HANDMIMIC: LEARNING DEXTEROUS MANIPULATIONS FROM VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

While humans naturally learn dexterous manipulation by watching others and practicing, robotic teleoperation of dexterous hands remains prohibitively expensive and non-trivial. We introduce HandMimic, a framework for learning dexterous manipulation policies directly from real-world monocular videos. We first reconstruct and align the observed hand-object interaction in simulation, while treating the resulting motion as an imperfect reference rather than a physically valid robot demonstration. We then practice in simulation using reinforcement learning to recover physically feasible behaviors from the reconstructed reference. To reduce dependence on a single reconstructed object, we further train across digital siblings with varying geometry and scale. Quantitative experiments show that policies learned from reconstructed references approach those trained from ground-truth references, while digital-sibling training improves generalization to unseen geometry and scale variations. We further validate the resulting motions on a real dexterous robot. Together, these results support using reconstructed human interactions as references for physical practice across varying object geometries.

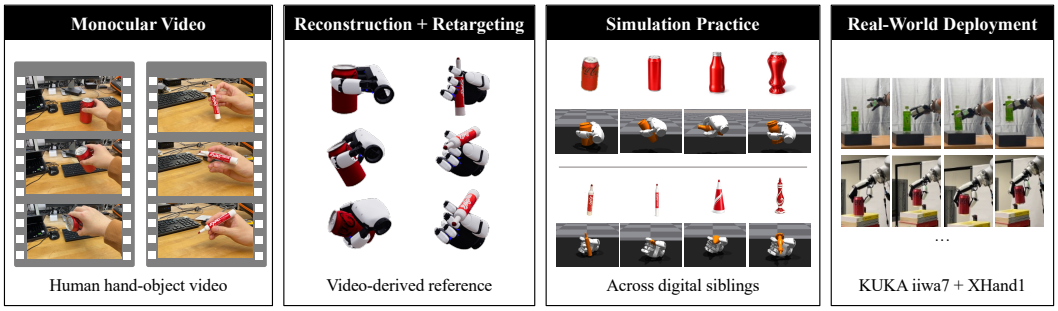


Figure 1: **HandMimic** reconstructs and retargets monocular human hand-object videos into video-derived references, practices them across digital siblings in simulation, and deploys the learned manipulation behaviors on a real dexterous hand.

1 INTRODUCTION

Learning dexterous robotic manipulation policies remains exceptionally challenging due to the scarcity of high-quality robot demonstration data. While teleoperation can provide valid robot trajectories, it is inherently complex, expensive, and difficult to scale. In contrast, large amounts of human manipulation experience are available in real-world videos. Humans naturally acquire dexterous skills by observing others and practicing; enabling robots to similarly learn from monocular videos and practice in simulation offers a scalable alternative for acquiring complex manipulation skills.

A direct way to use such videos is to reconstruct the observed hand-object interaction and transfer it to a dexterous hand. Recent advances in hand reconstruction, object reconstruction, and pose tracking make this increasingly practical. In this work, we develop a state-of-the-art monocular hand-object reconstruction pipeline that recovers the object geometry and motion together with the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

human hand trajectory in a common metric frame. Yet even state-of-the-art reconstruction still leaves errors that matter for dexterous manipulation, including inaccuracies in hand-object alignment, object pose, metric scale, occluded hand articulation, and unobserved object geometry. These errors may be small visually, but can substantially change the contacts required for manipulation. Moreover, even a perfectly reconstructed human interaction does not directly specify the motion of a dexterous hand: differences in hand morphology and kinematics make it difficult for retargeting to preserve the same contacts and object behavior. As a result, we treat the reconstructed and retargeted motion as an imperfect reference rather than a physically valid robot demonstration.

This motivates a different question: *instead of requiring the reconstructed and retargeted motion to be physically exact, can a dexterous hand learn from it through physical practice?* We introduce **HandMimic**, a framework for learning dexterous manipulation policies directly from real-world monocular videos. HandMimic first reconstructs the hand-object interaction and aligns the hand and object in a common metric frame. We then kinematically retarget the reconstructed hand motion to the robot hand without requiring the retargeted trajectory to fully resolve contact and physical feasibility. Instead, we train a closed-loop policy in simulation to reproduce the demonstrated manipulation, using the retargeted hand motion as a kinematic reference while placing greater emphasis on the demonstrated object behavior. In this way, physical feasibility is resolved during policy optimization in the simulator rather than being fully specified by reconstruction and retargeting alone.

A second challenge arises from the reconstructed object itself. A policy trained on a single reconstructed mesh can become specialized to errors in its geometry or scale, even when the reconstructed motion is otherwise useful. This is particularly problematic for dexterous manipulation, where small geometric differences can substantially alter the contacts required to achieve the same object motion. To reduce this dependence, we train across a family of *digital siblings* with varying object geometry and scale. Rather than practicing only with the nominal reconstructed object, the policy experiences multiple geometric variations of the same manipulation task and learns behavior that is less tied to any single reconstructed mesh.

We evaluate HandMimic on DexYCB and in-the-wild monocular videos. Under a matched DexYCB evaluation, our reconstruction pipeline achieves state-of-the-art performance on interaction-centric reconstruction metrics while remaining competitive in articulated hand-pose accuracy. Despite the remaining reconstruction and retargeting errors, policies trained from reconstructed references approach the performance of policies trained from ground-truth references.We further show that training across digital siblings improves generalization to unseen object geometry and scale variations compared with training on a single reconstructed object. Finally, we validate the resulting motions on a physical KUKA iiwa 7 arm equipped with an XHand1 dexterous hand.

In summary, our main contributions are threefold:

- We develop a state-of-the-art monocular hand-object reconstruction pipeline that recovers metric and aligned hand motion, object motion, and object geometry from real-world videos.
- We introduce HandMimic, which treats the reconstructed and retargeted interaction as an imperfect reference for physical practice and uses simulation-based reinforcement learning to recover physically feasible dexterous manipulation behavior.
- We introduce digital-sibling training, which practices the demonstrated manipulation across variations in object geometry and scale to reduce dependence on a single reconstructed object and improve generalization to unseen geometry.

2 Related Works

Learning Dexterous Manipulation from Human Demonstrations. Dexterous manipulation with multi-fingered hands is challenging due to high-dimensional control and complex contact dynamics OpenAI et al. (2019); Rajeswaran et al. (2018). Teleoperation systems Handa et al. (2020); Ding et al. (2024) provide high-quality demonstrations, but their scalability is limited by hardware cost and embodiment mismatch. Recent methods therefore learn from human hand data. Given accurate MoCap or ground-truth hand-object trajectories, ManipTrans Li et al. (2025) learns a hand trajectory imitator and refines it with a residual interaction policy; DexMachina Mandi et al. (2025) performs functional retargeting with curriculum-based reinforcement learning; DexTrack Liu et al.

(2025) trains a generalizable neural tracking controller; and OmniGrasp Luo et al. (2024) learns diverse humanoid grasping skills. These methods achieve strong results, but they typically rely on clean hand-object references, such as MoCap data or accurate object poses.

To reduce this dependency, recent work explores learning dexterous manipulation from monocular videos. VideoDex Shaw et al. (2023) uses videos as visual and action priors. QuasiSim Liu et al. (2024) optimizes through quasi-physical simulation, but requires expensive per-sequence optimization. DexMan Hsieh et al. (2025), VideoManip Chen et al. (2026a), and DexImit Mu et al. (2026) reconstruct hand-object interactions from monocular videos and use them for policy learning. However, monocular reconstruction inevitably introduces errors in hand pose, object geometry, and hand-object alignment. Our method explicitly targets this noisy video-to-policy setting: instead of treating reconstructed contacts as reliable physical constraints, we use kinematic retargeting to preserve the hand motion and rely on closed-loop policy learning with shape and scale augmentation to handle geometric inaccuracies.

Motion Retargeting from Human to Robot Hands. Human-to-robot hand retargeting is often formulated as keypoint-based inverse kinematics Qin et al. (2022); Handa et al. (2020); Sivakumar et al. (2022), which is efficient but does not guarantee physical feasibility. Recent methods incorporate physics or task objectives into retargeting. Pan et al. (2025); Paliwal et al. (2026) use physics-based sampling and contact guidance, while DexH2R Zhao et al. (2024) combines retargeting with a task-oriented residual policy. Physics-aware retargeting can improve feasibility when geometry and contact estimates are accurate, but in our setting the object mesh and hand-object alignment are reconstructed from monocular video and remain noisy. Enforcing penetration, collision, or signed-distance constraints from these estimates can distort the retargeted hand motion. We therefore use retargeting only to obtain a kinematic robot-hand reference, and leave physical correction to policy learning in simulation.

Simulation-Based Policy Learning and Sim-to-Real Transfer. Simulation has become a standard platform for training dexterous manipulation policies Makoviychuk et al. (2021); Todorov et al. (2012). OpenAI OpenAI et al. (2019) demonstrated sim-to-real in-hand manipulation with automatic domain randomization, and recent work extends simulation-based training to humanoid manipulation Lin et al. (2025) and tool use with procedurally generated objects Kedia et al. (2026). Residual policy learning has also been effective for manipulation transfer Jiang et al. (2024); Li et al. (2025); Zhao et al. (2024). Our policy similarly tracks a reference motion in simulation, but faces a different source of uncertainty: the reference object geometry comes from imperfect monocular reconstruction. A policy trained on a single reconstructed mesh can overfit to its scale and shape artifacts. To address this, we broaden the training distribution with digital siblings spanning object geometry and scale variations, encouraging policies that are less tied to a single reconstructed mesh.

3 METHOD

Overview Our method consists of four sequential stages: (1) reconstructing the human hand motion, object motion, and object geometry from a monocular RGB video; (2) aligning the reconstructed interaction in a common metric frame and retargeting the human hand motion to the dexterous hand; (3) constructing digital siblings with varying object geometry and scale; and (4) training a closed-loop policy in simulation to reproduce the demonstrated object behavior. Crucially, we use reconstruction and retargeting to obtain a kinematic interaction reference without requiring them to fully resolve physical feasibility, which is instead recovered through policy learning in simulation.

3.1 VIDEO-DERIVED HAND-OBJECT REFERENCE

Given video frames $\{I_t\}_{t=1}^{T}$, we reconstruct a video-derived interaction reference

$$\mathcal{D}^{\mathrm{vid}} = \left\{ \mathcal{H}_t^{\mathrm{vid}}, \mathbf{T}_t^{o,\mathrm{vid}}, \mathcal{M}^o \right\}_{t=1}^{T}, \tag{1}$$

where $\mathcal{H}_t^{\mathrm{vid}} = \mathrm{MANO}(\theta_t, \beta, \tau_t)$ is the reconstructed human hand trajectory, $\mathbf{T}_t^{o,\mathrm{vid}} \in SE(3)$ is the object trajectory, and $\mathcal{M}^o$ is the reconstructed object mesh. We segment the object using SAM2 Ravi et al. (2025), reconstruct $\mathcal{M}^o$ using SAM3D Chen et al. (2026b) via SpatialTrackerV2 Xiao et al. (2025) camera parameters, track $\mathbf{T}_t^{o,\mathrm{vid}}$ using FoundationPose Wen et al. (2024), and extract $\mathcal{H}_t^{\mathrm{vid}}$

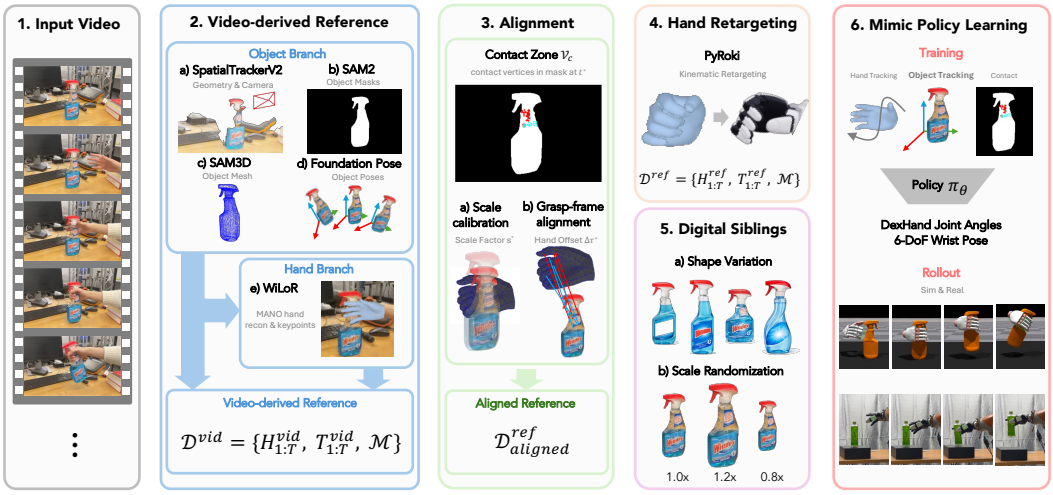


Figure 2: **System overview.** From a monocular video, HandMimic reconstructs object geometry and motion together with the human hand trajectory, aligns them in a common metric frame, and retargets the hand motion to the dexterous hand. We then construct digital siblings with shape and scale variations and optimize an object-centric manipulation policy through practice in simulation.

using WiLoR Potamias et al. (2025). Since the hand and object are reconstructed independently, we calibrate their metric scale and relative placement before retargeting.

Contact-based object scale calibration. WiLoR provides a metric hand reconstruction, whereas the reconstructed object mesh and trajectory inherit the scale of the monocular scene reconstruction. We therefore use the observed grasp contact to refine the object scale. Let $t^\star$ denote the grasp frame and let $\mathcal{V}_c$ be the predefined MANO contact-zone vertices whose projections lie inside the object mask. For a candidate scale s, we place the object surface at

$$\mathcal{S}^o_{t^\star}(s) = \{s\,(\mathbf{R}^o_{t^\star}\mathbf{p} + \mathbf{t}^o_{t^\star}) \mid \mathbf{p} \in \mathcal{M}^o\}\,, \tag{2}$$

and estimate

$$s^\star = \arg\min_s \frac{1}{|\mathcal{V}_c|} \sum_{\mathbf{h}_i \in \mathcal{V}_c} d(\mathbf{h}_i,\, \mathcal{S}^o_{t^\star}(s))\,, \tag{3}$$

where $d(\cdot,\cdot)$ measures the hand-to-object surface distance. The resulting scale is applied consistently to the object mesh and the translational component of its full trajectory.

Grasp-frame alignment. Even after scale calibration, residual translation errors can remain between the independently reconstructed hand and object. At the grasp frame $t^\star$, we establish correspondences $\mathcal{C} = \{(\mathbf{h}_i, \mathbf{o}_i)\}$ between MANO contact-zone vertices and points on the rendered object surface. We then optimize a single translation offset $\Delta\boldsymbol{\tau}$ for the hand:

$$\begin{aligned}\Delta\boldsymbol{\tau}^\star = \arg\min_{\Delta\boldsymbol{\tau}} \frac{1}{N}\sum_i \|\Pi(\mathbf{h}_i + \Delta\boldsymbol{\tau}) - \Pi(\mathbf{h}_i)\|_2^2 \\ + \lambda_c \frac{1}{|\mathcal{C}|} \sum_{(\mathbf{h}_i,\mathbf{o}_i)\in\mathcal{C}} \|\mathbf{h}_i + \Delta\boldsymbol{\tau} - \mathbf{o}_i\|_2^2\,,\end{aligned} \tag{4}$$

where $\Pi(\cdot)$ denotes perspective projection and $\lambda_c = 5$ balances image-space consistency and hand-object contact. The same translation offset is applied to the entire hand trajectory, while the reconstructed hand articulation and orientation are left unchanged.

Finally, we transform the reconstructed interaction to a table-aligned world frame using the estimated supporting plane, producing the video-derived reference used by the retargeting stage.

3.2 Dexterous Hand Retargeting

We retarget the aligned MANO hand trajectory to the target dexterous hand using PyRoki Kim et al. (2025). We denote the retargeted hand state at time t as $\mathcal{H}^{\text{ref}}_t = \{\mathbf{w}^{\text{ref}}_t, \mathbf{q}^{h,\text{ref}}_t\}$, consisting

of the wrist pose and hand joint configuration. Together with the calibrated object trajectory and reconstructed mesh, the resulting reference is

$$\mathcal{D}^{\text{ref}} = \left\{ \mathcal{H}^{\text{ref}}_{1:T}, \mathbf{T}^{o,\text{ref}}_{1:T}, \mathcal{M}^{o} \right\}. \tag{5}$$

PyRoki optimizes the dexterous-hand motion to match the reconstructed MANO hand geometry while preserving grasp-frame contact correspondences and temporal smoothness. We do not require the resulting reference to be a physically executable trajectory; instead, $\mathcal{D}^{\text{ref}}$ provides the kinematic and object-motion targets for policy learning.

3.3 Digital Siblings for Geometry Variation

The reconstructed mesh $\mathcal{M}^o$ provides a nominal geometry for the observed interaction, but a policy trained only on this mesh can become specialized to reconstruction-specific shape and scale. We therefore construct a family of digital siblings

$$\mathcal{G}^o = \{\mathcal{M}^o, \mathcal{M}^o_1, \ldots, \mathcal{M}^o_K\}, \tag{6}$$

which contains the reconstructed object together with related geometric variants. During policy training, we sample an object geometry from $\mathcal{G}^o$ and additionally randomize its scale by $s_o \sim \mathcal{U}(0.8, 1.2)$. The same interaction reference is therefore practiced across different object geometries rather than only the nominal reconstructed mesh.



Figure 3: **Generated digital siblings.** Shape variants used for policy training.

3.4 Mimic Policy Learning

Given the retargeted interaction reference $\mathcal{D}^{\text{ref}}$, we train a closed-loop policy in simulation to physically realize the demonstrated manipulation. Following ManipTrans Li et al. (2025), we formulate learning as a reference-tracking task, but adapt the objective to video-derived references: the retargeted hand motion provides a kinematic tracking target, while the demonstrated object trajectory is prioritized as the task-level objective. We additionally remove hand-tracking early termination after grasp, allowing policy optimization to resolve physical interaction without requiring exact reproduction of the retargeted hand trajectory.

State and action. At time t, the policy state is $\mathbf{s}_t = \{\mathbf{s}^{\text{prop}}_t, \mathbf{s}^{\text{obj}}_t, \mathbf{s}^{\text{ref}}_t, \mathbf{g}^o\}$, where the proprioceptive state $\mathbf{s}^{\text{prop}}_t = \{\mathbf{q}^h_t, \dot{\mathbf{q}}^h_t, \mathbf{w}_t, \dot{\mathbf{w}}_t\}$ consists of robot hand joint angles $\mathbf{q}^h_t$, velocities $\dot{\mathbf{q}}^h_t$, wrist pose $\mathbf{w}_t$, and wrist velocity $\dot{\mathbf{w}}_t$. The object state $\mathbf{s}^{\text{obj}}_t$ contains the simulated object pose and velocity relative to the wrist. The reference state $\mathbf{s}^{\text{ref}}_t$ contains tracking targets from $\mathcal{D}^{\text{ref}}$ (wrist motion, hand keypoints, object pose trajectory), and $\mathbf{g}^o$ is the BPS encoding of the object mesh.

The policy outputs action $\mathbf{a}_t = [\mathbf{a}^w_t, \mathbf{a}^h_t]$, where $\mathbf{a}^w_t$ specifies incremental commands for the virtual 6-DoF wrist and $\mathbf{a}^h_t$ specifies absolute joint-position targets for the dexterous hand.

Reward. The total reward is defined as:

$$r_t = w_{\text{hand}} r^t_{\text{hand}} + w_{\text{object}} r^t_{\text{object}} + w_{\text{contact}} r^t_{\text{contact}} + r^t_{\text{reg}}. \tag{7}$$

The hand tracking reward is computed over selected robot-hand keypoints $\mathcal{K}$ rather than joint angles:

$$r^t_{\text{hand}} = \sum_{k \in \mathcal{K}} \omega_k \exp\left(-\lambda_k \left\| \mathbf{j}^t_{d,k} - \mathbf{j}^{t,\text{ref}}_{d,k} \right\|^2_2 \right), \tag{8}$$

where $\mathbf{j}^t_{d,k}$ and $\mathbf{j}^{t,\text{ref}}_{d,k}$ are the simulated and retargeted robot-hand keypoints.

The object tracking reward encourages the simulated object to follow the reference trajectory:

$$r^t_{\text{object}} = \exp\left(-\lambda_p \left\| \mathbf{p}^{o,\text{sim}}_t - \mathbf{p}^{o,\text{ref}}_t \right\|^2_2 - \lambda_R d_R \left(\mathbf{R}^{o,\text{sim}}_t, \mathbf{R}^{o,\text{ref}}_t \right)^2 \right), \tag{9}$$

Method	PA-MPJPE ↓	MRPE_{obj} ↓	F5_{posed} ↑	$\text{F10}_{\text{posed}}$ ↑	CD_{posed} ↓	JM ↓
DexImit Mu et al. (2026)	5.90 / 5.73	126.37 / 75.99	15.63 / 12.07	28.64 / 24.21	180.99 / 21.82	6.53 / 5.53
Do as I Do Paliwal et al. (2026)	**3.60 / 3.44**	81.61 / 69.68	14.28 / 15.92	28.67 / 33.26	25.41 / 20.57	3.84 / 3.70
HandMimic	3.87 / 3.80	**60.57 / 39.85**	**22.02 / 22.09**	**42.81 / 45.23**	**16.28 / 13.57**	**0.63 / 0.61**

Table 1: **HOI reconstruction comparison on DexYCB.** All methods are evaluated on the same 28 sequences using the same evaluation protocol. Values are reported as mean / median. PA-MPJPE, MRPE_{obj}, and Chamfer distance (C.D.) are in mm; F5 and F10 are reported as percentages; and jerk magnitude (JM) is in mm/frame^3.

where $\mathbf{T}_t^{o,\text{sim}} = (\mathbf{p}_t^{o,\text{sim}}, \mathbf{R}_t^{o,\text{sim}})$ and $\mathbf{T}_t^{o,\text{ref}} = (\mathbf{p}_t^{o,\text{ref}}, \mathbf{R}_t^{o,\text{ref}})$ are the simulated and reference object poses. We set $w_{\text{object}} > w_{\text{hand}}$ to prioritize object manipulation over hand imitation. The contact reward encourages fingertip contact after the pre-grasp phase. We additionally use standard motion regularization terms. Full reward details and weights are provided in the appendix.

Training strategy. We apply three modifications to the baseline training setup: (1) post-grasp, we remove the hand-tracking early-termination condition and maintain only object-centric termination; (2) we train the policy from scratch to handle the high motion variation of in-the-wild videos; and (3) we train with the shape and scale augmentations defined in Sec. 3.3 to prevent overfitting to reconstruction errors.

4 Experiments

Our experiments address three questions. First, how accurate is the video-derived hand-object reference produced by our reconstruction pipeline, and what errors remain even with a strong reconstruction frontend? Second, can the resulting imperfect reference still support successful dexterous manipulation? Third, does training across digital siblings improve generalization beyond the single reconstructed object? We conclude with real-world deployment on a real dexterous hand.

Datasets and policy metrics. We evaluate on DexYCB Chao et al. (2021) and six in-the-wild monocular iPhone videos. For DexYCB, we start from 32 sequences covering 8 YCB objects with 4 sequences per object under the master-camera convention. To avoid sequences in which the manipulated object is largely unobservable, we apply a reconstruction-independent visibility filter based solely on DexYCB ground truth: for each sequence, we compute the average fraction of the projected object silhouette that is visible in the annotated segmentation and retain sequences with an average visibility score of at least 0.65. This yields 28 sequences, which are used for both reconstruction and policy evaluation. For policy evaluation, we report object rotation error E_R, translation error E_T, and success rate (SR), with a rollout considered successful when $E_R < 30^\circ$ and $E_T < 60\,\text{mm}$. Each DexYCB policy result is evaluated over 600 rollouts.

4.1 How Good Is the Video-Derived Interaction Reference?

We first evaluate the reconstructed hand-object interaction before studying policy learning. Our goal is to establish that HandMimic starts from a strong reconstruction frontend, while identifying the residual errors that remain relevant to dexterous manipulation.

Metrics. PA-MPJPE measures local hand articulation error after Procrustes alignment, while MRPE_{obj} measures the hand-root error in the object frame and thus captures hand-object alignment. F5_{posed} and $\text{F10}_{\text{posed}}$ measure posed object-surface accuracy at 5 mm and 10 mm thresholds, CD_{posed} measures the corresponding Chamfer distance, and jerk magnitude (JM) measures temporal smoothness. All values are reported as mean / median.

Under the same 28-sequence DexYCB evaluation, HandMimic achieves the strongest performance on most interaction-centric metrics, including hand-object alignment, posed object geometry, and temporal smoothness, while Do as I Do obtains slightly lower PA-MPJPE. This highlights that accurate local hand articulation does not necessarily imply an accurate hand-object interaction. Overall, our pipeline provides a strong video-derived reference, while still leaving residual errors relevant to physical manipulation.

Demo Source	Method	Object Geometry	DexYCB			In-the-Wild
			E_R (deg) ↓	E_T (mm) ↓	SR (%) ↑	SR (%) ↑
GT Demonstration	ManipTrans	GT Mesh	14.0	10.3	**75.7**	N/A
	Ours	GT Mesh	7.6	16.0	71.4	N/A
Reconstructed Demo	ManipTrans	GT Mesh	30.9	44.7	6.0	N/A
	ManipTrans	Recon. Mesh	25.3	34.8	35.4	38.5
	Ours	GT Mesh	10.5	13.6	**67.8**	N/A
	Ours	Recon. Mesh	4.6	10.1	**85.7**	**83.3**

Table 2: **Policy performance.** We report E_R (deg), E_T (mm), and SR (%) under different demonstration sources and object geometries. For in-the-wild videos, only SR is reported.

Residual errors and failure modes. We observe recurring errors from object-pose ambiguity or drift under occlusion and symmetry, occluded hand articulation, residual hand-object misalignment, and inaccurate object geometry or scale, particularly on unobserved surfaces. Figure 4 shows representative cases. These errors motivate treating the reconstruction as a reference for downstream learning rather than as an already physically valid robot demonstration.

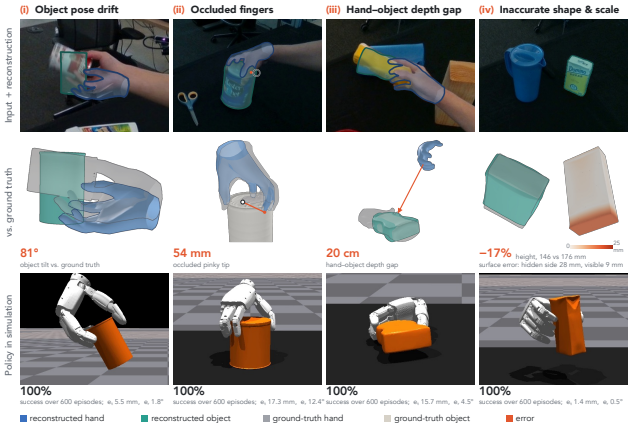


Figure 4: **Residual reconstruction errors and simulation practice.** Representative errors in object pose, occluded hand articulation, hand-object alignment, and unseen geometry. Despite these errors, simulation-based practice recovers manipulation behaviors that reproduce the demonstrated object motion.

4.2 Can Imperfect Video-Derived References Produce Successful Manipulation?

We next evaluate whether reconstructed and retargeted interactions remain useful for manipulation, and whether HandMimic can effectively learn from the imperfections introduced by video reconstruction.

With ground-truth demonstrations and object geometry, HandMimic achieves 71.4% SR, comparable to ManipTrans at 75.7%, showing that our learning formulation remains competitive when given clean references. More importantly, when replacing the ground-truth demonstration with our reconstructed reference while keeping the same ground-truth geometry, HandMimic retains 67.8% SR-about 95% of its ground-truth-demo performance. This indicates that the reconstructed and retargeted interaction preserves most of the motion information needed for manipulation.

The benefit of HandMimic becomes more pronounced when learning from these imperfect references. With the reconstructed demonstration, HandMimic reaches 67.8% SR on the ground-truth mesh compared with 6.0% for ManipTrans, and 85.7% on the reconstructed mesh compared with 35.4%. On the in-the-wild videos, HandMimic similarly achieves 83.3% SR. Together, these results show that reconstructed interactions can provide effective manipulation references, provided that the learning procedure can accommodate their residual errors.

Comparison with Do as I Do. Having established in Sec. 4.1 that HandMimic improves interaction-centric reconstruction metrics over Do as I Do Paliwal et al. (2026), we further compare their end-to-end manipulation performance. Since the two methods use different success definitions, we evaluate both under both protocols. Under our per-frame criterion (60 mm/30°), HandMimic succeeds on $24/28$ sequences compared with $0/28$ for Do as I Do. Under their trajectory-average criterion (100 mm/0.5 rad), HandMimic succeeds on $27/28$ sequences compared with $17/28$.

Training Geometry	DexYCB					In-the-Wild	
	GT SR (%) ↑	Recon. SR (%) ↑	Close SR (%) ↑	Medium SR (%) ↑	Far SR (%) ↑	Recon. SR (%) ↑	Held-out SR (%) ↑
Single reconstructed mesh	18.0	69.3	9.8	6.4	3.5	83.1	0.0
Scale randomization	62.1	85.6	72.5	57.3	21.6	**83.3**	30.7
Retrieved siblings + scale	**67.9**	**85.7**	**81.5**	**72.3**	**40.6**	**83.3**	36.9
Generated siblings + scale	67.8	**85.7**	80.3	70.2	36.6	**83.3**	**43.3**

Table 3: **Generalization across object geometry.** Success rate under different training-geometry strategies on DexYCB and in-the-wild videos. Gray rows denote digital-sibling training.

4.3 Does Digital-Sibling Training Generalize to New Geometry?

A policy trained only on the reconstructed object may specialize to errors in its geometry and scale. We therefore test whether practicing the same interaction across multiple object geometries improves transfer to unseen shapes.

Held-out geometry benchmark. For each sequence, we treat the reconstructed object mesh as the anchor geometry. On DexYCB, we additionally evaluate on the ground-truth YCB scan Calli et al. (2015) corresponding to the demonstrated object. To measure generalization beyond this instance, we construct held-out test sets from real object scans in YCB, Google Scanned Objects Downs et al. (2022), and OmniObject3D Wu et al. (2023). Test shapes are grouped by their geometric distance to the reconstructed anchor into *close*, *medium*, and *far* bands. We use a normalized symmetric Chamfer distance after canonical alignment, with close corresponding to $d \leq 2\%$, medium to $2\% < d \leq 3.5\%$, and far to $3.5\% < d \leq 10\%$. All test shapes are held out from policy training.

We compare four training distributions. **Single reconstructed mesh** uses only the anchor geometry at its reconstructed scale. **Scale randomization** uses the same mesh with scale sampled from $\mathcal{U}(0.8, 1.2)$. **Generated siblings** augment the anchor with GPT-edited object variants reconstructed using SAM3D, while **retrieved siblings** use category-related 3D assets retrieved with OpenShape Liu et al. (2023). For a fair comparison, both strategies use 12 siblings per demonstration in addition to the reconstructed anchor, with matched geometric-distance composition, the same scale randomization, policy architecture, and training budget.

Results. Table 3 shows that training on a single reconstructed mesh strongly overfits to the anchor geometry, while scale randomization and especially digital-sibling training substantially improve transfer to held-out shapes. Retrieved and generated siblings perform similarly overall: retrieval is slightly stronger on DexYCB, whereas generation performs better on the in-the-wild objects.

A likely reason is that DexYCB contains common YCB household objects, for which large 3D asset collections are more likely to contain closely matching shapes. For arbitrary in-the-wild objects, such matches are harder to retrieve, while image-conditioned generation can construct variations directly from the observed instance. We therefore use generated siblings in HandMimic, while the results suggest that the main benefit comes from practicing across multiple geometries rather than from a particular sibling source.

Figure 5 qualitatively shows that HandMimic remains stable under geometry variation, whereas ManipTrans trained on ground-truth data degrades on the changed geometry.

4.4 Real-World Deployment

Finally, we deploy the motions learned from monocular videos on a physical KUKA iiwa 7 equipped with an XHand1 dexterous hand. We consider three manipulation tasks and, for each task, evaluate both the same object instance observed in the source video and a held-out object with unseen geometry, giving six task-object settings. We perform 10 trials per setting using open-loop replay without online visual feedback. Across the six settings, four achieve $10/10$ successes, one achieves $9/10$, and one achieves $7/10$, for an overall success rate of $56/60$. Figure 6 shows representative deployments together with the corresponding source videos. These results show that motions obtained

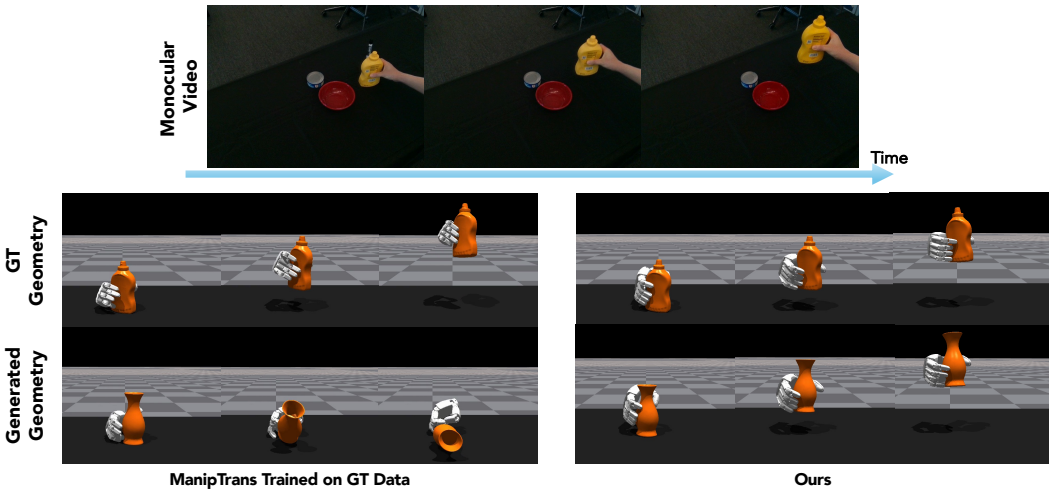


Figure 5: **Robustness to geometry variation.** ManipTrans trained on ground-truth data degrades under changed object geometry, while HandMimic remains stable across both geometries.

through video reconstruction, retargeting, and simulation practice transfer reliably to the physical hand and remain effective under changes in object geometry.

5 Conclusion

We present HandMimic, a framework for learning dexterous manipulation directly from monocular human videos. Rather than treating reconstructed and retargeted interactions as physically valid robot demonstrations, HandMimic uses them as imperfect kinematic references for physical practice in simulation. Our experiments show that strong video-derived reconstructions still contain manipulation-relevant errors, yet policies trained from these references retain performance close to those trained from ground-truth demonstrations. Training across digital siblings further reduces dependence on a single reconstructed object and improves generalization to unseen geometry. Finally, we validate the learned motions through real-world replay on a KUKA iiwa 7 with an XHand1, achieving reliable manipulation across both demonstrated and unseen object geometries.



Figure 6: **Real-world motion replay.** Learned motions from monocular video references are replayed on a KUKA iiwa 7 with an XHand1 dexterous hand.

Limitations HandMimic currently assumes rigid objects with temporally consistent geometry and therefore does not directly handle articulated or deformable objects. The reconstruction pipeline also depends on reliable object segmentation, pose tracking, and hand estimation, which can degrade under severe occlusion, fast motion, symmetry, or challenging appearance. These perception errors can propagate into the reconstructed interaction and make subsequent retargeting and policy learning more difficult. Extending the framework to articulated or deformable objects and improving robustness to challenging videos are natural directions for future work.

AI USE STATEMENT

We used generative AI tools to assist with paper writing, code development, and debugging. All AI-assisted content and code were reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

REFERENCES

Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In IEEE (ed.), *Proceedings of IEEE International Conference on Advanced Robotics (ICAR '15)*, pp. 510 – 517, July 2015.

Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan 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*, pp. 9044–9053, 2021.

Hongyi Chen, Tony Dong, Tiancheng Wu, Liquan Wang, Yash Jangir, Yaru Niu, Yufei Ye, Homanga Bharadhwaj, Zackory Erickson, and Jeffrey Ichnowski. Dexterous manipulation policies from rgb human videos via 3d hand-object trajectory reconstruction. *arXiv preprint arXiv:2602.09013*, 2026a.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7220–7232, 2026b.

Runyu Ding, Yuzhe Qin, Jiyue Zhu, Chengzhe Cheng, Shiqi Yang, Ruihan Yang, Xiaojuan Qi, and Xiaolong Wang. Bunny-VisionPro: Real-time bimanual dexterous teleoperation for imitation learning. *arXiv preprint arXiv:2407.03162*, 2024.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 international conference on robotics and automation (ICRA)*, pp. 2553–2560. Ieee, 2022.

Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan D. Ratliff, and Dieter Fox. DexPilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9164–9170, 2020.

Jhen Hsieh, Kuan-Hsun Tu, Kuo-Han Hung, and Tsung-Wei Ke. DexMan: Learning bimanual dexterous manipulation from human and generated videos. *arXiv preprint arXiv:2510.08475*, 2025.

Yunfan Jiang, Chen Wang, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. TRANSIC: Sim-to-real policy transfer by learning from online correction. In *Conference on Robot Learning (CoRL)*, 2024.

Kushal Kedia, Tyler Ga Wei Lum, Jeannette Bohg, and C. Karen Liu. SimToolReal: An object-centric policy for zero-shot dexterous tool manipulation. *arXiv preprint arXiv:2602.16863*, 2026.

Chung Min Kim, Brent Yi, Hongsuk Choi, Yi Ma, Ken Goldberg, and Angjoo Kanazawa. Pyroki: A modular toolkit for robot kinematic optimization. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1312–1319. IEEE, 2025.

Kailin Li, Puhao Li, Tengyu Liu, Yuyang Li, and Siyuan Huang. ManipTrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6991–7003, 2025.

Toru Lin, Kartik Sachdev, Linxi Fan, and Jitendra Malik. Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids. *Proceedings of The 9th Conference on Robot Learning (CoRL)*, pp. 4926–4940, 2025.

Minghua Liu, Ruoxi Shi, Kaiming Kuang, Yinhao Zhu, Xuanlin Li, Shizhong Han, Hong Cai, Fatih Porikli, and Hao Su. Openshape: Scaling up 3d shape representation towards open-world understanding. *Advances in neural information processing systems*, 36:44860–44879, 2023.

Xueyi Liu, Kangbo Lyu, Jieqiong Zhang, Tao Du, and Li Yi. QuasiSim: Parameterized quasi-physical simulators for dexterous manipulations transfer. In *European Conference on Computer Vision (ECCV)*, 2024.

Xueyi Liu, Jianibieke Adalibieke, Qianwei Han, Yuzhe Qin, and Li Yi. DexTrack: Towards generalizable neural tracking control for dexterous manipulation from human references. In *International Conference on Learning Representations (ICLR)*, 2025.

Zhengyi Luo, Jinkun Cao, Sammy Christen, Alexander Winkler, Kris Kitani, and Weipeng Xu. OmniGrasp: Grasping diverse objects with simulated humanoids. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High performance GPU-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

Zhao Mandi, Yifan Hou, Dieter Fox, Yashraj Narang, Ajay Mandlekar, and Shuran Song. DexMachina: Functional retargeting for bimanual dexterous manipulation. *arXiv preprint arXiv:2505.24853*, 2025.

Juncheng Mu, Sizhe Yang, Yiming Bao, Hojin Bae, Tianming Wei, and Linning Xu. DexImit: Learning bimanual dexterous manipulation from monocular human videos. *arXiv preprint arXiv:2602.10105*, 2026.

OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving Rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Bhawna Paliwal, Haritheja Etukuru, William Liang, Pieter Abbeel, Nur Muhammad Mahi Shafiullah, and Jitendra Malik. Do as i do: Dexterous manipulation data from everyday human videos. *arXiv preprint arXiv:2606.19333*, 2026.

Chaoyi Pan, Changhao Wang, Haozhi Qi, Zixi Liu, Homanga Bharadhwaj, Akash Sharma, Tingfan Wu, Guanya Shi, Jitendra Malik, and Francois Hogan. SPIDER: Scalable physics-informed dexterous retargeting. *arXiv preprint arXiv:2511.09484*, 2025.

Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. Wilor: End-to-end 3d hand localization and reconstruction in-the-wild. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12242–12254. IEEE, 2025.

Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. DexMV: Imitation learning for dexterous manipulation from human videos. *arXiv preprint arXiv:2108.05877*, 2022.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Robotics: Science and Systems (RSS)*, 2018.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. VideoDex: Learning dexterity from internet videos. In *Conference on Robot Learning (CoRL)*, pp. 654–665, 2023.

Aravind Sivakumar, Kenneth Shaw, and Deepak Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on YouTube. *Robotics: Science and Systems (RSS)*, 2022.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5026–5033, 2012.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17868–17879, 2024.

Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 803–814. IEEE, 2023.

Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: Advancing 3d point tracking with explicit camera motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6726–6737, 2025.

Shuqi Zhao, Xinghao Zhu, Yuxin Chen, Chenran Li, Xiang Zhang, Mingyu Ding, and Masayoshi Tomizuka. DexH2R: Task-oriented dexterous manipulation from human to robots. *arXiv preprint arXiv:2411.04428*, 2024.