Do as I Do: Dexterous Manipulation Data from Everyday Human Videos

Bhawna Paliwal* Haritheja Etukuru* William Liang*
Pieter Abbeel Nur Muhammad "Mahi" Shafiullah Jitendra Malik

UC Berkeley

https://do-as-i-do.com

Abstract: How can we scalably generate data for robotic manipulation, especially on human-like platforms such as dexterous multi-fingered hands? Learning from human videos has recently emerged as a likely answer to this question. However, difficulties in estimating hand-object interaction and crossing the human-to-robot embodiment gap have hindered the adoption of abundant monocular RGB-only human videos as the *primary* source of robot manipulation data. In this work, we present DO AS I DO, an algorithm to reconstruct and retarget monocular RGB human videos to multi-fingered dexterous robotic hands. DO AS I DO reconstructs hand-object interactions from various egocentric and exocentric in-the-wild video sources. The algorithm then retargets these hand-object interaction estimates into a sequence of actions executable in the real world, yielding robot-complete manipulation data from disparate human videos. Overall, DO AS I DO outperforms previous state of the art in estimating hand-object interactions and extracting dexterous manipulation trajectories from RGB videos, as we show in experiments on datasets with ground truths and on a dataset of video clips collected online. Our experiments enable us to propose an efficacy playbook for practitioners collecting human data for manipulation.

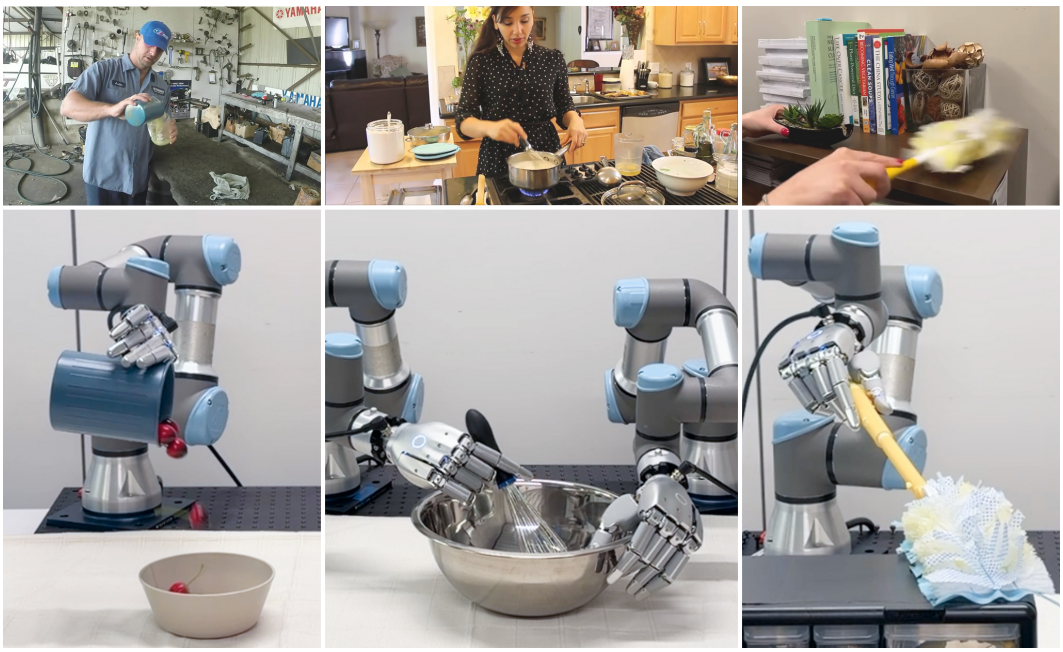

Figure 1: We introduce DO AS I DO, an algorithm that takes in-the-wild monocular RGB videos of hand-object interaction (top) and generates dexterous hand manipulation data (bottom).

*Denotes equal contribution. Correspondence to: bhawna_paliwal@berkeley.edu.

1 Introduction

Data is a critical component of any learning algorithm, including robot learning. For a novice intelligent actor, be it a child, an apprentice, or a robot, however, available data for any task is mostly *observational*: experienced not via *doing* but via *watching* experts do it. This gap in watching and doing is easily closed by children [1, 2], but presents an insurmountable barrier for today's robots, which rely primarily on *experiential* data collected via real-world teleoperation or simulated exploration. Consider the challenges of generating experiential data for dexterous behaviors as shown in Fig. 1. Teleoperation is bottlenecked by operator expertise, cost of operation, and mechanical transparency of the teleoperation rig. Exploration in simulation is similarly bottlenecked by complexities in designing diverse environments and reward functions. The natural question we ask in this work is how to close this gap by converting the accessible, observational data from humans into experiential data for robots.

Investigations into inferring robot actions from watching humans, also known as "Do as I do", are almost as old as the field of artificial intelligence, seen notably in the 1970 MIT "copy-demo" [3], "robot instruction" by direct human demonstration [4] and "Do as I do" with action synthesis and retargeting [5]. The two primary algorithmic challenges are recognition or reconstruction of the human behavior, and retargeting of this behavior on a (possibly vastly different) robotic embodiment. Assumptions on the behaviors, i.e. pick-and-place only, and on the objects, i.e. 3D-scanned only, have made inroads into this challenge possible [6, 7]. Over time, further assumptions on the data modality, such as availability of depth, 3D, or hand keypoints in data collected with specialized hardware, have made more advances in "Do as I do" possible [8, 9]. However, most of our observational data today is stored as monocular RGB videos of humans, and therefore algorithms that can address this general case stand to deliver the largest increase in robotic data.

Here, two recent advances in adjacent fields provide us with a novel approach to this problem. First, 3D computer vision models today can take 2D RGB images and reconstruct depth [10], objects [11], and hands [12] in 3D from purely monocular RGB videos, enabling 4D hand-object estimation. Simultaneously, GPU-parallel physical simulators such as Mujoco Warp [13] and Isaac [14] enable fast sampling-based optimization algorithms [15] that we can use to infer robotic dexterous hand actions in minutes from 4D hand-object states. Note that both of these approaches make minimal assumptions on the problem structure, such as the observed behavior or target objects. Inspired by such advances, in this work we present DO AS I DO, aiming to bridge this gap between observational and experiential data for dexterous manipulation. DO AS I DO recognizes and retargets dexterous human actions from everyday monocular RGB videos and produces multi-fingered robotic hand and arm actions performing the same task on the environment. More importantly, we are able to do so without making limiting assumptions on the displayed behavior (e.g. no grasping priors) or object classes, supporting arbitrary rigid bodies. Concretely, the contributions of this work are as follows:

1. We introduce DO AS I DO, a two-step algorithm to reconstruct and retarget behaviors from monocular RGB videos to multi-fingered dexterous hands.
2. Our hand-object reconstruction process outperforms SOTA on relevant metrics and handles diverse videos — ego- or exo-centric, ranging from in-the-wild internet clips to outputs of generative video models.
3. Our retargeting process improves upon existing scalable dynamics-aware retargeting techniques by introducing novel components that robustify the noisy reconstructed reference trajectories.
4. Our robot data is playable on a dexterous robot hand and arm, completing, to the best of our knowledge, the first pipeline that can go from an internet video to real dexterous hand rollouts.

2 Related Work

Dexterous Manipulation from Human Videos. Many past works explore ways to leverage semantics and motions from human videos. Some extract priors via pretraining, either as visual represen-

Table 1: **Related Work.** We summarize methodology and data sources used by prior work, in rough order of difficulty. Self refers to data collected by the authors.

	Method		Data Sources			
	Reconstruction	Retargeting	Self	Gen.	Ego.	Internet
H2Sim2Robot [16]	LiDAR Scan + FPose [17]	RL	✓			
VideoManip [18]	MeshyAI [19] + FPose	DRO [20] + DP3 [21]	✓		✓	
DexMan [22]	TRELLIS [23] + FPose + SpaTrack [24]	RL	✓	✓		
DexImit [7]	SAM 3D [11] + FPose++ [25]	Motion planning	✓	✓		
Ours	SAM 3D + Guided Diffusion	Sampling-based opt.	✓	✓	✓	✓

tations [26, 27, 28], dexterous policies [29, 30, 31, 32, 33], or forward dynamics models [34, 35]. Other approaches compute more structured priors, such as affordances [36, 37], flows [38, 39], and 3D reconstructions for retargeting [6, 40, 41, 42, 16, 7, 22, 18]. Our work falls in the final category and pushes the frontier in utilizing diverse in-the-wild data sources, as shown in Table 1.

Hand-Object Reconstruction. Reconstructing 4D hand-object interaction from monocular RGB video decomposes along three complementary axes: hand pose estimation, object shape and pose estimation, and their joint modeling. The structured shape and motion of the human hand [43] has enabled hand tracking models [12, 44, 45] robust to motion blur, occlusion, and low resolution. *Object reconstruction and tracking* under the same in-the-wild noisy conditions remain substantially harder because of diversity in everyday objects. Recent progress has come from (1) image-conditioned 3D generative foundation models [46, 23, 11] with robust priors over shape and pose, which can handle occlusions and lower resolution, and (2) model-based 6-DoF trackers [17, 47] that estimate object pose given a known or jointly reconstructed mesh. However, these methods were largely validated on clean lab videos and struggle on in-the-wild noisy videos, as we show below. Finally, *joint hand-object reconstruction* methods reason about both hand and object signals simultaneously, and have shown progress on lab data [48] as well as in-the-wild videos [49, 50, 51]. Recent video methods remain narrow in scope: some are egocentric-only [52], while category-conditioned approaches [53, 54] assume closed object taxonomies. In contrast to these joint approaches, DO AS I DO adopts a *modular decomposition*: HaWoR for hand tracking, SAM3D for single-image object meshing, and our SAM3D-based tracking (Sec. 3.1) for object pose evolution.

Human-to-Robot Retargeting. Dexterous retargeting algorithms map human hand poses onto robot embodiments with vastly different geometries (e.g., finger link lengths and articulations). *Kinematic retargeting* approaches do so by solving geometric, task-space, or joint-space optimizations [55, 56, 57, 58], but operate solely on the robot configuration and do not account for forces between hand and object, often causing penetration, fingertip sliding, and grasp instability. To address this, *dynamics-aware retargeting* frameworks optimize physically-simulated hand-object trajectories, generally via reinforcement learning to track a reference-based reward [59, 60, 16, 61] or sampling-based optimization [62, 63, 15]. However, a key assumption shared by most prior retargeting approaches is availability of clean references with ground-truth hand-object poses from, e.g., MoCap. In contrast, we tackle the more difficult setting of noisier reconstructed references, with potential temporal discontinuities and severe hand-object misalignments.

3 Method

DO AS I DO consists of two parts, shown in Fig. 2. First, we reconstruct the 3D hand and object, and track them through time (Section 3.1). Then, we retarget the reconstructions onto the robot embodiment, producing dynamically-feasible trajectories that are effective in the real world (Section 3.2).

3.1 Reconstruction

Recognizing and reconstructing hand-object interactions from in-the-wild videos can be decomposed into two components: (1) tracking the human hand, and (2) determining object shape and

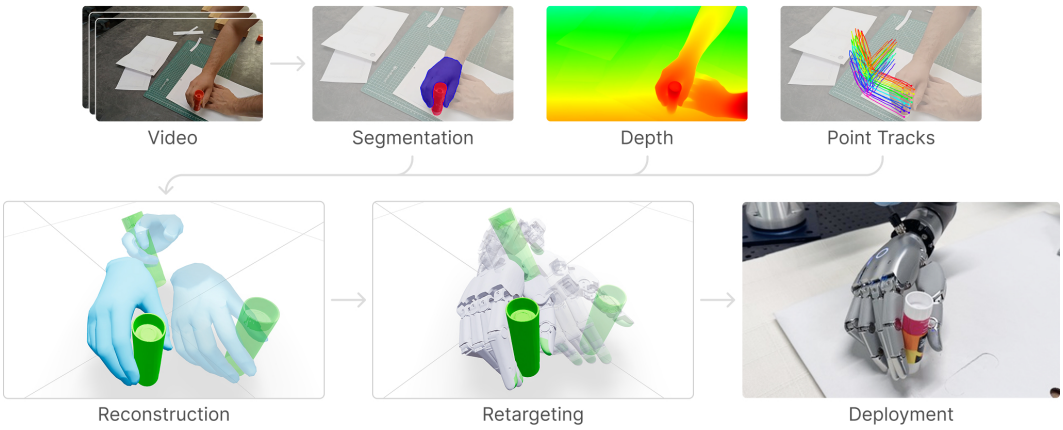

Figure 2: **Method Overview.** Our method leverages vision foundation models to reconstruct the hand and object, and retargets them onto the robot via sampling-based optimization in simulation.

tracking its pose. Critically, these capabilities need to be robust to diverse visual conditions found in noisy internet videos. We find that existing models such as HawoR [45] satisfy this criterion and can directly be used for our hand tracking with reasonable performance. For open-ended rigid objects with hand occlusions, however, prior works [17, 47] tend to lose pose lock, drift, or fail to re-acquire the object once visual evidence degrades as shown in 5. Therefore, we develop our own tracking method based on 3D generative foundation models trained with occlusions, namely SAM 3D [11]. Once tracked, the 3D hand, object, and camera are composed into a consistent near metric-space. We run three processing steps: (a) hand and object segmentation using SAM 3 [64], (b) depth and camera intrinsics estimation with MoGe [10], and (c) object 3D mesh generation using SAM 3D [11].

Object Tracking via Guided Diffusion. We repurpose SAM 3D [11], an image-to-3D generative model robust to occlusions and low resolution, into a video object tracker. It learns the joint distribution over shape and pose $p_\theta(x^s, x^p \mid c)$ given a single 2D image and object mask. As a result, it produces a *different* mesh each frame and a pose sequence with no temporal coherence if applied independently on frames. Our key observation is that shape and pose share the same latent space in the learned joint distribution; we can therefore fix the shape and obtain an updated pose entirely at inference time for each frame. Specifically, we fix the shape $\bar{x}^s$ at an *anchor* frame and, given the pose x^p_{k-1} from the previous frame, predict the pose x^p_k at frame k. Tracking thus reduces to drawing from $p_\theta(x^p_k \mid x^s_k = \bar{x}^s, c_k)$ biased toward x^p_{k-1}. Marginalizing over all possible poses is infeasible in the 6-DoF continuous pose space with the large generative backbone of SAM 3D. We instead exploit the flow matching inference itself: a sample is produced by integrating the ODE $\dot{x} = v_\theta(x_t, t, c)$ from $x_0 \sim \mathcal{N}(0, \mathbf{I})$ along the linear path $x_t = (1-t)\,x_0 + t\,x_1$. In a flow model, the forward-noised target is simply its *interpolant* along the model's own probability path [65, 66]. At each Euler step, we take the model's free Euler update of each block and blend it toward target interpolants, nudging towards canonical shape $\bar{x}^s$ for the shape block, and previous-frame pose x^p_{k-1} for the pose block:

$$x^s_t = \underbrace{(1-\alpha_s)(x^s_{t-\Delta} + \Delta v^s_\theta)}_{\text{denoising}} + \underbrace{\alpha_s\, z^s_{\text{ref}}(t)}_{\text{blending}}, \quad x^p_t = \underbrace{(1-\alpha_p)(x^p_{t-\Delta} + \Delta v^p_\theta)}_{\text{denoising}} + \underbrace{\alpha_p\, z^p_{\text{ref}}(t)}_{\text{blending}} \tag{1}$$

where $\alpha_s, \alpha_p \in [0,1]$ are guidance strength parameters; $z^s_{\text{ref}}(t) = (1-t)\,\epsilon^s + t\,\bar{x}^s$ and $z^p_{\text{ref}}(t) = (1-t)\,\epsilon^p + t\,x^p_{k-1}$ are target interpolants; ϵ^s, ϵ^p are the blocks' initial noise and v^s_θ, v^p_θ are the shape and pose components of the velocity $v_\theta(x_{t-\Delta}, t-\Delta, c)$.

Adaptive Guidance Parameters. As we focus on rigid objects, any fixed shape guidance $\alpha_s \in [0.9, 1]$ works well. Instead of defining a fixed pose guidance α_p, which may cause over-rigidity or spurious flips, we derive it from the data using rotational velocity of the object estimated from

Figure 3: **Verbs and Objects.** We visualize 20 distinct actions from our pipeline: placing, picking, scrubbing, spreading, squeezing, ironing, painting, dusting, digging, erasing, pouring, writing, whisking, stirring, poking, tamping, drilling, hammering, cutting, and basting.

2D point tracks [67]. This adds one offline tracking pass per video but noticeably improves pose tracking as shown in Appendix.

Sampling Per-frame Poses. As explained above, the guided pose sampling in Eq. 1 is stochastic. So, at each frame k, our algorithm draws N candidates $\{x^p_{k,i}\}_{i=1}^N$ that share the fixed shape $\bar{x}^s$ and needs to pick one of the samples per frame. The principled choice is to rank candidates by the model's own conditional log-density over poses given a shape. However, this process takes orders of magnitude more compute over generation itself and becomes prohibitive at video scale. We hence sample and cluster poses under a weighted $\mathrm{SE}(3)$ distance. Empirically, confident samples concentrate on the same pose mode while estimator noise scatters across $\mathrm{SE}(3)$, so consensus filtering and mask-IoU recovers the mode-best pose without ever re-invoking the diffusion backbone.

Hand-Object Alignment. After we independently reconustruct hand and object at possibly different scales, we need to align them. We treat the hand reconstruction scale as ground-truth and scale the object translation s to be aligned with the hand. We compute hand and object centroids: $\mathbf{c}^{\mathrm{M}}_{\mathrm{hand}}$, $\mathbf{c}^{\mathrm{M}}_{\mathrm{obj}}$ in object scale, and centroid of the *visible* portion of the hand mesh $\mathbf{c}^{\mathrm{H}}_{\mathrm{hand}}$ in hand mesh scale (near metric). Now, given the scaling ratio from the centroid z values, $k = z^{\mathrm{H}}_{\mathrm{hand}}/z^{\mathrm{M}}_{\mathrm{hand}}$, we optimize for the target object position $\mathbf{obj}_{\mathrm{target}} = \mathbf{c}^{\mathrm{H}}_{\mathrm{hand}} + k\left(\mathbf{c}^{\mathrm{M}}_{\mathrm{obj}} - \mathbf{c}^{\mathrm{M}}_{\mathrm{hand}}\right)$, where the per-frame translation scale is solved by least squares. Finally, we align the trajectory with gravity using GeoCalib [68].

3.2 Retargeting

Next, we aim to reproduce the reconstructed hand-object trajectory on a robot hand. However, this reference is incomplete: human and robot morphologies differ, and contact information and forces are absent from the kinematic signal. Prior works address this with kinematic solvers [22] or robotic heuristics [18, 7], but they do not ensure physical plausibility or lose general-purpose expressiveness.

DO AS I DO instead performs dynamics-aware retargeting, which follows the reference while ensuring realism within physics simulation. Building on the framework from Pan et al. [15], we perform an MPPI-style sampling-based optimization with a kernel annealed across both iterations and the prediction horizon, which shifts from broad exploration to local refinement. To enable retargeting for noisy reconstructed references, we further highlight several novel innovations, as shown in Fig. 4.

Warmup Steps. Two issues lie in the initial trajectory horizon H: (1) a noisy first frame may initialize the hand and object in a state that's impossible to recover from (e.g., if object is not grasped), and (2) annealed sampling does not fully explore these H steps since they appear only at the start

Figure 4: **Retargeting.** Our method succeeds in common failure modes (top) and excels at handling noisy references (bottom), despite, e.g., incorrect depth estimation causing poor alignment.

of the rollout horizon. Thus, we introduce additional H warmup steps prepended to the reference. During warmup, the object is held in place (e.g., in mid-air) while the robot hand is free to move; afterwards, the weld is dropped and simulation proceeds as normal. This allows the robot to adjust its pose before tracking the reference (e.g., to avoid dropping the object in Fig. 4), and naturally guides the optimizer in maximizing its tracking objective. Crucially, this warmup design does not assume any grasp sampling or heuristics, and simply utilizes the pre-existing core optimization procedure.

Random Force Perturbation. Second, the rollout horizon may trap optimization in local minima, with unstable object interactions that track briefly but cannot recover (e.g., balancing on fingertips in Fig. 4). To address this, we assert that interactions should be robust against minor disturbances: drawing inspiration from sim-to-real [69, 70], we introduce random forces to sample rollouts, thus encouraging controls robust to such perturbations. Importantly, this solution is general-purpose and does not assume high-fidelity references, unlike alternatives (e.g., contact guidance [15]).

Transition Reward. Third, object transitions between "rest" and "in-hand" mark critical inflection points in the trajectory, but with noisy references, tracking reward alone is too imprecise and soft to encourage the step-function interaction induced by these transitions (e.g., failed pickup in Fig. 4). Thus, we add a constant penalty term for failed transitions: (1) lack of object-floor contact during resting reference timesteps and (2) lack of hand-object contact during in-hand reference timesteps. We define reference timestep stages by measuring reference hand-object distance under threshold ϵ.

4 Experiments

4.1 Experimental Setup

We evaluate each step of our framework on standard benchmarks for their respective tasks. First, for hand-object reconstruction, we follow the setup from prior work [51, 53, 54]: we evaluate on DexYCB [71] and HOI4D [72] datasets with 160 and 12 annotated videos respectively, and isolate object-level performance by supplying ground-truth hands and measuring object reconstruction and tracking quality. We compare against two groups of baselines: (1) joint hand-object reconstructions including both image-based [48, 49, 50] and video-based [51, 54] approaches, and (2) object trackers [17, 47], where we replace our object tracking approach with these baselines keeping every other component fixed in our pipeline. To additionally assess performance on a distribution closer to everyday videos, we collect a benchmark of 150 videos drawn from in-the-wild internet videos, egocentric datasets, and generated videos. Since ground-truth object poses are unavailable for these videos, we evaluate via human preference, asking 3 volunteers per video to compare object poses from our SAM 3D-based tracking method against those from the current state of the art. Details of each of the baselines and human evaluation setup have been provided in the Appendix.

Finally, for retargeting, we evaluate on our in-the-wild reconstruction dataset of 655 reconstructed references, as well as OakInk2 [73], a large MoCap dataset with 1,352 clean bimanual human-object task trajectories. We compare against SPIDER [15], the state-of-the-art for dexterous retargeting and

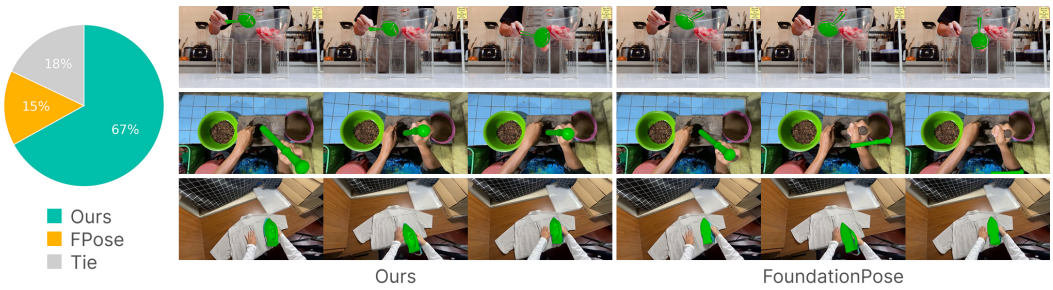

Figure 5: **Object Tracking Comparison.** We compare Ours and FoundationPose [17] for object tracking with head-to-head human evaluations on 150 videos (left), and visualize samples (right).

Table 2: **Reconstruction Results.** F-5, F-10, and Chamfer distance (CD) on two hand-object tracking datasets.

	DexYCB			HOI4D		
	F-5↑	F-10↑	CD↓	F-5↑	F-10↑	CD↓
HO [48]	0.24	0.48	4.76	0.28	0.51	3.86
IHOI [49]	–	–	–	0.42	0.70	2.7
HORSE [50]	0.23	0.42	6.97	0.26	0.45	6.69
MCC-HO [51]	0.36	0.60	3.74	0.52	0.78	1.36
G-HOP [54]	0.31	0.49	8.11	0.69	**0.91**	0.63
FoundationPose [17]	0.69	0.89	0.89	0.71	**0.91**	**0.49**
Any6D [47]	0.69	0.88	0.97	0.71	**0.91**	0.50
Ours	**0.71**	**0.93**	**0.66**	**0.72**	**0.91**	**0.49**

the only prior method designed for this scale; SPIDER serves as the Annealed Sampling baseline, and we progressively introduce our three components to assess their contributions. Following the recent literature [15, 59, 60], we evaluate successful trajectories as those with mean position error $E_{\text{pos}} < 0.1$ m and mean rotation error $E_{\text{rot}} < 0.5$ rad. Technical details are in the Appendix.

Across all tasks, we use the 22-DoF Sharpa Wave hand. Real-world deployment results shown here are on a bimanual setup with Sharpa Wave hands and UR3e arms, both commanded at 50 Hz.

4.2 Reconstruction Results

As shown in Fig. 5 (left), on the 150-video in-the-wild benchmark, human raters prefer our object tracking over the state-of-the-art FPose 67% of the time, with most videos receiving unanimous preferences. Qualitatively (Fig. 5), FPose loses the object under mild motion blur and occlusion, whereas our method recovers temporally consistent translations and rotations across the full clip. This advantage carries over to the standard hand-object reconstruction benchmarks in Table 2: we establish a new state-of-the-art on both DexYCB and HOI4D, outperforming all baselines.

In addition, we ablate design choices for our object tracking in the Appendix. We see that adaptive pose guidance via point tracking consistently improves reconstruction quality, and clustering-based selection performs on par with pose-likelihood selection while being up to 30× faster.

4.3 Retargeting Results

We report retargeting results in Table 3. On our reconstructed in-the-wild data, DO AS I DO reaches a 71% success rate, significantly improving over the baseline of 25%. The main differentiator is warmup, which discovers initial states that are much more stable and natural than the noisy initial frame, thereby leading to successful tracking in subsequent timesteps. Additionally, we find that perturbation noticeably improves the qualitative results (e.g., natural grasps) despite marginally af-

Table 3: **Retargeting Results.** Success rate and average position and orientation error on two human-object reference datasets.

	Reconstruction			OakInk2		
Method	Success ↑	Pos ↓	Rot ↓	Success ↑	Pos ↓	Rot ↓
Annealed Sampling	0.25	0.08	0.40	0.72	0.08	0.32
+ Warmup	0.66	0.06	**0.28**	0.77	0.06	0.25
+ Perturbation	0.67	0.06	0.30	0.79	**0.03**	**0.14**
+ Transition Reward	**0.71**	**0.05**	**0.28**	**0.81**	**0.03**	0.15

Figure 6: **Real-World Deployment.** We showcase trajectories for 10 tasks: whisking, pouring, dusting, squeezing, tamping, erasing, stirring, hammering, spreading, and picking.

fecting the quantitative metrics, and our transition reward encourages successful picks and places for trajectories that otherwise would've missed the object during crucial transition timesteps.

Further validating our method on OakInk2, we also see consistent improvement with the introduction of each component, moving from a baseline of 72% up to 81%. This shows that our retargeting, despite being designed for imperfect reconstructed references, produces effective gains even with clean MoCap trajectories, and scales well to the 1,000+ bimanual tasks in this benchmark.

4.4 Real-World Deployment

In total, our pipeline produced 500 high-quality, human-verified dexterous manipulation trajectories across internet (53%), egocentric (31%), and generated (16%) videos. To demonstrate quality, we execute a representative set of trajectories in the real world. We choose 10 motions with various object geometries and grasp classes [74], including writing tripod, power, ventral, and parallel extension grasps. After gravity alignment, the reconstructions still follow the videos' camera coordinates, so we manually align the initial pose (x, y, z, yaw) with the robot workspace in simulation before computing arm IK and deploying in the real world. Results are shown in Fig. 6, further film strips are presented in the Appendix, and videos are presented on our webpage.

4.5 Human Data Filtering Playbook

Given recent interest in human data for scaling up robotics [30, 33, 75], this section aims to highlight common quality issues in online human data sources. We noticed these patterns while analyzing common datasets, and present an analysis performed on 100DOH [76]. However, these lessons should be applicable more generally. We start with 2,000 10-second clips sampled from 100DOH, which has already been filtered for hand-object interaction, and find that only 187 clips (9%) have meaningful hand-object interaction present. Out of these 187 candidate clips, 41 clips have the hand or object outside the video boundary, and 29 clips have no activity or activity that spans across shot boundaries. Further 14 clips fail due to camera motion, and another 10 clips fail because of

SAM 3D, both of which may be fixed in the future with better models. We lose another 10 clips for other reasons. Out of the 2,000 videos sampled from 100DOH, only 83 (4%) survive our quality check for the reconstruction pass. Even in the best case, we foresee 107 clips, or roughly 5% of the data, being directly relevant for learning dexterous manipulation, implying a 20× penalty in not properly preprocessing and filtering internet videos for robot learning.

5 Conclusion

We introduced DO AS I DO, a framework for reconstructing and retargeting everyday human videos onto dexterous robot hands. Our method is effective across egocentric, exocentric, and online video sources, showing a path towards scaling robot data by simply observing humans. We hope DO AS I DO pushes us closer to making human videos first-class citizens in the robot learning data landscape.

Limitations. Our approach assumes rigid objects and semi-accurate metric depth predictions from monocular RGB, and may fail when either assumption doesn't hold. Monocular observations also suffer from ambiguity in the true hand-object distance, making it difficult to distinguish physical contact from mere visual occlusion. In addition, our method reconstructs only the hand and an object, rather than the full scene. As a result, it cannot reason about environmental constraints such as obstacles or articulations. Such scene-level reasoning remains important even with perfectly accurate references, since human intention is expressed through not only hand-object but also hand-scene interactions. Finally, the current physics simulators model the real world dynamics only approximately, which places an upper bound on the achievable real-world performance of our framework.

Acknowledgments

We thank Kyutai for providing us with the compute resources for this project. We are grateful to Chaoyi Pan for guidance and insightful discussions on retargeting. We also thank Jane Wu and Hongsuk Choi for helpful advice and discussions on hand-object reconstruction. This work was supported by ONR MURI N00014-21-1-280. Haritheja Etukuru and William Liang are supported by the NSF Graduate Research Fellowship Program under Grant DGE 2146752. Pieter Abbeel holds concurrent appointments as a Professor at UC Berkeley and as an Amazon Scholar at Amazon. This paper describes work performed at UC Berkeley and is not associated with Amazon.

References

[1] A. N. Meltzoff and M. K. Moore. Imitation of Facial and Manual Gestures by Human Neonates. *Science*, 198(4312):75–78, Oct. 1977. doi:10.1126/science.198.4312.75. URL https://www.science.org/doi/10.1126/science.198.4312.75.

[2] A. N. Meltzoff. Infant imitation after a 1-week delay: Long-term memory for novel acts and multiple stimuli. *Developmental Psychology*, 24(4):470–476, 1988. ISSN 1939-0599, 0012-1649. doi:10.1037/0012-1649.24.4.470. URL https://doi.apa.org/doi/10.1037/0012-1649.24.4.470.

[3] D. M. Bernard Meltzer. *Machine Intelligence 7*. 1972. URL http://archive.org/details/mi7_20200519.

[4] S. B. Kang and K. Ikeuchi. Toward automatic robot instruction from perception-mapping human grasps to manipulator grasps. *IEEE Transactions on Robotics and Automation*, 13(1): 81–95, 1997. doi:10.1109/70.554349.

[5] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, pages 726–733, Nice, France, 2003.

[6] 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. In *European Conference on Computer Vision*, pages 570–587. Springer, 2022.

[7] J. Mu, S. Yang, Y. Bao, H. Bae, T. Wei, L. Xu, B. Li, H. Xu, and J. Pang. Deximit: Learning bimanual dexterous manipulation from monocular human videos. *arXiv preprint arXiv:2602.10105*, 2026.

[8] I. Guzey, H. Qi, J. Urain, C. Wang, J. Yin, K. Bodduluri, M. Lambeta, L. Pinto, A. Rai, J. Malik, et al. Dexterity from smart lenses: Multi-fingered robot manipulation with in-the-wild human demonstrations. *arXiv preprint arXiv:2511.16661*, 2025.

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

[10] R. Wang, S. Xu, Y. Dong, Y. Deng, J. Xiang, Z. Lv, G. Sun, X. Tong, and J. Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details, 2025. URL https://arxiv.org/abs/2507.02546.

[11] S. D. Team, X. Chen, F.-J. Chu, P. Gleize, K. J. Liang, A. Sax, H. Tang, W. Wang, M. Guo, T. Hardin, X. Li, A. Lin, J. Liu, Z. Ma, A. Sagar, B. Song, X. Wang, J. Yang, B. Zhang, P. Dollár, G. Gkioxari, M. Feiszli, and J. Malik. Sam 3d: 3dfy anything in images, 2025. URL https://arxiv.org/abs/2511.16624.

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

[13] Google DeepMind and NVIDIA. Mujoco warp (MJWarp). https://mujoco.readthedocs.io/en/latest/mjwarp/, 2025. GPU-accelerated implementation of the MuJoCo physics engine built on NVIDIA Warp.

[14] NVIDIA. NVIDIA Isaac Sim: Robotics simulation and synthetic data generation. https://developer.nvidia.com/isaac/sim, 2025. GPU-accelerated robotics simulator built on NVIDIA Omniverse.

[15] C. Pan, C. Wang, H. Qi, Z. Liu, H. Bharadhwaj, A. Sharma, T. Wu, G. Shi, J. Malik, and F. Hogan. Spider: Scalable physics-informed dexterous retargeting, 2026. URL https://arxiv.org/abs/2511.09484.

[16] T. G. W. Lum, O. Y. Lee, C. K. Liu, and J. Bohg. Crossing the human-robot embodiment gap with sim-to-real rl using one human demonstration, 2025. URL https://arxiv.org/abs/2504.12609.

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

[18] H. Chen, T. Dong, T. Wu, L. Wang, Y. Jangir, Y. Niu, Y. Ye, H. Bharadhwaj, Z. Erickson, and J. Ichnowski. Dexterous manipulation policies from rgb human videos via 3d hand-object trajectory reconstruction. *arXiv preprint arXiv:2602.09013*, 2026.

[19] Meshy AI. Meshy ai: The #1 ai 3d model generator for creators. https://www.meshy.ai/, 2025. Accessed: 2025-04-17.

[20] Z. Wei, Z. Xu, J. Guo, Y. Hou, C. Gao, Z. Cai, J. Luo, and L. Shao. $\mathcal{D}(\mathcal{R}, \mathcal{O})$ grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping, 2025. URL https://arxiv.org/abs/2410.01702.

[21] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[22] J. Hsieh, K.-H. Tu, K.-H. Hung, and T.-W. Ke. Dexman: Learning bimanual dexterous manipulation from human and generated videos. *arXiv preprint arXiv:2510.08475*, 2025.

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

[24] Y. Xiao, J. Wang, N. Xue, N. Karaev, Y. Makarov, B. Kang, X. Zhu, H. Bao, Y. Shen, and X. Zhou. Spatialtrackerv2: 3d point tracking made easy, 2025. URL https://arxiv.org/abs/2507.12462.

[25] W. Yan and J. Chu. Foundationpose-plus-plus: Real-time 6d pose tracker in high-dynamic scenes. GitHub repository, 2025. URL https://github.com/teal024/FoundationPose-plus-plus.

[26] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training, 2023. URL https://arxiv.org/abs/2210.00030.

[27] Y. J. Ma, W. Liang, V. Som, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman. Liv: Language-image representations and rewards for robotic control, 2023. URL https://arxiv.org/abs/2306.00958.

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

[29] K. Shaw, S. Bahl, and D. Pathak. Videodex: Learning dexterity from internet videos, 2022. URL https://arxiv.org/abs/2212.04498.

[30] R. Zheng, D. Niu, Y. Xie, J. Wang, M. Xu, Y. Jiang, F. Castañeda, F. Hu, Y. L. Tan, L. Fu, T. Darrell, F. Huang, Y. Zhu, D. Xu, and L. Fan. Egoscale: Scaling dexterous manipulation with diverse egocentric human data, 2026. URL https://arxiv.org/abs/2602.16710.

[31] R. Yang, Q. Yu, Y. Wu, R. Yan, B. Li, A.-C. Cheng, X. Zou, Y. Fang, X. Cheng, R.-Z. Qiu, H. Yin, S. Liu, S. Han, Y. Lu, and X. Wang. Egovla: Learning vision-language-action models from egocentric human videos, 2025. URL https://arxiv.org/abs/2507.12440.

[32] H. Luo, Y. Feng, W. Zhang, S. Zheng, Y. Wang, H. Yuan, J. Liu, C. Xu, Q. Jin, and Z. Lu. Being-h0: Vision-language-action pretraining from large-scale human videos, 2025. URL https://arxiv.org/abs/2507.15597.

[33] R. Punamiya, S. Kareer, Z. Liu, J. Citron, R.-Z. Qiu, X. Cai, A. Gavryushin, J. Chen, D. Liconti, L. Y. Zhu, P. Aphiwetsa, B. Li, A. Cheluva, P. Kuppili, Y. Liu, D. Patel, A. Gao, H.-Y. Chung, R. Co, R. Zbizika, J. Liu, X. Xu, H. Xiong, G. Chen, S. Oliani, C. Yang, X. Wang, J. Fort, R. Newcombe, J. Gao, J. Chong, G. Matsuda, A. Doriwala, M. Pollefeys, R. Katzschmann, X. Wang, S. Song, J. Hoffman, and D. Xu. Egoverse: An egocentric human dataset for robot learning from around the world, 2026. URL https://arxiv.org/abs/2604.07607.

[34] R. G. Goswami, A. Bar, D. Fan, T.-Y. Yang, G. Zhou, P. Krishnamurthy, M. Rabbat, F. Khorrami, and Y. LeCun. World models for learning dexterous hand-object interactions from human videos, 2026. URL https://arxiv.org/abs/2512.13644.

[35] S. Gao, W. Liang, K. Zheng, A. Malik, S. Ye, S. Yu, W.-C. Tseng, Y. Dong, K. Mo, C.-H. Lin, Q. Ma, S. Nah, L. Magne, J. Xiang, Y. Xie, R. Zheng, D. Niu, Y. L. Tan, K. R. Zentner, G. Kurian, S. Indupuru, P. Jannaty, J. Gu, J. Zhang, J. Malik, P. Abbeel, M.-Y. Liu, Y. Zhu, J. Jang, and L. J. Fan. Dreamdojo: A generalist robot world model from large-scale human videos, 2026. URL https://arxiv.org/abs/2602.06949.

[36] J. Shi, Z. Zhao, T. Wang, I. Pedroza, A. Luo, J. Wang, J. Ma, and D. Jayaraman. Zeromimic: Distilling robotic manipulation skills from web videos, 2025. URL https://arxiv.org/abs/2503.23877.

[37] A. Agarwal, S. Uppal, K. Shaw, and D. Pathak. Dexterous functional grasping, 2023. URL https://arxiv.org/abs/2312.02975.

[38] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation, 2024. URL https://arxiv.org/abs/2405.01527.

[39] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play, 2023. URL https://arxiv.org/abs/2302.12422.

[40] H. G. Singh, A. Loquercio, C. Sferrazza, J. Wu, H. Qi, P. Abbeel, and J. Malik. Hand-object interaction pretraining from videos, 2024. URL https://arxiv.org/abs/2409.08273.

[41] Y. Qin, H. Su, and X. Wang. From one hand to multiple hands: Imitation learning for dexterous manipulation from single-camera teleoperation, 2023. URL https://arxiv.org/abs/2204.12490.

[42] J. Li, Y. Zhu, Y. Xie, Z. Jiang, M. Seo, G. Pavlakos, and Y. Zhu. Okami: Teaching humanoid robots manipulation skills through single video imitation, 2024. URL https://arxiv.org/abs/2410.11792.

[43] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017.

[44] R. A. Potamias, J. Zhang, J. Deng, and S. Zafeiriou. Wilor: End-to-end 3d hand localization and reconstruction in-the-wild, 2024.

[45] J. Zhang, J. Deng, C. Ma, and R. A. Potamias. Hawor: World-space hand motion reconstruction from egocentric videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1805–1815, 2025.

[46] M. Liu, C. Xu, H. Jin, L. Chen, M. Varma T, Z. Xu, and H. Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36, 2024.

[47] T. Lee, B. Wen, M. Kang, G. Kang, I. S. Kweon, and K.-J. Yoon. Any6D: Model-free 6d pose estimation of novel objects. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, 2025.

[48] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11807–11816, 2019.

[49] Y. Ye, A. Gupta, and S. Tulsiani. What's in your hands? 3d reconstruction of generic objects in hands. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3895–3905, 2022.

[50] A. Prakash, M. Chang, M. Jin, R. Tu, and S. Gupta. 3d reconstruction of objects in hands without real world 3d supervision. In *European Conference on Computer Vision*, pages 126–145. Springer, 2024.

[51] J. Wu, G. Pavlakos, G. Gkioxari, and J. Malik. Reconstructing hand-held objects in 3d. *arXiv preprint arXiv:2404.06507*, 2024.

[52] Y. Ye, J. Li, R. Rong, and C. K. Liu. Whole: World-grounded hand-object lifted from egocentric videos. *CVPR Findings*, 2026.

[53] Y. Ye, P. Hebbar, A. Gupta, and S. Tulsiani. Diffusion-guided reconstruction of everyday hand-object interaction clips. In *ICCV*, 2023.

[54] Y. Ye, A. Gupta, K. Kitani, and S. Tulsiani. G-hop: Generative hand-object prior for interaction reconstruction and grasp synthesis. In *CVPR*, 2024.

[55] K. Zakka. Mink: Python inverse kinematics based on MuJoCo, Feb. 2026. URL https://github.com/kevinzakka/mink.

[56] C. M. Kim, B. Yi, H. Choi, Y. Ma, K. Goldberg, and A. Kanazawa. Pyroki: A modular toolkit for robot kinematic optimization, 2025. URL https://arxiv.org/abs/2505.03728.

[57] Y. Qin, W. Yang, B. Huang, K. V. Wyk, H. Su, X. Wang, Y.-W. Chao, and D. Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system, 2024. URL https://arxiv.org/abs/2307.04577.

[58] Z.-H. Yin, C. Wang, L. Pineda, K. Bodduluri, T. Wu, P. Abbeel, and M. Mukadam. Geometric retargeting: A principled, ultrafast neural hand retargeting algorithm, 2025. URL https://arxiv.org/abs/2503.07541.

[59] K. Li, P. Li, T. Liu, Y. Li, and S. Huang. Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning, 2025. URL https://arxiv.org/abs/2503.21860.

[60] Z. Mandi, Y. Hou, D. Fox, Y. Narang, A. Mandlekar, and S. Song. Dexmachina: Functional retargeting for bimanual dexterous manipulation, 2025. URL https://arxiv.org/abs/2505.24853.

[61] S. Xu, Y.-W. Chao, L. Bian, A. Mousavian, Y.-X. Wang, L.-Y. Gui, and W. Yang. Dexplore: Scalable neural control for dexterous manipulation from reference-scoped exploration, 2025. URL https://arxiv.org/abs/2509.09671.

[62] L. Yang, H. J. T. Suh, T. Zhao, B. P. Graesdal, T. Kelestemur, J. Wang, T. Pang, and R. Tedrake. Physics-driven data generation for contact-rich manipulation via trajectory optimization, 2026. URL https://arxiv.org/abs/2502.20382.

[63] Z. Si, J. E. Chen, M. E. Karagozler, A. Bronars, J. Hutchinson, T. Lampe, N. Gileadi, T. Howell, S. Saliceti, L. Barczyk, I. O. Correa, T. Erez, M. Shridhar, M. F. Martins, K. Bousmalis, N. Heess, F. Nori, and M. Bauza. Exostart: Efficient learning for dexterous manipulation with sensorized exoskeleton demonstrations, 2025. URL https://arxiv.org/abs/2506.11775.

[64] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, J. Lei, T. Ma, B. Guo, A. Kalla, M. Marks, J. Greer, M. Wang, P. Sun, R. Rädle, T. Afouras, E. Mavroudi, K. Xu, T.-H. Wu, Y. Zhou, L. Momeni, R. Hazra, S. Ding, S. Vaze, F. Porcher, F. Li, S. Li, A. Kamath, H. K. Cheng, P. Dollár, N. Ravi, K. Saenko, P. Zhang, and C. Feichtenhofer. Sam 3: Segment anything with concepts, 2026. URL https://arxiv.org/abs/2511.16719.

[65] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. V. Gool. Repaint: Inpainting using denoising diffusion probabilistic models, 2022. URL https://arxiv.org/abs/2201.09865.

[66] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

[67] C. Doersch, P. Luc, Y. Yang, D. Gokay, S. Koppula, A. Gupta, J. Heyward, I. Rocco, R. Goroshin, J. Carreira, and A. Zisserman. Bootstap: Bootstrapped training for tracking-any-point, 2024. URL https://arxiv.org/abs/2402.00847.

[68] A. Veicht, P.-E. Sarlin, P. Lindenberger, and M. Pollefeys. GeoCalib: Single-image Calibration with Geometric Optimization. In *ECCV*, 2024.

[69] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang. Solving rubik's cube with a robot hand, 2019. URL https://arxiv.org/abs/1910.07113.

[70] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning, 2022. URL https://arxiv.org/abs/2109.11978.

[71] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. Van Wyk, U. Iqbal, S. Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9044–9053, 2021.

[72] Y. Liu, Y. Liu, C. Jiang, K. Lyu, W. Wan, H. Shen, B. Liang, Z. Fu, H. Wang, and L. Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21013–21022, 2022.

[73] X. Zhan, L. Yang, Y. Zhao, K. Mao, H. Xu, Z. Lin, K. Li, and C. Lu. Oakink2: A dataset of bimanual hands-object manipulation in complex task completion, 2024. URL https://arxiv.org/abs/2403.19417.

[74] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic. The grasp taxonomy of human grasp types. *IEEE Transactions on human-machine systems*, 46(1):66–77, 2015.

[75] R. Hoque, P. Huang, D. J. Yoon, M. Sivapurapu, and J. Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 2025.

[76] D. Shan, J. Geng, M. Shu, and D. F. Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9869–9878, 2020.

[77] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

[78] X. Wei, M. Liu, Z. Ling, and H. Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics*, 41(4):1–18, 2022. ISSN 1557-7368. doi:10.1145/3528223.3530103. URL http://dx.doi.org/10.1145/3528223.3530103.

[79] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi:10.1109/IROS.2012.6386109.

A Reconstruction

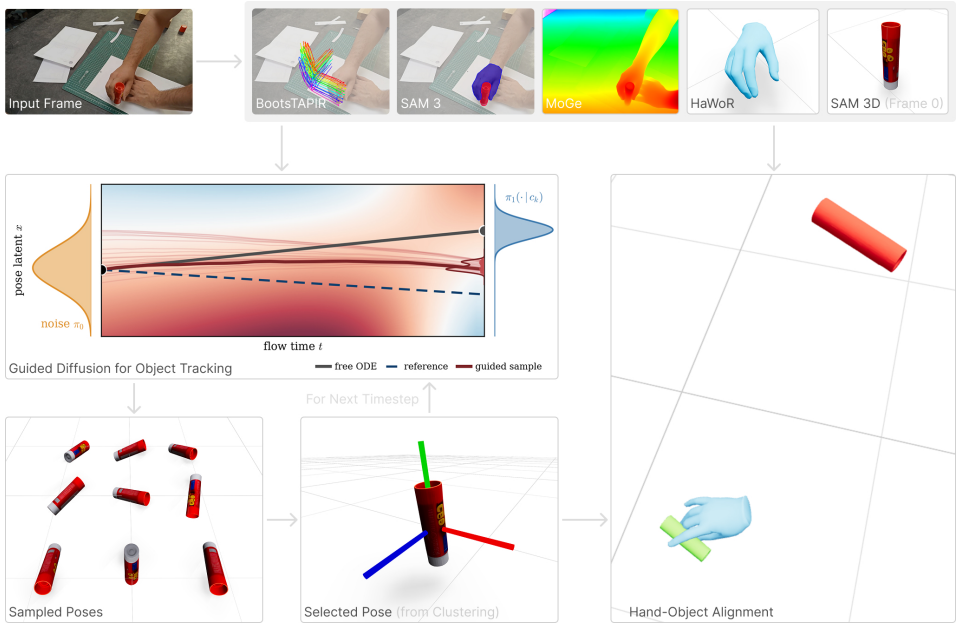

Figure 7: **Reconstruction Architecture.** SAM 3D [11] generates the object mesh from a single frame, while HaWoR [45] tracks the hand across the video. We then track the object frame-by-frame via guided diffusion (Section 3.1), anchoring each step to the predicted object shape and the previous frame's pose. Per frame, we sample N candidate poses and select the best using a clustering-based heuristic. Finally, a depth-map-based alignment step registers the HaWoR hand to our predicted object, yielding a consistent 4D hand-object trajectory.

Adaptive Guidance. Here, we explain how we get the pose guidance strength parameter α_p using point tracks. For each pair of consecutive frames, we sample 20 points inside the object mask and track them with BootsTAPIR [67]. We retain the points that remain inside the next mask and fit a 2D rigid transform to them in closed form by SVD. At every frame k, this yields the object's estimated in-plane rotation $\Delta\theta_k$ together with its centroid translation. We then set the pose guidance strength α_p as a clamped affine function of the rotation magnitude. In general, we use the following function to estimate $\alpha_p(k)$: $\alpha_p(k) = \max(0.1,\ 0.7 - 0.09\,|\Delta\theta_k|)$. For standard benchmarks we evaluate on DexYCB and HOI4D [71, 72], and show that adaptive pose guidance consistently improves over fixed pose guidance experiments in Table 5.

Sampling Per-frame Poses. At each frame k, our algorithm draws N candidates $\{x^p_{k,i}\}_{i=1}^N$ that share the fixed shape $\bar{x}^s$ and needs to pick one of the samples per frame. The principled choice is to rank candidates by the model's own conditional log-density over poses given a shape. For a flow model this density is exactly computable through the instantaneous change-of-variables formula [77], restricted to the pose block:

$$\log p_\theta\left(x^p_{k,i} \,\middle|\, \bar{x}^s,\, c_k\right) \;=\; \log p_0(x^p_0) \;+\; \int_0^1 \operatorname{tr}\left(\frac{\partial v^p_\theta(x_t, t, c_k)}{\partial x^p_t}\right) dt, \tag{2}$$

where x^p_0 is the noise pre-image of $x^p_{k,i}$ obtained by integrating the ODE backward from $t{=}1$ to $t{=}0$. Evaluating the trace exactly requires one vector-Jacobian product per pose coordinate at every Euler step; with $D_p{=}13$ pose dimensions, $T{=}25$ ODE steps, and $N{=}25$ candidates per frame, scoring costs $N\,T\,(1{+}D_p) \approx 8.7$k forward+backward passes through the diffusion backbone *per frame* — roughly two orders of magnitude above generation itself and prohibitive at video scale.

As a result, we go with a clustering based heuristic which is almost real-time once the candidates have been generated. We sample and cluster $N = 25$ poses under a weighted $\mathrm{SE}(3)$ distance described below:

$$d(x_i^p, x_j^p) = w_t \|t_i - t_j\|_2 + w_r \cdot 2 \arccos|\langle q_i, q_j \rangle|, \tag{3}$$

where $(t_i, q_i) := x_i^p$ are the translation and unit-quaternion components of x_i^p and the second term is the geodesic angle on $\mathrm{SO}(3)$. We discard clusters below a minimum size as outliers and rank the remaining clusters' 2D silhouette IoU against the input mask. Empirically, confident samples concentrate on the same pose mode while estimator noise scatters across $\mathrm{SE}(3)$. We show that this heuristic performs on part with pose likelihood based ranking in Table 5 while adding essentially zero computational cost over generation.

Hand-Object Alignment. Since SAM 3D has been trained using pointmaps obtained from monocular geometry estimation model MoGe [10], we use MoGe pointmaps during inference as well for best performance. To align the pose predictions with the HaWoR hand predictions, as seen in Figure 8, we first compute the centroid of the *visible* portion of the HaWoR hand mesh $\mathbf{c}^{\mathrm{H}}_{\mathrm{hand}}$ by casting a ray for each pixel in the segmentation mask, and then averaging over the first hits onto the hand mesh. Then, in the per-frame MoGe pointmap, the hand centroid $\mathbf{c}^{\mathrm{M}}_{\mathrm{hand}}$ is the mean over the 3D values of the same pixels used to recover $\mathbf{c}^{\mathrm{H}}_{\mathrm{hand}}$ (the rays that strike the HaWoR mesh), and the object centroid $\mathbf{c}^{\mathrm{M}}_{\mathrm{obj}}$ is the mean of the 3D values over all segmentation mask pixels. Letting $z^{\mathrm{H}}_{\mathrm{hand}}$ and $z^{\mathrm{M}}_{\mathrm{hand}}$ denote the depth (z) components of the hand centroids, the per-frame scale is

$$k = \frac{z^{\mathrm{H}}_{\mathrm{hand}}}{z^{\mathrm{M}}_{\mathrm{hand}}}.$$

For each frame we place the object relative to the hand: we take the hand-to-object offset in pointmap space, rescale it to the near-metric (HaWoR) units by k, and add it to this HaWoR hand centroid,

$$\mathbf{obj}_{\mathrm{target}} = \mathbf{c}^{\mathrm{H}}_{\mathrm{hand}} + k \left(\mathbf{c}^{\mathrm{M}}_{\mathrm{obj}} - \mathbf{c}^{\mathrm{M}}_{\mathrm{hand}}\right).$$

Given this target object position, we hold the SAM 3D predicted mesh orientation fixed and optimize a single scalar, the translation scale s, on the mesh's camera-frame translation $\mathbf{t}$. The object's visible-surface centroid is $\mathbf{obj}_{\mathrm{pos}}(s) = \mathbf{c}_{\mathrm{mesh}} + s\,\mathbf{t}$, where $\mathbf{c}_{\mathrm{mesh}}$ is the centroid of *visible* mesh vertices that project inside the object mask (with translation excluded). We solve the one-dimensional least-squares problem

$$s^\star = \arg\min_s \left\|\mathbf{obj}_{\mathrm{pos}}(s) - \mathbf{obj}_{\mathrm{target}}\right\|^2 = \frac{\mathbf{t}^\top\left(\mathbf{obj}_{\mathrm{target}} - \mathbf{c}_{\mathrm{mesh}}\right)}{\mathbf{t}^\top\mathbf{t}},$$

which slides the object along its viewing ray to the depth that best matches the target while preserving its recovered orientation. Repeating this for every frame yields an aligned metric 4D hand-object trajectory.

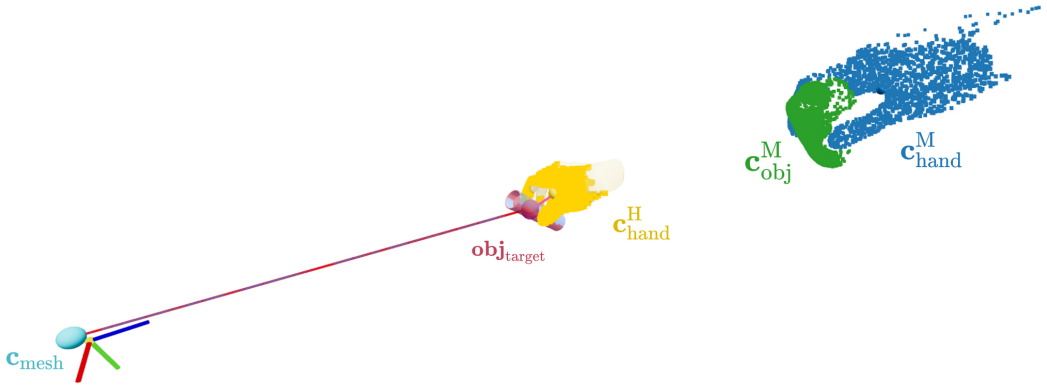

Figure 8: **Hand-Object Alignment.** The translation and scale of the object mesh are converted from MoGe pointmap space to HaWoR hand mesh space using relative distance between hand and object.

B Retargeting

Simulation Setup. We perform retargeting in the MuJoCo Warp simulator, with a simulation timestep duration of 0.005s (200 Hz). We convex decompose object meshes using CoACD [78], and to stabilize hand-object interactions especially for tasks that involve many contacts, we thicken and dilate the object meshes by 2mm.

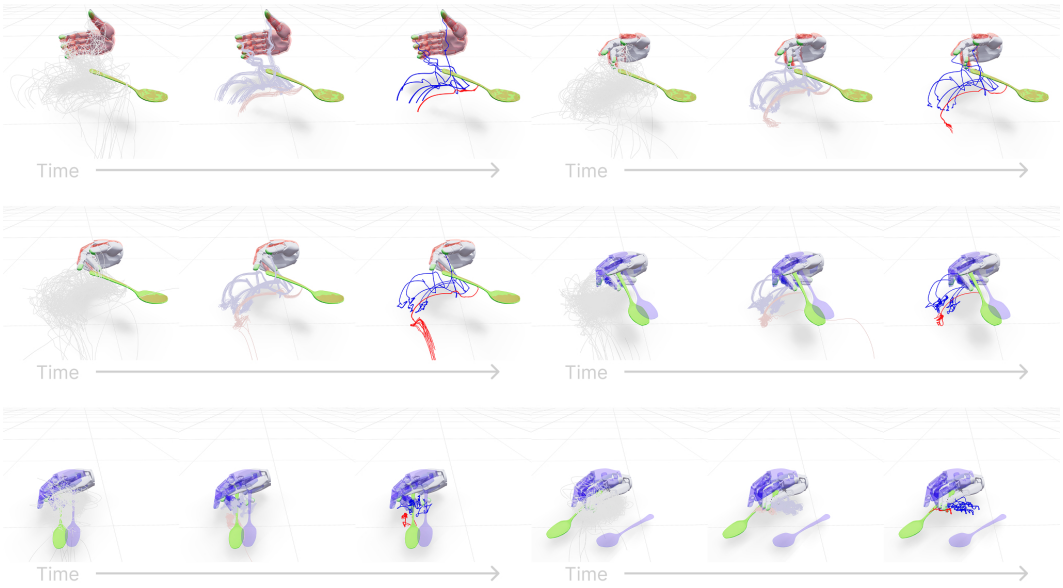

Figure 9: **Retargeting Optimization.** We visualize multiple iterations of the sampling-based optimization process for a trajectory: blue and red traces indicate converged fingertip and object trajectories, respectively. The ghost hand and object indicate reference (blue) and warmup (red).

Algorithm Details. To prepare for dynamics-aware retargeting, we first compute the reference trajectory by kinematically retargeting the human hand onto the robot hand (without considering forces, contacts, etc), using mink to match fingertip positions [55]. Next, we run our sampling-based dynamics-aware retargeting algorithm, visualized in Fig. 9, with planning every 0.5s (2 Hz) on a horizon of 3s. Each planning step, we evaluate 1024 samples and optimize over 32 iterations, and rollouts are rewarded for tracking the object (position and orientation) and hand (position, orientation, and finger joints), with penalties on excessive penetration (to avoid exploiting the simulator) and the transition reward introduced in the main text. Further hyperparameter details are in Table 4.

Finally, we list several important details and lessons besides those described in the main text.

1. **Reference Blending.** At each planning timestep, the initial samples are centered around the previous plan's controls, appended at the end with the next chunk of reference steps. Naively appending the reference to previously optimized controls results in sharp transitions and jerky motions, and to avoid this, we blend the optimized controls into the reference via interpolation.
2. **Robust Kinematic Retargeting.** Retargeting draws samples centered around the reference, and thus requires a reasonable reference in order to find dynamically-feasible controls. Since kinematic retargeting is computationally inexpensive, a simple yet effective way to improve reference quality is to compute multiple kinematic retargeting results starting from different random initial poses, thus avoiding potential local minima.
3. **Object Base.** In some videos, objects of interest may not be lying flat on a surface (e.g., spoon standing upright in a container). To faithfully stabilize their initial poses, we add a

Table 4: **Retargeting Hyperparameters.** All results from our method use the following setup.

Parameter	Value	Parameter	Value
Algorithm		*Sampling*	
num_samples	1024	knot_dt	0.2
max_num_iterations	32	pos_noise_scale	0.01
sim_dt	0.005	rot_noise_scale	0.01
ctrl_dt	0.5	joint_noise_scale	0.1
horizon	3.0	final_noise_scale	0.01
		first_ctrl_noise_scale	1.0
		last_ctrl_noise_scale	4.0
Rewards		*Perturbation*	
pos_rew_scale	1.0	num_perturb_samples	4
rot_rew_scale	0.3	perturb_force_scale	0.5
base_pos_rew_scale	0.1	perturb_torque_scale	0.5
base_rot_rew_scale	0.03	perturb_prob	0.05
joint_rew_scale	0.01	perturb_continue_prob	0.95
terminal_rew_scale	10.0		
penetration_penalty_scale	3000.0		
transition_penalty_scale	0.5		

flat base "plate" to the bottom of the object mesh, which only has contact with the floor (and not the robot). This allows us to retarget objects starting from any initial pose, without any task-specific assumptions or additional scene reconstructions.

C Experimental Setup and Results

Reconstruction Evaluation. We compare against two groups of baselines: (1) joint hand-object reconstructions, and (2) object trackers [17, 47]. In the first group, we compare against HO [48], IHOI [49], HORSE [50], and MCC-HO [51], whose numbers are taken from Wu et al. [51]. We also compare against a more recent video-based approach for RGB joint hand-object reconstruction G-HOP [54]. For HOI4D, we evaluate the authors' released per-clip checkpoints directly. These checkpoints were optimized with the HOI4D-only diffusion prior provided. For DexYCB, the original GHOP paper trains the diffusion prior on (among other datasets) the DexYCB shape and pose data, but does not report video-reconstruction results on DexYCB. We therefore run the per-clip test-time optimization ourselves on the full 160-clip test set using the authors' released mix-data prior. In the next group of baselines, we have state-of-the-art 6-DoF object trackers, FoundationPose [17] and Any6D [47]. Since the latter methods only perform object tracking, we slot each into our framework in place of our object-tracking module while holding every other component fixed, providing a controlled comparison against our tracker.

For our evaluation on in-the-wild internet videos, ground-truth object poses are unavailable, so we rely on human preference. Each rater is shown the original hand-object interaction video alongside two reprojections, where the posed object mesh from our tracker and from FoundationPose [17] is projected back into 2D, and asked which video tracks the object more consistently with its true motion. We randomize the left-right ordering of the two methods per video to avoid position bias, and collect three ratings, each from a distinct rater in a pool of five, for each of the 150 videos. Raters prefer our method 67% of the time, compared to 18% for FoundationPose and 15% rated as ties. This corresponds to a 79% win rate among non-tie judgments. 75% of videos received unanimous agreement across all three raters, and inter-rater agreement was substantial (Fleiss' $\kappa = 0.65$). Figure 10 shows a screenshot of the user interface presented to each rater.

Table 5: **Object Tracking Ablation.** We ablate two design axes of our object tracking method.

Pose Guidance Strategy	Candidate Selection	DexYCB			HOI4D		
		F-5↑	F-10↑	CD↓	F-5↑	F-10↑	CD↓
Fixed	Clustering	0.70	0.91	0.74	0.69	0.91	0.50
Adaptive	Random	0.70	0.91	0.74	0.62	0.87	0.66
Adaptive	Log-likelihood	0.72	0.93	0.65	0.72	0.91	0.49
Adaptive	Clustering	0.71	0.93	0.66	0.72	0.91	0.49

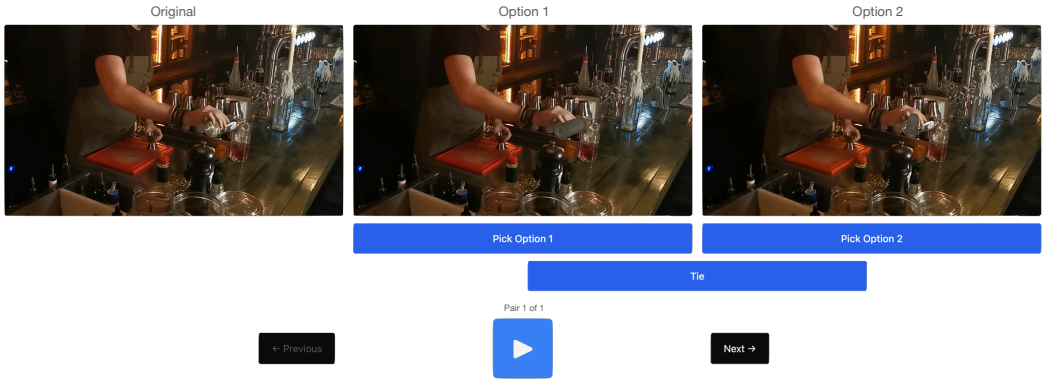

Figure 10: A screenshot of the user interface shown to the human evaluators for in-the-wild object tracking. In this instance shown in the figure, option 1 is regarded as better because the object pose is aligned with true object position.

D Robot Deployment

Simulation Setup for Dual UR3e arms + Sharpa hands. We replay the retargeted trajectories in simulation on a dual-arm setup with UR3e arms and Sharpa Wave hands, using the MuJoCo physics simulator [79] for dynamics and Viser for visualization. The trajectories from our retargeting stage (Section 3.2) cannot be executed on the robot directly; we first map them onto the arm-and-hand setup via inverse kinematics using mink [55]. As shown in Figure 11, we build a digital twin of our real-world setup, letting us visually validate each trajectory for self-collisions, table contacts, and similar issues in simulation before real-world execution.

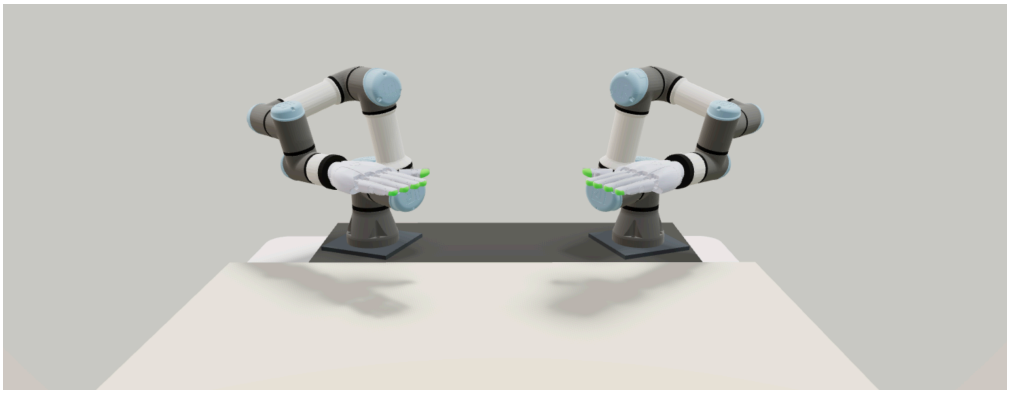

Figure 11: **Digital Twin.** A simulated replica of our real-world bimanual setup (UR3e arms with Sharpa Wave hands) in MuJoCo, visualized with Viser.

Real-World Rollouts. Once validated in simulation, we roll out the trajectories on the real-world dual-arm setup at roughly half speed, with both arms and hands commanded at 50 Hz. Figure 12 shows the real-world rollout strips for 6 different tasks:

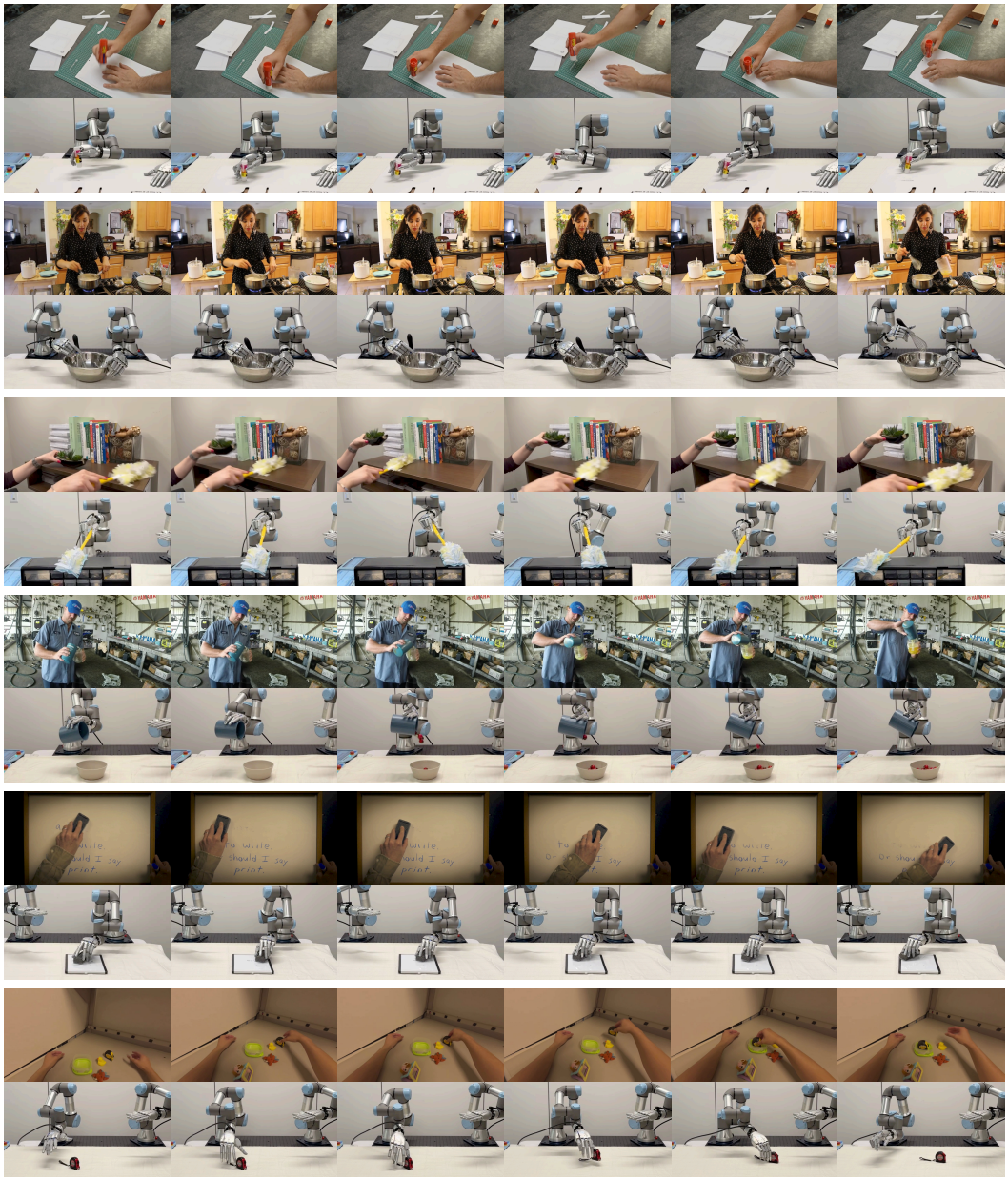

Figure 12: **Real-World Rollouts.** Frames from our robot rollouts for spreading, whisking, dusting, pouring, erasing, and picking. More tasks and videos are available on our webpage.