This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

EC-Flow: Enabling Versatile Robotic Manipulation from Action-Unlabeled Videos via Embodiment-Centric Flow

Yixiang Chen[1,2] Peiyan Li[1,2] Yan Huang[1,2,3†] Jiabing Yang[1,2,3] Kehan Chen[1,2] Liang Wang[1,2†]
[1]New Laboratory of Pattern Recognition (NLPR),
Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences [3]FiveAges
yixiang.chen@cripac.ia.ac.cn, {yhuang, wangliang}@nlpr.ia.ac.cn

Abstract

*Current language-guided robotic manipulation systems often require low-level action-labeled datasets for imitation learning. While object-centric flow prediction methods mitigate this issue, they remain limited to scenarios involving rigid objects with clear displacement and minimal occlusion. In this work, we present Embodiment-Centric Flow (EC-Flow), a framework that directly learns manipulation from action-unlabeled videos by predicting embodiment-centric flow. Our key insight is that incorporating the embodiment's inherent kinematics significantly enhances generalization to versatile manipulation scenarios, including deformable object handling, occlusions, and non-object-displacement tasks. To connect the EC-Flow with language instructions and object interactions, we further introduce a goal-alignment module by jointly optimizing movement consistency and goal-image prediction. Moreover, translating EC-Flow to executable robot actions only requires a standard robot URDF (Unified Robot Description Format) file to specify kinematic constraints across joints, which makes it easy to use in practice. We validate EC-Flow on both simulation (Meta-World) and real-world tasks, demonstrating its state-of-the-art performance in occluded object handling (**62%** improvement), deformable object manipulation (**45%** improvement), and non-object-displacement tasks (**80%** improvement) than prior state-of-the-art object-centric flow methods. More results can be found on our project website: https://ec-flow1.github.io.*

1. Introduction

Language-guided robotic manipulation has garnered significant attention in recent years. Current state-of-the-art methods have predominantly adopted imitation learning frameworks, where end-to-end Vision-Language-Action (VLA) models have been trained to directly map visual observations and language instructions to low-level robot actions. However, a critical limitation of these approaches stems from their heavy reliance on large-scale datasets annotated with low-level robot action data. Such data are not only difficult to collect at scale but also inherently prone to noise, potentially degrading policy robustness.



Figure 1. Limitations of object-centric flow: (1) Deformable object failure; (2) Struggles with occlusions; (3) Cannot handle non-displacement actions (e.g., rotations or fine motions).

Beyond limited numbers of action-labeled datasets, abundant manipulation videos without action labels offer rich motion priors that could potentially alleviate data scarcity. Recent works [3, 5, 8, 18, 19, 23, 40] have utilized image and video generation models to infer subgoals

†Corresponding authors.

for policy learning, improving sample efficiency and generalization. However, these methods still rely on imitation learning, requiring low-level action-labeled data for training. A parallel line of research [1, 10, 16, 28, 33, 35, 39] has attempted to bypass action supervision entirely through object-centric flow prediction, where robot motions are inferred from inter-frame transformations of predicted object-centric flow. While promising, these methods have three major limitations in challenging scenarios (Figure 1): (1) *Rigidity assumption*: They presume uniform transformations across object components, failing to handle deformable objects; (2) *Occlusion vulnerability*: Actions are derived solely from object state changes, which makes them ineffective in scenarios with partial observability; (3) *Non-object-displacement manipulation*: They fail to capture actions when objects undergo non-translational state changes (e.g., rotating a round switch) or remain nearly static (e.g., pressing a mouse button) during manipulation. These limitations potentially constrain their generalization to a broader range of real-world manipulation scenarios.

Our approach is motivated by a critical observation: We can fundamentally avoid the limitations by shifting the prediction from object-centric flow to embodiment-centric flow. It eliminates dependence on object properties or states, enabling generalization across deformable and rigid objects, and can also handle occlusions and non-object-displacement manipulation. Moreover, the robot's kinematic structure is visible in most real-world manipulation scenarios – and even under partial occlusion, sufficient motion cues can be extracted from visible joints to infer actions. However, the implementation of embodiment-centric flow poses two significant challenges. (1) Although embodiment-centric flow simplifies the task by emphasizing the robot's motion dynamics, most manipulation tasks still require precise and meaningful interactions with specific objects. Naive embodiment-centric flow prediction might generate inaccurate robot actions that fail to interact with objects appropriately, especially when language instructions include specific object-related cues. (2) The different joints of the embodiment typically have distinct kinematic constraints such as different ranges of motion, degrees of freedom, and dynamic capabilities, which could not be treated as a unified entity when calculating actions.

To resolve these issues, we propose a new framework named **EC-Flow** (**E**mbodiment-**C**entric **F**low) that predicts embodiment-centric flow from action-unlabeled videos with two key designs. First, a goal-conditioned alignment module ensures the predicted flow reconstructs both the robot's motion trajectory and the goal image specified by language instructions, enabling task-relevant object interactions while adhering to kinematic constraints. Second, a physics-aware action calculation framework leverages the embodiment's URDF (Unified Robot Description Format) file to decompose global EC-Flow into joint-specific components via forward kinematics and apply joint-specific transformations, bridging visual predictions with executable actions. We conducted experiments in both the simulation benchmark Meta-World and the real-world scenarios to validate the effectiveness of the method.

Our main contributions can be summarized as follows:

- We present the **EC-Flow** method, which enables manipulation policy learning solely from action-unlabeled video demonstrations.
- We design a goal-image-conditioned flow alignment mechanism that aligns embodiment-centric flow with language instruction and object interactions.
- We propose a URDF-aware action calculating paradigm, which establishes a physics-grounded translation from visual flow predictions to executable actions.
- Experimental results demonstrate that EC-Flow effectively handles diverse scenarios, exhibiting strong generalization across deformable objects, occluded environments, and non-object-displacement manipulation tasks. Specifically, it achieves success rate improvements of 62%, 45%, and 80%, respectively, compared to prior state-of-the-art object-centric flow-based methods.

2. Related Works

2.1. Imitation Learning

Imitation learning has been a predominant framework for robotic manipulation, enabling agents to acquire skills through supervised learning from expert demonstrations. While recent Vision-Language-Action (VLA) models [2, 4, 15, 17, 20, 24, 25, 42] have adopted this paradigm, they rely on large-scale low-level action-labeled datasets. For instance, RDT-1B [20] has leveraged over 1 million trajectories (21TB) and OpenVLA [15] has utilized the Open X-Embodiment dataset [32] comprising 2M+ trajectories from 70+ distinct robot platforms. Although such extensive datasets could enable generalizable policies, they have incurred substantial collection costs and introduced challenges including sensor noise aggregation and multi-modal action distributions. These limitations motivate the exploration of action-unlabeled datasets that can circumvent the need for precise low-level action supervision.

2.2. Learning Manipulation from Videos

Learning robotic manipulation from action-unlabeled videos leverages abundant, naturally occurring motion priors, eliminating the need for explicit low-level action labels. Traditional methods rely on video prediction [5, 8, 19, 21, 34, 36, 40] and image generation [18, 23] to infer intermediate subgoals, but ultimately depend on action labels for policy learning.

Another line of work leverages object-centric flow to

entirely get rid of the action-labeled dataset, which first predicts object flow and calculates the transformation of the object to infer actions of the object. However, these methods suffer from fundamental scenario-specific limitations: rigidity assumption, occlusion vulnerability, and non-object-displacement manipulation that cannot handle deformable objects, occlusion, or objects with rotational/micro movements scenarios. Among them, General Flow [39] and MT-π [28] are most relevant to our approach. In addition to the major difference between object-centric and embodiment-centric, General Flow requires RGBD cameras to predict 3D flow fields, while our EC-Flow uses only RGB videos. MT-π relies on tracking 5 predefined gripper keypoints that must remain fully visible during operation, whereas EC-Flow naturally handles partial arm occlusion through embodiment-centric flow prediction. This key difference enables our method to manipulate deformable objects and operate in occluded environments where object-centric approaches fail.

3. EC-Flow Method

3.1. Overview

Our framework addresses language-conditioned manipulation tasks using a video-only dataset $\mathcal{D} = \{\zeta_i\}_{i=1}^{n}$, where each demonstration $\zeta_i = (\{o_t\}_{t=1}^{m_i}, l_i)$ contains: (1) RGB visual observations $\{o_t\}_{t=1}^{m_i}$ captured from a fixed camera with a total of m_i frames, and (2) a language instruction l_i, while explicitly excluding low-level action data.

The proposed **EC-Flow** overcomes key limitations of prior work through two modules: (1) *Embodiment-Centric Flow Prediction* (Sec 3.2): Instead of tracking objects, we predict pixel-wise flow (i.e., future locations) for randomly sampled points of the embodiment; (2) *Kinematic-Aware Action Calculation* (Sec 3.3): Leveraging only the embodiment's URDF model (standard in robotic systems) and consecutive predicted EC-Flow, we compute executable actions without requiring action labels for supervision. This dual-stage approach enables deformable objects, occlusion-robust, and non-object-displacement manipulation purely from RGB video demonstrations.

3.2. Embodiment-Centric Flow Prediction

To predict embodiment-centric flow, we first construct a flow prediction dataset derived from original RGB videos, then develop a flow prediction model. The primary challenge lies in ensuring that the predicted embodiment-centric flow can effectively capture meaningful interactions between the embodiment and environmental objects for successful instruction execution. To address this fundamental alignment problem, we introduce an auxiliary goal image prediction task, which guarantees that the generated motions adhere to kinematic constraints while also guiding the system toward interactions with objects that are relevant to the instruction. The detailed embodiment-centric flow prediction framework is shown in Figure 2.

3.2.1. Dataset Construction

Let o_0 denote the initial video frame. Our pipeline begins with extracting a pixel-wise mask of the embodiment following the structure of Grounded Sam [30], which first obtains a bounding box of the embodiment using text prompt (*"the robotic arm"* in our case) via GroundingDINO [29] and then refine to pixel-wise masks using SAM2 [27] within detected regions. Subsequently, we initialize randomly sampled N_p points within the embodiment mask for tracking. We further analyze the sensitivity to the segmentation results in Figure 6, which demonstrates that EC-Flow remains robust to segmentation errors due to the subsequent action calculation procedure.

Ground truth point tracking results are acquired using the off-the-shelf CoTracker [14] model, which simultaneously predicts future *uv* coordinates and visibility of the initial query points across frames. While we save the tracking results across the whole video, our training protocol samples temporal windows of fixed-length T frames from complete trajectories. All coordinates undergo normalization to ensure scale invariance across varying video resolutions. The constructed dataset could be noted as $\mathcal{D} = \{(p_i, o_0^i, l_i)\}_{i=1}^{n}$, where $p_i \in \mathbb{N}^{N_p \times T \times 3}$.

3.2.2. Model Architecture

Diffusion-Based Flow Prediction Our framework (branch (a) in Figure 2) implements a conditional diffusion process [13] for flow denoising, formalized through the following Markov chain with discrete timesteps $t \in \{0, \cdots, T\}$,

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

where $\mathbf{z}_0$ represents the ground truth flow (*uv* coordinates and visibility states), and $\{\beta_t\}_{t=1}^{T}$ defines a cosine noise schedule that gradually corrupts the flow to isotropic Gaussian. The reverse process learns to iteratively denoise the flow through the parameterized transition:

$$p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_{t-1}; \mu_\theta(\mathbf{z}_t, t, \mathbf{c}), \Sigma_\theta(\mathbf{z}_t, t, \mathbf{c})) \tag{2}$$

where $\mathbf{c}$ denotes our multimodal conditioning signals. The training objective minimizes the distance between the predicted noise and the ground truth:

$$\mathcal{L}_{flow} = \mathbb{E}_{t,\mathbf{z}_0,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t, \mathbf{c})\|_2^2] \tag{3}$$

where ϵ represents the actual noise added at timestep t.

Our multimodal conditioning signals $\mathbf{c} = [\tilde{\mathbf{v}}, \tilde{\mathbf{l}}, \tilde{\mathbf{s}}]$ include three parts: (1) Visual context $\tilde{\mathbf{v}}$: initial frame encoded through ResNet-50 [12]; (2) Language guidance $\tilde{\mathbf{l}}$: instruction embeddings from CLIP text encoder [26]; (3)

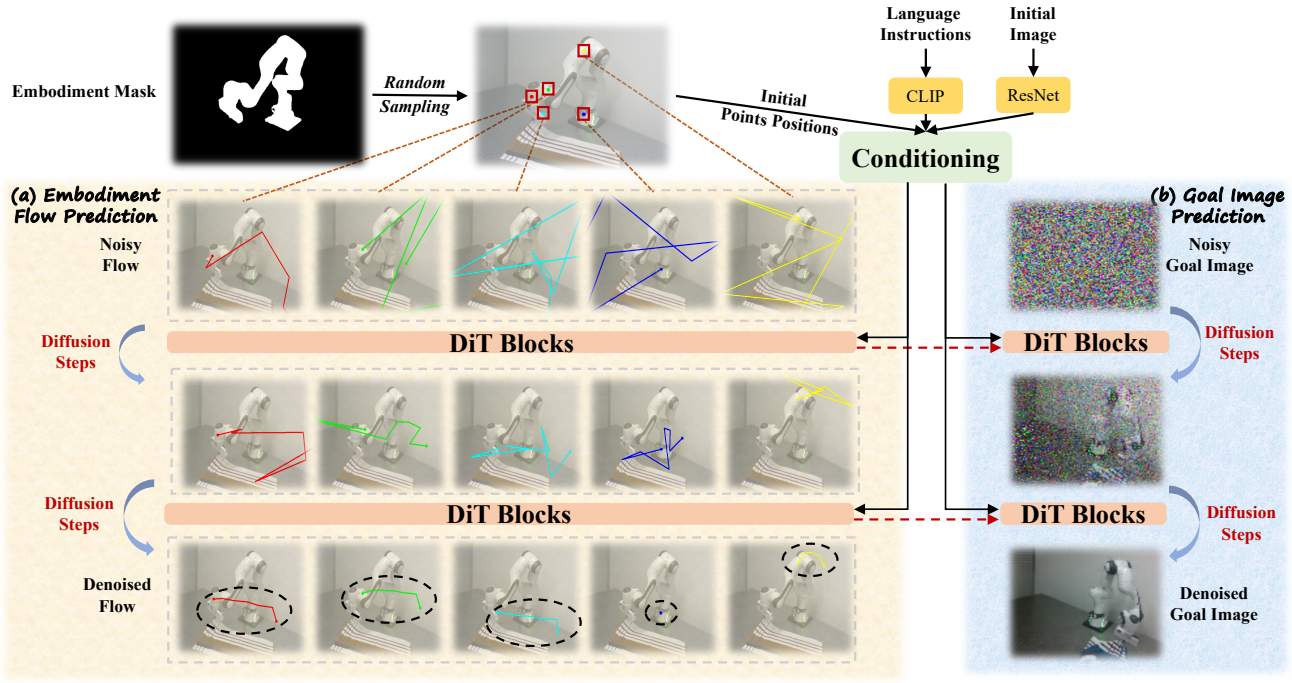


Figure 2. Embodiment-centric flow prediction network architecture. *Branch (a)*: prediction of embodiment flow. *Branch (b)*: prediction of the goal image which is used as an auxiliary task for aligning flow to object interactions and language instruction.

Starting state $\tilde{\mathbf{s}}$: initial point coordinates. Rather than applying diffusion to the starting point, we treat it as ground truth throughout the generation process by setting the corresponding noise to zero. These heterogeneous features are projected to a unified latent space through learnable adapter layers before cross-attention fusion with trajectory tokens.

During inference, we employ DDIM [31] sampling for efficient trajectory generation. The model output contains the whole trajectory of embodiment point flow during a fixed horizon T.

Goal Image Prediction While the predicted embodiment-centric flow captures motion dynamics, it may not inherently ensure proper object interactions or semantic alignment with language instructions. To bridge this gap, we introduce an auxiliary goal image prediction task that serves as a visual grounding mechanism (branch (b) in Figure 2).

The goal image generator shares the core architecture of our flow prediction model but operates in the pixel space. During training, we corrupt the goal image (corresponding to the trajectory endpoint) with progressive noise and task the model with reconstructing the denoised image. Crucially, we synchronize the diffusion timesteps between the flow prediction and goal image generation branches, allowing the embodiment flow noise patterns to condition the goal-image generation process. Specifically, at each timestep, the goal-image generator receives augmented conditioning $\mathbf{c_t}^{img} = [\tilde{\mathbf{v}}, \tilde{\mathbf{l}}, \tilde{\mathbf{s}}, f_t^{flow}]$, where f_t^{flow} denotes the flow prediction output at time t, creating information transfer between motion estimation and visual synthesis. We denote the loss of the goal image as L_{image}, and the total loss could be formulated as:

$$L = L_{flow} + \lambda L_{image}$$

This co-training strategy establishes implicit constraints: (1) Object interaction validity: the generated goal image must depict physically plausible object states resulting from the predicted flow; (2) Instruction grounding: visual semantics in the goal image are forced to align with language instructions; (3) Temporal consistency: intermediate flow trajectories must bridge the initial and goal states coherently. By jointly optimizing these complementary objectives, the framework learns to generate the embodiment-centric flow that is both geometrically coherent and semantically aligned with task requirements.

3.3. Kinematic-Aware Action Calculation

Translating predicted flow into executable actions requires addressing the articulated nature of embodiment kinematics. Contrary to prior works that compute unified object transformations, our embodiment-centric flow captures heterogeneous motion patterns across distinct joints, a critical property demanding specialized calculation mechanics.

Our kinematic-aware calculation pipeline (Algorithm 1) operates through two core phases, we first allocate sampled

Algorithm 1 Kinematic-Aware Action Calculation

Input:

$\{\mathcal{P}^{(t)_{2D}}\}_{t=0}^{T}$: predicted 2D EC-Flow of horizon T, $\mathcal{K}$: camera intrinsic/extrinsic parameters, $\mathcal{J}$: embodiment joint configuration, $\pi(\cdot)$: camera projection function, M: number of joints, $\{\mathbf{T}_{ee}^{(t)}\}_{t=0}^{T-1}$: pose matrix of the end-effector to be optimized.

Output:

$\{\mathbf{T}_{ee}^{*(t)}\}_{t=0}^{T-1}$: optimal pose matrix of the end-effector for each time step

for $t = 0$ to $T-1$ **do**

$\mathcal{P}_{filt}^{(t)_{2D}} \leftarrow \text{FilterPoints}(\mathcal{P}^{(t)_{2D}})$ ▷ Filter points for stable calculation

$\{\mathbf{b}_j\}_{j=1}^{M} \leftarrow \text{ProjectJointPositions}(\mathcal{J}, \mathcal{K})$ ▷ Get 2D joint bounding box using URDF geometry

$\{\mathcal{P}_j^{(t)_{2D}}\}_{j=1}^{M} \leftarrow \text{AssignPointsToJoints}(\mathcal{P}_{filt}^{(t)_{2D}}, \{\mathbf{b}_j\}_{j=1}^{M})$ ▷ Establish joint-point correspondences

$\{\mathcal{P}_j^{(t)_{3D}}\}_{j=1}^{M} \leftarrow \text{LiftTo3D}(\{\mathcal{P}_j^{(t)_{2D}}\}_{j=1}^{M}, \textit{Depth}, \mathcal{K})$ ▷ Depth-aware reconstruction

$\{{}_j^{ee}\mathbf{T}^{(t)}\}_{j=1}^{M} \leftarrow \text{ComputeJointTransforms}(\mathbf{T}_{ee}^{(t)}, \mathcal{J})$ ▷ Compute joint transforms for a given $\mathbf{T}_{ee}^{(t)}$ via IK

$\mathbf{T}_{ee}^{*(t)} \leftarrow \arg\min_{\mathbf{T}_{ee}^{(t)}} \sum_{j=1}^{M}\sum_{i=1}^{N_j} \left\| \pi(\mathbf{T}_{ee}^{(t)} \cdot {}_j^{ee}\mathbf{T}^{(t)} \cdot \mathbf{P}_{ji}^{(t)_{3D}}) - \mathbf{P}_{ji}^{(t+1)_{2D}} \right\|_2$ ▷ Optimization of the EEF pose at time t

points to joints and then calculate the transformation of the end-effector.

3.3.1. Allocate Sampled Points to Joints

We begin by filtering the sampled points according to three criteria: (1) consistent visibility across consecutive frames, (2) significant displacement surpassing a minimum motion threshold, and (3) valid depth information from the camera.

For joint assignment (shown in Figure 3), we first utilize the geometric attributes provided in the URDF file to determine the 3D positions of the joints. Subsequently, the corresponding 2D bounding boxes of joints on the image plane are computed using the intrinsic and extrinsic parameters of the camera. A point is considered to belong to a joint if it lies within the computed bounding box. Due to the potential overlap of joints in the 2D observation, only points uniquely assigned to a single joint are selected for further processing.

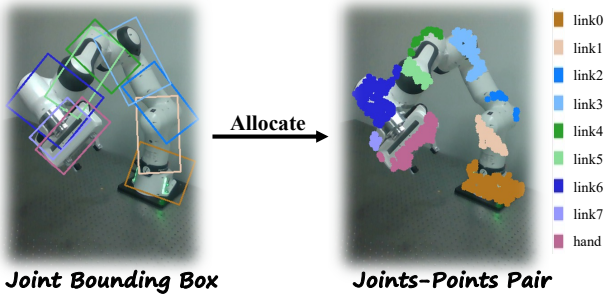


Figure 3. Process of allocating sampled points to specific joints. We only select points that only belong to a single joint.

3.3.2. Calculate Transformation of End-Effector

Given the established joint-point correspondences, we compute executable end-effector actions. The pipeline involves three core operations:

(1) Depth-aware Reconstruction: Lift the 2D point flow to 3D coordinates using calibrated depth maps and camera parameters. (2) Joint Transformation Calculation: For candidate end-effector poses, compute associated joint transformations via URDF-based inverse kinematics. (3) Reprojection Alignment: Optimize the proposed end-effector pose by minimizing the discrepancy between (a) predicted embodiment flow trajectories and (b) reprojected trajectories generated from the estimated poses, ensuring consistency through iterative refinement.

Specifically, given a target end-effector pose T_{ee}, we first compute each joint's transformation relative to end-effector ${}_j^{ee}\mathbf{T}$ via inverse kinematics, and compute the reprojection error:

$$\epsilon = \sum_{j=1}^{M}\sum_{i=1}^{N_j} \left\| \pi(\mathbf{T}_{ee} \cdot {}_j^{ee}\mathbf{T} \cdot \mathbf{P}_{ji}^{(t)_{3D}}) - \mathbf{P}_{ji}^{(t+1)_{2D}} \right\|_2$$

where $\pi(\cdot)$ denotes camera projection.

The optimal pose $\mathbf{T}_{ee}^{*}$ is obtained through the minimization of ϵ, constrained by the embodiment's mechanical limits. This geometrically grounded approach ensures physical plausibility while maintaining alignment with visual observations. The resulting optimal end-effector pose, $\mathbf{T}_{ee}^{*}$, is then transmitted to the control systems for action execution.

	Avg. Success ↑	door open	door close	shelf place	btn press	btn-top press	faucet close	faucet open	handle press	hammer strike
BC-Scratch	0.204	0.24	0.36	0	0.36	0.12	0.20	0.20	0.36	0
BC-R3M [22]	0.360	0.04	0.60	0	0.36	0.04	0.24	0.68	0.52	0.76
UniPi [7]	0.093	0	0.36	0	0.12	0	0.04	0.12	0.16	0.04
Diffusion Policy [6]	0.298	0.48	0.48	0	0.40	0.20	0.24	0.60	0.24	0.04
AVDC [16]	0.489	0.72	0.92	**0.20**	0.60	0.24	**0.56**	0.24	**0.84**	0.08
Track2Act [1]	0.556	0.88	0.76	0.04	0.56	0.40	0.48	0.88	0.76	0.24
EC-Flow(ours)	**0.720**	**0.96**	**1.00**	0.04	**0.64**	**1.00**	0.44	**1.00**	0.52	**0.88**

Table 1. Simulation results on Meta-World benchmark.

4. Experiment Results

4.1. Simulation Evaluation

Simulation Setup We conduct our experiment within the Meta-World simulation benchmark [37], which provides a simulated environment for a series of tabletop manipulation tasks using a Sawyer robot arm. This benchmark involves interactions with various objects and tools, where the object locations are randomly initialized at the start of each task. In the Meta-World setup, the robot's gripper is assumed to remain in a downward orientation throughout, resulting in an action space with four dimensions: the position of the end effector and the gripper's open/close state.

We evaluated nine tasks from the Meta-World benchmark. For each task, we used a side-mounted camera in the environment to capture five video demonstrations, resulting in a total of 45 videos. The policy is evaluated 25 times for each task with randomly initialized object positions. Success is defined as reaching a pre-defined goal within a predefined maximum number of steps. We trained our EC-Flow policy using 8 Nvidia 4090 GPUs for one day.

Compared Methods We compare our method against six state-of-the-art methods:

(1) BC-Scratch: Standard behavior cloning baseline with dual-stream encoder architecture, which processes RGB observations through ResNet-18 [12] and language instructions via CLIP [26] and is trained end-to-end on expert demonstrations.

(2) BC-R3M [22]: Enhances visual representation learning by initializing the image encoder with R3M pre-trained weights, while maintaining identical architecture and training protocol as BC-Scratch.

(3) UniPi [7]: Two-stage approach combining: (i) vision-language conditioned video prediction model, and (ii) inverse dynamics model that maps predicted frames to actions through behavioral cloning.

(4) Diffusion Policy [6]: Diffusion-based action predictor that models action sequences as denoising processes conditioned on observation-history windows, trained via imitation learning.

(5) AVDC [16]: Object-centric flow paradigm that first learns video prediction, then extracts optical flow trajectories using off-the-shelf point trackers, and finally computes action through object flows.

(6) Track2Act [1]: Approach that directly uses object-centric flow prediction. Unlike the original paper (incorporating an extra module using action-labeled data to refine actions), we here implement the approach for computing actions solely from the flow.

Implementation Details Our grasping strategy adopts the object-centric approach from [16], the grasp point is the centroid of the segmented mask. For motion planning robustness, we also implement an adaptive replanning protocol, as described in [16], which is triggered when the actions do not change beyond a given threshold.

For EC-Flow, we predict $T = 8$ steps over $N_p = 400$ sampled points. Both the start and goal images are resized to 128×128. We perform 250 sampling iterations for DDIM during the diffusion inference process. The batch size is set to 56, with a learning rate of 5×10^{-5} for the flow prediction and 1×10^{-4} for the goal image prediction. To balance the contributions of the flow prediction and the goal image prediction, we set λ to 0.4.

Results The simulation results on the Meta-World benchmark are presented in Table 1. Methods (1)–(4) rely on traditional imitation learning using low-level action-labeled datasets, while methods (5)–(6) utilize only action-unlabeled video data. Specifically, method (5) employs video prediction followed by flow tracking, whereas method (6) directly predicts object-centric flow.

The results demonstrate a significant improvement, particularly in the *btn-top-press* and *hammer-strike* tasks, where our approach achieves an average success rate increase of **65%** compared to prior object-centric flow methods. Moreover, our method surpasses previous state-of-the-art approaches on the overall Meta-World benchmark by **16.4%**. This improvement is primarily attributed to the challenges posed by object occlusions in these tasks, where previous methods struggle with accurately tracking occluded objects, often leading to failure. As illustrated in

Figure 4, prior methods fail because of occlusion of target objects or hallucination effects introduced by video prediction, which can lead to erroneous object tracking. In contrast, our EC-Flow method effectively handles such cases, even when parts of the embodiment remain unobservable.



Figure 4. Occlusion and hallucination problems of object-centric flow and video prediction methods.

Ablation Study We systematically evaluate four critical design variants to isolate component contributions. The *Video+GT Flow* configuration replaces our end-to-end flow prediction with flow estimates from a pre-trained video prediction model. Removing the goal image prediction auxiliary task (*Goal Img Pred*) tests multimodal alignment necessity. For action generation, disabling the point filtering strategy (*Point Filter*) before action calculation reveals motion reliability criteria importance while restricting tracked points to the end-effector (*EEF Points*) evaluates full-arm motion modeling benefits.

Model #2 shows that combining a video prediction model with an off-the-shelf point tracking method performs worse than our end-to-end EC-Flow approach. As illustrated on the right side of Figure 4, video prediction can suffer from hallucination issues—predicting multiple robotic arms—which then introduces errors in point tracking. This problem is avoided by using end-to-end flow prediction. Model #3 highlights the necessity of the goal-image prediction module; directly predicting embodiment-centric flow may be misaligned with object interactions or language instructions, whereas the additional module alleviates this issue. Model #4 demonstrates the effectiveness of the point filtering strategy since static, invisible, or invalid-depth points can degrade the overall performance of the action calculation process. Finally, model #5 underscores the importance of computing the complete set of embodiment points. In scenarios where the end-effector is occluded, considering the full embodiment helps reduce errors.

We also evaluate various configurations of the sample point count N_p and the prediction horizon T in the EC-Flow prediction network. Our results indicate that using fewer sample points degrades performance, as inaccuracies from outlier points impair the action calculation. Additionally, a shorter prediction horizon may miss critical stages of the manipulation process, and larger gaps between consecutive actions can compound errors during calculation, further reducing performance. Increasing N_p and T does not lead to substantial differences, we select the current configuration to balance performance with training and inference costs.

#	Flow Pred		Action Calc		Avg
	Video + GT Flow	Goal Img Pred	Point Filter	EEF Points	
1	-	✓	✓	-	**0.720**
2	-	-	✓	-	0.636
3	-	✓	-	-	0.582
4	-	✓	✓	✓	0.604
5	✓	✓	✓	-	0.667

Table 2. Ablation analysis of EC-Flow design.

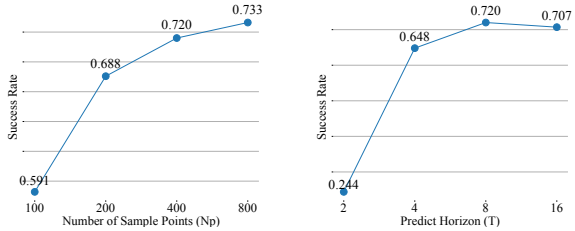


Figure 5. *Left*: Influence of choice of sampled points N_p. *Right*: Influence of choice of predict horizon T.

4.2. Real-world Evaluation

Real-world Setup Our real-world experiment setup consists of a statically mounted Franka Research 3 robotic arm and a static third-person view Intel Realsense D435i RGB-D camera. We calibrate the robot-camera extrinsic and convert the perceived 3D points into the robot's base frame before calculating actions. For a given target gripper action, we utilize Deoxys [41] to guide the robot to the target through trajectory generation and feedback control.

Seven manipulation tasks are constructed across three categories: (a) rigid objects with prismatic/revolute joints (3 tasks), (b) deformable objects (2 tasks), and non-object-displacement manipulation (2 tasks). Each task is trained on five action-unlabeled video demonstrations.

We adopt two baselines: Behavior Cloning (BC) using action-labeled data and Track2Act[1] using object-centric flow, which is consistent with the simulation baselines. For all baselines, we manually position the robotic arm for task initiation and we test each task 10 times with objects of randomly initialized positions.

Results The results of the real-world evaluation are presented in Table 3. The results of the real-world evaluation are presented in Table 3. Our findings show that EC-Flow achieves a **45%** improvement in deformable object manipulation and an **80%** improvement in non-object-displacement manipulation tasks compared to prior object-centric flow-based methods, which are incapable of handling these tasks.

	All Tasks ↑	open fridge	open drawer	open oven	fold clothes	fold towel	press mouse	rotate switch
BC	31/70	5/10	6/10	6/10	**4/10**	2/10	3/10	5/10
Track2Act [1]	21/70	8/10	8/10	5/10	0/10	0/10	0/10	0/10
EC-Flow(ours)	**54/70**	**9/10**	**10/10**	**10/10**	3/10	**6/10**	**9/10**	**7/10**

Table 3. Results on real-world manipulation tasks.

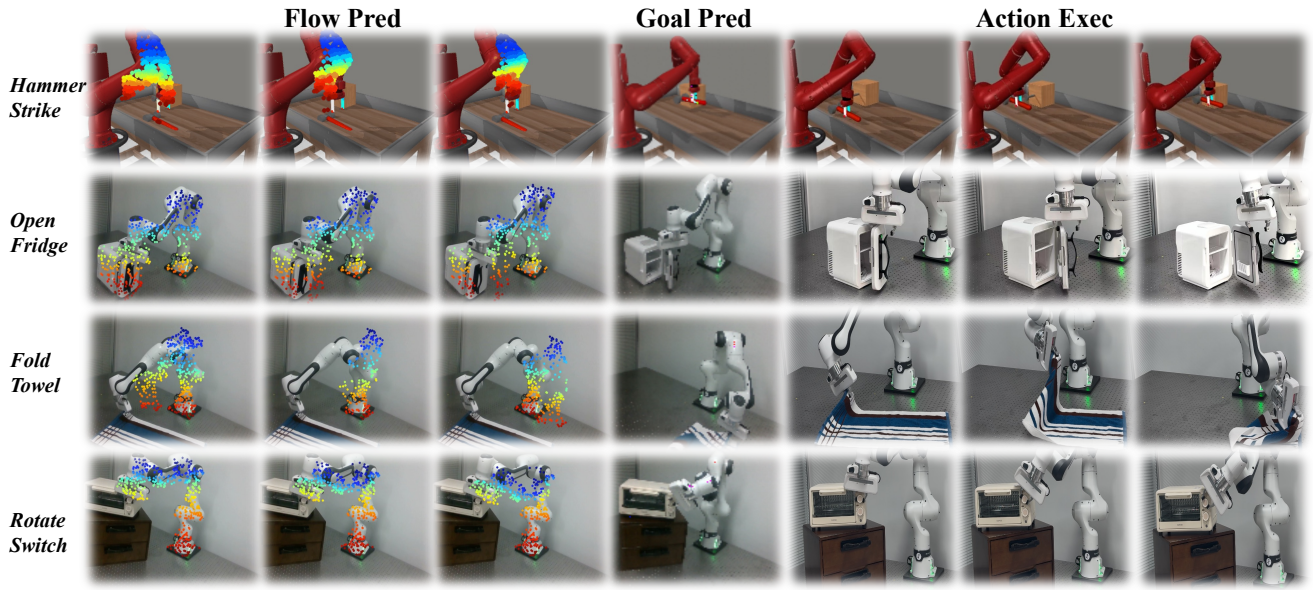


Figure 6. Visualization of the EC-Flow method for learning robotic manipulation from action-unlabeled video data. *Left* (3 cols): flow prediction results. *Middle* (1 col): goal image prediction results. *Right* (3 cols): final action execution results. View differences originate from distinct camera positions for data capture (*Left/Middle*) and execution recording (*Right*).

Moreover, our method surpasses behavior cloning approaches that depend on action-labeled data for supervision. These methods typically require a large number of expert demonstrations to learn an effective policy, making them less effective in data-limited scenarios. In contrast, our approach achieves superior performance with only five videos per task. This is enabled by leveraging embodiment-centric flow in the observation space, allowing EC-Flow to generalize more effectively to new objects and robot positions without relying on low-level action supervision.

The relatively low success rate in the *fold clothes* task may be partially attributed to inaccuracies of depth perception from the low-cost D435i camera. The action accuracy could potentially be improved by using a sensor with more precise depth-sensing capabilities.

Qualitative Analysis We visualize our EC-Flow method in Figure 6. The process begins with utilizing goal image prediction to assist in generating the embodiment-centric flow, followed by directly computing actions from the flow. In the second row of Figure 6, we further demonstrate the robustness of our method to inaccuracies in segmentation results produced by off-the-shelf models. Specifically, in that case, the segmentation model mistakenly identifies a part of the fridge as part of the embodiment. However, our method effectively filters out these erroneous points by leveraging the spatial positions of the robotic arm's joints, as defined in the URDF file. We demonstrate that EC-Flow is capable of handling deformable objects and performing non-object-displacement manipulation.

5. Conclusion

We propose EC-Flow, a framework that learns embodiment-centric flow directly from action-unlabeled videos. By jointly predicting flow and goal images, it captures both the robot's motion and object interactions. Moreover, actions are computed using only a standard URDF file, making the method easy to deploy. Experiments in simulation and the real world show that EC-Flow effectively handles diverse manipulation tasks, including deformable objects, occlusions, and non-object-displacement scenarios.

6. Acknowledgments

This work was jointly supported by National Natural Science Foundation of China (62322607, 62236010 and 62276261), and Youth Innovation Promotion Association of Chinese Academy of Sciences under Grant 2021128.

References

[1] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision*, 2024. 2, 6, 7, 8, 13

[2] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 2

[3] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Rich Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 2

[5] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 1, 2

[6] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023. 6

[7] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36, 2024. 6

[8] Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, brian ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Pack Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2

[9] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2020. 12

[10] Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Cai Zhehao, and Lin Shao. FLIP: Flow-centric generative planning as general-purpose manipulation world model. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[11] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5842–5850, 2017. 12

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 3, 6, 13

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[14] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024. 3

[15] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, pages 2679–2713. PMLR, 2025. 2

[16] Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 6

[17] Peiyan Li, Yixiang Chen, Hongtao Wu, Xiao Ma, Xiangnan Wu, Yan Huang, Liang Wang, Tao Kong, and Tieniu Tan. Bridgevla: Input-output alignment for efficient 3d manipulation learning with vision-language models. *arXiv preprint arXiv:2506.07961*, 2025. 2

[18] Peiyan Li, Hongtao Wu, Yan Huang, Chilam Cheang, Liang Wang, and Tao Kong. Gr-mg: Leveraging partially-annotated data via multi-modal goal-conditioned policy. *IEEE Robotics and Automation Letters*, 2025. 1, 2

[19] Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. In *8th Annual Conference on Robot Learning*, 2024. 1, 2

[20] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1b: a diffusion foundation model for bimanual manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[21] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. In *Robotics: Science and Systems*, 2023. 2

[22] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Proceedings of The 6th Conference on Robot Learning*, pages 892–909. PMLR, 2023. 6

[23] Fei Ni, Jianye Hao, Shiguang Wu, Longxin Kou, Jiashun Liu, Yan Zheng, Bin Wang, and Yuzheng Zhuang. Generate subgoal images before act: Unlocking the chain-of-thought reasoning in diffusion model for robot manipulation with multimodal prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13991–14000, 2024. 1, 2

[24] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024. 2

[25] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025. 2

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3, 6, 13

[27] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollar, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations*, 2025. 3

[28] Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. *arXiv preprint arXiv:2501.06994*, 2025. 2, 3

[29] Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, et al. Grounding dino 1.5: Advance the" edge" of open-set object detection. *arXiv preprint arXiv:2405.10300*, 2024. 3

[30] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 3

[31] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 4

[32] Quan Vuong, Sergey Levine, Homer Rich Walke, Karl Pertsch, Anikait Singh, Ria Doshi, Charles Xu, Jianlan Luo, Liam Tan, Dhruv Shah, Chelsea Finn, Max Du, Moo Jin Kim, Alexander Khazatsky, Jonathan Heewon Yang, Tony Z. Zhao, Ken Goldberg, Ryan Hoque, Lawrence Yunliang Chen, Simeon Adebola, Gaurav S. Sukhatme, Gautam Salhotra, Shivin Dass, Lerrel Pinto, Zichen Jeff Cui, Siddhant Haldar, Anant Rai, Nur Muhammad Mahi Shafiullah, Yuke Zhu, Yifeng Zhu, Soroush Nasiriany, Shuran Song, Cheng Chi, Chuer Pan, Wolfram Burgard, Oier Mees, Chenguang Huang, Deepak Pathak, Shikhar Bahl, Russell Mendonca, Gaoyue Zhou, Mohan Kumar Srirama, Sudeep Dasari, Cewu Lu, Hao-Shu Fang, Hongjie Fang, Henrik I Christensen, Masayoshi Tomizuka, Wei Zhan, Mingyu Ding, Chenfeng Xu, Xinghao Zhu, Ran Tian, Youngwoon Lee, Dorsa Sadigh, Yuchen Cui, Suneel Belkhale, Priya Sundaresan, Trevor Darrell, Jitendra Malik, Ilija Radosavovic, Jeannette Bohg, Krishnan Srinivasan, Xiaolong Wang, Nicklas Hansen, Yueh-Hua Wu, Ge Yan, Hao Su, Jiayuan Gu, Xuanlin Li, Niko Suenderhauf, Krishan Rana, Ben Burgess-Limerick, Federico Ceola, Kento Kawaharazuka, Naoaki Kanazawa, Tatsuya Matsushima, Yutaka Matsuo, Yusuke Iwasawa, Hiroki Furuta, Jihoon Oh, Tatsuya Harada, Takayuki Osa, Yujin Tang, Oliver Kroemer, Mohit Sharma, Kevin Lee Zhang, Beomjoon Kim, Yoonyoung Cho, Junhyek Han, Jaehyung Kim, Joseph J Lim, Edward Johns, Norman Di Palo, Freek Stulp, Antonin Raffin, Samuel Bustamante, João Silvério, Abhishek Padalkar, Jan Peters, Bernhard Schölkopf, Dieter Büchler, Jan Schneider, Simon Guist, Jiajun Wu, Stephen Tian, Haochen Shi, Yunzhu Li, Yixuan Wang, Mingtong Zhang, Heni Ben Amor, Yifan Zhou, Keyvan Majd, Lionel Ott, Giulio Schiavi, Roberto Martín-Martín, Rutav Shah, Yonatan Bisk, Jeffrey T Bingham, Tianhe Yu, Vidhi Jain, Ted Xiao, Karol Hausman, Christine Chan, Alexander Herzog, Zhuo Xu, Sean Kirmani, Vincent Vanhoucke, Ryan Julian, Lisa Lee, Tianli Ding, Yevgen Chebotar, Jie Tan, Jacky Liang, Igor Mordatch, Kanishka Rao, Yao Lu, Keerthana Gopalakrishnan, Stefan Welker, Nikhil J Joshi, Coline Manon Devin, Alex Irpan, Sherry Moore, Ayzaan Wahid, Jialin Wu, Xi Chen, Paul Wohlhart, Alex Bewley, Wenxuan Zhou, Isabel Leal, Dmitry Kalashnikov, Pannag R Sanketi, Chuyuan Fu, Ying Xu, Sichun Xu, brian ichter, Jasmine Hsu, Peng Xu, Anthony Brohan, Pierre Sermanet, Nicolas Heess, Michael Ahn, Rafael Rafailov, Acorn Pooley, Kendra Byrne, Todor Davchev, Kenneth Oslund, Stefan Schaal, Ajinkya Jain, Keegan Go, Fei Xia, Jonathan Tompson, Travis Armstrong, and Danny Driess. Open x-embodiment: Robotic learning datasets and RT-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition @ CoRL2023*, 2023. 2

[33] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023. 2

[34] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, 2024. 2

[35] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the

cross-domain manipulation interface. In *8th Annual Conference on Robot Learning*, 2024. 2

[36] Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[37] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, 2019. 6, 12

[38] XinQiang Yu, Jialiang Zhang, Haoran Liu, Danshi Li, Haoran Geng, He Wang, Yufei Ding, and Jiayi Chen. Dexgraspnet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes. In *2nd Workshop on Dexterous Manipulation: Design, Perception and Control (RSS)*, 2024. 12

[39] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. In *8th Annual Conference on Robot Learning*, 2024. 2, 3

[40] Wentao Zhao, Jiaming Chen, Ziyu Meng, Donghui Mao, Ran Song, and Wei Zhang. Vlmpc: Vision-language model predictive control for robotic manipulation. In *Robotics: Science and Systems*, 2024. 1, 2

[41] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. VIOLA: Object-centric imitation learning for vision-based robot manipulation. In *6th Annual Conference on Robot Learning*, 2022. 7

[42] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023. 2