Generative Visual Foresight Meets Task-Agnostic Pose Estimation in Robotic Table-top Manipulation

Chuye Zhang[*1] **Xiaoxiong Zhang**[*1] **Wei Pan**[1] **Linfang Zheng**[†2,3] **Wei Zhang**[†1,2]

[1]Southern University of Science and Technology
[2]LimX Dynamics
[3]The University of Hong Kong
{12110807, 12433017, 12211810}@mail.sustech.edu.cn
lfzheng@hku.hk, zhangw3@sustech.edu.cn

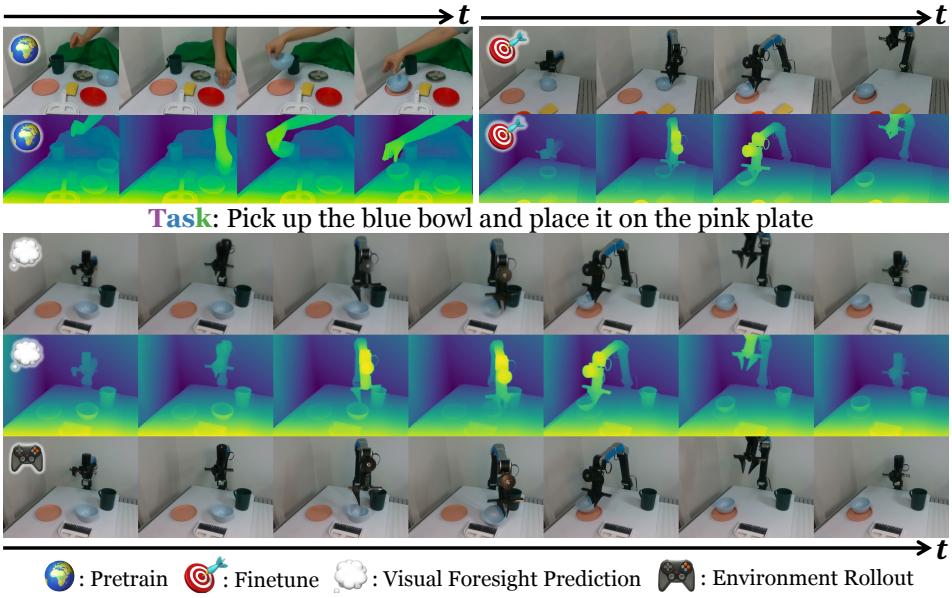


Figure 1: High-level illustration of GVF-TAPE. Given a single RGB observation and a task description, GVF-TAPE predicts future RGB-D frames via a generative foresight model. A decoupled pose estimator then extracts end-effector poses, enabling closed-loop manipulation without action labels.

Abstract:

Robotic manipulation in unstructured environments requires systems that can generalize across diverse tasks while maintaining robust and reliable performance. We introduce GVF-TAPE, a closed-loop framework that combines generative visual foresight with task-agnostic pose estimation to enable scalable robotic manipulation. GVF-TAPE employs a generative video model to predict future RGB-D frames from a single side-view RGB image and a task description, offering visual plans that guide robot actions. A decoupled pose estimation model then extracts end-effector poses from the predicted frames, translating them into executable commands via low-level controllers. By iteratively integrating video foresight and pose estimation in a closed loop, GVF-TAPE achieves real-time, adaptive manipulation across a broad range of tasks. Extensive experiments in both simulation and real-world settings demonstrate that our approach reduces reliance on task-specific action data and generalizes effectively, providing a practical and scalable solution for intelligent robotic systems. The video and code can be found at https://clearlab-sustech.github.io/gvf-tape/.

Keywords: Robotic Manipulation, Action-Label-Free, Generative Foresight

[*]Equal contribution, order by dice rolling.
[†]The corresponding authors.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

1 Introduction

Humans develop an intuitive understanding of hand kinematics through continuous interaction with their environment [1, 2]. Studies have highlighted the strong sensory coupling between vision and body awareness [3, 4], enabling people to predict the visual consequences of their actions before executing them. Inspired by this capability, we propose to enable robots to **imagine future visual scenes and infer their end-effector states to guide actions**. This insight motivates the design of **GVF-TAPE** (**G**enerative **V**isual **F**oresight with **T**ask-**A**gnostic **P**ose **E**stimation), a closed-loop framework that combines generative video prediction with task-agnostic pose estimation to achieve scalable, real-time robotic manipulation.

Recent advances in robotic manipulation leverage large-scale, vision-language-action models [5, 6, 7, 8, 9]. However, scaling such models is challenging due to the cost and effort required for human-annotated demonstrations. To address this, action-free datasets have gained increasing attention. Some approaches learn general representations for policy learning [10, 11] or label action-free dataset with latent action [12], while others guide actions by predicting intermediate visual cues such as future frames [13, 14, 15], point tracks [16, 17] and sphere pose [18]. Despite these advances, many methods still depend on task-specific action supervision during downstream learning or require rigid setups, limiting their scalability and adaptability. Recent efforts to eliminate action labels through dense correspondence [19], goal-conditioned exploration [20], or stereo-based pose estimation [21] have shown promise, but often face challenges related to real-world flexibility, data collection efficiency, or closed-loop deployment. These limitations motivate the need for a **task-agnostic**, **action-label-free framework** that can plan through future visual prediction and execute actions reliably in real-time, without relying on specialized hardware or task-specific supervision.

In this work, we introduce **GVF-TAPE**, a novel video-based framework for robotic manipulation that decouples the phases of visual planning and action execution. Our approach leverages a **generative video model** to predict future RGB-D frames from a single side-view RGB image and a task description, providing a rich visual plan for decision making. A **task-agnostic pose estimation model** then extracts 6-DoF end-effector poses from the generated frames and translates them into executable actions through low-level controllers via inverse kinematics. Crucially, the pose estimation model is trained solely on random exploration data, making it simple to collect and scalable across different robots and environments. By integrating video foresight and task-agnostic pose estimation in a closed-loop system, GVF-TAPE enables robust, real-time manipulation across a wide range of tasks. Extensive experiments in both simulation and real-world settings demonstrate that our method matches or outperforms prior video-pretrained, action-labeled, and self-exploration-based approaches while requiring significantly less task-specific data.

The main contributions of this work are:

- We propose GVF-TAPE, a closed-loop, action-label-free framework that combines generative visual foresight and task-agnostic pose estimation for real-time robotic manipulation.
- We develop a scalable training pipeline by leveraging random exploration data for pose learning and large-scale video pretraining for foresight, eliminating the need for expert-labeled demonstrations.
- We demonstrate that GVF-TAPE achieves real-time deployment in both simulation and real-world environments, and significantly outperforms prior action-labeled, video-pretrained, and self-exploration-based methods across diverse manipulation tasks.

2 Related Work

Visual foresight for robotic manipulation. The research on visual foresight models have become a hotspot for robotic manipulation by using it as auxiliary loss, guidance feature or sub-goal. [14, 19, 13, 16, 17, 21, 22, 23, 24]. Approaches like [23, 25, 24] integrate visual foresight as auxiliary loss for policy learning to obtain better dynamics comprehension. Methods like [16, 17, 22, 14, 13]

choose to train a model that generates temporal feature like point track [16, 17], sphere pose [22] or sub-goal image [14, 13, 15] to guide the policy learning. While these methods exploited visual foresight to enhance policy learning, they still rely on action-labeled data to train a inverse dynamic model mapping visual foresight to executable action. Methods like [19, 20, 21] bridged this gap by eliminating the need for action-labeled data. AVDC [19] uses optical flow and dense matching to obtain action, suffering from manipulation precision and dependence on the robot mask. V2A [20] obtains an inverse dynamic model by self-exploration and bootsrapping, facing challenges in task specificity, data-efficiency and real-world safety constraints. Dreamitate [21] estimates the pose of the robotic arm in the predicted video, which requires manipulator CAD model, stereo setup and camera calibration, as well as struggles in inference time and precision. Our research aims to develop an agile and close-loop video prediction and execution framework, additionally it's easy-to-obtain, less-dependent and practical.

Pose Estimation in Robotics Pose estimation has been extensively studied in both Computer Vision and Robotics. Object pose estimation can generally be categorized into instance-level pose estimation, which requires CAD models [26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38], and category-level pose estimation, which generalizes across object instances within a given category [39, 40, 41, 42, 43, 44]. In the context of tabletop manipulators and articulated robots, important pose estimation tasks include object poses, end-effector poses, and joint angles (1D poses) [19, 45, 46] Keypoint-based approaches have also been widely adopted for estimating 6D camera-to-robot poses and joint angles [47, 48, 49, 46, 50]. Additionally, [51] introduced a render-and-compare method that overlays articulated CAD models for pose estimation. Foundation pose model [52] create a general and easy-to-adopt workflow for object pose estimation. However, the constrain to articulated objects makes it less convenient in estimating gripper aperture. FEEPE [53] construct a training free foundation model for robot end effector pose estimation. However, we focus on end-to-end robot-centric approach which utilize extensive proprioception data which are more adaptable for certain manipulator embodiment. Similarly, [54] also use deep neural network for end effector pose estimation. Keeping the advantages of end-to-end approaches, it needs additional ground truth depth information. In contrast to these approaches, our work employs a lightweight, end-to-end deep learning model that requires only easily and automatically collected random exploration data, and synthesized depth generated by ready-to-use Video Depth Anything Model [55], making it both efficient and practical for real-world robotic applications.

3 Method

3.1 Problem Formulation

Our goal is to develop a closed-loop robotic manipulation system for tabletop environments that combines visual foresight with task-agnostic pose estimation. Given a single side-view RGB observation x_0 of the scene and a task description c, the system predicts a sequence of future robot actions in the form of end-effector poses. Specifically, the system generates a pose trajectory $\mathcal{T} = T_1, T_2, \ldots, T_h$, where each $T_i = (\mathbf{p}_i, \mathbf{q}_i, g_i)$ consists of the 3D position $\mathbf{p}_i \in \mathbb{R}^3$, the orientation $\mathbf{q}_i \in \mathbb{R}^4$ represented as a unit quaternion, and the gripper state $g_i \in [0, 1]$ indicating the gripper opening. Thus, we aim to learn a mapping function $f : (x_0, c) \rightarrow \mathcal{T}$ that allows the robot to execute tasks robustly while continuously adapting to dynamic environments.

3.2 Framework Overview

We propose GVF-TAPE (Generative Visual Foresight and Task-Agnostic Pose Estimation model), a decoupled two-stage framework for closed-loop robotic manipulation, as illustrated in Fig. 2. GVF-TAPE plans directly in the visual space by first predicting future observations and then inferring the corresponding end-effector poses through task-agnostic pose estimation. This design enables greater generalization and eliminates the need for expert demonstrations.

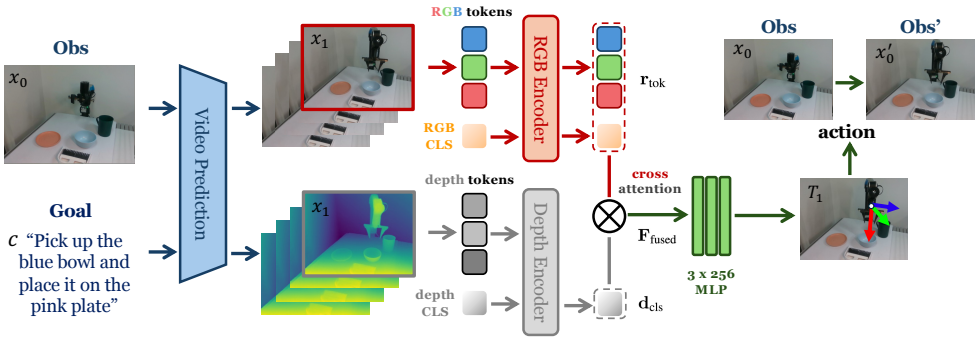


Figure 2: Framework Overview. GVF-TAPE first generates a future RGB-D video conditioned on the current RGB observation and task description. A transformer-based pose estimation model then extracts the end-effector pose from each predicted frame and sends it to a low-level controller for execution. After completing the predicted trajectory, the system receives a new observation and repeats the process in a closed-loop manner.

3.3 Text-Conditioned Visual Foresight for RGB-D Prediction

Our visual foresight module predicts future RGB-D frames conditioned on the current RGB observation x_0 and a task description c. Prior models [14, 19] predict only RGB frames, while CLOVER [15] generates RGB-D but requires explicit depth inputs, limiting scalability. In contrast, our approach infers depth implicitly, enabling training and deployment without depth sensors. Using off-the-shelf depth estimators [55], we also enable pretraining on large-scale RGB-only datasets.

To model the conditional distribution $p(x_{1:h} \mid x_0, c)$ efficiently, we adopt rectified flow [56, 57], which transforms an initial noisy sequence $x^1_{1:h} \sim \mathcal{N}(0, I)$ toward a clean video prediction $x^0_{1:h}$ following:

$$dx^t_{1:h} = (x^0_{1:h} - x^1_{1:h})dt, \tag{1}$$

where $x^t_{1:h}$ interpolates between noise and ground truth. The velocity model v_θ is trained to predict the displacement between the noisy and clean sequences by minimizing:

$$\mathcal{L} = \|v_\theta(x^t_{1:h}, x_0, c, t) - (x^0_{1:h} - x^1_{1:h})\|^2, t \sim U(0, 1). \tag{2}$$

Here, $x^t_{1:h}$ is a linear interpolation between $x^0_{1:h}$ and $x^1_{1:h}$, given by $x^t_{1:h} = tx^1_{1:h} + (1-t)x^0_{1:h}$, with t denoting the noise level.

Since future prediction requires modeling both spatial and temporal dynamics, we adopt a lightweight 3D U-Net [19] as the backbone for the velocity model v_θ, and encode the task description c using a CLIP text encoder [58]. This architecture enables efficient and scalable visual foresight for real-time robotic planning.

3.4 Task-Agnostic Pose Estimation Model

To translate the generated video frames into executable robot actions, we employ a task-agnostic pose estimation model. Unlike previous methods [14, 13, 19] that rely on inverse dynamics and temporal dependencies, our approach processes each frame independently, improving flexibility and generalization across different tasks.

Given an RGB image x_i and its corresponding depth map from the foresight model, the pose estimator π_ϕ predicts the end-effector pose $T_i = (\mathbf{p}_i, \mathbf{q}_i, g_i)$, as defined in Section 3.1. The model is trained to minimize a Smooth L1 loss:

$$\mathcal{L} = \text{SmoothL1}(\pi_\phi(x_i) - T_i). \tag{3}$$

To fuse RGB and depth information effectively, we adopt two separated pretrained ViT-base encoders [59] for RGB and depth modalities, and apply a multi-head cross-attention mechanism:

$$\mathbf{f}_{\text{fused}} = \mathcal{A}\left(\mathbf{Q} = \mathbf{d}_{\text{cls}},\ \mathbf{K} = \mathbf{r}_{\text{tok}},\ \mathbf{V} = \mathbf{r}_{\text{tok}}\right), \tag{4}$$

where $\mathbf{f}_{\text{fused}}$ denotes the fused feature representation, and $\mathcal{A}$ is the multi-head attention module. The query $\mathbf{Q}$ is the CLS token from the depth encoder ($\mathbf{d}_{\text{cls}}$), while the keys and values $(\mathbf{K}, \mathbf{V})$ are patch

tokens from the RGB image ($\mathbf{r}_{\text{tok}}$), including its CLS token $\mathbf{r}$cls. Further architectural details are provided in the Supplementary Material.

To collect training data for the pose estimation model, we use a random exploration strategy: sampling $T_i = (\mathbf{p}_i, \mathbf{q}_i, g_i)$ uniformly within a predefined workspace range. An off-the-shelf controller drives the robot to each sampled pose, with safety constraints enforced in real-world settings. Details are included in the Supplementary Material.

4 Experiment

We evaluate GVF-TAPE through extensive simulation and real-world experiments to answer the following key questions: (1) How does our approach, trained on random exploration data, compare with state-of-the-art video pre-training imitation learning methods that require action labels? (2) How does it compare with other video prediction methods that map future or goal images to actions? (3) Can GVF-TAPE benefit from pre-train on external video data (*e.g.*, human manipulation videos)? (4) How effective are our design choices, such as rectified flow and depth inference?

Datasets. We evaluate GVF-TAPE in both simulated and real-world settings. In simulation, we adopt the LIBERO benchmark [60]—a suite of language-conditioned manipulation tasks designed for benchmarking generalizable robotic agents. Detailed information about LIBERO is provided in Sec.7.2. For training the task-agnostic pose estimation model, we generate over 400k RGB-D/pose pairs per task suite via simulation. The datasets used to train the video generation model in simulation are described in Sec.4.2 and Sec.4.1. For real-world experiments, we collect 18k RGB-D/pose pairs through random exploration (Sec.3.4) for pose estimation, and acquire 20 teleoperated demonstrations per task to train the video generation model.

Baselines. We compare GVF-TAPE against two categories of prior work. First, we evaluate against video pretraining methods including *R3M-finetune* [10], *VPT* [61], *UniPi* [14], and *ATM* [16], all of which rely on action-labeled demonstrations for policy learning. Second, we compare with video prediction-based approaches such as *DP* [62], *GCDP* [20], *AVDC* [19], *SuSIE* [13], and *V2A* [20], which learn from videos by predicting intermediate visual representations or subgoals. Notably, unlike V2A, our method does not require goal-conditioned exploration and can be trained entirely offline. Baseline training and evaluation protocols follow those reported in [16, 20].

Method	Side View	Eye-in-hand View	Action Data	Libero-Spatial	Libero-Object	Libero-Goal	Overall
R3M-finetune	✓	✓	20%	49.17 ± 3.79	52.83 ± 8.2	59.2 ± 7.80	53.73 ± 8.04
VPT	✓	✓	20%	37.83 ± 4.29	19.50 ± 0.82	3.33 ± 2.36	20.22 ± 14.37
UniPi	✓	✓	20%	69.17 ± 3.75	59.83 ± 3.01	11.83 ± 2.02	46.94 ± 25.30
ATM	✓	✓	20%	68.50 ± 1.78	68.00 ± 6.18	$\mathbf{77.83 \pm 0.82}$	71.44 ± 5.87
GVF-TAPE(Ours)	✓	✗	0%	$\mathbf{95.50 \pm 0.87}$	$\mathbf{86.70 \pm 1.26}$	66.80 ± 2.00	$\mathbf{83.00 \pm 12.01}$

Table 1: Performance comparison with state-of-the-art methods across three LIBERO evaluation suites. Success rates (mean ± standard deviation) are reported over three random seeds. GVF-TAPE achieves the highest performance on two of the three suites and outperforms the next-best overall average by 11.56%.

4.1 Comparison with Video Pre-training Methods

To evaluate the effectiveness of our proposed method, we compare our method with state-of-the-art video pre-training imitation learning approaches [10, 14, 16, 61] on LIBERO-spatial, LIBERO-object, and LIBERO-goal, covering a total of 30 language-conditioned manipulation tasks. For baselines, each task is trained with 50 video demonstrations and 10 action-labeled trajectories.

The results, presented in Table 1, show that our method (GVF-TAPE) outperforms all baselines requiring action-labeled data in LIBERO-spatial and LIBERO-object, achieving 27.00% and 18.70% performance gains, respectively. In LIBERO-goal, our method ranks second, being 11.03% lower than the ATM. Upon further analysis, we found that tasks in LIBERO-goal often require precise manipulation in gripper occluded scenes, such as opening drawers.

Task	DP*	GCDP*	SuSIE*	AVDC	V2A w/ SuSIE	V2A	Ours
LR-Scene5-put-red-mug-left	33.6 ± 3.2	24.8 ± 4.7	18.4 ± 2.0	0.0 ± 0.0	23.2 ± 3.0	38.4 ± 15.3	**83.6 ± 6.2**
LR-Scene5-put-red-mug-right	33.6 ± 8.2	22.4 ± 7.4	32.0 ± 8.4	0.0 ± 0.0	**60.0 ± 6.7**	40.8 ± 7.8	56.0 ± 5.7
LR-Scene5-put-white-mug-left	59.2 ± 7.8	16.0 ± 8.8	43.2 ± 4.7	0.0 ± 0.0	**68.8 ± 4.7**	51.2 ± 3.9	64.0 ± 8.0
LR-Scene5-put-Y/W-mug-right	57.6 ± 5.4	3.2 ± 3.0	25.6 ± 11.5	0.0 ± 0.0	**67.2 ± 8.9**	38.4 ± 8.6	60.0 ± 3.6
LR-Scene6-put-choc-left	42.4 ± 5.4	45.6 ± 6.0	17.6 ± 9.3	1.3 ± 1.9	44.0 ± 7.6	70.4 ± 12.8	**96.8 ± 1.6**
LR-Scene6-put-choc-right	50.4 ± 5.4	32.0 ± 8.8	32.8 ± 9.9	0.0 ± 0.0	54.4 ± 5.4	79.2 ± 3.9	**92.8 ± 4.8**
LR-Scene6-put-red-mug-plate	32.8 ± 9.3	5.6 ± 4.1	16.0 ± 2.5	0.0 ± 0.0	66.4 ± 12.0	72.8 ± 6.4	**90.4 ± 3.2**
LR-Scene6-put-white-mug-plate	71.2 ± 5.3	7.2 ± 6.4	10.4 ± 4.1	0.0 ± 0.0	36.0 ± 7.6	25.6 ± 11.5	**91.6 ± 1.7**
Overall	47.6 ± 13.4	19.6 ± 13.7	19.2 ± 6.5	0.3 ± 0.5	52.5 ± 16.6	52.1 ± 19.7	**79.4 ± 16.6**

Table 2: Comparison of methods on eight tasks in two LIBERO-100 living room scenes. * indicates use of action-label expert demos. Ours outperforms the second-best by 26.9%.

Our current setup uses a single fixed camera, which can limit visibility of fine-grained interactions (see Fig. 3 (c–d)). Incorporating wrist-mounted or multi-view inputs may help mitigate this limitation, which we leave for future work.

Despite these challenges, GVF-TAPE achieves the best average performance across all suites, surpassing ATM by 11.56%. These results show that our framework can achieve competitive or superior performance compared to action-labeled methods, without requiring any expert actions, and highlight its potential for scalable and label-free robot learning.

Figure 3: Challenging scenarios in LIBERO. The left two panels show tasks from LIVING-ROOM-SCENE-5, where the robot's end effector moves outside the camera's field of view, making pose estimation unreliable. The right two panels illustrate limited gripper visibility from a fixed side-view camera, which affects accuracy in fine-grained tasks from LIBERO-Goal.

Data Efficiency Reducing reliance on large quantities of robot data is increasingly important due to the cost of human teleoperation and annotation.

To evaluate GVF-TAPE's data efficiency, we pretrain the video generation model on LIBERO-90 and fine-tune it on LIBERO-Spatial, LIBERO-Object, and LIBERO-Goal using 20%, 50%, and 90% of available task data. We assess both video generation quality, using LPIPS and SSIM metrics, and downstream task performance, comparing against models trained from scratch. As shown in Fig. 5, the pretrained model consistently outperforms the scratch model across all data proportions, demonstrating that pretraining on external video sources improves video fidelity. For task success, as depicted in Fig. 4 GVF-TAPE achieves 68% with only 20% of demonstration data (10 demonstrations per task), and further improves to 77% when pretrained on LIBERO-90, surpassing the previous state-of-the-art by 5.43%. These results highlight the strong data efficiency and transferability of our approach.

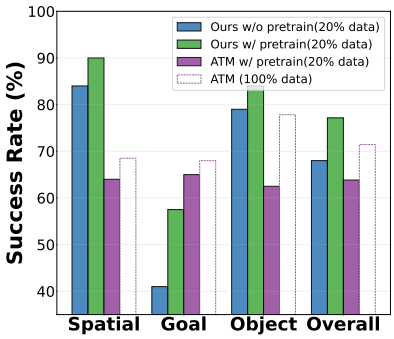


Figure 4: Performance of our method with and without pretraining. Using only 20% of the video data, our method matches prior SOTA (ATM); pretraining on LIBERO-90 boosts performance by 9.2%, outperforming ATM by 5.43%.

4.2 Comparison with Video Prediction Methods

We further compare GVF-TAPE against video prediction-based approaches [20, 19, 13] on eight tasks from two living room scenes in the LIBERO-100 suite, following the evaluation protocol in [20]. Each method is trained on 20 video demonstrations per task (160 total). Baselines include DP [62], GCDP [20], and SuSIE [13], which rely on action-labeled data, and V2A [20] and AVDC [19], which eliminate action labels through goal-conditioned exploration or dense matching.

As shown in Table 2, GVF-TAPE achieves the highest performance in 5 tasks and ranks second in the remaining 3. In the 3 tasks, there exists some challenging scenario for our method like robot reach out of camera, we summarize these situation in Fig. 3. On average, our overall performance surpasses the second-best approach by 26.9%. The performance of DP, GCDP, and SuSIE appears

relatively low, which may be attributed to their reliance on action-labeled data. Given that only 20 demonstrations per task are available in this experiment, the limited supervision may constrain their effectiveness. These results demonstrate the strong generalization ability of GVF-TAPE across diverse manipulation tasks. Moreover, unlike V2A, which requires costly online exploration for each task, GVF-TAPE operates fully offline by learning from random exploration data, offering improved efficiency and scalability.

Although GVF-TAPE performs robustly across most tasks, some failure cases occur when the end-effector moves outside the camera field of view, particularly in LIVING-ROOM-SCENE-5 (Fig. 3 (a–b)). Addressing this via multi-view setups or improved pose estimation is left for future work.

Overall, GVF-TAPE provides a flexible and scalable alternative to video prediction frameworks, fully eliminating the need for action labels or goal-conditioned exploration.

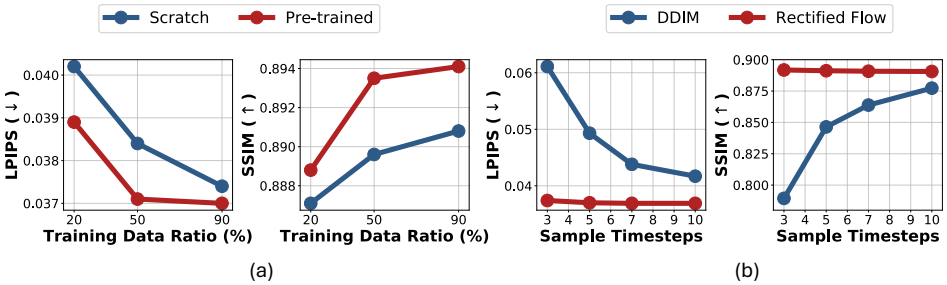


Figure 5: Pretraining and model choice critically affect video generation quality and efficiency. (a) Pretrained models consistently outperform models trained from scratch across different proprioception data ratios. (b) While diffusion improves with more sampling steps, it incurs high inference cost; rectified flow achieves strong results with just three steps, motivating our design choice.

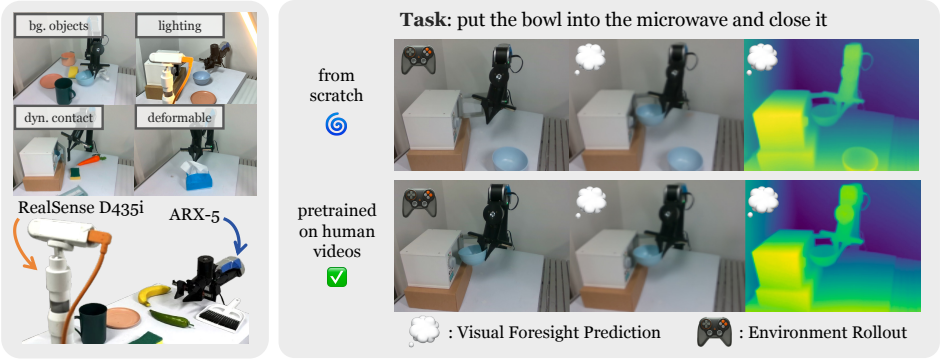
4.3 Real World Performance

We evaluate GVF-TAPE on five real-world tasks involving rigid, deformable, and articulate objects as shown in Fig. 6. The tasks include: 1) pick up the blue bowl and place it on the pink plate, 2) grab a tissue, 3) place the sponge on the plate. 4) put the blue bowl into the microwave and close it, and 5) put the pepper in the basket. For each task, we conduct 10 independent trials, with success rates summarized in Tab. 3. GVF-TAPE achieve an average success rate of 56% across all tasks with only 20 video per task, using the same task-agnostic pose estimation model without task-specific fine-tuning. Notably, the objects and configurations encountered during evaluation differ from those seen during random exploration training. Despite this domain shift, GVF-TAPE demonstrates strong generalization to unseen object positions, highlighting its robustness and practicality for real-world deployment.

Task	Ours w/o pt.	Ours w/ pt.
put-bowl-plate	80%	**100%**
grab-tissue	30%	**70%**
put-sponge-plate	70%	**90%**
bowl-into-micro.	60%	**100%**
pepper-in-basket	40%	**70%**
average	56%	**86%**

Table 3: Real-world task success rates of our method, with and without pretraining on human hand data. Pretraining leads to consistently higher performance, reaching 100% success on several tasks and boosting the overall average by 30%.

Cross-embodiment Transfer. We investigate whether GVF-TAPE can leverage human demonstration videos to improve robot manipulation while reducing reliance on robot-specific data. To this end, we pre-train the video generation module using 50 additional human hand manipulation videos per task, followed by fine-tuning on robot data. This cross-embodiment pre-training improves the model's ability to capture task-relevant visual structure and generalize across varying spatial configurations. As shown in Fig. 6, it reduces hallucinations and enhances real-world robustness. Table 3 reports consistent performance gains, demonstrating that GVF-TAPE can effectively transfer knowledge from human to robot domains and improve generalization with limited robot data.



(a) (b)

Figure 6: (a) Real-world setup. We use an ARX-5 robotic arm equipped with a fixed side-view Intel RealSense D435i camera. The evaluation environment includes dynamic contacts, deformable objects, background clutter, and varying lighting conditions. **(b) Effect of human video pre-training.** Pre-training on human hand manipulation videos significantly reduces hallucinations and improves prediction stability.

4.4 Ablation Study

Effect of Rectified Flow. To validate our choice of rectified flow [56] for video generation, we compare it with diffusion-based approaches [63, 64] used in prior work [19, 14, 20]. We evaluate structural similarity (SSIM) and perceptual similarity (LPIPS) across LIBERO-Spatial, LIBERO-Object, and LIBERO-Goal, averaging the metrics across suites (Fig. 5). To accelerate diffusion sampling, we adopt DDIM [64]. As shown, while increasing sampling steps improves diffusion video quality, it significantly increases inference time. In contrast, rectified flow achieves comparable video quality with only three steps, drastically reducing latency. This efficiency is critical for real-time closed-loop deployment. Detailed timing results are provided in the appendix.

Effect of integrating monocular depth estimation. We evaluate the impact of incorporating relative depth by comparing GVF-TAPE under two settings: one using RGB-D video generated with supervision from a monocular depth estimator [55], and the other using RGB-only video when depth estimation is unavailable. As shown in Table 4, integrating depth consistently improves performance across all test environments, with particularly notable gains in spatially complex tasks. Additional experimental results and analyses are included in the supplementary material.

Method	Video Depth Anything	Libero-Spatial	Libero-Object	Libero-Goal	Overall
Ours w/o depth	✗	91.83 ± 1.52	80.33 ± 3.33	56.5 ± 0.00	76.22 ± 14.71
Ours w/ depth	✓	$\mathbf{95.50 \pm 0.87}$	$\mathbf{86.70 \pm 1.26}$	$\mathbf{66.8 \pm 2.00}$	$\mathbf{83.00 \pm 12.01}$

Table 4: Performance comparison on three test suites using RGB-D vs. RGB-only input in GVF-TAPE. Incorporating relative depth significantly boosts performance across all cases, highlighting the benefit of depth information.

5 Conclusion

We present GVF-TAPE, a real-time manipulation framework that decouples visual planning from action execution by combining generative video prediction with task-agnostic pose estimation. Unlike prior methods, GVF-TAPE learns from unlabeled videos and random exploration, removing the need for action-labeled data. This design allows robots to predict future visual outcomes and infer executable poses, enabling robust closed-loop control across diverse tasks. Experiments in both simulation and the real world show that GVF-TAPE outperforms action-supervised and video-based baselines, demonstrating the potential of label-free, foresight-driven frameworks for scalable manipulation. We hope this work encourages further research in video-guided, action-free robot learning.

6 Limitations and Future Works

While GVF-TAPE achieves strong performance, several limitations remain. First, the system relies exclusively on visual feedback, omitting dynamic signals such as force or tactile feedback that are critical for stable contact-rich manipulation. Incorporating additional sensing modalities, such as proprioception or touch, could improve robustness and interaction awareness. Second, our current single-view video generation model may struggle with fine-grained spatial reasoning due to limited scene coverage. Multi-view foresight could help resolve occlusions and improve accuracy in cluttered or partially observed environments. Finally, although Rectified Flow provides fast and high-quality video prediction, further architectural or optimization improvements could reduce inference latency and enable more agile closed-loop control.

Acknowledgments

This work was supported by the Guangdong Science and Technology Program under Grant No. 2024B1212010002.

References

[1] A. Pilacinski, A. Vandenberghe, G. Andrietta, and G. Vannuscorps. Humans underestimate the movement range of their own hands. *Communications Psychology*, 2(1):104, 2024. ISSN 2731 - 9121. doi:10.1038/s44271-024-00153-x. URL https://doi.org/10.1038/s44271-024-00153-x.

[2] K. C. Dieter, B. Hu, D. C. Knill, R. Blake, and D. Tadin. Kinesthesis can make an invisible hand visible. *Psychological Science*, 25(1):66 – 75, 2014. doi:10.1177/0956797613497968. URL https://doi.org/10.1177/0956797613497968.

[3] N. Faivre, R. Salomon, and O. Blanke. Visual consciousness and bodily self - consciousness. *Curr Opin Neurol*, 28(1):23–28, 02 2015. doi:10.1097/WCO.0000000000000160.

[4] T. Yokosaka, S. Kuroki, S. Nishida, and J. Watanabe. Apparent time interval of visual stimuli is compressed during fast hand movement. *PLoS One*, 10(4):e0124901, 04 2015. doi:10.1371/journal.pone.0124901.

[5] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

[6] 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.

[7] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023. URL https://arxiv.org/abs/2212.06817.

[8] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023. URL https://arxiv.org/abs/2307.15818.

[9] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024. URL https://arxiv.org/abs/2410.24164.

[10] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation, 2022. URL https://arxiv.org/abs/2203.12601.

[11] G. Jiang, Y. Sun, T. Huang, H. Li, Y. Liang, and H. Xu. Robots pre-train robots: Manipulation-centric robotic representation from large-scale robot dataset. *arXiv preprint arXiv:2410.22325*, 2024.

[12] S. Ye, J. Jang, B. Jeon, S. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin, L. Liden, K. Lee, J. Gao, L. Zettlemoyer, D. Fox, and M. Seo. Latent action pretraining from videos, 2024. URL https://arxiv.org/abs/2410.11758.

[13] 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, 2023. URL https://arxiv.org/abs/2310.10639.

[14] Y. Du, M. Yang, B. Dai, H. Dai, O. Nachum, J. B. Tenenbaum, D. Schuurmans, and P. Abbeel. Learning universal policies via text-guided video generation. *arXiv e-prints*, pages arXiv–2302, 2023.

[15] Q. Bu, J. Zeng, L. Chen, Y. Yang, G. Zhou, J. Yan, P. Luo, H. Cui, Y. Ma, and H. Li. Closed-loop visuomotor control with generative expectation for robotic manipulation, 2024. URL https://arxiv.org/abs/2409.09016.

[16] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point trajectory modeling for policy learning, 2024. URL https://arxiv.org/abs/2401.00025.

[17] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song. Flow as the cross-domain manipulation interface, 2024. URL https://arxiv.org/abs/2407.15208.

[18] M. Shridhar, Y. L. Lo, and S. James. Generative image as action models, 2024. URL https://arxiv.org/abs/2407.07875.

[19] P.-C. Ko, J. Mao, Y. Du, S.-H. Sun, and J. B. Tenenbaum. Learning to Act from Actionless Videos through Dense Correspondences. *arXiv:2310.08576*, 2023.

[20] Y. Luo and Y. Du. Grounding video models to actions through goal conditioned exploration, 2024. URL https://arxiv.org/abs/2411.07223.

[21] J. Liang, R. Liu, E. Ozguroglu, S. Sudhakar, A. Dave, P. Tokmakov, S. Song, and C. Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation, 2024. URL https://arxiv.org/abs/2406.16862.

[22] M. Shridhar, Y. L. Lo, and S. James. Generative image as action models, 2024. URL https://arxiv.org/abs/2407.07875.

[23] Y. Tian, S. Yang, J. Zeng, P. Wang, D. Lin, H. Dong, and J. Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation, 2024. URL https://arxiv.org/abs/2412.15109.

[24] C.-L. Cheang, G. Chen, Y. Jing, T. Kong, H. Li, Y. Li, Y. Liu, H. Wu, J. Xu, Y. Yang, H. Zhang, and M. Zhu. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation, 2024. URL https://arxiv.org/abs/2410.06158.

[25] 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, 2023. URL https://arxiv.org/abs/2312.13139.

[26] N. Merrill, Y. Guo, X. Zuo, X. Huang, S. Leutenegger, X. Peng, L. Ren, and G. Huang. Symmetry and uncertainty-aware object slam for 6dof object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14901–14910, June 2022.

[27] M. Rad and V. Lepetit. BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects Without Using Depth. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[28] W. Chen, X. Jia, H. J. Chang, J. Duan, and A. Leonardis. G2L-Net: Global to Local Network for Real-Time 6D Pose Estimation With Embedding Vector Features. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[29] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[30] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield. Deep object pose estimation for semantic robotic grasping of household objects, 2018. URL https://arxiv.org/abs/1809.10790.

[31] Y. Su, M. Saleh, T. Fetzer, J. Rambach, N. Navab, B. Busam, D. Stricker, and F. Tombari. Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation, 2022. URL https://arxiv.org/abs/2203.09418.

[32] S. Zakharov, I. Shugurov, and S. Ilic. DPOD: 6D Pose Object Detector and Refiner. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[33] J. Sun, Z. Wang, S. Zhang, X. He, H. Zhao, G. Zhang, and X. Zhou. Onepose: One-shot object pose estimation without cad models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6825–6834, 2022.

[34] H. Chen, P. Wang, F. Wang, W. Tian, L. Xiong, and H. Li. Epro-pnp: Generalized end-to-end probabilistic perspective-n-points for monocular object pose estimation, 2022. URL https://arxiv.org/abs/2203.13254.

[35] R. L. Haugaard and A. G. Buch. Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. *CoRR*, abs/2111.13489, 2021. URL https://arxiv.org/abs/2111.13489.

[36] B. Tekin, S. Sinha, and P. Fua. Real-time seamless single shot 6d object pose prediction. pages 292–301, 06 2018. doi:10.1109/CVPR.2018.00038.

[37] J. Zhou, K. Chen, L. Xu, Q. Dou, and J. Qin. Deep fusion transformer network with weighted vector-wise keypoints voting for robust 6d object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13967–13977, October 2023.

[38] Z. Linfang, L. Ales, T. Tze Ho, Elden, H. Nora, C. Hua, Z. Wei, and C. Hyung Jin. Tp-ae: Temporally primed 6d object pose tracking with auto-encoders. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[39] M. Tian, M. H. Ang, and G. H. Lee. Shape prior deformation for categorical 6d object pose and size estimation. In *European Conference on Computer Vision (ECCV)*, pages 530–546. Springer, 2020.

[40] T. Lee, B. Lee, I. Shin, J. Choe, U. Shin, I. S. Kweon, and K. Yoon. UDA-COPE: unsupervised domain adaptation for category-level object pose estimation. *CoRR*, abs/2111.12580, 2021. URL https://arxiv.org/abs/2111.12580.

[41] D. Chen, J. Li, Z. Wang, and K. Xu. Learning canonical shape space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[42] M. Z. Irshad, S. Zakharov, R. Ambrus, T. Kollar, Z. Kira, and A. Gaidon. Shapo: Implicit representations for multi object shape appearance and pose optimization. 2022. URL https://arxiv.org/abs/2207.13691.

[43] L. Zheng, T. H. E. Tse, C. Wang, Y. Sun, H. Chen, A. Leonardis, and W. Zhang. Georef: Geometric alignment across shape variation for category-level object pose refinement, 2024. URL https://arxiv.org/abs/2404.11139.

[44] L. Zheng, C. Wang, Y. Sun, E. Dasgupta, H. Chen, A. Leonardis, W. Zhang, and H. J. Chang. Hs-pose: Hybrid scope feature extraction for category-level object pose estimation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17163–17173, 2023. doi:10.1109/CVPR52729.2023.01646.

[45] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation, 2024. URL https://arxiv.org/abs/2409.01652.

[46] Y. Zuo, W. Qiu, L. Xie, F. Zhong, Y. Wang, and A. L. Yuille. Craves: Controlling robotic arm with a vision-based, economic system. *CVPR*, 2019.

[47] T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, and S. Birchfield. Camera-to-robot pose estimation from a single image. *CoRR*, abs/1911.09231, 2019. URL http://arxiv.org/abs/1911.09231.

[48] J. Lu, Z. Liang, T. Xie, F. Ritcher, S. Lin, S. Liu, and M. C. Yip. Ctrnet-x: Camera-to-robot pose estimation in real-world conditions using a single camera, 2024. URL https://arxiv.org/abs/2409.10441.

[49] A. Simoni, G. Borghi, L. Garattoni, G. Francesca, and R. Vezzani. D-spdh: Improving 3d robot pose estimation in sim2real scenario via depth data. *IEEE Access*, 12:166660–166673, 2024. doi:10.1109/ACCESS.2024.3492812.

[50] Y. Tian, J. Zhang, G. Huang, B. Wang, P. Wang, J. Pang, and H. Dong. Robokeygen: Robot pose and joint angles estimation via diffusion-based 3d keypoint generation, 2024. URL https://arxiv.org/abs/2403.18259.

[51] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic. Single-view robot pose and joint angle estimation via render & compare, 2021. URL https://arxiv.org/abs/2104.09359.

[52] B. Wen, W. Yang, J. Kautz, and S. Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects, 2024. URL https://arxiv.org/abs/2312.08344.

[53] T. Wu, J. Zhang, S. Liang, Z. Han, and H. Dong. Foundation feature-driven online end-effector pose estimation: A marker-free and learning-free approach, 2025. URL https://arxiv.org/abs/2503.14051.

[54] H. Cheng, Y. Wang, and M. Q.-H. Meng. Real-time robot end-effector pose estimation with deep network. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10921–10926, 2020. doi:10.1109/IROS45743.2020.9341760.

[55] S. Chen, H. Guo, S. Zhu, F. Zhang, Z. Huang, J. Feng, and B. Kang. Video depth anything: Consistent depth estimation for super-long videos. *arXiv:2501.12375*, 2025.

[56] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL https://arxiv.org/abs/2209.03003.

[57] Z. Zheng, X. Peng, T. Yang, C. Shen, S. Li, H. Liu, Y. Zhou, T. Li, and Y. You. Open-sora: Democratizing efficient video production for all, March 2024. URL https://github.com/hpcaitech/Open-Sora.

[58] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

[59] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

[60] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL https://arxiv.org/abs/2306.03310.

[61] B. Baker, I. Akkaya, P. Zhokhov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos, 2022. URL https://arxiv.org/abs/2206.11795.

[62] 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.

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

[64] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[65] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

[66] S. Xu, Y. Wang, C. Xia, D. Zhu, T. Huang, and C. Xu. Vla-cache: Towards efficient vision-language-action model via adaptive token caching in robotic manipulation. *arXiv preprint arXiv:2502.02175*, 2025.

[67] R. Zheng, Y. Liang, S. Huang, J. Gao, H. Daumé III, A. Kolobov, F. Huang, and J. Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024.

7 Appendix

7.1 Performance comparison with VLA methods

To further evaluate the performance of GVF-TAPE, we compare it with several VLA-based methods, as summarized in Table 5. These baselines are trained with 100% action-labeled data, while our method uses no action labels. Despite this significant difference in supervision, GVF-TAPE achieves competitive performance, demonstrating the effectiveness of our action label-efficient approach.

	Libero-Spatial	Libero-Object	Libero-Goal	Avg.
Octo[6]	78.90	85.70	**84.60**	83.07
OpenVLA[5]	84.70	88.40	79.20	84.10
SpatialVLA[65]	88.20	**89.90**	78.60	**85.57**
VLA-Cache[66]	83.80	85.80	76.40	82.00
TraceVLA[67]	84.60	85.20	75.10	81.63
GVF-TAPE(ours)	**95.50**	86.70	66.80	83.00

Table 5: Performance comparison with VLA-based methods trained on 100% action-labeled data. GVF-TAPE achieves competitive results without requiring action labels, highlighting its label efficiency.

7.2 Overview of the LIBERO benchmark

As illustrated in Fig. 7, the LIBERO benchmark [60] comprises four task suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-100. Each of the first three suites contains 10 tasks, while LIBERO-100 includes 100 diverse tasks spanning a wide range of object types and environments. Every task is accompanied by 50 expert demonstrations.

LIBERO-Spatial focuses on spatial variation, such as placing a bowl on a plate at different locations. LIBERO-Object involves manipulating different objects (e.g., pick-and-place tasks), while LIBERO-Goal keeps the object and location fixed but varies the intended goal. LIBERO-100 significantly expands the benchmark with greater diversity in both object types and scene configurations.

The dataset provides side-view and eye-in-hand RGB images at a resolution of 128×128, along with robot proprioception data, supporting both visual and embodied learning tasks.

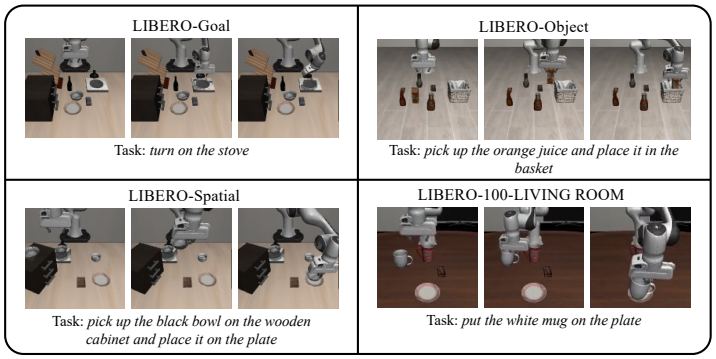


Figure 7: An overview of the LIBERO benchmark.

7.3 Inference Time

All real-world evaluations were performed on an NVIDIA RTX 4080 GPU. To improve inference speed, we utilized mixed-precision computation with TensorFloat-32 (TF32) tensor operations. The

average computation times are summarized in Table 6. GVF-TAPE generates visual plans at an average rate of 1.6 Hz and estimates object poses at 43.5 Hz, as summarized in Table 6.

Table 6: Inference time.

Module	Video Generation	Pose Estimation
Cost Time (s)	0.61 ± 0.0037	0.023 ± 0.0096

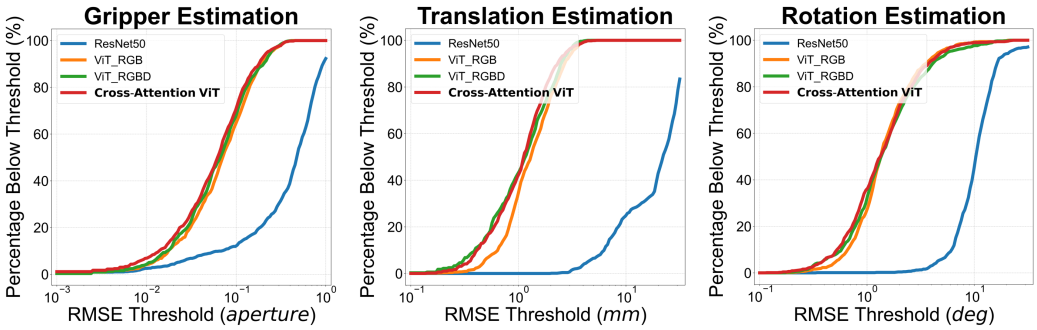


Figure 8: Comparison of model architecture. Performance evaluation using AUC for ResNet50, RGB 3-channel ViT, RGBD 4-channel ViT, and Depth-RGB cross attention model as pose estimation network, trained separately on same amount of random exploration data. Each point on the curve represents the percentage of test points within a given threshold, with a larger AUC indicating better performance.

7.4 Influences of image encoder structure and modality

We evaluated several image encoder architectures to assess their impact on pose estimation performance in real-world manipulation tasks. All models were trained on the same dataset, collected via randomized exploration. As shown in Fig. 8, the cross-attention-based encoder consistently outperforms alternative architectures. Specifically, it surpasses a single Vision Transformer (ViT) applied to stacked RGB-D inputs, a standard ViT trained solely on RGB images, and a ResNet-50 backbone. These results suggest that cross-attention mechanisms are particularly effective at integrating and utilizing depth information, making them well-suited for multimodal visual representations in downstream pose estimation tasks.

7.5 Pose Estimation Model Variants and Implementation

To evaluate the impact of different visual encoders on pose estimation performance, we implemented and compared several architectural variants. Each model predicts an 8-dimensional end-effector pose, T, based on visual observations, and all models were trained using the same dataset and optimization protocol unless otherwise noted.

Cross-Attention ViT Our primary architecture leverages two separate ViT-Base encoders to independently process RGB and depth inputs. These modalities are fused via a multi-head attention mechanism, $\mathcal{A}$, which uses the depth [CLS] token as the query and RGB patch tokens as keys and values. The resulting fused feature is a 768-dimensional vector, passed through a three-layer MLP (256 units per layer) to regress the pose T.

Plain RGB ViT In this configuration, a single ViT-Base model is used to encode the RGB images. The extracted [CLS] token is input into a three-layer MLP, each with 256 hidden units, to produce the predicted pose.

Plain RGBD ViT Here, the RGB and depth images are concatenated into a four-channel RGB-D input, which is processed by a modified ViT-Base model with a 4-channel input stem. The resulting [CLS] token is used as the feature vector and passed through the same MLP as above.

ResNet50 Baseline As a convolutional baseline, we use a ResNet-50 model to encode RGB images into a 2048-dimensional vector, which is then mapped to the pose through the same three-layer MLP.

Table 7: Pose Estimation Model and Training Parameters for real world experiment

Parameter	**Value**
Activation Function	ReLU
Optimizer	Adam
Learning Rate	1×10^{-4}
β Values	[0.9, 0.999]
Weight Decay	1×10^{-8}
Epochs	100
Batch Size	512
Color Jitter (B, C, S, H)	(0.3, 0.2, 0.3, 0.2)
Image Resize (H, W)	(224, 224)
Normalize RGB (Mean, Std)	([0.5, 0.5, 0.5], [0.5, 0.5, 0.5])
Normalize Depth (Mean, Std)	([0.5], [0.5])

Table 8: Pose Estimation Model and Training Parameters for Simulation

Parameter	**Value**
Activation Function	ReLU
Optimizer	Adam
Learning Rate	1×10^{-4}
β Values	[0.9, 0.999]
Weight Decay	1×10^{-8}
Epochs	100
Batch Size	128
Color Jitter (B, C, S, H)	(0.3, 0.2, 0.3, 0.2)
Image Resize (H, W)	(224, 224)
Normalize RGB (Mean, Std)	([0.5, 0.5, 0.5], [0.5, 0.5, 0.5])
Normalize Depth (Mean, Std)	([0.5], [0.5])

7.6 Visual Foresight Model Implementation

We implement our visual foresight module using a 3D-UNet architecture [19] for velocity-based video prediction. To incorporate semantic guidance, we encode textual inputs using CLIP [58], producing latent embeddings that condition the generation process. The 3D-UNet output is modified from 3 to 4 channels to support RGB-D frame generation. This lightweight yet expressive architecture enables the synthesis of spatially-consistent and high-fidelity RGB-D sequences.

The model is trained using an L2 reconstruction loss, optimized with the AdamW optimizer. We employ a cosine annealing learning rate schedule, starting at 1×10^{-4} and decaying to zero over the course of training. Training is performed with a batch size of 8 for 100,000 steps. The model generates 6 future frames per input sequence, and rectified flow fields are computed during inference using an Euler integration scheme. Full implementation details are provided in Table 9.

Table 9: Visual Foresight Model Implementation

Parameter	Value
Loss Function	L2
Optimizer	AdamW
LR Scheduler	CosineAnnealing
Init Learning Late	1×10^{-4}
Weight Decay	1×10^{-8}
Decay Period	100000
Steps	100000
Batch Size	8
Generation Frames	6
Rectified Flow Solver	Euler Solver

For pre-training on the Libero-90 dataset, we follow a similar training protocol but increase the batch size to 32 and distribute the training across 4 A100 GPUs in parallel. Pre-training on real-world human hand video data follows the exact procedure outlined in Table 9.

7.7 Real World Experiment Setting

For real-world experiments, we use an ARX-5 robotic arm paired with a fixed, side-mounted Intel RealSense D435i camera to capture RGB observations. For each manipulation task, we collect 20 teleoperated demonstrations and 50 human hand manipulation videos for pre-training purposes. The video generation model is trained at a resolution of 128×128 and conditioned on both the current visual observation and a task-specific language instruction. It generates six future frames per inference. For pose estimation, we collect 18,000 RGB frames (at a resolution of 224×224) paired with corresponding robot poses using random exploration. Depth maps are labeled using the Video-Depth-Anything model, resulting in an RGB-D and pose dataset.

During real-world evaluation, our system runs using ROS. The generated RGB-D images are resized to 224×224 and used for pose estimation. The predicted poses are then sent to the ARX-5 robot as control commands through ROS topic. To make the robot's movements smoother, we employs sinusoidal interpolation for smooth transitions of position and orientation, combined with linear interpolation for the gripper state.

Each task is evaluated over 10 rollouts with varying conditions: 5 with random object placement near the initial position, 2 with objects placed far from the starting location, 2 with added distractor objects on the table, and 1 under altered lighting conditions. Each rollout is limited to 15 video generation cycles; failure to complete the task within this limit is counted as a failure.

7.8 Simulation Experiment Setting

For the simulation environments, we use a resolution of 128 for video generation and a resolution of 224 for pose estimation. To ensure the quality of video generation by Video Depth Anything, we rendered the environment at a resolution of 256×256 during random exploration data collection. After generating the depth maps, the data were resized to 224×224 for training the pose estimation network. Our video generation model is trained on the demonstration dataset, while for the pose estimation model, we employ the same random exploration method to collect RGB-D and pose pairs, accumulating over 400,000 such pairs for each suite.

During evaluation, we conduct 20 rollouts for each seed and test a total of 3 evaluation seeds. In the LIBERO environment, we use an additional PID controller to ensure the robotic arm moves to the target pose. Furthermore, to facilitate safe object grasping, we implement a gripper threshold, which set gripper aperture to zero when the generated gripper aperture is below the threshold value.

(a) (b)

Figure 9: Real-world task setups for deformable object manipulation. (a) Fold the cloth: The robot is required to grasp one edge of the cloth and fold it, and **(b) put the rag in the trash bin:** the robot is required to grasp the rag and put it in the trash bin. Each task is shown in its initial and final state. These setups highlight the complexity and variability of real-world deformable object manipulation.

7.9 Additional Experiments on Deformable Object Manipulation

To further evaluate GVF-TAPE's ability to handle deformable objects, we conducted two additional real-world tasks: (1) folding a cloth, and (2) placing a rag into a trash bin. The task setups and examples of initial and final object states are illustrated in Fig. 9. For each task, we collected 20 teleoperated demonstration videos, which were combined with the demonstrations from Section 4 to train the video generation model. The pose estimation model remained unchanged from Section 4. We performed 10 evaluation trials per task, and the results are summarized in Table 10.

Task	fold the cloth	place-rag-bin
Success Rate	70%	80%

Table 10: Success rates for deformable object manipulation tasks. GVF-TAPE achieves promising performance on real-world tasks involving deformable objects.

7.10 More Qualitative Results

7.10.1 Failure recovery ability.

By following video-generation guidance in a closed loop, our system can recover from failures. As shown in Fig. 10, the video generation model detects the current task state—recognizing that the tissue has not been grasped. After two attempts, it successfully retrieves the tissue using text-prompted instructions.

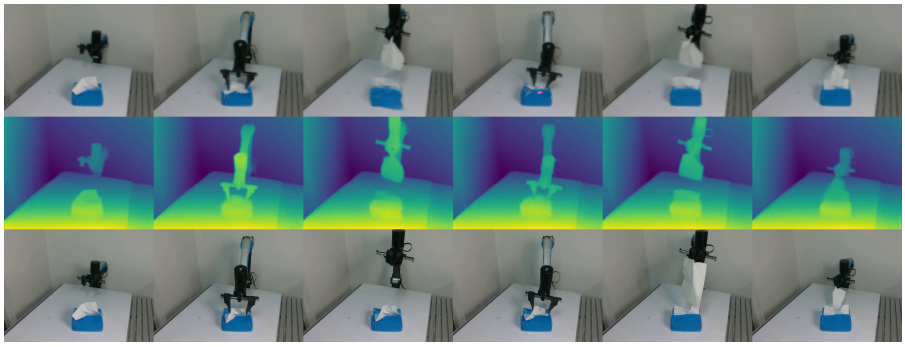


Figure 10: Eval environment roll out of successfully grabbing a tissue through multiple replans. The first and second rows show generated RGB and depth frames, respectively; the third row shows the real world environment. The robot arm fail to grab out the tissue during the first trial; Video generation model as a planner in this process notice the tissue hasn't been grabbed, so the new sampled image will still direct the robot to do so, leading the final success.

7.10.2 Qualitative Comparison of GVF-TAPE w/ and w/o Relative Depth

The impact of incorporating relative depth is further demonstrated through specific examples comparing the performance of GVF-TAPE under two settings: one using RGB-D video generated with supervision from a monocular depth estimator [55], and the other using RGB-only video when depth estimation is unavailable. The inclusion of depth information significantly enhances the system's performance, particularly in spatial pose estimation. Accurate estimation of spatial relationships is critical for successful manipulation. As shown in Fig.12, the RGB-only model produces biased or inaccurate pose estimations, leading to task failure. In contrast, the RGB-D version, demonstrated in Fig.11, achieves correct pose estimation and successfully completes the tasks.

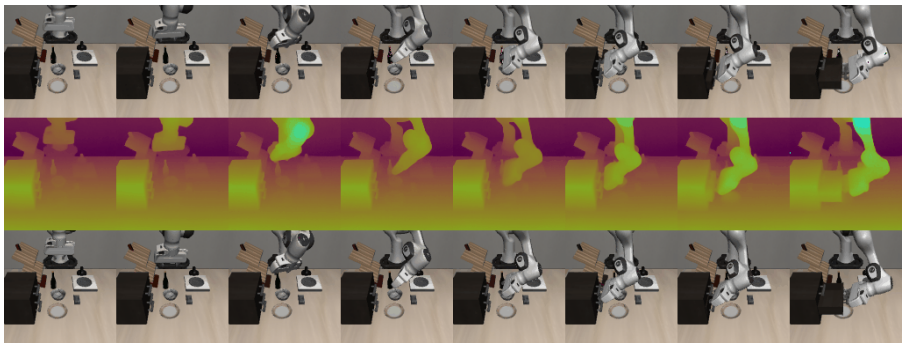


Figure 11: Evaluation rollout of the system with Video-Depth-Anything successfully opening the drawer. The first and second rows show generated RGB and depth frames, respectively; the third row shows the simulation environment.

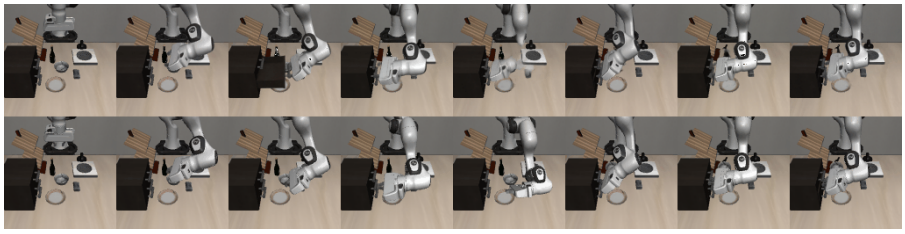


Figure 12: Evaluation rollout of the system without Video-Depth-Anything failing to open the drawer due to biased spatial pose estimation. The first row shows generated RGB frames; the second row shows the simulation environment.

7.10.3 Failure Analysis

We summarize several factors contributing to failure, outlined as follows:

Hallucination. The video generation model may produce physically implausible frames, such as introducing novel objects or causing the robotic arm to become occluded. As shown in 13, the robot may exhibit erratic movement, leading to task failure.

Occultation. In certain manipulation tasks, the robotic arm may move behind an object or obstruct its gripper, making pose estimation challenging. As shown in 14, the proposed method is unable to manipulate the object effectively.

Pose Estimation Error. Errors in the pose estimation model can result in incorrect contact positions, preventing the robot from successfully grasping the object. As shown in 15, the robot arm fails to pick up the bowl due to pose estimation error.

To better understand the prevalence of different failure modes, we conducted an evaluation on the LIBERO-Spatial suite. As shown in Tab. 11, out of 200 trials, we observed a total of 11 failures,

including 3 due to hallucination, 5 due to pose estimation errors, and 3 due to system-level issues. These results suggest that GVF-TAPE is generally robust in scenarios without significant occlusion. However, its performance may degrade in settings involving occlusion, where accurate pose estimation becomes more difficult.

Total Trials	Success	Pose Est. Error	Hallucination	Sys.-Level Error
200	189	5	3	3

Table 11: Failure Analysis

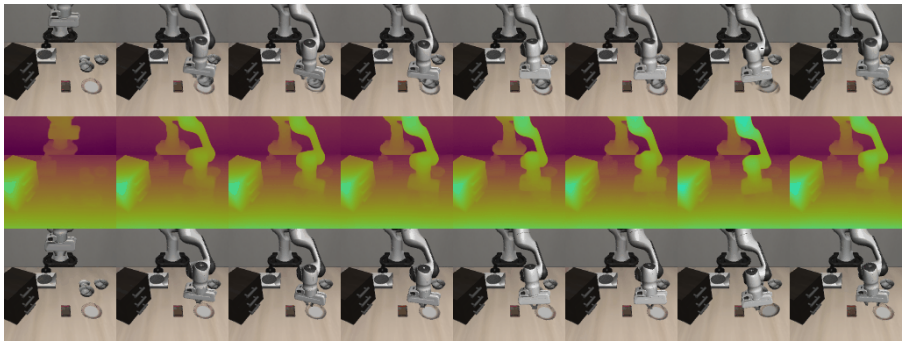


Figure 13: Hallucination in the video generation model leads to task failure. The figure above illustrates a scenario where the model generates a novel bowl, resulting in failure to complete the task. The first and second rows display the generated RGB and depth frames, respectively, while the third row depicts the simulation environment.

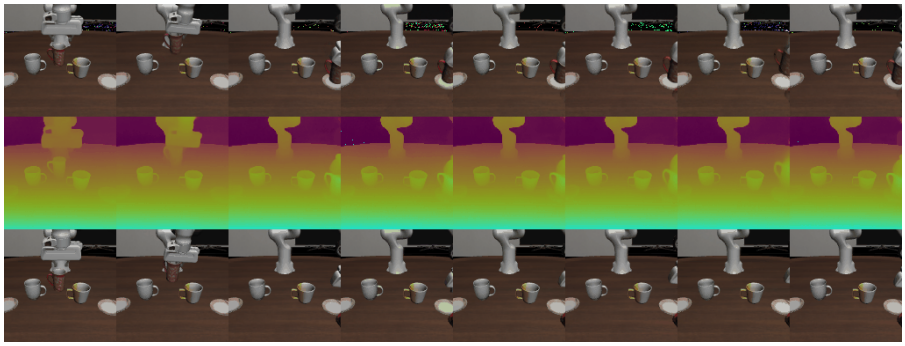


Figure 14: Occultation of the gripper leads to failure. The figure above demonstrates a scenario where the robotic arm moves out of the camera's view, resulting in unreliable pose estimation. The first and second rows display the generated RGB and depth frames, respectively, while the third row depicts the simulation environment.

7.11 Random Exploration

The random exploration process employs a randomized sampling strategy to acquire diverse end-effector poses within the robot's operational workspace and within FOV of the agentview camera. In real-world settings, to ensure safety, we incorporate several safeguards, including joint limit checks and unexpected stop detection. The entire sampling process runs autonomously at 10 Hz, enabling stable and continuous operation. This approach enables efficient exploration of the reachable workspace while maintaining continuous operation stability. In real world settings, we collect around 18k pose-image pair data. The pseudo code real world sampling strategy of our method is provided in Algorithm 1.

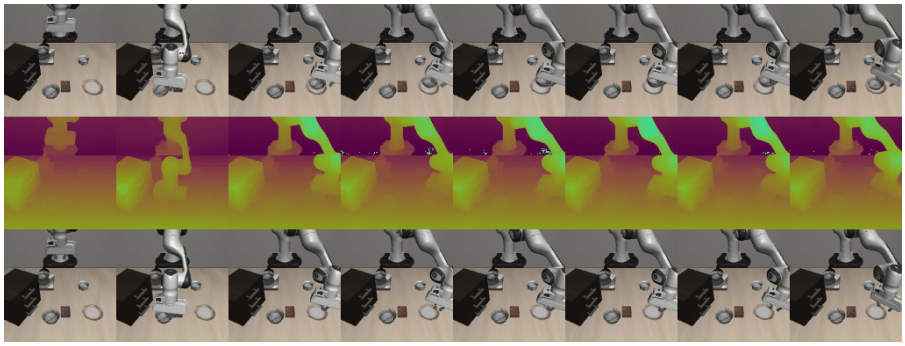


Figure 15: Pose estimation errors lead to failure. The figure above illustrates a scenario where the robotic arm fails to grasp the bowl due to inaccurate pose estimation. The first and second rows display the generated RGB and depth frames, respectively, while the third row depicts the simulation environment.

Algorithm 1 Random Exploration Algorithm in the Real World

Require: Workspace bounds $\mathcal{W}$, arrival threshold $\Delta\mathcal{T}$, number of frames N
Start a parallel thread to continuously check safety
while $num_{frames} < N$ **do**
  Sample a desired end-effector pose $p_{desire} \in \mathcal{W}$
  while $\|p_{current} - p_{desire}\|_2 < \Delta\mathcal{T}$ **do**
    Resample p_{desire}
    Set p_{desire} as the new goal and publish to the robot arm controller
    if $p_{current} \notin \mathcal{W}$ **then**
      Resample $p_{desire} \in \mathcal{W}$ and publish to the robot arm controller
    end if
  end while
end while

7.11.1 Qualitative Results of Real World Tasks

The following are visualizations of real-world tasks: 1) pick up the blue bowl and place it on the pink plate 16, 2) grab a tissue 17, 3) place the sponge on the plate 18. 4) put the blue bowl into the microwave and close it 19, and 5) put the pepper in the basket 20. 6)fold the cloth 21. 7) put the rag in the trash bin 22.

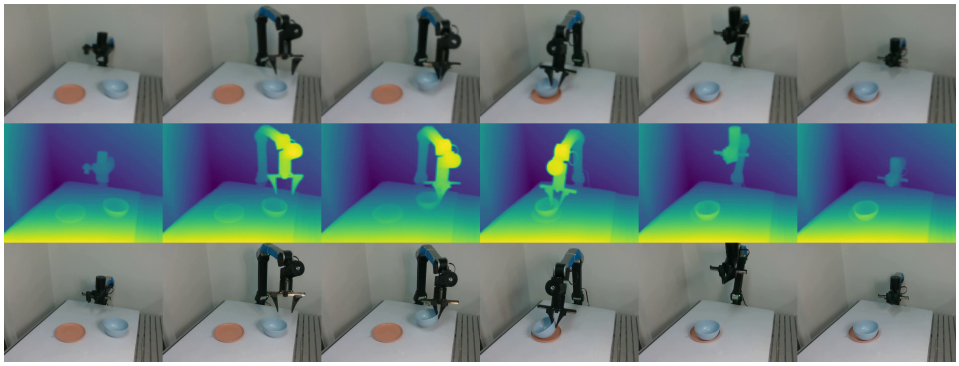


Figure 16: Evaluation rollout of real world task pick up the blue bowl and place it on the pink plate. The first and second rows show generated RGB and depth frames, respectively; the third row shows the real world environment.

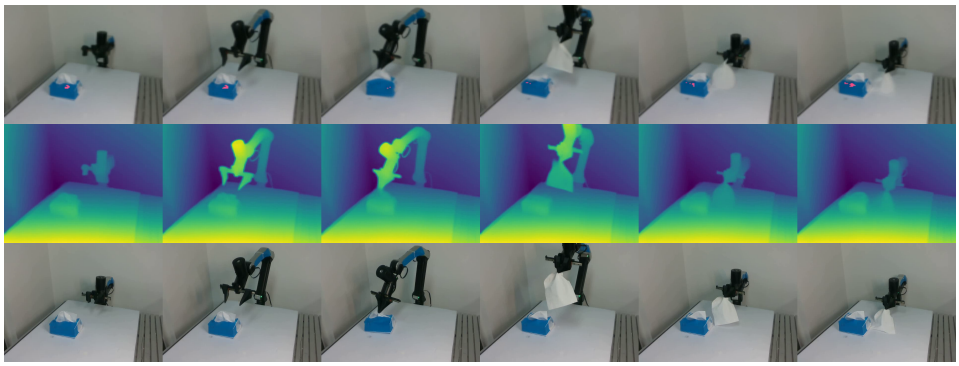


Figure 17: Evaluation rollout of real world task grab a tissue. The first and second rows show generated RGB and depth frames, respectively; the third row shows the real world environment.

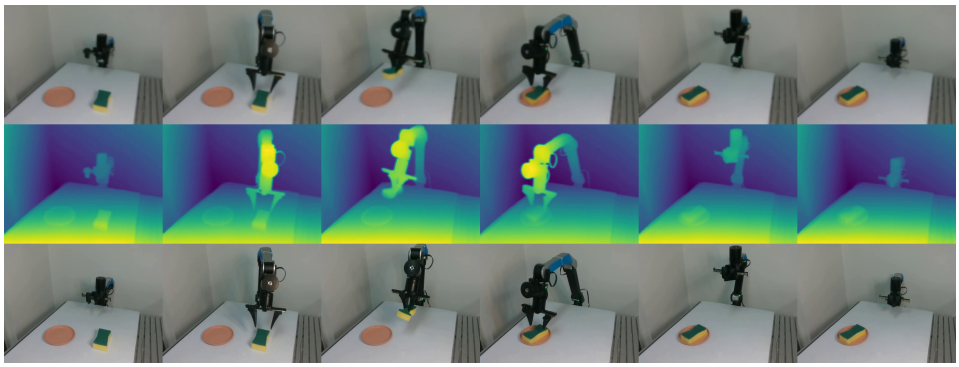


Figure 18: Evaluation rollout of real world task place the sponge on the plate. The first and second rows show generated RGB and depth frames, respectively; the third row shows the real world environment.

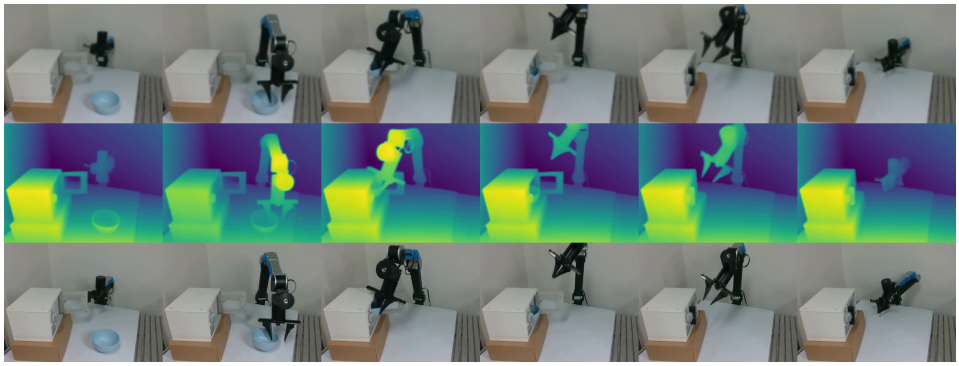


Figure 19: Evaluation rollout of real world task put the blue bowl into the microwave and close it. The first and second rows show generated RGB and depth frames, respectively; the third row shows the real world environment.

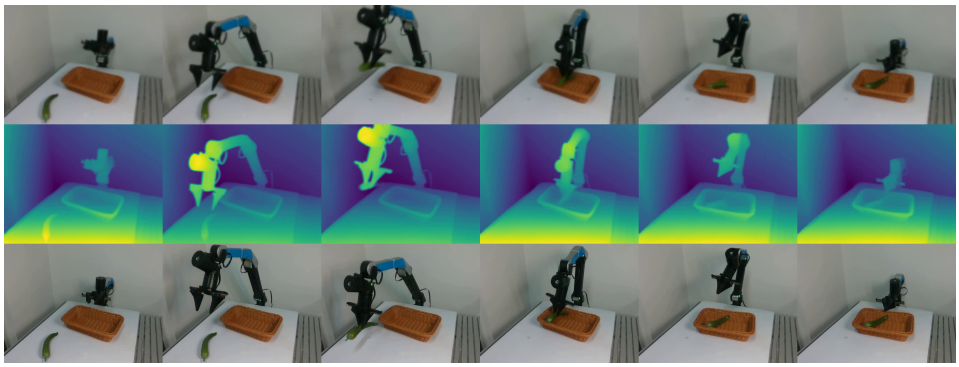


Figure 20: Evaluation rollout of real world task put the pepper in the basket. The first and second rows show generated RGB and depth frames, respectively; the third row shows the real world environment.

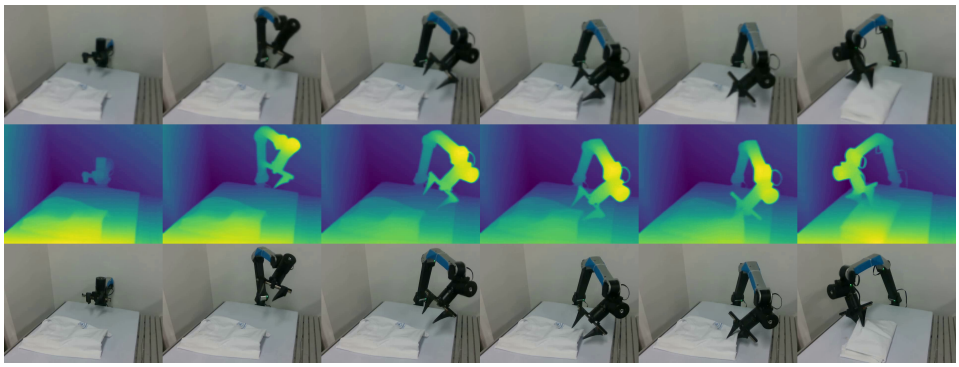


Figure 21: Evaluation rollout of real world task fold the cloth. The first and second rows show generated RGB and depth frames, respectively; the third row shows the real world environment.

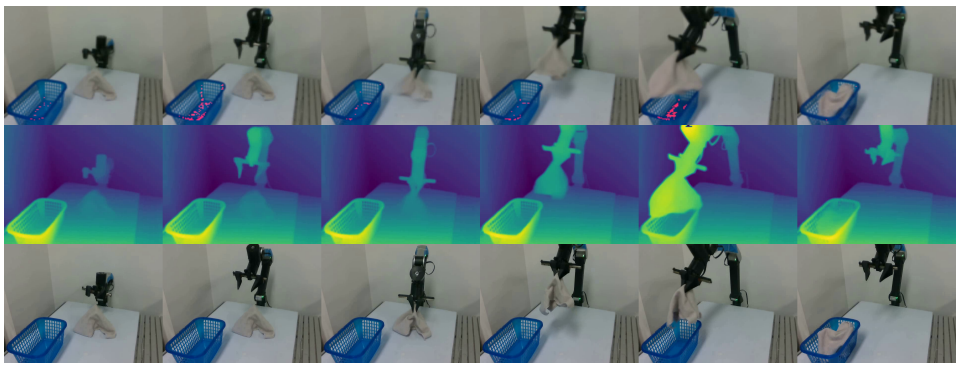


Figure 22: Evaluation rollout of real world task put the rag in the trash bin. The first and second rows show generated RGB and depth frames, respectively; the third row shows the real world environment.