Object-centric 3D Motion Field for Robot Learning from Human Videos

Zhao-Heng Yin[1] **Sherry Yang**[1,2] **Pieter Abbeel**[1]
[1]BAIR, UC Berkeley EECS
[2]Google DeepMind

https://zhaohengyin.github.io/3DMF

Abstract

Learning robot control policies from human videos is a promising direction for scaling up robot learning. However, how to extract action knowledge (or action representations) from videos for policy learning remains a key challenge. Existing action representations such as video frames, pixelflow, and pointcloud flow have inherent limitations such as modeling complexity or loss of information. In this paper, we propose to use object-centric 3D motion field to represent actions for robot learning from human videos, and present a novel framework for extracting this representation from videos for zero-shot control. We introduce two novel components in its implementation. First, a novel training pipeline for training a "denoising" 3D motion field estimator to *extract* fine object 3D motions from human videos with noisy depth robustly. Second, a dense object-centric 3D motion field *prediction architecture* that favors both cross-embodiment transfer and policy generalization to background. We evaluate the system in real world setups. Experiments show that our method reduces 3D motion estimation error by over 50% compared to the latest method, achieve 55% average success rate in diverse tasks where prior approaches fail ($\lesssim$ 10%), and can even acquire fine-grained manipulation skills like insertion.

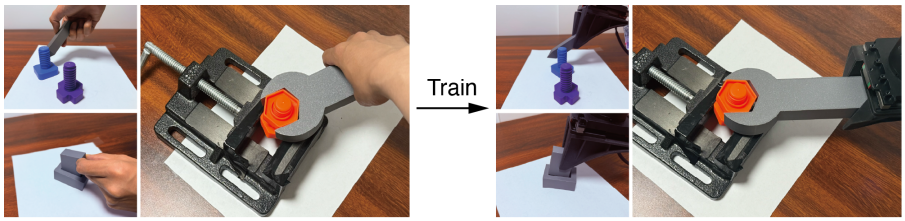

Figure 1: We propose a novel framework for robot learning from human demonstration videos *without* relying on any robot-collected data. Our approach learns to control robots by extracting and modeling 3D object motion fields from RGBD human videos.

1 Introduction

Data is the primary bottleneck in robot learning – collecting large-scale high quality robotic data in real world at scale for training control policies is not only expensive but also mentally challenging for humans in complex tasks [1, 55]. Recently, human-object interaction videos stand out as a particularly promising avenue to overcome this challenge. These videos are not only scalable—given the vast

Correspondence to: zhaohengyin@cs.berkeley.edu

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

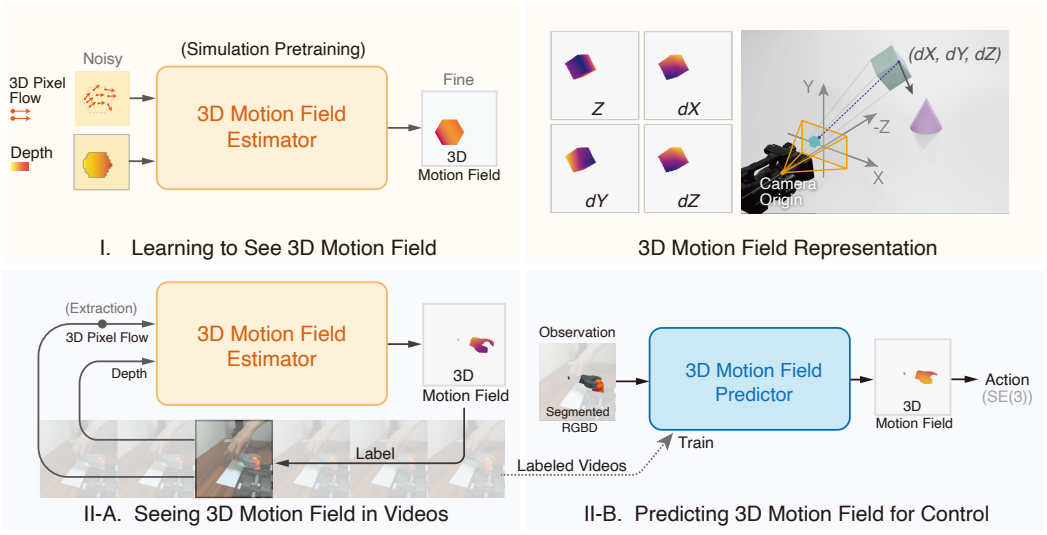

Figure 2: Overview of proposed learning framework. We first pretrain a 3D motion field estimator in simulation (Phase I) and use it to estimate the 3D motion field in *noisy* RGBD human videos (Phase II-A). Then, we train policies to predict the estimated 3D motion field and use them to control robots in a zero-shot manner (Phase II-B). *Unlike* existing 3D tracking works that assume depth as a groundtruth reference, we recover accurate 3D object motion from noisy depth.

amount of footage available from internet or wearable device recordings — but they also capture rich, naturalistic demonstrations of complex tasks. Moreover, collecting data through human hands is inherently easier, cheaper, and more intuitive than through robotic teleoperation [44, 8, 39].

Due to this data collection challenge, many works look into the feasibility of using real-world action-free videos for robot learning. Some recent works, such as UniPi [9] and UniSim [54], directly apply video prediction for control, which essentially view future video frames as action representations and use inverse dynamics model to translate the predicted future frame to actions. Although this line of work achieved some preliminary success, video frames turn out to be an overly noisy, redundant action representation, which not only unnecessarily complicates training and inference but also makes policy non-robust due to blurry details in videos. Therefore, recent works also explored more compact action representations extracted from videos, such as pixel-flow [49, 52, 3], point-cloud flow [59], and SE(3) pose transformation [15]. Nevertheless, each of these representations suffers from certain drawbacks. Pixel-flow is a 2D representation and drops the important 3D movement information for recovering actions. Point-cloud 3D flow is noisy and cannot represent motion accurately. SE(3) pose extraction relies on object 3D models and limits themselves to rigid bodies. We refer readers to the related works (Section 6) for a complete discussion and comparisons. In conclusion, so far, it remains unclear what serves as a good and feasible action representation for video-based robot learning.

In this work, we propose to use an object-centric 3D motion field representation as the action representation for control. Specifically, it is a dense position and motion field over the object pixels in the input image, representing how each observable point on an object should move in each task (Figure 2). This action representation has many advantages compared to previous works: 1. It preserves minimal sufficient 3D information for robot control; 2. It is a image-based representation and allows the use of well-studied powerful image generative models; 3. It is object-centric and embodiment-agnostic, simplifying cross embodiment transfer; 4. Its extraction fully depends on RGBD video and does not assume additional information like object 3D models.

While it may sound appealing, extracting such information from video turns out to be challenging. Although it is relatively simple to extract accurate 2D pixel flow movement part using the latest tracking model, the depth channel is full of noisy measurement values (e.g. missing or wrong values) and directly using the raw depth values with for computing 3D motion will only result in inaccurate motion. Our key insight is that we can build a "denoising" 3D motion field estimator to reconstruct 3D position and motion flow robustly from the noisy depth measurement with simulation data, as depth noise has some easy-to-simulate characteristics. Since this task is fully geometrical and does

not involve complex RGB textures, this simulation-trained estimator can transfer to real world well. Utilizing this, we can train control policies to predict reconstructed high-quality 3D motion flow, which is then translated to robot actions in downstream tasks for control. We evaluate our proposed components and system in several real world object manipulation and tool-use tasks. Our system reduces 3D motion field estimation error by over 50% compared to latest works and achieves $\sim$55% zero-shot success rate on average where prior approaches fall short ($\lesssim$ 10%). Remarkably, we also show that the policy trained solely on human (hand) videos is capable of performing fine-grained manipulations, such as precise insertions – a level of skill not previously shown in this setting.

In summary, our main contributions are as follows. 1). We propose to use object-centric 3D motion field for robot learning from videos and present a novel learning framework for extracting this representation for control. 2). We present a simple and novel architecture that can learn to see and predict object-centric 3D motion field in the real world for control. With this, we can teach robots new skills with human videos as the **only** training data. 3). We validate our proposed components in the real world. We demonstrate that our motion extraction pipeline can significantly reduce motion estimation error by over 50%. For its robotics application, we significantly outperformed existing approaches and show that our policies trained solely on human videos can achieve finegrained manipulation skill for the first time, demonstrating the potential of video-based robot learning.

2 Preliminaries

2.1 What Can Be Learned from Videos?

We begin by discussing what knowledge can be learned from videos and why we focus on extracting and learning object movements. Recently, some works propose to extract low-level human finger movement from videos and aim at directly retargeting them to robots [51, 36, 32, 20, 21]. Although this might be a feasible approach if robot embodiment is highly similar to human hand, we argue that learning direct embodiment control is both unnecessary and challenging. On one hand, the ultimate knowledge we want to learn is how each object in the view should move in each task, and it does not matter too much how we control a robot to generate such object movement as long as we can achieve it. The state-of-the-art robots have built-in functionality to realize arbitrary object movements: i.e., by calling well-established foundational grasping policy and task-space SE(3) movement commands, and therefore we should focus on extracting object motion. On the other hand, generating reliable action through naive retargeting is hard [57, 24]. In some cases, the movement performed by a human hand might not be realizable by a robot (e.g. when robot has different or fewer fingers) and action retargeting will be undefined. Actually, existing methods typically require the human hand to act as a gripper and avoid in-hand manipulations in the training videos. This is not natural for human at work and introduces extra burdens. We conclude with the following assumption and proposal.

Assumption 1 We assume we have access to a reliable robot grasping/ungrasping (i.e. object attaching/detaching) policy. This is a reasonable assumption given success in previous work on robot grippers [11] or hands [61, 56]. Different robots should have their specific grasping policy.

Proposal We propose to focus on extracting and learning object 3D motions for robot learning from videos. Combining an object flow prediction model with the assumed grasping policies above, a robot can solve a wide range of tasks.

2.2 Necessity of Depth Perception

To extract the object 3D motion from video effectively, we assume RGBD videos as training data. While there is growing interest in leveraging RGB-only information for action extraction, we argue that incorporating depth information is necessary for accurate estimation and learning. Otherwise, there exist infinitely many 3D transformations to realize observed pixel motion. Even if we assume some priors over underlying 3D movement, this is still extremely challenging and brittle. For example, suppose that we have a point $(X, 0, Z)$ in the camera frame and it is translating along z-axis. Our goal is to recover this small z-axis motion ΔZ from pixels. Assuming a pinhole camera with focal length f, the observed pixel movement on x-axis is $\Delta x_p = \frac{fX}{Z+\Delta Z} - \frac{fX}{Z} \approx -\frac{fX}{Z^2}\Delta Z$, and we

However, we may require some "functional knowledge" as constraints: e.g. do not hold the sides of a cup/ should grasp the tool handle. Therefore in the long run, besides object motion, we also need to consider extracting contact (semantical affordance).

rearrange it as $\Delta Z = -\frac{Z^2}{fX}\Delta x_p$. The problem is that Δx_p might be noisy in practice and it can lead to huge estimation error in ΔZ due to the large slope $-Z^2/fX$, especially near $X = 0$ (i.e. when the camera is looking at the object), and even a 1-pixel error can result in over 1 centimeter difference! This error is disastrous for robot manipulation task requiring high accuracy – to overcome this we need robust subpixel-level trackers. Although this might be possible in the future, having RGBD video instead of RGB video can make learning much easier. Therefore:

Assumption 2 We require access to RGBD videos collected through pinhole cameras with known camera intrinsics (i.e. camera focal length f_x, f_y and center c_x, c_y), instead of general RGB videos collected by unknown camera with distortions. Although it might seem like a waste not to reuse RGB videos already available on the internet, accumulating new RGBD videos can be easier than one may think of – Million hours of videos are being produced every day [58] and many existing daily camera devices are already equipped with depth sensing (e.g. latest mobile phones and wearable devices).

2.3 3D Motion Field

As we have discussed, our goal is to extract an object-centric 3D motion field from RGBD videos for control. In this paper, a 3D motion field [42, 23, 40] is a dense, image-based representation defined as follows.

Definition (3D Motion Field) Given a pair of $H \times W$ images I_0, I_1 (current frame and next frame), the 3D motion field F of I_0 is defined as 4-channel image tensor in $\mathbb{R}^{H\times W\times 4}$. The first channel $F[:,:,0]$, denoted as F_{depth}, is the depth value of each pixel in I_0. The remaining three channels $F[:,:,1:4]$, denoted as F_{motion}, represent the underlying 3D movement of each pixel between the 2 frames under the I_0 camera frame.

Note that we include depth in this definition. Then, given the camera intrinsics and the 3D motion field, we can reconstruct the position and movement of every pixel in the 3D space.

3 Phase I: Seeing 3D Motion Field in Noise

Our first step is to extract accurate 3D motion fields from noisy RGBD videos. We first discuss a very simple pipeline for this purpose as suggested by latest works [59] and its fundamental limitations, and then we introduce our improved approach. The discussion below assumes the depth observation of two consecutive images I_0 and I_1, and a dense pixel correspondence computed by a video tracker.

3.1 Direct Approach

We first assigns the camera depth to the $F_{depth} = F[:,:,0]$ depth channel. Then, we run a pixel tracker (e.g. Cotracker [17]) to decide the pixel correspondence between I_0 and I_1. Let a pixel (x, y) in I_0 correspond to (x', y') in I_1. Since we have camera intrinsics and their depth values Z and Z', we apply camera inverse projection to get their 3D coordinates (X, Y, Z) and (X', Y', Z'), and set $F_{motion} = F[y, x, 1:4] = (X', Y', Z') - (X, Y, Z)$, i.e., the 3D space movement. The procedure above assumes a static camera but it also applies to moving camera (transform (X', Y', Z') to I_0 coordinate frame first).

The direct method can be effective if we have perfect depth and perfect tracker, but unfortunately, this is untrue in practice. First, the commonly used depth cameras are sensitive to lighting and particularly erroneous when it comes to moving objects – there are numerous holes (missing values) and white noises across the depth image. Second, even if we might have high quality learning-based binocular depth sensing method to improve the depth accuracy, noises from the pixel tracker can also lead to significant errors in motion. For instance, if the pixel tracker mistakenly maps a foreground object pixel in I_0 to a background pixel or a hidden pixel in the next frame I_1 (which is common for object boundary pixels), we will obtain incorrect depth Z' and consequently wrong X' and Y' as $X' = Z'x'/f_x$ and $Y' = Z'y'/f_y$. In both cases, we will obtain a wrong motion field. Due to this, prior works typically use intensive heuristics-based filtering to reduce outliers in observations. Nevertheless, heuristics-based filtering can be unreliable, hence not eliminating the problem thoroughly. This will result in a noisy motion field representation and make the subsequent prediction a hard problem.

3.2 Our Improved Approach: Learning to See 3D Motion Field

To remedy this issue, we propose to learn a 3D motion field estimator to reconstruct the groundtruth smooth 3D motion field with the noisy sensor measurements. As shown in Figure 2 and 4, the

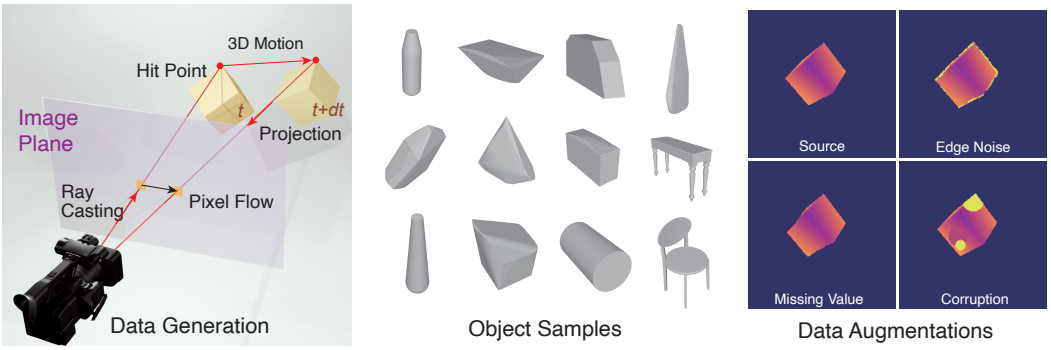

Figure 3: (Left) Phase I Synthetic Data Generation. We randomly generate object and 3D motions, and use ray casting and projection to obtain 3D pixel flow input and the 3D motion field label. (Middle) Random Objects for Data Generation. (Right) Depth Samples and Data Augmentations.

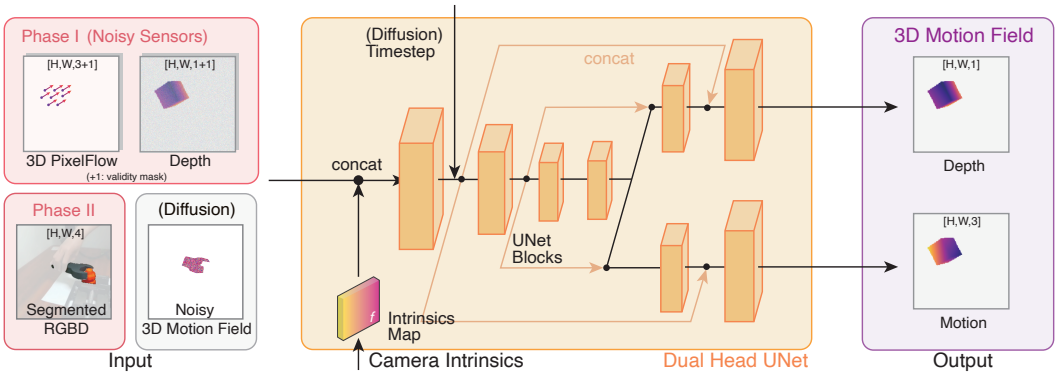

Figure 4: Model Architecture. The most important design is a dense intrinsics map feature concatenated to the input, which contains key information for reconstructing the groundtruth 3D flow. Note that Phase I and II train separate models: Phase I uses noisy object depth and 3D pixel flow as input and Phase II uses segmented RGBD and a noisy 3D motion field (in the case of Diffusion Model). For Phase I, we also use a validity mask value channel to indicate which feature values are defined.

main inputs to this model are noisy dense object 3D pixel flow (2D pixel flow (x, y-axis) and depth flow (z-axis)) and the noisy depth image. The output is the reconstructed 3D motion field. Since all the input and output information are geometrical (no RGB textures), we propose to use a simulator to generate training data due to minimal sim-to-real gap for geometrical data. The dataset, model, and training method are as follows.

Dataset Generation We use the objects in the ShapeNet dataset [5] and some randomly generated regular rigid bodies as training objects. We rescale all the objects to regular daily object sizes, which is about 4$\sim$20cm on each dimension. Then, we randomly generate a camera with a random field of view (FoV) between 40$\sim$55° (common camera FoV), and randomly place the object in the camera view to render the initial depth frame. Then, we generate a random twist motion $\mathcal{S} = [v, w]$ to move (translate and rotate) this object for several steps. We calculate both the 3D pixel movement and the groundtruth 3D motion of each observed pixel in the initial frame through ray casting followed by transformations as shown in Figure 3. We generate 8M samples at 256×256 resolution for training, which can be produced with 1 NVIDIA L40 GPU in less than 12 hours.

Data Augmentation During training, we use diverse data augmentations to simulate the noise effect of each sensor observations, and the underlying idea is relevant to the Denoising Autoencoder [43] which reconstructs signals from sensor noises for downstream processing. The most common noise for depth is random missing values, white noises, and wrong values (especially for moving objects), and we apply these effects on the depth image. For the pixel flow input, since we find existing trackers are usually off by a few pixels, we apply random Gaussian noise as augmentation. Besides, we also apply random dropout on the pixel flow input, i.e., only using part of the flow vectors for prediction. This also allows us to use partial, sparse pixel flow for inferring 3D motion field, so that we can speed

up the data labeling process. Finally, we also apply subset masking to the input feature map. With this, we can approximate complex object contours with simple objects.

Model: 3D Motion Field Estimator We use a dual head UNet [33] model as our 3D motion field estimator f. This model predicts F_{depth} and F_{motion} through two separate low-level decoder branches f_{depth} and f_{motion}. This is to reduce interference near the output as these predicted values have different semantical meanings. Besides, we also append a dense, "intrinsic" map feature $I_{map} \in \mathbb{R}^{H \times W \times 4}$ to the image, whose elements are given by

$$I_{map}[y, x] = ((y - c_y)/f_y, (x - c_x)/f_x, 1/f_y, 1/f_x). \tag{1}$$

I_{map} contains crucial low-level information for accurate F_{motion} prediction. To understand this, let us consider motion prediction along x-axis as an example. Since $X = (x - c_x)Z/f_x$, by taking differential we have $dX = (Z/f_x)dx + [(x - c_x)/f_x]dZ$. The (noisy) 2D pixel motion dx, depth motion dZ, and depth Z are already included as input, while the remaining $1/f_x$, $(x - c_x)/f_x$ terms cannot be inferred by a CNN architecture without position embeddings. Hence, we also include them explicitly as input as well for predicting x-axis motion dX. These features will directly propagate to the last layers in UNet through skip concatenations. We show their importance in Figure 8.

Training We apply a weighted Huber loss ($\|\cdot\|$) as a stable supervision to train this model:

$$\mathcal{L} = \mathbb{E}_{x,F,M \sim \mathcal{D}_{sim}} \|M \odot (f_{depth}(x) - F_{depth})\| + \alpha \|M \odot (f_{motion}(x) - F_{motion})\|. \tag{2}$$

In the loss function above, $\mathcal{D}_{sim}$ is the generated synthetic dataset. M is the object mask and $\odot$ is the (broadcast) elementwise product, so the dense motion field prediction loss is only applied on the object part. α is a weighting hyperparameter. We will also present a diffusion-based architecture and objective in the next section, but for this motion estimation task, a direct prediction without further refinement is already good enough. We use the AdamW optimizer [22] to train this model and the training procedure takes about 1 day with 16 NVIDIA A100-40GB GPUs.

Discussion I: Motion and Geometry Synergy A key idea in our prediction task is the synergy between the motion and geometry (depth) – the motion clue can be used to recover the missing or wrong depth value. One example is shown in Figure 5. If we observe that some pixels move together with others whose depth values are known, we can reasonably infer their depth — particularly when the observed object is rigid. This formulation can also be considered as masked pretraining of 4D (time plus 3D) geometrical data. Most importantly, the tracks on the 2D plane are usually very visually accurate (< 3 pixel error), making such "masked" reconstruction feasible. In the experiments, we find that adding motion and camera intrinsics information can significantly improve depth prediction accuracy.

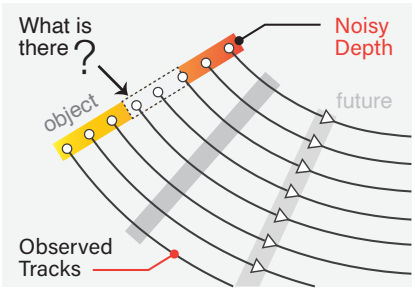

Figure 5: Guess what is there? Object tracks can be used to recover missing or wrong depth values.

Discussion II: Extension to Moving Camera In our current implementation, we assumed a static camera. However, since the data is produced in simulation, this framework can be extended to estimating 3D motions in videos captured by moving cameras (through simulating camera movement). Then again, one can append an additional dimension of (noisy) camera motion as input to the estimator and use it to process general videos (the camera motion in videos can be inferred via monocular simultaneous localization and mapping (SLAM) [10]). We leave this extension as a future work.

4 Phase II: Predicting Object 3D Motion Field for Control

Now that we have a model to see the groundtruth 3D motions from noisy sensors, we are ready to build our control policies with human videos.

Dataset: Human Videos We only require human object interaction video dataset $\mathcal{D}_{human}$ to train our control policy. We first process the dataset through our learned estimator and existing foundation models. Specifically, we first apply foundation segmentation model SAM2 [31] in video mode to extract the segmentation of all the task-relevant objects in each frame. Then, we use pixel trackers (CoTracker3 [17]) to extract the noisy 3D pixel flow of each object point and lift them to an

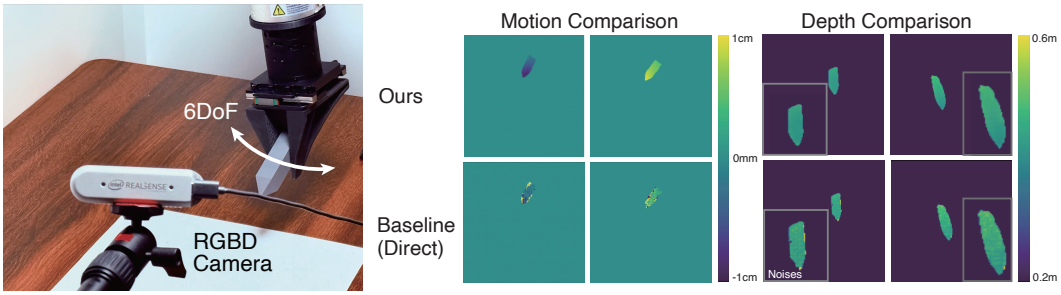

Figure 6: (Left) Experimental Setup. (Right) Qualitative Results on "Pen" (Left Figure). Our method produces smoother motion field and depth compared to baseline. This is essential for accurate control.

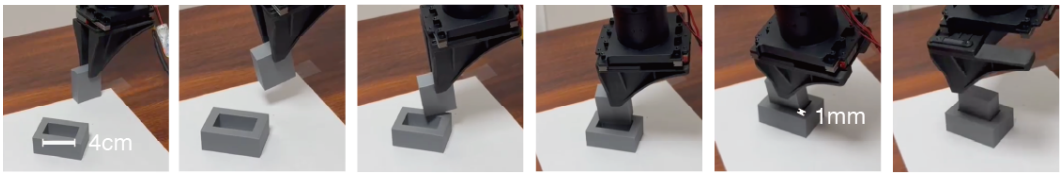

Figure 7: A rollout of fine-grained insertion. Our method can achieve high precision, even if we are observing the motion from 40cm away without a wrist camera. However, we observe an adjustment behavior resembling bang-bang control – it is still challenging for the policy to insert accurately in one-shot as human. Fortunately, it still captures rough directions for adjustment to finish the task.

accurate 3D motion field through our pretrained estimator. However, an underlying requirement is that we require the object consistently visible (i.e. not fully occluded) throughout the video segments, so we have to discard the fully occluded segments and learn from all the remaining partially occluded and fully visible segments. We leave the full occlusion case to future investigation.

Model and Training Then, we train a policy network π to predict these labeled 3D motion field with the segmented RGBD image as input. Since our motion field is image-shaped, we can use either a Gaussian or a diffusion model (policy) [14, 38, 7] for its accurate prediction. We reuse most of the dual-head UNet architecture as our policy π. We apply the following general regression objective for training (for both diffusion and Gaussian policy):

$$\mathcal{L}_\pi = \mathbb{E}_{o,F,M\sim\mathcal{D}_{human}} \|M \odot (\pi_{depth}(o, \tilde{F}, t) - F_{depth})\| + \alpha \|M \odot (\pi_{motion}(o, \tilde{F}, t) - F_{motion})\|. \tag{3}$$

In this objective above, o is the segmented RGBD image observation, F is the groundtruth object 3D motion field (desired action over the object) labeled by our estimator, and M is the mask of the corresponding object extracted by SAM2. $(\tilde{F}, t)$ is the noised motion field sample and timestep and only apply to diffusion model. As our policy network only uses task-relevant object information as input and does not observe any embodiment-specific information, the gap between human domain and robot domain is minimal. However, we still find it important to apply a random masking data augmentation to objects. Besides, for diffusion model we also find it useful to use "masked noise sample" as input – this simplifies training and makes the model more robust.

Deployment In the inference time, we need to convert the predicted 3D motion field F to the robot action. As the object is already firmly grasped by the robot, this conversion is straightforward. For each pixel on the object mask, given F, we compute their current and future 3D coordinates in the camera frame through camera inverse projections, resulting in two point clouds $P_0, P_1 \in \mathbb{R}^{N\times 3}$. Importantly, these two point clouds have point-wise correspondence (as opposed to unordered point clouds that require ICP [2]) so that we can directly solve a SE(3) transformation $\mathbf{T}_o = \{\mathbf{R}, \mathbf{t}\}$ for aligning them. We minimize $\|\mathbf{R}P_0^T + \mathbf{t} - P_1^T\|^2$, which has a closed form solution (Kabsch method [16]). As there are outliers in P_0 and P_1 inevitably, so we also use RANSAC [13] to improve the quality of $\mathbf{T}_o$. Then, suppose the camera pose in robot base frame $\{b\}$ is $\mathbf{T}_{bc}$, our desired robot action can be computed as $\mathbf{T}_a = \mathbf{T}_{bc}\mathbf{T}_o\mathbf{T}_{bc}^{-1}$. The conversion from object 3D motion field to SE3 action is very fast at around 300-1000Hz, introducing minimal overhead to control loop.

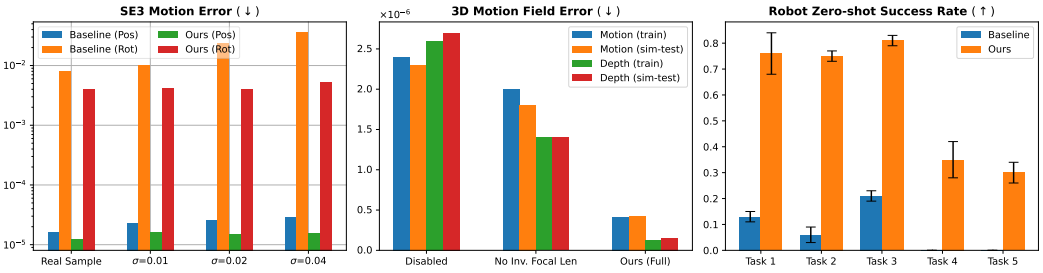

Figure 8: (Left) SE3 motion estimation performance in real world. Our method achieves lower error compared to baseline. (Middle) Intrinsics Map Ablation Studies. Both inverse focal length and coordinate map are crucial. (Right) Real world Task Success Rate (3 seeds). Baseline is the latest General Flow [59]. Other recent methods fail on our setup due to their limitations (Table 2).

5 Experiments

In this section, we demonstrate the effectiveness of our 3D motion field estimator and our control policy through real world experiments.

System Setup We use a widely-used Intel D435 RGBD camera at 640×480 resolution for video dataset collection at 30Hz (Figure 6). During training, we crop the image to 480×480 and rescale it to 256×256. To get a high-quality raw depth, we use the native temporal and spatial filters provided by the camera SDK. We use an XArm7 robot arm with a parallel-jaw gripper for the test dataset collection and robot experiments. We did not use a wrist camera in the experiments as previous works. In the experiments, the manipulated object is typically 40$\sim$50cm away from the camera.

5.1 Evaluating 3D Motion Field Estimator

Setup We capture RGBD video of randomly moving objects in the real world with their corresponding ground truth transformations as the test set. We program the robot arm to grasp random objects and wave them in front of the camera. Since the robot gripper grasps the object, we can directly obtain the groundtruth object 3D transformation by reading and calculating robot gripper pose transformation. We use objects of diverse aspect ratios and apply diverse robot motions.

Main Results We show the results of our method and the latest baseline "direct" method in Figure 8. We report the MSE of recovered object translation and the F-norm of object rotation matrix error. Note that we use the gripper frame as the base for representing these transformations. We find that our approach has significantly lower error compared to the baseline. We also visualize some examples in Figure 6. Our method successfully reproduces smooth depth and motion field even if the input depth is very noisy. We refer readers to appendix for more details.

Adversarial Robustness We test robustness further through adversarial attack in real world experiments by injecting Gaussian noise of different intensities into the depth observation (which then propagates into the computed 3D pixel flow input). We find that the baseline performance deteriorates quickly, while the error of our method remains at the same level. However, this is somewhat expected since the same kind of noise is also applied during our training process.

Ablation Studies We also ablate intrinsic maps as shown in Figure 8 (Middle). We find that both coordinates and *inverse focal length* are crucial for successful learning, confirming our derivation. Surprisingly, the focal length value plays a critical role in motion prediction even for a relatively small FoV variation around 10 degrees.

5.2 Robot Learning from Videos

Real world Tasks In this section, we test if our method can acquire object manipulation skills from human videos. We introduce the following tasks to benchmark the performance:

1. *Pick, Rotate, and Place*. The robot is required to pick up an object and/or rotate it and put it at a target location. This task is considered as successful only if the object is in correct pose in the end.
2. *Line Tracking*. The robot is required to pick up a pen-like flashlight and control it to track a cable on the table. This task is considered successful if the robot can finish the tracking trajectory with spotlight focusing on the cable in the process. The procedure is monitored by a human.

Table 2: Technical Feature Comparisons. Our method is free from many limitations of existing works. Note that this list is non-exhaustive and we refer readers to the text for more discussion.

Method	UniPi [9]'23	ATM [49]'24	Track2Act [3]'24	GFlow [59]'24	Im2Flow2Act [52]'24	SPOT [15]'24	TI [6]'25	Ours
No Pose Estimation	✓	✓	✓	✓	✓	×	×	✓
No Robot Data (Zero-shot)	×	×	×	✓	×	✓	✓	✓
Close-loop *Motion* Control	✓	✓	✓	✓	×	✓	✓	✓
Depth Robustness	N/A	N/A	N/A	×	N/A	N/A	N/A	✓
Distractor Generalization	×	×	×	×	×	×	×	✓

3. *Tool Use I: Pushing*. The robot is required to pick up a tool and push one object to a goal.
4. *Tool Use II: Wrench*. In this task, the robot is required to pick up a wrench and use it to tighten a nut by a round. This task can be viewed as a more challenging version of pick, rotate, and place since the process is kinematically constrained and requires precision.
5. *Insertion*. In this task, the robot is required to pick, rotate, and insert an item into a slot (hole). There is only 2.5mm tolerance so this requires very high precision.

We collect around 50-150 human videos for each of these tasks. The data collection procedure for each of the tasks lasts around 2-15 minutes, depending on task complexity. During the evaluation, we ensure that the grasping and the object segmentation is correct for each of the evaluated method (otherwise we restart that trial).

Main Results We show the success rate of different methods in Figure 8 Right. We find that our method significantly outperformed the other evaluated methods. During the deployment, the baseline method will quickly deviate from the correct moving direction/trajectory. Our method not only solves the tasks but also follows the human-demonstrated path accurately throughout the evaluation (see Appendix). We attribute its success to accurate and smooth motion estimation. Besides this, we also observe the expected robustness to the background variations during experiments due to the use of object-centric input representation.

Ablation Studies We also study the design choices of our policy architecture and training. We find that for fine-grained tasks, it is important to apply a diffusion model even if the human has tried to act as consistently as possible. Compared to the Gaussian policy head, the diffusion policy can produce high-quality, accurate motion fields which is important for success. Besides, we also find it important to mask out unnecessary regions in the diffusion model reverse step. Otherwise, the irrelevant noise in non-object regions can slow down training and harm performance. Finally, we find it important to apply object masking augmentation during training, as the object's silhouette under the robot gripper differs from that under a human hand, which leads to a subtle domain gap.

Table 1: Policy Learning Ablation for Fine-grained Tasks.

Setting	**Success**
w/o Diffusion (Diff.)	0.0%
w/o Diff. Masking.	0.0%
w/o Masking Data Aug.	5.0%
Full	**35.0%**

6 Related Works

Robot Learning from Videos We summarize the most relevant studies in Table 2. A key aspect of existing methods is how they represent actions. Some approaches rely on direct video frame prediction [12, 9, 54, 4] for control while recent research explores compact and informative representations, such as 2D pixel flow [49, 41, 52, 3], point cloud flow [35, 59], and 3D poses [15, 6]. While these approaches offer certain advantages, each has notable limitations, as previously discussed. Among them, point cloud flow is most relevant to our approach. However, we distinguish our work by employing an image-based 3D motion representation with a denoising architecture to refine the extracted 3D information. Another related line of research involves embodiment-specific action retargeting [51, 32, 20], which can be infeasible in general setups. In contrast, our embodiment-agnostic approach aims to extract common motion knowledge for control.

In addition to action representation, another challenge is domain alignment—ensuring that the learned model can effectively transfer to the robot domain. Recent works tackle this by employing ego

masking or inpainting [20, 18]. We have a similar idea but adopt a more object-centric approach by using task-relevant objects as input. While not entirely new [62], we are the first to apply this concept within the context of learning from human video demonstrations. In a broader context, human videos have also been used as auxiliary data source to pretrain vision-language-action models/policies [**?**, 37] or define RL tasks [29, 30, 28] in simulation, which are orthogonal directions.

3D Vision and Video Analysis Our method is related to 3D reconstruction from videos [10, 26, 34]. Some works in this area have utilized 3D representations such as Neural Radiance Fields (NeRF) [25], Gaussian Splatting [19], and Pointmaps [47, 60, 46] for scene *geometry* reconstruction, while our work focuses on *motion* reconstruction. In recent 3D motion extraction research, some works propose both optimization-based and end-to-end tracking method [27, 50, 45]. Our method shares a similar goal but differs significantly in focus and underlying assumptions. These works typically focus on long-range (time) tracking and assume the raw depth as the reference even if it can be noisy and inaccurate. In contrast, we focus on recovering depth and precise instantaneous motion from noisy temporal depth sensing. In this sense, our work is more aligned with depth estimation [53, 48] — areas traditionally explored within computational photography for 3D. While many existing approaches aim to recover depth from monocular or stereo RGB images as well, our method takes a distinct direction by leveraging the interplay between depth and motion to improve 3D understanding. Finally, we note that our approach is compatible with the methods above – it can be integrated with existing depth and track estimation methods by using their depth and track outputs as inputs.

7 Conclusion

In this paper, we have demonstrated a novel framework for learning robot control policies from human videos using object-centric 3D motion field representations. Our approach overcomes key limitations of existing representations by introducing a robust 3D motion estimator and a dense flow prediction architecture, enabling better cross-embodiment transfer and background generalization. Experiments demonstrate substantial improvements in motion estimation and success rates across diverse real-world tasks compared to prior works, and the unprecendented effectiveness in handling precise manipulation tasks. Our approach opens up new possibilities for leveraging scalable human video datasets to train versatile and generalizable robotic agents.

Acknowledgement

This work is part of the Google-BAIR Commons project. The authors gratefully acknowledge Google for providing computational resources. Zhao-Heng Yin is supported by the ONR MURI grant N00014-22-1-2773 at UC Berkeley. Pieter Abbeel holds concurrent appointments as a Professor at UC Berkeley and as an Amazon Scholar. This research was conducted at UC Berkeley and is not affiliated with Amazon.

Broader Impact

Our method targets one of the most significant challenges in robot learning: data. By addressing this issue through human video data, we open the door to scaling up data for the development of foundational robotic agents.

Limitations

We identify the following limitations in this work as directions for future research. First, we did not consider knowledge extraction in the case of full occlusion – we may need separate action representations for mining action knowledge from fully occluded data. Second, we mainly consider grippers and it would be interesting to study the case of robotic hands – for which we may need to train motion-conditioned control policy rather than solving an optimization problem as we did. Third, in this study, we assumed a fixed camera although we have shown a potential way to scale it to moving cameras (Section 4 Discussion II). Finally, it would be interesting to handle the case of (very) soft-bodies like cloth – for which we should not extract an SE(3) transformation from the whole object movement, but instead only consider a subset of motion field around the contact point. In summary, there are several directions for extending this work and making it a useful industry-level solution.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See the summary part of introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See the limitation section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: There is some short discussion on the limit of existing assumptions.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide details on collecting dataset and model training.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce

the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release the trained model and learning pipeline.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We discuss the setup thoroughly in experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Yes, this is reproducible on several tasks.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discussed them in implementation.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See broader impact section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite them properly.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We use some simulation generated assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

References

[1] S. Belkhale, Y. Cui, and D. Sadigh. Data quality in imitation learning. In *Neural Information Processing Systems (NeurIPS)*, 2023.

[2] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.

[3] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani. Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation. In *European Conference on Computer Vision (ECCV)*, 2024.

[4] 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. In *International Conference on Learning Representations (ICLR)*, 2024.

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

[6] H. Chen, C. Zhu, Y. Li, and K. Driggs-Campbell. Tool-as-interface: Learning robot policies from human tool usage through imitation learning. *arXiv preprint arXiv:2504.04612*, 2025.

[7] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[8] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Robotics: Science and Systems (RSS)*, 2024.

[9] Y. Du, S. Yang, B. Dai, H. Dai, O. Nachum, J. Tenenbaum, D. Schuurmans, and P. Abbeel. Learning universal policies via text-guided video generation. In *Neural Information Processing Systems (NeurIPS)*, 2023.

[10] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[11] H.-S. Fang, C. Wang, M. Gou, and C. Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[12] C. Finn and S. Levine. Deep visual foresight for planning robot motion. In *International Conference on Robotics and Automation (ICRA)*, 2017.

[13] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[14] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Neural Information Processing Systems (NeurIPS)*, 2020.

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

[16] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Foundations of Crystallography*, 32(5):922–923, 1976.

[17] N. Karaev, I. Makarov, J. Wang, N. Neverova, A. Vedaldi, and C. Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024.

[18] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu. Egomimic: Scaling imitation learning via egocentric video. In *International Conference on Robotics and Automation (ICRA)*, 2024.

[19] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):139–1, 2023.

[20] M. Lepert, J. Fang, and J. Bohg. Phantom: Training robots without robots using only human videos. *arXiv preprint arXiv:2503.00779*, 2025.

[21] V. Liu, A. Adeniji, H. Zhan, R. Bhirangi, P. Abbeel, and L. Pinto. Egozero: Robot learning from smart glasses. *arXiv preprint arXiv:2505.20290*, 2025.

[22] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*, 2018.

[23] W.-C. Ma, S. Wang, R. Hu, Y. Xiong, and R. Urtasun. Deep rigid instance scene flow. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[24] Z. Mandi, Y. Hou, D. Fox, Y. Narang, A. Mandlekar, and S. Song. Dexmachina: Functional retargeting for bimanual dexterous manipulation. *arXiv preprint arXiv:2505.24853*, 2025.

[25] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[26] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[27] T. D. Ngo, P. Zhuang, C. Gan, E. Kalogerakis, S. Tulyakov, H.-Y. Lee, and C. Wang. Delta: Dense efficient long-range 3d tracking for any video. In *International Conference on Learning Representations (ICLR)*, 2025.

[28] A. Patel, A. Wang, I. Radosavovic, and J. Malik. Learning to imitate object interactions from internet videos. *arXiv preprint arXiv:2211.13225*, 2022.

[29] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

[30] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. 2022.

[31] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations (ICLR)*, 2025.

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

[33] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[34] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[35] D. Seita, Y. Wang, S. J. Shetty, E. Y. Li, Z. Erickson, and D. Held. Toolflownet: Robotic manipulation with tools via predicting tool flow from point clouds. In *Conference on Robot Learning (CoRL)*, 2023.

[36] K. Shaw, S. Bahl, A. Sivakumar, A. Kannan, and D. Pathak. Learning dexterity from human hand motion in internet videos. *The International Journal of Robotics Research*, 43(4):513–532, 2024.

[37] H. G. Singh, A. Loquercio, C. Sferrazza, J. Wu, H. Qi, P. Abbeel, and J. Malik. Hand-object interaction pretraining from videos. In *International Conference on Robotics and Automation (ICRA)*, 2025.

[38] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

[39] T. Tao, M. K. Srirama, J. J. Liu, K. Shaw, and D. Pathak. Dexwild: Dexterous human interactions for in-the-wild robot policies. In *Robotics: Science and Systems (RSS)*, 2025.

[40] Z. Teed and J. Deng. Raft-3d: Scene flow using rigid-motion embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8375–8384, 2021.

[41] M. Vecerik, C. Doersch, Y. Yang, T. Davchev, Y. Aytar, G. Zhou, R. Hadsell, L. Agapito, and J. Scholz. Robotap: Tracking arbitrary points for few-shot visual imitation. In *International Conference on Robotics and Automation (ICRA)*, 2024.

[42] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 1999.

[43] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning (ICML)*, 2008.

[44] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024.

[45] Q. Wang, V. Ye, H. Gao, J. Austin, Z. Li, and A. Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024.

[46] Q. Wang, Y. Zhang, A. Holynski, A. A. Efros, and A. Kanazawa. Continuous 3d perception model with persistent state. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[47] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud. Dust3r: Geometric 3d vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[48] B. Wen, M. Trepte, J. Aribido, J. Kautz, O. Gallo, and S. Birchfield. Foundationstereo: Zero-shot stereo matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[49] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel. Any-point trajectory modeling for policy learning. In *Robotics: Science and Systems (RSS)*, 2024.

[50] Y. Xiao, Q. Wang, S. Zhang, N. Xue, S. Peng, Y. Shen, and X. Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[51] H. Xiong, Q. Li, Y.-C. Chen, H. Bharadhwaj, S. Sinha, and A. Garg. Learning by watching: Physical imitation of manipulation skills from human videos. In *International Conference on Intelligent Robots and Systems (IROS)*, 2021.

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

[53] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. Depth anything v2. In *Neural Information Processing Systems (NeurIPS)*, 2024.

[54] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations (ICLR)*, 2024.

[55] Z.-H. Yin and P. Abbeel. Offline imitation learning through graph search and retrieval. In *Robotics: Science and Systems (RSS)*, 2024.

[56] Z.-H. Yin and P. Abbeel. Lightning grasp: High performance procedural grasp synthesis with contact fields. *arXiv preprint arXiv:2511.07418*, 2025.

[57] Z.-H. Yin, C. Wang, L. Pineda, F. Hogan, K. Bodduluri, A. Sharma, P. Lancaster, I. Prasad, M. Kalakrishnan, J. Malik, et al. Dexteritygen: Foundation controller for unprecedented dexterity. In *Robotics: Science and Systems (RSS)*, 2025.

[58] YouTube. You know what's cool? a billion hours, February 2017.

[59] C. Yuan, C. Wen, T. Zhang, and Y. Gao. General flow as foundation affordance for scalable robot learning. In *Conference on Robot Learning (CoRL)*, 2024.

[60] J. Zhang, C. Herrmann, J. Hur, V. Jampani, T. Darrell, F. Cole, D. Sun, and M.-H. Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. In *International Conference on Learning Representations (ICLR)*, 2025.

[61] J. Zhang, H. Liu, D. Li, X. Yu, H. Geng, Y. Ding, J. Chen, and H. Wang. Dexgraspnet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes. In *Conference on Robot Learning (CoRL)*, 2024.

[62] Y. Zhu, Z. Jiang, P. Stone, and Y. Zhu. Learning generalizable manipulation policies with object-centric 3d representations. In *Conference on Robot Learning (CoRL)*, 2023.