by the language instructions.

V. CONCLUSION

The limited availability of robot data and compounding errors in IL make it difficult to generalize long-horizon tasks to unseen poses and environments. We introduce a diffusion trajectory-guided framework that leverages RGB-domain diffusion trajectories to improve policy learning in robot manipulation. Our method augments training data through data augmentation or manually crafted labels, producing more accurate trajectories. It has two stages: (i) training a diffusion trajectory model to generate task-relevant trajectories, and (ii) using them to guide the robot's manipulation policy. On the CALVIN benchmark, our method outperforms state-of-the-art baselines by an average success rate of 25%, and also shows strong improvements using only robot data as well as in real-world experiments.

For future work, we plan to extend our framework to other state-of-the-art policies, as we believe diffusion trajectories can further enhance their effectiveness. Another direction is to obtain trajectory labels using camera intrinsic and extrinsic parameters, which are often missing in open-source datasets [40]. Track-Anything [29] offers strong object-tracking ability for generating labels, while EgoMimic [41] captures detailed 3D hand tracks with Aria glasses that can be projected into 2D

particle trajectories. Such hardware also enables large-scale video pretraining for diffusion trajectory tasks.

REFERENCES

[1] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," in *Conference on Robot Learning*. PMLR, 2023, pp. 1199–1210.

[2] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[3] T. Gao, S. Nasiriany, H. Liu, Q. Yang, and Y. Zhu, "Prime: Scaffolding manipulation tasks with behavior primitives for data-efficient imitation learning," *IEEE Robotics and Automation Letters*, 2024.

[4] 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.

[5] 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.

[6] Y. Ma, Z. Song, Y. Zhuang, J. Hao, and I. King, "A survey on vision-language-action models for embodied ai," *arXiv preprint arXiv:2405.14093*, 2024.

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

[8] 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.

[9] Y. Du, M. Yang, P. Florence, F. Xia, A. Wahid, B. Ichter, P. Sermanet, T. Yu, P. Abbeel, J. B. Tenenbaum, *et al.*, "Video language planning," *arXiv preprint arXiv:2310.10625*, 2023.

[10] Y. Du, S. Yang, B. Dai, H. Dai, O. Nachum, J. Tenenbaum, D. Schuurmans, and P. Abbeel, "Learning universal policies via text-guided video generation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[11] J. Gu, S. Kirmani, P. Wohlhart, Y. Lu, M. G. Arenas, K. Rao, W. Yu, C. Fu, K. Gopalakrishnan, Z. Xu, *et al.*, "Rt-trajectory: Robotic task generalization via hindsight trajectory sketches," in *International Conference on Learning Representations*, 2024.

[12] S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh, "Rt-h: Action hierarchies using language," *arXiv preprint arXiv:2403.01823*, 2024.

[13] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7327–7334, 2022.

[14] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[15] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[16] X. Li, M. Liu, H. Zhang, C. Yu, J. Xu, H. Wu, C. Cheang, Y. Jing, W. Zhang, H. Liu, H. Li, and T. Kong, "Vision-language foundation models as effective robot imitators," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=lFYj0oibGR

[17] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6892–6903.

[18] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, "Octo: An open-source generalist robot policy," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[19] A. Escontrela, A. Adeniji, W. Yan, A. Jain, X. B. Peng, K. Goldberg, Y. Lee, D. Hafner, and P. Abbeel, "Video prediction models as rewards for reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[20] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar, "Mimicplay: Long-horizon imitation learning by watching human play," *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.

[21] K. Wu, Y. Zhu, J. Li, J. Wen, N. Liu, Z. Xu, and J. Tang, "Discrete policy: Learning disentangled action space for multi-task robotic manipulation," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[22] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel, "Any-point trajectory modeling for policy learning," *arXiv preprint arXiv:2401.00025*, 2023.

[23] J. Ren, P. Sundaresan, D. Sadigh, S. Choudhury, and J. Bohg, "Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[24] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song, "Flow as the cross-domain manipulation interface," *Proceedings of the Conference on Robot Learning (CoRL)*, 2024.

[25] Y. Su, N. Liu, D. Chen, Z. Zhao, K. Wu, M. Li, Z. Xu, Z. Che, and J. Tang, "Freqpolicy: Efficient flow-based visuomotor policy via frequency consistency," *arXiv preprint arXiv:2506.08822*, 2025.

[26] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[27] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[28] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, "Perceiver: General perception with iterative attention," in *International conference on machine learning*. PMLR, 2021, pp. 4651–4664.

[29] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng, "Track anything: Segment anything meets videos," *arXiv preprint arXiv:2304.11968*, 2023.

[30] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[31] H. Wu, Y. Jing, C. Cheang, G. Chen, J. Xu, X. Li, M. Liu, H. Li, and T. Kong, "Unleashing large-scale video generative pre-training for visual robot manipulation," in *International Conference on Learning Representations*, 2024.

[32] 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.

[33] O. Mees, L. Hermann, and W. Burgard, "What matters in language conditioned robotic imitation learning over unstructured data," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 205–11 212, 2022.

[34] X. Li, M. Liu, H. Zhang, C. Yu, J. Xu, H. Wu, C. Cheang, Y. Jing, W. Zhang, H. Liu, *et al.*, "Vision-language foundation models as effective robot imitators," in *International Conference on Learning Representations*, 2024.

[35] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," *arXiv preprint arXiv:2402.10885*, 2024.

[36] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, *et al.*, "Ego4d: Around the world in 3,000 hours of egocentric video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 995–19 012.

[37] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel, "Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 12 156–12 163.

[38] 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.

[39] S. Haldar, Z. Peng, and L. Pinto, "Baku: An efficient transformer for multi-task policy learning," *arXiv preprint arXiv:2406.07539*, 2024.

[40] 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.

[41] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu, "Egomimic: Scaling imitation learning via egocentric video," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.