Trajectory-based Diffusion from One-shot Human Video for Generalizable Manipulation

Anonymous Author(s)
Affiliation
Address
email

Abstract: Imitation learning enables robots to perform complex tasks by leveraging human-collected demonstrations. However, most existing approaches require labor-intensive data collection with expensive teleoperation systems, which hinders scalability. Human videos represent a natural source of data that implicitly contain knowledge of manipulation behaviors. However, transferring them to robots remains challenging due to the visual domain gap and the absence of explicit action labels. In this work, we introduce TDV, a trajectory-based framework that leverages data generation to enable generalizable manipulation. TDV focuses on task-relevant objects by leveraging plug-and-play 6D pose estimation techniques to extract object trajectories from human videos, thereby mitigating the impact of irrelevant backgrounds and varying viewpoints. As a modality-agnostic intermediate representation, trajectories are inherently independent of the visual domains and different embodiments, enabling efficient data generation. Our method outperforms previous approaches on RLBench simulation tasks, achieving near 100% success rates across seven tasks. In real-world experiments, TDV enables cross-embodiment generalization manipulation from one-shot human demonstrations by leveraging efficient trajectory generation techniques.

Keywords: CoRL, Robots, Learning

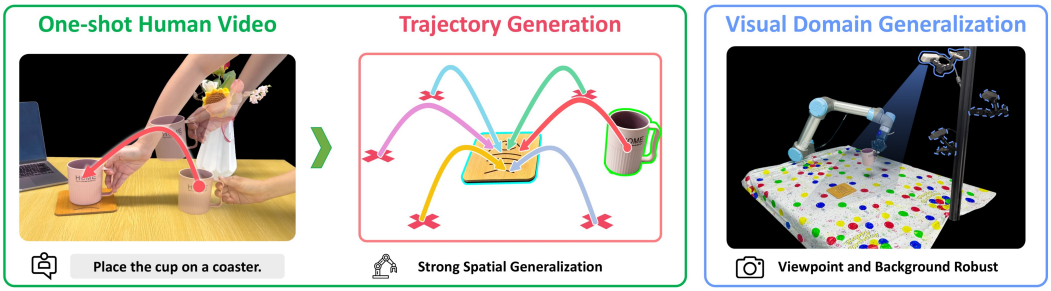

Figure 1: Our method is capable of generating a large number of valid trajectories from a single human video, enabling strong spatial generalization. Moreover, it exhibits robust performance under variations in the visual domain during deployment.

1 Introduction

Imitation learning enables robots to perform complex tasks without the need for manually configured heuristic rules. Most existing imitation learning approaches rely on proprioceptive states and visual data collected directly from the robot system, typically obtained through additional teleoperation systems operated by humans [1, 2, 3]. However, this approach is inefficient and presents

Submitted to the 9th Conference on Robot Learning (CoRL 2025). Do not distribute.

significant challenges in scaling up data collection. In addition to images and text, there is a vast amount of human demonstration videos available on the internet, which inherently encode procedural knowledge for performing various manipulation tasks. Nevertheless, extracting meaningful perceptual information and actions from these videos for robot manipulation remains a non-trivial challenge.

This problem involves two main aspects: the significant gap in visual domains and the lack of robot action information. To address the visual domain gap, the most straightforward approach is to leverage generative techniques for image editing, replacing the human with a robot in the images or erasing the human entirely [4, 5]. However, this method heavily depends on the performance of generative models, and there is currently a lack of studies that ensure both good consistency and compliance with physical laws. Some studies utilize flow as an intermediate representation [6, 7, 8, 9], which captures the motion trends of task-relevant objects and exhibits robustness to visual discrepancies introduced by different embodiments. However, these methods require the camera viewpoint during deployment to be consistent with the one used during data acquisition, which limits their scalability in practical applications [10]. For manipulation tasks, the primary focus is on the state changes of task-relevant objects, rather than the agent executing the actions, thereby enabling cross-embodiment deployment. We find that object-centric representations, such as 6D object pose [11, 12, 13], can effectively capture the real-time state of the object while mitigating the effects of irrelevant background and different viewpoints. A trajectory composed of a sequence of 6D object poses encapsulates the state transitions during the manipulation process. The previous method [12] requires pre-generated grasp poses for the object and fails to fully exploit the advantages of trajectory representations in scaling datasets [14, 15], which is crucial for enhancing the generalization of manipulation.

To overcome these shortcomings, we propose a trajectory-based diffusion framework, TDV, which enables generalizable manipulation across different embodiments from one-shot human demonstrations. Our framework is structured into three stages. First, we parse human videos to extract motion trajectories relevant to the manipulation task. Depending on whether a manipulated object is present, we classify the tasks into two categories: hand tasks and object tasks. In hand tasks that involve only hand motion, we leverage a 3D hand pose estimation model [16] to reconstruct the human hand trajectories. In contrast, for object tasks where a manipulated object is present, we utilize a plug-and-play 6D pose estimation model [17] to extract its trajectory. To eliminate the influence of viewpoint changes, we represent trajectories relative to the reference object frame. Then, the resulting trajectories provide a modality-agnostic intermediate representation, enabling efficient data augmentation through simple trajectory scaling and interpolation. Finally, We then use the augmented dataset to train a trajectory-based diffusion policy, which achieves robust generalization across diverse manipulation tasks and supports multi-procedure execution conditioned on high-level task descriptions. During deployment, the pose of the moving object is detected in real time and fed into the history sequence of the diffusion policy, enabling closed-loop execution.

Our contributions include: 1) proposed a trajectory-based diffusion framework, TDV, which learns manipulation policies solely from human videos, without requiring any robot demonstrations. 2) leveraged a modality-agnostic intermediate representation based on trajectories to enable effective learning and generalization across different data sources. 3) demonstrated the capability to achieve spatial generalization from a single human video across diverse manipulation tasks.

2 Related Works

2.1 Learning from Human Videos

Many endeavors have been made to learn scalable manipulation policies from non-robotic data, particularly human videos[8, 18, 19, 20]. The main idea is to extract meaningful representations from massive visual data and transfer them to robotic tasks. For instance, R3M[21] and MVP[22] pretrain visual representations on the large-scale internet human video dataset Ego4D[23], aiming to extract manipulation knowledge beneficial for robot learning from first-person perspective videos. How-

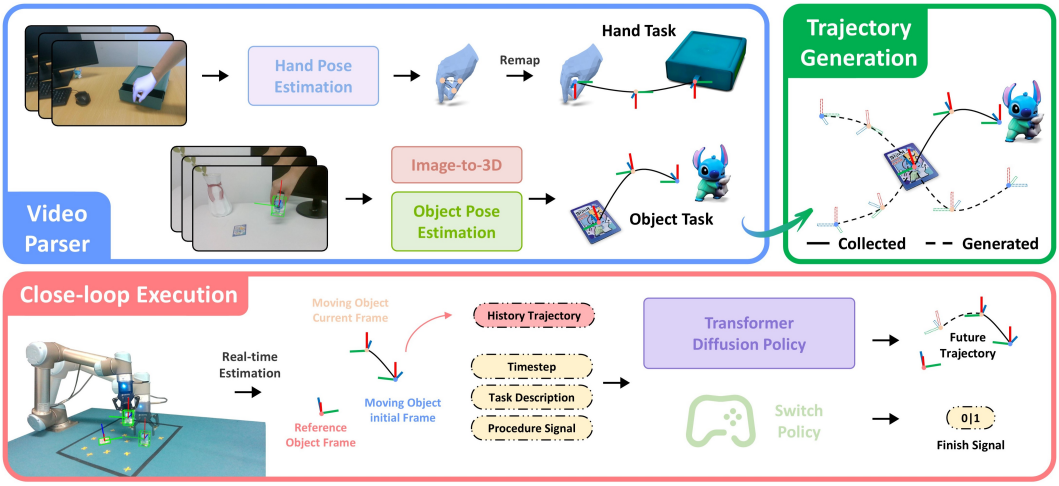

Figure 2: **Overview of TDV.** Our framework consists of three stages: video parsing, trajectory generation, and closed-loop execution. For hand tasks, we extract 3D hand trajectories and remap them to a two-finger gripper. For object tasks, we apply 6D object pose estimation and perform data augmentation by sampling diverse initial poses. A trajectory-based diffusion policy is trained on the resulting data. During deployment, real-time pose estimation and a switch policy enable closed-loop control and multi-procedure task execution.

ever, due to the heterogeneous nature of the data sources and significant domain gaps, transferring pre-trained representations to specific manipulation remains challenging. To address this issue, some methods[24, 25] leverage in-domain human videos, where humans and robots perform the same manipulation tasks in identical environments. Another approach leverages transferable representations, such as affordances[26, 5] and flows[6, 8, 7], combined with additional action policies.

2.2 Object-Centric Manipulation

To exclude background and task-irrelevant objects, many researchers focus on developing effective object-centric representations, which possess visual robustness and cross-scene generalization capabilities. The primary representation include point clouds [27], Gaussian splatting[28], 6D object pose estimation, neural implicit fields[29], and other related methodologies [30, 6, 31, 12]. Owing to the development of visual foundation models, previous work [27] has segmented the object point cloud from the scene point cloud in a modular manner and used it as input for the imitation policy, which are robust to background changes. Rather than directly using point clouds, Kerr et al. [30] combined 3D Gaussians with object part motion to construct a 4D Differentiable Part Model. Among object-centric representations, the 6D object pose is comparatively straightforward and efficient. Pan et al. [13] instructed the LLM to construct constraint relations based on the coordinate of task-relevant objects, and performed real-time 6D pose tracking during closed-loop execution to optimize the trajectory.

3 Methods

Our overall framework consists of three stages, as illustrated in Figure 2. To generate training data, we first extract trajectories from human videos based on task descriptions. Tasks are categorized into hand tasks and object tasks depending on the type of moving object involved. Since the relative motion of the hand with respect to the reference object in hand tasks is typically consistent, we apply data augmentation only to object tasks. Specifically, we randomly sample valid initial poses of the moving object within the workspace to construct a large and diverse set of training instances. All generated data are then used to train a trajectory-based diffusion transformer policy. To support

autonomous execution of multi-procedure manipulation tasks, we additionally train a switch policy that determines when to transition between subtasks. During deployment, we perform real-time pose estimation of both the moving and reference objects to enable closed-loop control.

3.1 Video Parser

3.1.1 Hand Pose Trajectory

To extract hand pose trajectories from video, we adopt the HaMeR [16] model, which reconstructs a MANO hand model with 21 articulated joints. Since our experimental setup employs a two-finger gripper as the end-effector, we remap the finger joint poses to the gripper configuration. Considering typical grasping gestures, we select the thumb MCP joint, thumb IP joint, and index MCP joint, whose positions are denoted as $P_{t_{\text{MCP}}}$, $P_{t_{\text{IP}}}$, and $P_{i_{\text{MCP}}}$, respectively. Based on these three points, we compute the remapped pose T_{remap}, with the detailed derivation provided in the supplementary.

For object tasks, we extract the hand pose and the reference object pose from the keyframe in which the object is successfully grasped. Given their relative transformation and the remapped pose T_{remap}, we compute the corresponding gripper pose for execution.

3.1.2 Object Pose Trajectory

Image-to-3D. Given a close-up image of the task-relevant object, the image-to-3D generation model TRELLIS [17] is capable of generating high-quality 3D models resembling the object, a capability acquired through training on a large-scale dataset of diverse objects. However, the generated 3D assets lack scale information, which we calibrate by computing the 3D bounding box corresponding to the segmented point cloud with SAM2 or through manual measurement.

Oject Pose Estimation. To enable real-time object tracking in videos based on the generated object mesh, we divide the process into two steps: initial mask generation and mask tracking. We use the task description as a text prompt for GroundingSAM [32] to obtain the initial mask corresponding to the task-relevant object. Once the initial frame mask is obtained, it is used as a mask prompt for the video predictor of SAM2 [33], enabling consistent mask propagation across subsequent frames based on through the introduction of the memory mechanism. Given the object mesh and the corresponding mask, we utilize FoundationPose [34] to estimate the 6D object pose from raw RGB-D inputs.

3.2 Trajectory Generation.

Although directly collecting human videos as demonstrations provides an efficiency improvement over teleoperation, achieving generalizable manipulation still requires large-scale data acquisition. Compared to visual images, using trajectories as an intermediate representation offers significant advantages in terms of computational efficiency and data augmentation [14]. To maintain the feasibility of the original trajectory, we only randomize the initial pose of the target object within the workspace to generate a new initial pose $\hat{T}_0^g$. Since the trajectory is referenced to the coordinate frame of the reference object, changes in the reference object pose can be equivalent to changes in the target object pose.

Smooth transition between two initial poses, T_0^g and $\hat{T}_0^g$, is achieved by applying warping [35] for position and spherical linear interpolation for orientation. For the position component P, we first compute the vectors from the original and new starting point to the fixed endpoint, respectively denoted as v_o^g, $\hat{v}_o^g$ and project them onto the XY-plane, resulting in the projected vectors $v_{o,xy}^g$ and $\hat{v}_{o,xy}^g$. The original trajectory is then normalized by $|v_{o,xy}^g|$. Using the scale factor $|\hat{v}_{o,xy}^g|$, we rescale the normalized trajectory, and apply the rotation of angle r, which represents the angle between $v_{o,xy}^g$ and $\hat{v}_{o,xy}^g$, to obtain a generated trajectory with its starting point at the origin. Finally, we align the starting point by translating the trajectory with the vector $\vec{M}$, which represents the offset from the original to the new starting point, yielding the final position component of generated trajectory. For the orientation component O, We retain the original orientation over a fixed-proportion region

near the end of the trajectory, defined by a ratio parameter $\alpha \in (0, 1)$, to ensure consistency in the final operation. Between the new start orientation and this terminal region, we interpolate the orientation using the Slerp algorithm. This approach provides a smooth transition in orientation along the trajectory.

3.3 Trajectory-based Diffusion Transformer

To adapt to the trajectory representation, we model the middle process of the task as $f(T^{\text{target}}_{t:t+n}|T^{\text{target}}_{t-h:t}, l, s)$, where the future trajectory $T^{\text{target}}_{t:t+n}$ of length n is predicted based on the history trajectory $T^{\text{target}}_{t-h:t}$ of length h of the target object, conditioned on the task description l and the task procedure signal s. We employ a Denoising Diffusion Probabilistic Model (DDPM) [36] to approximate the conditional distribution $p(T^{\text{target}}_{t:t+n}|\mathbf{O}_t)$, where the observations $\mathbf{O}_t$ are composed of the history trajectory $T^{\text{target}}_{t-h:t}$, task description l, and procedure signal s. Unlike previous work, we use the historical trajectory as input rather than just the current and past poses, which is crucial for tasks that require revisiting the same poses. We decouple the encoder from the denoising process, which leads to a significant improvement in inference efficiency. The history trajectory is mapped to the latent space via a two-layer MLP. The task description l is tokenized using BERT [37], while the procedure signal s is represented as a learnable embedding. The features are then encoded with positional information according to their modalities and fused through a transformer encoder. To better integrate the observations, we employ a DiT-block [35] as the decoder backbone, with the fused features and timestep embedding serving as the conditional information. Building upon previous work, we modify the denoising process as follows:

$$\mathbf{T_n}^{k-1} = \alpha(\mathbf{T_n}^{k} - \gamma\varepsilon_\theta(\mathbf{O_t}, \mathbf{T_n}^{k}, k) + \mathcal{N}(0, \sigma^2 I)) \tag{1}$$

where $\mathbf{T_n}$ is a simplified notation for $T^{\text{target}}_{t:t+n}$ and ε_θ is the noise prediction network with parameters θ. The parameters α, γ, and σ of the denoising scheduler are functions of the iteration step k. This is the reverse process of the diffusion process, which is detailed in previous work. The loss function is modified as follows:

$$\mathscr{L} = \text{MSE}(\varepsilon^k, \varepsilon_\theta(\mathbf{O_t}, \mathbf{T_n}^{0} + \varepsilon^k, k)) \tag{2}$$

To achieve task autonomy, we introduce a switch policy that determines whether to increment the procedure signal based on the history trajectory and task description. When the procedure signal reaches the maximum number of procedures, it indicates the completion of the task. The final state of the trajectory for each procedure is set to 1, while the others are set to 0. To address data imbalance, we set the loss weight ratio to 1:100. The switch policy is implemented as a classification network built with MLPs, which is beneficial to improving inference efficiency.

3.4 Close-loop Execution

To account for temporal changes in object states, we perform real-time pose estimation for both the moving object and the reference object during deployment. In cases where the moving object corresponds to a human hand in the dataset, we utilize the end-effector pose T_{ee} as the observation input to the transformer diffusion policy. In object tasks, once the end-effector grasps the object, they maintain a fixed transformation T_{attach} throughout the subsequent motion. However, since the gripper may occlude the manipulated object during the task execution, potentially leading to pose estimation drift, we instead compute the object pose by combining the end-effector pose with the fixed attachment transformation.

4 Experiments

We conduct a series of experiments in both simulation and real-world environments to evaluate our method, aiming to answer the following questions: (1) How effective is the trajectory as an RGB-free intermediate representation for generalizing across diverse manipulation tasks? (2) Can

Table 1: Quantitative results on RLBench. We report success rates (%) on 12 selected tasks in the format of mean ± standard deviation, along with the overall average across all tasks.

Method	Close Jar	Drag Stick	Insert Peg	Meat off Grill	Place Cups	Place Wine	Put in Cupboard
OpenVLA	1.3 ± 1.2	1.3 ± 1.2	0.0 ± 0.0	4.0 ± 2.0	2.0 ± 0.0	4.6 ± 1.2	0.0 ± 0.0
RVT2	**100.0 ± 0.0**	**100.0 ± 0.0**	41.3 ± 4.6	94.6 ± 2.3	40.0 ± 10.5	86.6 ± 2.3	65.3 ± 4.6
3D-DA	49.3 ± 0.2	**100.0 ± 0.0**	64.0 ± 4.0	94.6 ± 6.1	22.6 ± 4.6	92.0 ± 4.0	86.6 ± 2.3
SPOT	98.7 ± 2.3	80.0 ± 0.0	78.7 ± 2.3	**100.0 ± 0.0**	62.7 ± 6.1	**100.0 ± 0.0**	42.7 ± 4.6
TDV	**100.0 ± 0.0**	**100.0 ± 0.0**	**98.7 ± 2.3**	**100.0 ± 0.0**	**73.3 ± 6.1**	98.7 ± 2.3	**98.7 ± 2.3**

Method	Put in Safe	Screw Bulb	Sort Shape	Stack Blocks	Stack Cups	Avg. Success
OpenVLA	4.6 ± 1.2	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	1.5 ± 0.57
RVT2	96.0 ± 4.0	**88.0 ± 0.0**	48.0 ± 4.0	68.0 ± 8.0	76.0 ± 4.0	71.6 ± 3.69
3D-DA	**100.0 ± 0.0**	76.0 ± 0.0	49.3 ± 2.3	71.0 ± 4.0	53.3 ± 8.3	75.3 ± 2.79
SPOT	**100.0 ± 0.0**	48.0 ± 8.0	32.0 ± 4.0	**94.0 ± 3.4**	**96.0 ± 0.0**	79.4 ± 2.56
TDV	98.7 ± 2.3	81.3 ± 2.3	**69.3 ± 2.3**	88.0 ± 4.0	94.7 ± 2.3	**91.8 ± 2.18**

our method achieve generalizable manipulation from one-shot human video via efficient data augmentation across embodiments? (3) Does our method generalize to visual domain shifts, including changes in camera viewpoints and background scenes?

4.1 Simulation Experiments

4.1.1 Setup

Benchmark. We evaluate our method on RLBench[38], a standard multi-task manipulation benchmark. We select 12 representative tasks, each consisting of 2 to 60 variants, with differences in the color, shape, and target position of the objects in each variant. The selected tasks include those that require high precision and multi-step execution. Each task is specified by a language description. Following prior work [31], we use 100 demonstrations per task for training and reserve 25 demonstrations for evaluation.

Baselines. We compare our method with several strong baselines that have demonstrated competitive performance on RLBench: 3D Diffuser Actor (3D-DA)[31], RVT2[39], and SPOT [12]. Additionally, we include OpenVLA [40], a 7B-parameter vision-language-action model specifically designed for generalization across diverse manipulation tasks. We fine-tune the released OpenVLA checkpoint using LoRA. For 3D-DA and RVT2, we follow their official training pipelines and retrain the models on our 12 selected tasks. As SPOT's implementation is not publicly available, we report its results directly from the original paper.

4.1.2 Results

Quantitative results on the RLBench tasks are presented in Table 1. We report the average performance over three random seeds, each with 25 trials. TDV ranks first in 8 out of 12 tasks and achieves competitive results on the remaining 4. Notably, TDV outperforms other methods by over 20% on Insert Peg and Sort Shape, which demand high-precision manipulation. It also demonstrates more than a 10% improvement on Place Cups and Put in Cupboard, highlighting the method's strength in long-horizon planning and execution. In tasks such as Screw Bulb and Stack Blocks, where TDV shows relatively lower performance, the main failure cases are attributed to infeasible motion planning in RLBench and errors arising from unseen relative spatial configurations. Overall, TDV achieves an average success rate of 91.8%, which surpasses the previous state-of-the-art by 13.4% and clearly answers question (1).

4.2 Real-world Experiments

4.2.1 Setup

For real-world experiments, we set up a tabletop robotic system consisting of a UR5 arm equipped with a Robotiq 2F-85 gripper. Perception is provided by an Intel RealSense D435 camera mounted on a 6-DoF articulating Magic Arm, allowing flexible adjustment of the camera viewpoint according to the experimental configuration. We evaluate our method on three object-centric and three hand-centric tasks: (1) placing the cup on a coaster (Cup-Coaster), (2) placing the Stitch toy on a card (Stitch-Card), (3) pouring water into the cup (Pour-Water), (4) pulling open the drawer (Pull-Drawer), (5) unplugging the charger from the socket (Unplug-Charger), and (6) pulling a tissue from the box (Pull-Tissue). These tasks are used to assess the generalization capabilities of our method across different object types and interaction modes.

4.2.2 Spatial Generalization

To evaluate the spatial generalization capabilities of our method, we define a workspace of 45 cm × 50 cm and randomly sample test cases within this region. Given that our approach relies on only a single human demonstration, we design two levels of spatial generalization: position generalization (PG) and pose generalization (PoG), progressing from simpler to more challenging settings. For position generalization, we randomly select 13 points whose spatial distribution covers diverse relative configurations across the workspace. For pose generalization, we apply random orientation perturbations at each of the 13 positions, generating novel configurations unseen in the demonstration trajectory.

Since the relative motion in hand tasks is typically consistent, and the reference object pose can be estimated in real time, a single demonstration is often sufficient for spatial generalization. Therefore, we focus our spatial generalization evaluation on three object tasks. We report results under three training settings: one-shot (a single demonstration), 15-shot (augmented from one), and 50-shot (further augmented), as summarized in Table 2. Although trained on a single trajectory, TDV demonstrates a certain degree of spatial generalization. However, it tends to overfit to the single demonstration, consistently reverting to the initial pose seen during training at the beginning of each execution. Such behavior not only reduces efficiency but also increases the likelihood of collisions, as shown in Fig. 3. Through efficient trajectory generation, we augment the dataset to 15 and 50 trajectories. As the number of training samples increases, TDV's spatial generalization improves significantly. With 15 trajectories, the coverage of the workspace remains relatively sparse, leading to invalid executions when the initial pose lies far from the augmented data distribution. When the dataset is expanded to 50 trajectories, the success rate approaches 100%. pose estimation. These results confirm that TDV is capable of achieving spatial generalization from one-shot demonstration, thereby providing strong empirical evidence in support of Question 2.

Table 2: Success rates under varying demonstration counts across object tasks. PG: position generalization, PoG: pose generalization.

Num of Demo	Cup-Coaster		Stitch-Card		Pour-Water	
	PG	PoG	PG	PoG	PG	PoG
1	6/13	6/13	4/13	5/13	4/13	1/13
15	7/13	11/13	10/13	8/13	12/13	9/13
50	**13/13**	**12/13**	**12/13**	**12/13**	**13/13**	**9/13**

4.2.3 Visual Domain Generalization

For visual domain generalization, we primarily investigate three aspects: background variation, viewpoint variation, and cross-embodiment transfer. To evaluate background robustness, we collect human demonstration videos in environments with backgrounds distinct from the robot workspace,

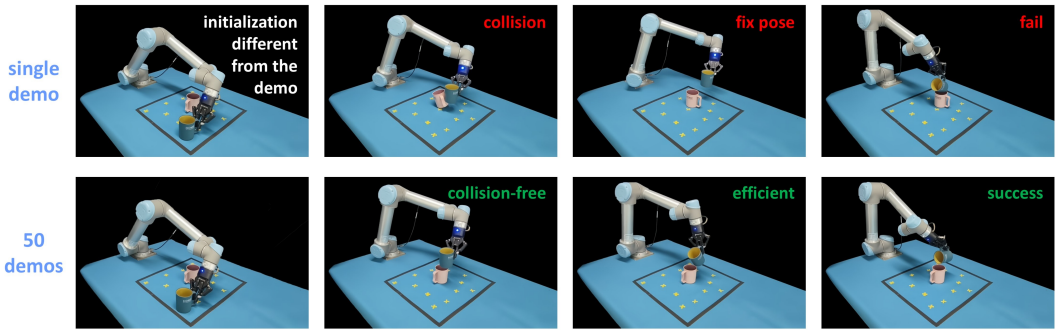

Figure 3: **Effect of data quantity on TDV generalization.** Both sequences start from an unseen test scenario. The top row, trained on a single demonstration, exhibits failure modes such as collisions and overfitting to the training trajectory. In contrast, the bottom row, trained on 50 augmented demonstrations, demonstrates reliable and efficient performance in completing the pouring task.

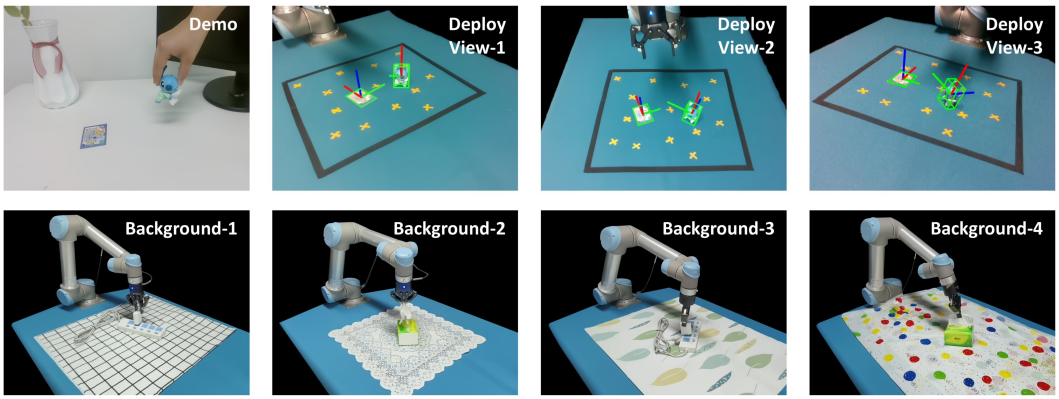

Figure 4: **Visual domain variation.** The top row shows the viewpoint of the human video demonstration, along with three different deployment viewpoints. The bottom row illustrates four types of background perturbations introduced during deployment.

and introduce four types of background perturbations. For viewpoint variation, we conduct evaluations using three camera viewpoints that are distinct from those used during data collection, as illustrated in Figure 4. For each visual domain variation, we perform five trials with randomized poses, and the results are summarized in the supplementary table. Overall, our method is robust to variations in camera viewpoints and background conditions. Failures occur primarily when the camera adopts a steep top-down angle, making it difficult to accurately localize small objects, such as the Stitch toy.

5 Conclusion

We propose a trajectory-based diffusion framework, TDV, which enables generalizable manipulation across different embodiments from one-shot human demonstrations. Through extensive simulation and real-world experiments, we validate that trajectories serve as an effective intermediate representation, allowing efficient data generation for spatial generalization. Furthermore, the modality-agnostic nature of trajectories contributes to strong robustness across visual domains, enabling consistent performance under varying camera viewpoints and background conditions.

However, our approach relies heavily on accurate 6D pose estimation, which limits its applicability to objects with complex geometry or deformable structures that are difficult to track reliably.

References

[1] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023.

[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 *Robotics: Science and Systems*, 2023.

[3] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. In *Conference on Robot Learning (CoRL)*, 2024.

[4] L. Smith, N. Dhawan, M. Zhang, P. Abbeel, and S. Levine. AVID: Learning Multi-Stage Tasks via Pixel-Level Translation of Human Videos. In *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020. doi:10.15607/RSS.2020.XVI.024.

[5] S. Bahl, A. Gupta, and D. Pathak. Human-to-robot imitation in the wild. 2022.

[6] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song. Flow as the cross-domain manipulation interface. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=cNI0ZkK1yC.

[7] C. Yuan, C. Wen, T. Zhang, and Y. Gao. General flow as foundation affordance for scalable robot learning. *CoRR*, abs/2401.11439, 2024.

[8] H. Bharadhwaj, D. Dwibedi, A. Gupta, S. Tulsiani, C. Doersch, T. Xiao, D. Shah, F. Xia, D. Sadigh, and S. Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *CoRR*, abs/2409.16283, 2024.

[9] C. Wen, X. Lin, J. I. R. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point Trajectory Modeling for Policy Learning. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi:10.15607/RSS.2024.XX.092.

[10] Z. Xue, S. Deng, Z. Chen, Y. Wang, Z. Yuan, and H. Xu. Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning. *arXiv preprint arXiv:2502.16932*, 2025.

[11] K. Rana, J. Abou-Chakra, S. Garg, R. Lee, I. D. Reid, and N. Sünderhauf. Affordance-centric policy learning: Sample efficient and generalisable robot policy learning using affordance-centric task frames. *CoRR*, abs/2410.12124, 2024.

[12] C.-C. Hsu, B. Wen, J. Xu, Y. Narang, X. Wang, Y. Zhu, J. Biswas, and S. Birchfield. Spot: Se (3) pose trajectory diffusion for object-centric manipulation. *arXiv preprint arXiv:2411.00965*, 2024.

[13] M. Pan, J. Zhang, T. Wu, Y. Zhao, W. Gao, and H. Dong. Omnimanip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints. *CoRR*, abs/2501.03841, 2025.

[14] 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. In *7th Annual Conference on Robot Learning*, 2023.

[15] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. Fan, and Y. Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. In *CoRL Workshop on Learning Robot Fine and Dexterous Manipulation: Perception and Control*, 2024. URL https://openreview.net/forum?id=KgUgavAl6Y.

[16] G. Pavlakos, D. Shan, I. Radosavovic, A. Kanazawa, D. Fouhey, and J. Malik. Reconstructing hands in 3D with transformers. In *CVPR*, 2024.

[17] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

[18] K. Shaw, S. Bahl, and D. Pathak. Videodex: Learning dexterity from internet videos. In *CoRL*, volume 205 of *Proceedings of Machine Learning Research*, pages 654–665. PMLR, 2022.

[19] Y. Ze, Y. Liu, R. Shi, J. Qin, Z. Yuan, J. Wang, and H. Xu. H-index: Visual reinforcement learning with hand-informed representations for dexterous manipulation. In *NeurIPS*, 2023.

[20] E. Chane-Sane, C. Schmid, and I. Laptev. Learning video-conditioned policies for unseen manipulation tasks. In *ICRA*, pages 909–916. IEEE, 2023.

[21] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3M: A universal visual representation for robot manipulation. In *CoRL*, volume 205 of *Proceedings of Machine Learning Research*, pages 892–909. PMLR, 2022.

[22] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. In *CoRL*, volume 205 of *Proceedings of Machine Learning Research*, pages 416–426. PMLR, 2022.

[23] K. G. et al. Ego4d: Around the world in 3, 000 hours of egocentric video. In *CVPR*, pages 18973–18990. IEEE, 2022.

[24] S. Kumar, J. Zamora, N. Hansen, R. Jangir, and X. Wang. Graph inverse reinforcement learning from diverse videos. In *CoRL*, volume 205 of *Proceedings of Machine Learning Research*, pages 55–66. PMLR, 2022.

[25] M. Sieb, X. Zhou, A. Huang, O. Kroemer, and K. Fragkiadaki. Graph-structured visual imitation. In *CoRL*, volume 100 of *Proceedings of Machine Learning Research*, pages 979–989. PMLR, 2019.

[26] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak. Affordances from human videos as a versatile representation for robotics. In *CVPR*, pages 1–13. IEEE, 2023.

[27] Y. Zhu, Z. Jiang, P. Stone, and Y. Zhu. Learning generalizable manipulation policies with object-centric 3d representations. *arXiv preprint arXiv:2310.14386*, 2023.

[28] Y. Li and D. Pathak. Object-aware gaussian splatting for robotic manipulation. In *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024.

[29] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann. Neural descriptor fields: Se(3)-equivariant object representations for manipulation. In *ICRA*, pages 6394–6400. IEEE, 2022.

[30] J. Kerr, C. M. Kim, M. Wu, B. Yi, Q. Wang, K. Goldberg, and A. Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. *CoRR*, abs/2409.18121, 2024.

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

[32] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[33] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

[34] J. K. S. B. Bowen Wen, Wei Yang. FoundationPose: Unified 6d pose estimation and tracking of novel objects. In *CVPR*, 2024.

[35] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

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

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[38] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics Autom. Lett.*, 5(2):3019–3026, 2020.

[39] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox. Rvt-2: Learning precise manipulation from few demonstrations. *arXiv preprint arXiv:2406.08545*, 2024.

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