Track, Articulate, Act

Generating Articulation from Casual Human Videos

Jiaming Zhang Homanga Bharadhwaj

Department of Computer Science, Johns Hopkins University

jzhan282@jhu.edu homanga@jhu.edu

track-articulate-act.github.io

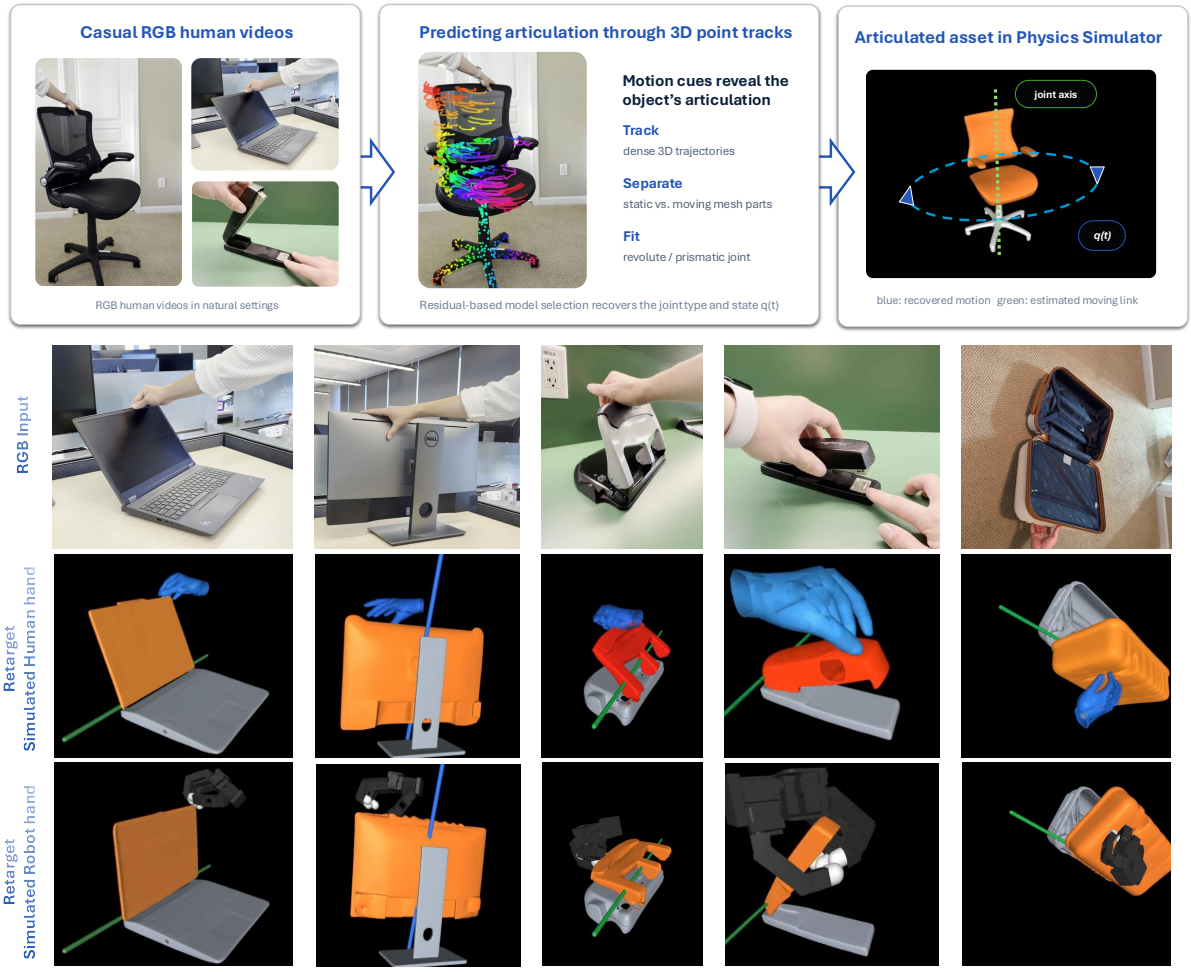

Figure 1: Overview. Starting from only a casual RGB human video, our method extracts motion cues, predicts articulation, constructs an approximate simulation-ready articulated object model, and replays the recovered hand-object interaction in a physics simulator. Results show that our framework is able to infer articulation for different types of joints including revolute and prismatic across diverse real-world objects.

Abstract. Human videos contain rich causal evidence for robot manipulation: they reveal how hand motion induces object motion and produces task-relevant changes in object state. In this work, we study articulated objects such as doors, drawers, cabinets, laptops, ovens, and hinged containers that are ubiquitous in daily life and present unique challenges for embodied interaction. These objects cannot be represented by a single pose; their motion depends on the underlying parts and joints. We introduce a real-to-sim framework that reconstructs a simulation-ready articulated object and hand–object interaction from a casual monocular RGB video, without RGB-D or multi-view input, prior scans, manually specified joints, or robot demonstrations. Our key insight is that dense 3D point tracks provide an embodiment-agnostic articulation cue: points on the fixed link remain approximately stationary, while points on the moving link follow coherent revolute or prismatic motion. Our method segments the links, estimates the joint and its state trajectory, reconstructs an articulated asset, and aligns the recovered 3D hand motion with the object. Central to our approach is a modular recipe that repurposes powerful pretrained models for single-image 3D reconstruction, mesh segmentation, and 3D scene flow, connecting their predictions through explicit geometric reasoning to infer articulation. We use the reconstructed articulated object and the human hand trajectory to replay interactions through contact in MuJoCo. The framework shows how pretrained vision models and explicit motion reasoning can turn casual human videos into articulated object models suitable for downstream embodied interactions.

1 Introduction

Everyday environments are filled with objects that open, slide, rotate, and fold. Cabinets, drawers, laptops, box lids, ovens, and hinged containers are central to household, office, and workshop activities. Interacting with them is a fundamental challenge for embodied intelligence: successful action requires understanding which part moves, which part remains fixed, how they are connected, and how contact changes the object state. Human interactions reveal this underlying structure very naturally. We study how to recover these embodiment-agnostic motion cues from ordinary human videos and use them for articulated object manipulation.

We ask whether this structure can be recovered from a single casually captured monocular video of a person manipulating an articulated object. Although easy to collect at scale, such a video provides no actions, depth observations, part labels, joint annotations, object meshes, or simulator-ready assets. Recent work has shown the value of converting human videos into explicit 3D, object-centric, and interaction-centric representations such as point tracks, hand–object trajectories, and scene flow [5, 6, 10, 15, 27]. We build on this principle, but must recover both the observed motion and the kinematic structure that explains it. A drawer front translates relative to a static frame, whereas a laptop screen or cabinet door rotates about a hinge. Unlike rigid-object pose, articulated-object pose is therefore not well-defined until the parts, joints, and moving link are identified [26, 58].

When a person opens a lid or pulls a drawer, points on the moving link follow a coherent low-dimensional trajectory, while points on the static structure remain approximately fixed. Dense 3D tracks therefore provide an interface between general video perception and explicit kinematic reasoning: their spatial grouping suggests the part decomposition, and their trajectories distinguish translation along an axis from rotation about a hinge. Recent progress in long-range tracking and monocular 4D reconstruction makes these cues increasingly accessible in ordinary videos [16, 21–23, 60]. The remaining challenge is to turn noisy tracks and incomplete visual observations into coherent link geometry, a joint model, and a metrically aligned interaction.

Recent work has made substantial progress on articulation recovery from videos, using learned part-and-joint predictors, dynamic reconstruction, point-track optimization, or geometric primi-

tives [2, 3, 14, 35]. Complementary systems reconstruct articulated hand–object interactions by coordinating geometry, pose, and contact priors [34, 57]. These methods establish that casual videos can support rich articulated reconstruction. Their primary focus, however, is recovering geometry and kinematics, typically evaluated through joint, reconstruction, tracking, or rendering accuracy. We study a complementary research question: whether the recovered scene can be instantiated as an explicit collision asset in physics simulators and whether the observed human hand motion can physically produce the demonstrated state change. *This requires not only plausible geometry and visual articulation, but also consistent scale, hand–object alignment, joint limits, and contact behavior.*

This goal naturally suggests a *modular approach*: although no single pretrained model recovers a simulation-ready articulated interaction, current models offer complementary estimates of object geometry, motion, and hand–object interaction. Starting from a casual monocular video, we select low-occlusion frames that span the observed articulation and reconstruct a mesh hypothesis from each. We use part masks to transfer the static–moving decomposition onto these meshes and register them through the static link, bringing the geometry into a common object frame. Within this frame, we ground dense 3D tracks using the same monocular geometry and fit multiple revolute and prismatic joint hypotheses to the observed moving-link trajectories. We refine and select the joint parameters and type by requiring the rendered link motion to agree with both the tracked points and the observed silhouettes. Undoing the recovered articulation then allows us to fuse the multi-frame link observations into a canonical articulated asset. Finally, we align the reconstructed 3D hand trajectory with the object and replay the interaction in a physics simulator (MuJoCo).

In this way, *we connect predictions from general-purpose pretrained models through geometric reasoning and optimization*, without training an articulation-specific model. This modular design *allows the framework to benefit from advances in its individual components* while preserving clear geometric and physical assumptions. In summary, our contributions are:

1. We develop a modular real-to-sim framework that converts a casual monocular human video into a simulation-ready articulated object and aligned hand–object interaction. The framework connects pretrained models for segmentation, single-image 3D reconstruction, monocular geometry, dense tracking, and hand reconstruction without training an articulation-specific model.

2. We propose a motion-grounded optimization for articulation inference that registers multi-frame object reconstructions, fits revolute and prismatic joint hypotheses to mesh-grounded 3D point trajectories, and jointly refines the joint parameters and state trajectory through track and silhouette re-projection objectives.

3. We enable interaction with the reconstructed object in physics simulation by constructing explicit link and collision geometry, aligning the recovered 3D hand trajectory with the articulated asset, and re-targeting the interaction to a simulated hand whose contact drives the passive object joint.

2 Related Work

Learning manipulation with human videos. Human videos offer manipulation experience at a scale that robot data cannot yet match, but they do not directly provide actions, metric 3D object state, or contact supervision. Prior work has used these videos to learn representations and rewards [37, 39], imitate human play or demonstrations [19, 50, 54, 55], and extract transferable cues such as affordances, hand poses, point trajectories, and future object motion [1, 4, 6, 27, 52]. Recent approaches also use generated videos and learned world models to provide visual plans

and action priors [9, 13, 18, 30, 43, 48]. Richer capture systems, including smart glasses, headsets, motion capture, and calibrated robot scenes, make human motion easier to transfer but introduce additional sensing and alignment assumptions [10, 15, 56]. We instead consider the setting of an ordinary monocular video to recover the articulated object model required for interaction: its static and moving links, joint type and parameters, state trajectory, and alignment with the human hand.

Hand–object reconstruction from video. Recent progress in monocular 3D reconstruction makes it possible to recover complementary elements of a hand–object interaction from ordinary video. Hand models estimate pose and shape from individual images [44, 46, 49], while video-based methods recover temporally consistent hand motion in a global frame [62, 63]. In parallel, single-image object reconstruction models such as SAM 3D Objects can infer complete mesh hypotheses from partial visual observations [11]. Joint hand–object approaches further reason about object geometry, pose, and contact, although most focus on rigid objects [7, 12, 17, 36, 61]. ArtHOI and Clay-to-Stone extend this setting to articulated interactions by coordinating geometric, motion, and contact priors [34, 57]. Our framework builds directly on these advances: we use SAM 3D Objects to reconstruct object geometry and HaWoR to recover the global 3D hand trajectory. We then connect these predictions with dense motion tracks and explicit kinematic reasoning to recover a low-DoF simulator joint, align the hand with the resulting collision geometry, and test whether replaying its trajectory physically actuates the passive joint.

Articulated object reconstruction. Articulation recovery has traditionally relied on explicit 3D observations or multiple views. Classical kinematic fitting and later reconstruction systems recover rearticulable models from rigid-part motion, interaction point clouds, dynamic 4D point clouds, two-state multiview images, or multiview videos [20, 32, 33, 41, 53]. RSRD similarly begins from a static multiview scan [24], while Video2Articulation and ArtiPoint use RGB-D video, with ArtiPoint explicitly handling camera motion and partial visibility through point tracking and factor-graph optimization [45, 59]. These settings provide stronger geometric observations than the casual monocular videos we consider in this work.

Recent work relaxes these assumptions in several complementary ways. Sim2Art learns monocular part and joint prediction from synthetic training data [2]; VideoArtGS and Articulat3D combine video motion cues with video-specific 3D reconstruction and explicit joint fitting [14, 35]; and Articulation in Prime fits visibility-aware geometric primitives without relying on long-term correspondences [3]. REACTO instead targets general rearticulable surface reconstruction without restricting the output to an explicit revolute or prismatic simulator joint [51]. Another line of work generates articulated assets from language, images, or independently reconstructed parts, rather than recovering the articulation demonstrated in a video [25, 29, 38, 47].

Compared with existing methods that recover articulated objects from casual monocular videos [14, 35], we target a *simulation-ready hand–object interaction* rather than object reconstruction alone. Instead of training an articulation-specific model, we connect predictions from replaceable, general-purpose vision models through multi-frame registration, dense tracking, and explicit low-dimensional kinematic fitting. This modular design allows inheriting improvements in the respective pretrained components that have wide community interest. We additionally reconstruct the 3D hand trajectory and align it with the recovered object and its collision geometry. The resulting links and passive joint are instantiated in MuJoCo, enabling interactive physics simulation and allowing us to test whether contact with the replayed hand produces the demonstrated state change. This physical evaluation complements the emphasis of prior work on joint, geometry, tracking, and rendering accuracy.

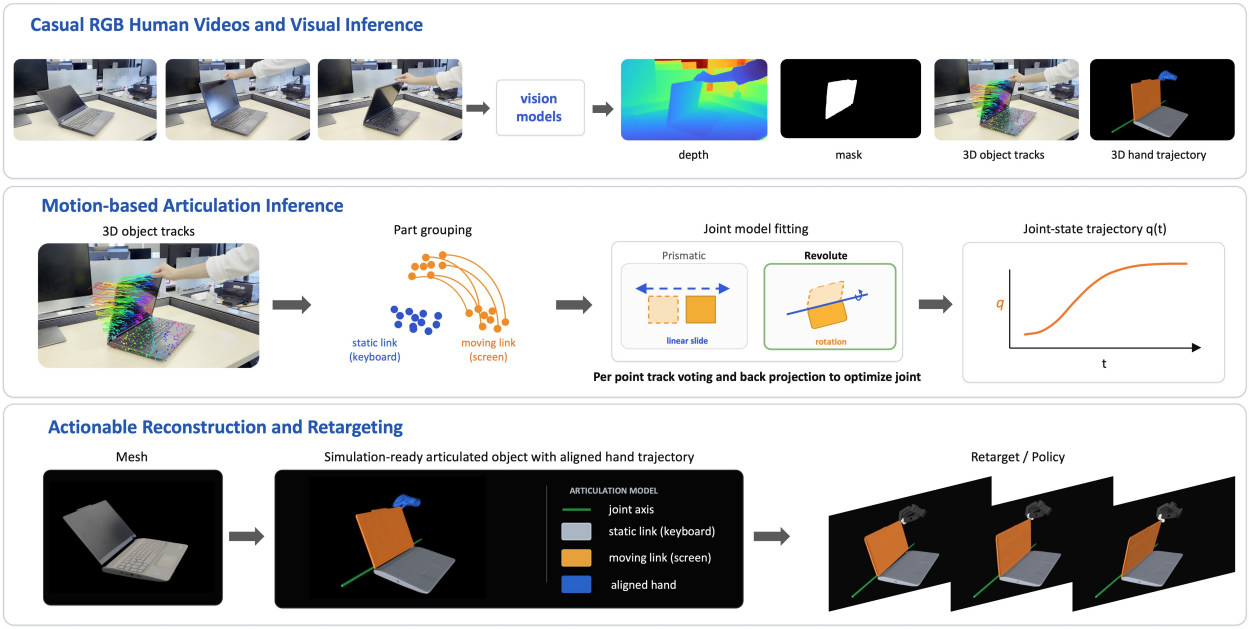

Figure 2: Method. Given a casual human manipulation video, we select frames that expose different articulation states, reconstruct and register part-level geometry, and fit joint hypotheses to metrically grounded 3D tracks. We refine the joint through track and silhouette reprojection, then align and replay the recovered 3D hand motion with the articulated object in MuJoCo.

3 Problem Formulation

We consider a monocular RGB video $V = \{I_t\}_{t=1}^{T}$ of a human manipulating an articulated object. From this video alone, we recover

$$\mathcal{O} = (\mathcal{M}_s, \mathcal{M}_m, \mathcal{J}, q_{1:T}), \tag{1}$$

where $\mathcal{M}_s$ and $\mathcal{M}_m$ are the static and moving links, $\mathcal{J}$ is a revolute or prismatic joint, and $q_{1:T}$ is the demonstrated joint trajectory. We also recover the 3D human hand motion and align it with the object in simulation. We assume one dominant joint per object, but no sensor depth, object scan, CAD model, part label, or joint annotation. Unlike rigid-object pose estimation, the object pose is not defined until its links and their kinematic constraint are known; our goal is therefore to infer the model that explains the observed motion.

4 Track, Articulate, Act

Our method turns a monocular human video into a simulation-ready articulated hand–object scene without training an articulation-specific model. We use pretrained models to recover temporally consistent masks, single-frame mesh hypotheses, monocular geometry, dense point correspondences, and global hand motion. We then connect these predictions through four explicit geometric interfaces. First, we select low-occlusion frames, reconstruct each independently, and register the resulting meshes through the static link (Sec. 4.1). Second, we place dense tracks in the same object frame using the shared monocular geometry and reconstructed surfaces (Sec. 4.2). Third, we fit revolute and prismatic joint hypotheses and refine them by requiring the rendered motion to agree with the observed tracks and silhouettes (Secs. 4.3–4.4). Finally, we align the recovered 3D hand motion

with the articulated object and re-target it open-loop in MuJoCo to a simulated human hand model, where contact with the hand drives the passive object joint (Sec. 4.5).

4.1 Object Masks and Multi-Frame Reconstruction from Video

Hands in videos often occlude precisely the link that moves, making an arbitrary video frame a poor source of object geometry. We use SAM 3 [8] to track the object and hand masks, choose the least-occluded valid frame as the reference, and add a few low-occlusion keyframes that span large object-track displacement. These frames typically capture open, intermediate, and closed configurations. We project the static and moving point clusters described in Sec. 4.2 into a clear frame and use them as point prompts to obtain the corresponding part masks. We then propagate these masks through the video, remove hand pixels, and assign every object pixel to at most one link.

At each keyframe, we estimate depth and camera geometry with Depth Anything 3 (DA3) [31] and reconstruct an object mesh from the crop and mask using SAM 3D [11]. We do not assume access to ground-truth sensor depth. We transfer the propagated static–moving decomposition to each mesh by guiding SegviGen [28] with the corresponding part masks. Projected vertex support across keyframes resolves ambiguous labels and keeps the two link meshes disjoint.

Because these meshes are reconstructed independently, they vary in shape, scale, and pose. We ground each prediction with the estimated depth and camera intrinsics, then register the keyframes through the link known to be static. The alignment uses a robust depth-aware similarity transform that takes into account any residual scale inconsistency and variability in single-image object reconstructions. We fuse the registered static observations into $\mathcal{M}_s$ and retain the moving-link meshes in their observed configurations. Their relative displacement provides evidence for the joint; once the joint is recovered, we undo this motion and fuse them into the canonical moving link $\mathcal{M}_m$.

4.2 Metric 3D Point Tracks

We recover dense reference-anchored correspondences and visibility over the complete video with TrackCraft3R [40]. Tracking operates directly on the full RGB frames and does not require object, part, or hand masks. After tracking, we use the propagated whole-object and hand masks only to retain object trajectories and discard hand and background trajectories. Let $\mathbf{u}_{i,t}$ be the image location of track i at time t. Using depth D_t, intrinsics $\mathbf{K}_t$, and a transform $\mathbf{G}_t$ from the camera to the canonical object frame, we first lift it to

$$\widetilde{\mathbf{p}}_{i,t} = \mathbf{G}_t \Big[D_t(\mathbf{u}_{i,t}) \mathbf{K}_t^{-1} \overline{\mathbf{u}}_{i,t} \Big] , \tag{2}$$

where $\overline{\mathbf{u}}_{i,t}$ is the homogeneous image coordinate. DA3 provides the initial depth and camera parameters. We separate the resulting camera-compensated trajectories into static and moving point clusters according to their motion: static points remain approximately fixed, whereas moving points follow a coherent motion. Projecting these clusters into the video provides the point prompts used to obtain the part masks in Sec. 4.1.

The tracker provides useful correspondences, but its framewise 3D coordinates can inherit some monocular depth fluctuations. After registering the reconstructed meshes through the static link, we refine $\mathbf{G}_t$ by aligning the observed static surface across frames. We then retain the tracker's image correspondences and, at the selected keyframes, replace the lifted depth with the first intersection between the corresponding camera ray and the registered mesh surface. We denote the resulting mesh-grounded points by $\mathbf{p}_{i,t}$. Using the same monocular geometry for mesh registration and track grounding keeps their coordinate frames and scales compatible.

4.3 Joint Hypotheses from Point Trajectories

We explain the moving-link trajectories with a single low-dimensional kinematic model. A prismatic candidate shares one translation direction across time, whereas a revolute candidate shares one axis and pivot. For joint class y, parameters $\boldsymbol{\theta}_y$, state trajectory $q_{1:T}$, and visibility $v_{i,t}$, we minimize

$$E_{\text{track}}(y) = \sum_{i,t} v_{i,t}\, \rho\Big(\|\mathbf{p}_{i,t} - g_y(q_t; \boldsymbol{\theta}_y)\mathbf{p}_{i,r}\|_2^2\Big) \tag{3}$$

over moving-link tracks, where g_y is the corresponding translation or rotation and ρ is a robust loss. We initialize prismatic candidates from the dominant displacement and revolute candidates from the observed velocity field. We built multiple subsets by sampling from the mesh-grounded points at key frames using RANSAC to produce several hypotheses, which is important when the motion is small or some depths are incorrect. The fit also recovers the scalar joint trajectory $q_{1:T}$, which we smooth over time while preserving its endpoints. E_{track} is calculated per joint type, and the actual joint type is determined by $y^* = \min\{E_{\text{track}}(y); \mid y \in \{\text{prismatic}, \text{revolute}\}\}$

4.4 Joint Refinement by Reprojection

A candidate can fit noisy 3D tracks while still moving the reconstructed link outside the observed object. We therefore replay every candidate joint hypothesis with the reconstructed mesh, which is articulated about the candidate joint with an estimated joint state $q_{1:T}$, and back-project the mesh onto the image plane. We jointly refine the joint parameters and $q_{1:T}$ using intersection over union (IoU) between the back-projected mesh and segmentation mask. A weak temporal penalty discourages abrupt changes in joint state. We select the candidate with the smallest normalized objective over the full sequence, requiring the final joint to explain both the point motion and the visible object shape.

There could be cases when a revolute joint located infinitely distant from the mesh could produce the same motion as a prismatic joint, which will result in a nearly identical cost between two joint types, i.e. $\|E_{\text{track}}(\text{revolute}) - E_{\text{track}}(\text{prismatic})\| \leq \tau$.

As a tie-breaker, we introduce a PCA-based naive joint estimation that extracts the principal axes of the moving link at key frames and consider it as a prismatic joint only when the PCA axes have less than 5° of variance across the articulation. Once the joint is fixed, we transform every moving-link reconstruction back to a common link frame and fuse the aligned observations into $\mathcal{M}_m$. The range of the recovered trajectory defines the joint limits, and we generate simple collision geometry for both links. The resulting MuJoCo asset contains the static and moving meshes, a single revolute or prismatic joint, and the demonstrated state trajectory.

4.5 Hand Alignment and Re-targeting

HaWoR [63] reconstructs a hand-mesh trajectory $\mathcal{H}_{1:T} = \{\mathbf{x}_{j,t}^h\}$ in its estimated world frame, while the articulation pipeline represents the moving-link trajectory in the canonical object frame as $\mathbf{T}_t^m = g_{\hat{y}}(q_t; \hat{\boldsymbol{\theta}})$. Although both are recovered from the same video, their independently estimated scale, pose, and depth can leave the hand displaced from the object. The main challenge is therefore to align the two trajectories while preserving their relative motion.

Since the hand is not visible across all frames and HaWoR sometimes confuses distance changes with scale changes, we first perform scale normalization of hand anatomy to the average value, then apply linear interpolation to make up for the frames where HaWoR fails to detect a hand. Thirdly, we align the depth of the hand such that it has the most intersection or the least distance to the generated mesh. The scale is simultaneously adjusted by preserving the same back-projected silhouette.

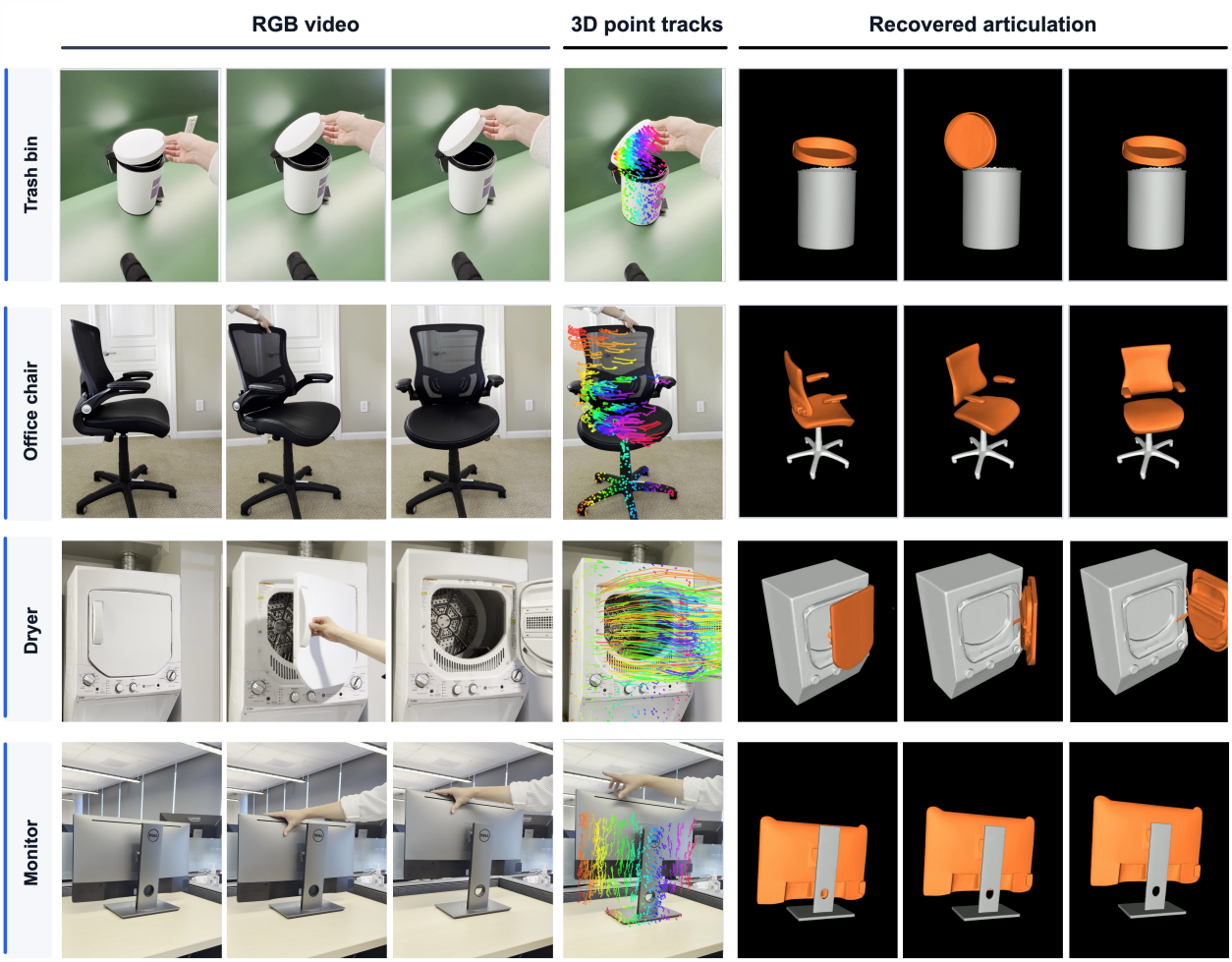

Figure 3: Articulation recovery from real human videos. From left to right, we show the RGB video input, 3D point tracks overlaid on an intermediate frame, and the reconstructed articulated asset. Our method handles both revolute and prismatic objects without any sensor depth or a prior object model.

We picked contact-relevant hand vertices and track $\mathbf{z}^*_{1:T}$ with position actuators in MuJoCo. The recovered object joint remains passive: its trajectory $q_{1:T}$ is used to align the demonstration and define the target state, but it is never commanded during a rollout. Any object motion must therefore arise from simulated hand contact.

Because our experiments use a human-shaped simulated hand, this kinematic mapping is sufficient for tasks that do not require grasping. The recovered hand trajectory, object trajectory, contact locations, and articulated asset also provide the inputs needed by cross-embodiment retargeting methods. For example, re-targeting approaches like SPIDER [42] and DexMachina [64] refine kinematic references into dynamically feasible robot trajectories using physics-based sampling and contact guidance. Such methods can replace our kinematic mapping when re-targeting the reconstructed interaction to a robot rather than the simulated human hand. We discuss additional details on physics-based re-targeting with articulated objects in the website.

Table 1: Articulated joint estimation on simulation videos. This evaluation uses simulator-derived depth, tracks, and masks, and analyzes the articulation optimization instead of the full pipeline. (n/a) means the method fails to produce any result at given condition.

	Method	Type acc. ↑	Axis/dir. err. (°) ↓	Axis loc. err. (mm) ↓
w/ Orbit	VideoArtGS [35]	0.93	1.81 ± 0.74	2.44 ± 3.86
	Articulat3D [14]	0.91	1.53 ± 1.17	2.58 ± 0.99
w/o Orbit	VideoArtGS	n/a	n/a	n/a
	Articulat3D	0.85	6.15 ± 9.03	7.82 ± 8.16
	Articulate-Anything [25]	0.66	30.66 ± 25.92	79.30 ± 125.58
	Ours	**0.97**	$\mathbf{1.96 \pm 0.45}$	3.93 ± 1.01

5 Experiments

Through experiments, we aim to understand two questions: 1) Can our approach recover the articulation observed in an RGB video, and 2) can the aligned 3D hand trajectory use the recovered asset to reproduce the interaction in physics simulation?

5.1 Experimental Setup

Real videos. We evaluate the full pipeline on eight casual RGB videos spanning laptop, hinged container, dryer, stapler, monitor, puncher, and office chair. Each contains one dominant revolute or prismatic interaction and is processed independently. Our approach receives only the RGB frames from the respective videos as input, and no additional auxiliary information.

Simulation videos. For quantitative articulation evaluation, a separate set of simulated models under known revolute and prismatic motion is chosen from the PARIS dataset [32]. To exclude the performance coupling across different modules, the depth map, point tracks, and segmentation masks are directly derived from the ground truth mesh. These simulator annotations provide ground truth joint parameters that are otherwise not available from casual real videos, and thus allow us to compare against the respective ground truths. We selected 10 simulated objects in PARIS, and for each joint, we drive the joint with 7 different combinations of motion range and starting pose to mimic the variability in real articulated objects.

Baselines. For articulation recovery, we compare with Articulate-Anything [25], VideoArtGS [35], and Articulat3D [14], which cover a VLM-based asset pipeline and the two closest monocular-video digital-twin methods. We do not include methods requiring an RGB-D capture or 4D point clouds in the main RGB-only table [24, 33, 45, 59].

5.2 Evaluation of the Full Framework from Human Videos to Articulated Objects

Figure 3 shows the full results of our framework for mesh estimation and recovering the articulation from real RGB human videos. We can see that the inferred mesh parts and articulation are both plausible and accurately reflect the intended motion of the object from the respective videos. We also evaluate the baselines on the real videos and find that our modular approach is significantly better in estimating the joints and also recovering the full interactable articulated object. We provide these additional qualitative comparisons in the website.

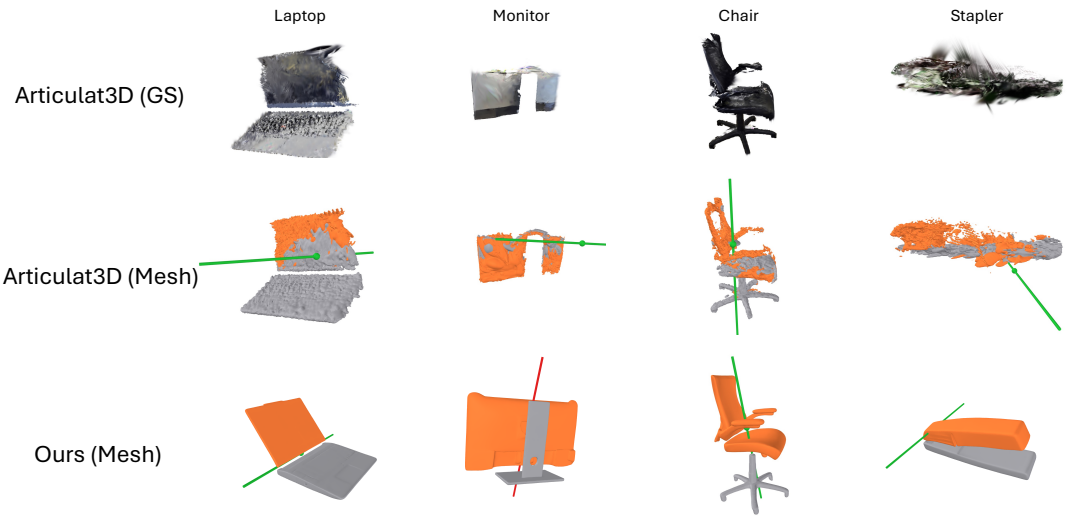

Figure 4: Qualitative results for articulated joint estimation on real data for our approach compared with Articulat3D. Each column shows the inferred joint for different objects.

5.3 Evaluation of Articulation Only

We isolate articulation recovery on the simulated objects, where every link and joint is known. We report joint-type accuracy, the sign-invariant angular error of the predicted axis or translation direction, and revolute-axis location error.

This evaluation uses simulator-derived depth, tracks, and masks, and analyzes the articulation optimization instead of the full pipeline. Within this setting, the key distinction is camera motion: VideoArtGS operates only on images captured while the camera orbits the object, and Articulat3D degrades substantially without an orbiting view (Table 1). Since removing this constraint is one of the core contributions of our pipeline, we evaluate both baselines under two settings: 1) with orbit views, to report their best-case performance, and 2) without, for a head-to-head comparison against our method. With casual RGB sequences, our method significantly outperforms both baselines; on real human videos that violate the orbit assumption, they often fail to recover a usable articulated object altogether, as shown qualitatively in Figure 4. Against the orbit-view upper bound, our joint parameters are slightly worse, but the absolute axis orientation and translation errors remain small relative to a mesh spanning tens of centimeters.

Figure 4 also shows two examples where our method could successfully estimate the articulation and mesh, but Articulat3D fails because of lacking orbit view image as inputs.

5.4 Hand–Object Interaction in Simulation

The preceding evaluations establish the quality of the reconstructed object geometry and recovered articulation. We now aim to understand how well the reconstructed hand trajectory can be aligned with this asset and feasibly re-targeted to reproduce the demonstrated interaction in a simulator. We consider opening, closing, pushing, pulling, and folding interactions that can be completed without requiring a stable grasp. The reconstructed human hand is aligned with the recovered object and kinematically re-targeted to a simulated human hand in MuJoCo. The hand trajectory is replayed open loop, while the object joint remains passive, so all object motion in the simulator must arise from contact with the hand.

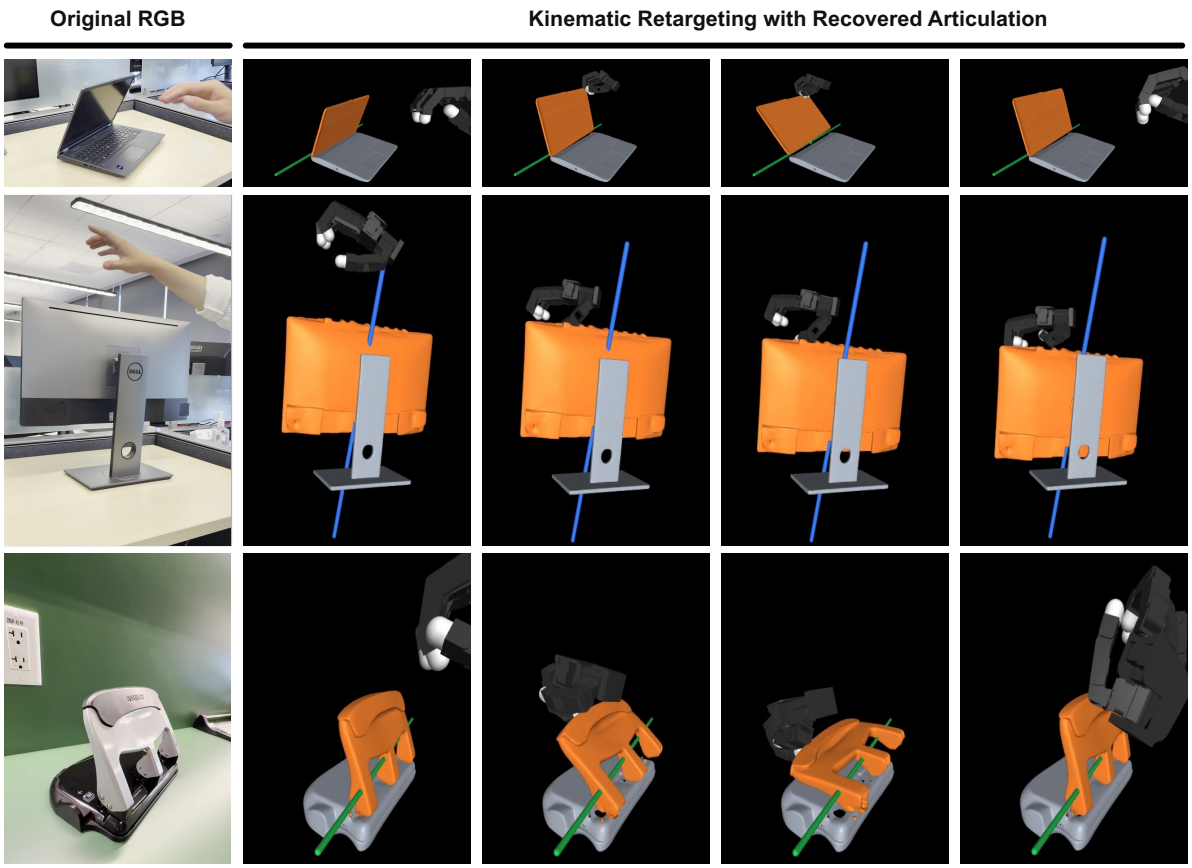

Figure 5: Interactive hand re-targeting in MuJoCo. We align the reconstructed human hand with the articulated asset and re-target its motion to a simulated robot hand. Results show that the re-targeted hand, when actuated, is able to plausibly manipulate the object.

Figure 5 shows the respective real video and the corresponding initial, intermediate, and final MuJoCo states. This evaluation directly probes the quality of the hand–object alignment: an inaccurate relative pose, scale, or trajectory would cause the simulated hand to miss the object, or attempt to move it in the wrong direction. We report normalized joint progress, final joint-state error relative to the demonstrated range. Table 2 shows that the aligned hand trajectory can reliably interact with the recovered object in simulation. These results establish the feasibility of aligning and re-targeting the recovered hand motion while preserving the contact needed to articulate the reconstructed object.

Table 2: Contact-driven interaction after re-targeting. We test whether the aligned and re-targeted hand can reproduce the demonstrated joint-state change in simulation, which is produced through contact and normalized against the demonstrated joint range.

Interaction	Joint progress (%) ↑	Final q err. (%) ↓
Laptop	63	35
Monitor	39	59
Stapler	92	4
Puncher	97	1
Mean	73	25

6 Discussion

We presented a *modular* real-to-sim framework that recovers an articulated object and aligned 3D hand trajectory from a casual monocular human video. By connecting pretrained predictions through multi-frame registration, motion-grounded joint fitting, and reprojection verification, the method produces an interactive MuJoCo asset without requiring sensor depth, prior object geometry, joint annotations, or robot demonstrations.

This work explores how far current best pre-trained vision models can take us toward recovering interactive articulated objects from human videos without the use of vision-language models. Our results show that combining visual predictions with geometric constraints and optimization can support both accurate articulation recovery and contact-driven hand replay in the settings studied here. These findings highlight the capabilities of existing vision models when their predictions are connected through the geometry and motion of the observed interaction.

The method remains limited by errors in monocular depth, tracking, segmentation, and single-image reconstruction, particularly under heavy occlusion or small joint motion. It currently handles one dominant revolute or prismatic joint and interactions in simulation that do not involve complex grasps; compound articulation, and deformable objects require richer contact and kinematic models. Looking ahead, extending our modular framework to recover more complex articulation and richer contact information from videos, together with advances in physics-based simulators, can enable dexterous manipulation across a wider range of objects.

References

[1] Ananye Agarwal, Shagun Uppal, Kenneth Shaw, and Deepak Pathak. Dexterous functional grasping. In *Conference on Robot Learning (CoRL)*, 2023.

[2] Arslan Artykov, Tom Ravaud, Corentin Sautier, and Vincent Lepetit. sim2art: Accurate articulated object modeling from a single video using synthetic training data only. *arXiv preprint arXiv:2512.07698*, 2025.

[3] Arslan Artykov, Tom Ravaud, Nicolás Violante-Grezzi, and Vincent Lepetit. Articulation in prime: Primitive-based articulated object understanding from a single casual video. *arXiv preprint arXiv:2605.18645*, 2026.

[4] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[5] Homanga Bharadhwaj and Yash Jangir. Motionforesight: Re-purposing video models for future 3d scene-flow prediction. *arXiv preprint arXiv:2607.16192*, 2026.

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

[7] Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. Reconstructing hand-object interactions in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[8] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra,

Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[9] Boyuan Chen, Tianyuan Zhang, Haoran Geng, Caiyi Zhang, Peihao Li, Kiwhan Song, William T. Freeman, Jitendra Malik, Pieter Abbeel, Russ Tedrake, Vincent Sitzmann, and Yilun Du. Large video planner enables generalizable robot control. *arXiv preprint arXiv:2512.15840*, 2025.

[10] 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*, 2026.

[11] Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J. Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jia-Wei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. SAM 3D: 3Dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7220–7232, 2026.

[12] Zicong Fan, Maria Parelli, Maria Eleni Kadoglou, Xu Chen, Muhammed Kocabas, Michael J. Black, and Otmar Hilliges. Hold: Category-agnostic 3d reconstruction of interacting hands and objects from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 494–504, 2024.

[13] Raktim Gautam Goswami, Amir Bar, David Fan, Tsung-Yen Yang, Gaoyue Zhou, Prashanth Krishnamurthy, Michael Rabbat, Farshad Khorrami, and Yann LeCun. World models for learning dexterous hand-object interactions from human videos. *arXiv preprint arXiv:2512.13644*, 2025.

[14] Lijun Guo, Haoyu Zhao, Xingyue Zhao, Rong Fu, Linghao Zhuang, Siteng Huang, Zhongyu Li, and Hua Zou. Articulat3d: Reconstructing articulated digital twins from monocular videos with geometric and motion constraints. *arXiv preprint arXiv:2603.11606*, 2026.

[15] Irmak Guzey, Haozhi Qi, Julen Urain, Changhao Wang, Jessica Yin, Krishna Bodduluri, Mike Lambeta, Lerrel Pinto, Akshara Rai, Jitendra Malik, Tingfan Wu, Akash Sharma, and Homanga Bharadhwaj. Dexterity from smart lenses: Multi-fingered robot manipulation with in-the-wild human demonstrations. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

[16] Adam W. Harley, Yang You, Xinglong Sun, Yang Zheng, Nikhil Raghuraman, Yunqi Gu, Sheldon Liang, Wen-Hsuan Chu, Achal Dave, Suya You, Rares Ambrus, Katerina Fragkiadaki, and Leonidas Guibas. AllTracker: Efficient dense point tracking at high resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5253–5262, 2025.

[17] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia 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.

[18] Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Li Fei-Fei. Pointworld: Scaling 3d world models for in-the-wild robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20765–20779, 2026.

[19] Vidhi Jain, Maria Attarian, Nikhil J. Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R. Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, Igor Gilitschenski, Yonatan Bisk, and Debidatta Dwibedi. Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers. In *Robotics: Science and Systems (RSS)*, 2024.

[20] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building digital twins of articulated objects from interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5616–5626, 2022.

[21] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision (ECCV)*, 2024.

[22] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[23] Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4d: Unified feed-forward metric 4d reconstruction. *arXiv preprint arXiv:2512.10935*, 2025.

[24] Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. In *Conference on Robot Learning (CoRL)*, 2024.

[25] Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model. *arXiv preprint arXiv:2410.13882*, 2024.

[26] Taeyeop Lee, Bowen Wen, Minjun Kang, Gyuree Kang, In So Kweon, and Kuk-Jin Yoon. Any6d: Model-free 6d pose estimation of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[27] Hongyu Li, Lingfeng Sun, Yafei Hu, Duy Ta, Jennifer Barry, George Konidaris, and Jiahui Fu. Novaflow: Zero-shot manipulation via actionable flow from generated videos. *arXiv preprint arXiv:2510.08568*, 2025.

[28] Lin Li, Haoran Feng, Zehuan Huang, Haohua Chen, Wenbo Nie, Shaohua Hou, Keqing Fan, Pan Hu, Sheng Wang, Buyu Li, and Lu Sheng. Segvigen: Repurposing 3d generative model for part segmentation. *ACM Transactions on Graphics (TOG)*, 45(4), 2026.

[29] Zizhang Li, Cheng Zhang, Zhengqin Li, Henry Howard-Jenkins, Zhaoyang Lv, Chen Geng, Jiajun Wu, Richard Newcombe, Jakob Engel, and Zhao Dong. ART: Articulated reconstruction transformer. *arXiv preprint arXiv:2512.14671*, 2025.

[30] Junbang Liang, Pavel Tokmakov, Ruoshi Liu, Sruthi Sudhakar, Paarth Shah, Rares Ambrus, and Carl Vondrick. Video generators are robot policies. *arXiv preprint arXiv:2508.00795*, 2025.

[31] Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[32] Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. Paris: Part-level reconstruction and motion analysis for articulated objects. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 352–363. IEEE, 2023.

[33] Shaowei Liu, Saurabh Gupta, and Shenlong Wang. Building rearticulable models for arbitrary 3d objects from 4d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21138–21147, 2023.

[34] Xingyu Liu, Pengfei Ren, Qi Qi, Haifeng Sun, Zirui Zhuang, Jianxin Liao, and Jingyu Wang. Clay-to-stone: Phase-wise 3d gaussian splatting for monocular articulated hand-object manipulation modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[35] Yu Liu, Baoxiong Jia, Ruijie Lu, Chuyue Gan, Huayu Chen, Junfeng Ni, Song-Chun Zhu, and Siyuan Huang. Videoartgs: Building digital twins of articulated objects from monocular video. *arXiv preprint arXiv:2509.17647*, 2025.

[36] Yumeng Liu, Xiaoxiao Long, Zemin Yang, Yuan Liu, Marc Habermann, Christian Theobalt, Yuexin Ma, and Wenping Wang. EasyHOI: Unleashing the power of large models for reconstructing hand-object interactions in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7037–7047, 2025.

[37] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

[38] Zhao Mandi, Yijia Weng, Dominik Bauer, and Shuran Song. Real2Code: Reconstruct articulated objects via code generation. *arXiv preprint arXiv:2406.08474*, 2024.

[39] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

[40] Jisu Nam, Jahyeok Koo, Soowon Son, Jaewoo Jung, Honggyu An, Junhwa Hur, and Seungryong Kim. Trackcraft3r: Repurposing video diffusion transformers for dense 3d tracking. *arXiv preprint arXiv:2605.12587*, 2026.

[41] Atsuhiro Noguchi, Umar Iqbal, Jonathan Tremblay, Tatsuya Harada, and Orazio Gallo. Watch it move: Unsupervised discovery of 3d joints for re-posing of articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3677–3687, 2022.

[42] 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.

[43] Shivansh Patel, Shraddhaa Mohan, Hanlin Mai, Unnat Jain, Svetlana Lazebnik, and Yunzhu Li. Robotic manipulation by imitating generated videos without physical demonstrations. In *International Conference on Learning Representations (ICLR)*, 2026.

[44] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[45] Weikun Peng, Jun Lv, Cewu Lu, and Manolis Savva. itaco: Interactable digital twins of articulated objects from casually captured rgbd videos. In *2026 International Conference on 3D Vision (3DV)*, pages 520–531. IEEE, 2026.

[46] 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)*, pages 12242–12254. IEEE, 2025.

[47] Xiaowen Qiu, Jincheng Yang, Yian Wang, Zhehuan Chen, Yufei Wang, Tsun-Hsuan Wang, Zhou Xian, and Chuang Gan. Articulate anymesh: Open-vocabulary 3d articulated objects modeling. *arXiv preprint arXiv:2502.02590*, 2025.

[48] Nadun Ranawaka, Josiah Wong, Wei-Lin Pai, Wei-Teng Chu, Tianyuan Dai, Masoud Moghani, Hang Yin, Yunfan Jiang, Wesley Durbano, Brandon Huynh, Yu Fang, Danfei Xu, Ruohan Zhang, Li Fei-Fei, Linxi Fan, Bowen Wen, Ajay Mandlekar, and Yuke Zhu. Simfoundry: Modular and automated scene generation for policy learning and evaluation. *arXiv preprint arXiv:2606.28276*, 2026.

[49] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration. *arXiv preprint arXiv:2008.08324*, 2020.

[50] Junyao Shi, Zhuolun Zhao, Tianyou Wang, Ian Pedroza, Amy Luo, Jie Wang, Jason Ma, and Dinesh Jayaraman. Zeromimic: Distilling robotic manipulation skills from web videos. *arXiv preprint arXiv:2503.23877*, 2025.

[51] Chaoyue Song, Jiacheng Wei, Chuan-Sheng Foo, Guosheng Lin, and Fayao Liu. REACTO: Reconstructing articulated objects from a single video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5384–5395, 2024.

[52] Rustin Soraki, Homanga Bharadhwaj, Ali Farhadi, and Roozbeh Mottaghi. Objectforesight: Predicting future 3d object trajectories from human videos. *arXiv preprint arXiv:2601.05237*, 2026.

[53] Jürgen Sturm, Cyrill Stachniss, and Wolfram Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41:477–526, 2011.

[54] Chao Tang, Anxing Xiao, Yuhong Deng, Tianrun Hu, Wenlong Dong, Hanbo Zhang, David Hsu, and Hong Zhang. Mimicfunc: Imitating tool manipulation from a single human video via functional correspondence. *arXiv preprint arXiv:2508.13534*, 2025.

[55] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. In *Conference on Robot Learning (CoRL)*, 2023.

[56] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C. Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. In *Robotics: Science and Systems (RSS)*, 2024.

[57] Zikai Wang, Zhilu Zhang, Yiqing Wang, Hui Li, and Wangmeng Zuo. ArtHOI: Taming foundation models for monocular 4d reconstruction of hand-articulated-object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[58] 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 (CVPR)*, 2024.

[59] Abdelrhman Werby, Martin Büchner, Adrian Röfer, Chenguang Huang, Wolfram Burgard, and Abhinav Valada. Articulated object estimation in the wild. In *Proceedings of the 9th Conference on Robot Learning*, pages 3828–3849. PMLR, 2025.

[60] Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Iurii Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: 3d point tracking made easy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[61] Yufei Ye, Poorvi Hebbar, Abhinav Gupta, and Shubham Tulsiani. Diffusion-guided reconstruction of everyday hand-object interaction clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[62] Yufei Ye, Yao Feng, Omid Taheri, Haiwen Feng, Shubham Tulsiani, and Michael J. Black. Predicting 4D hand trajectory from monocular videos. In *2026 International Conference on 3D Vision (3DV)*, pages 860–870. IEEE, 2026.

[63] Jinglei Zhang, Jiankang Deng, Chao Ma, and Rolandos Alexandros Potamias. Hawor: World-space hand motion reconstruction from egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[64] Mandi Zhao, Yifan Hou, Dieter Fox, Yashraj Narang, Ajay Mandlekar, and Shuran Song. DexMachina: Functional retargeting for bimanual dexterous manipulation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

Appendix

A Simulated Dataset

We use the PARIS dataset [32] to construct synthetic sequences for comparing joint estimation performance among VideoArtGS [35], Articulat3D [14], and our method. PARIS provides ground-truth joint parameters and meshes at the initial and final articulation states, with moving and static parts segmented. We interpolate between these states to generate complete articulation sequences, then render RGB images, depth maps, and segmentation masks from specified camera poses. We also obtain ground-truth point tracks by recording the trajectories of a fixed set of mesh vertices throughout each sequence.

As shown in Figure 1, we consider two camera configurations: an orbiting camera and a static camera. We use the orbiting configuration to evaluate VideoArtGS and Articulat3D under their intended camera-motion conditions. Because our method does not require camera motion, we report its joint estimation accuracy under the static-camera configuration.

B Joint Estimation

Track-based joint initialization. Given a video, per-frame part masks, camera parameters, segmented reference meshes, and 3D point trajectories of the moving part, we estimate its joint type, joint parameters, and articulation states. We select keyframes spanning diverse mask configurations within the available tracking window and reserve a subset for joint-type selection.

Let $\mathbf{x}_{i,t}$ denote the position of tracked point i at frame t, and let t_0 be the mesh reference frame. We use $\mathbf{x}_{i,t_0}$ as the reference point positions, accounting for any difference between the tracker and mesh reference frames. For each training keyframe, we estimate a rigid transformation by solving

$$(R_t, \mathbf{d}_t) = \arg \min_{R \in SO(3),\, \mathbf{d}} \sum_{i \in \mathcal{I}_t} \|R\mathbf{x}_{i,t_0} + \mathbf{d} - \mathbf{x}_{i,t}\|_2^2 . \tag{1}$$

We solve this alignment using Kabsch registration with iterative residual trimming to reduce the influence of unreliable tracks. Non-finite correspondences are excluded, and registration residuals provide confidence weights for subsequent joint fitting.

Joint parameterization and fitting. We fit both prismatic and revolute hypotheses to the estimated rigid transformations. Their motion models are

$$\mathcal{T}_{\mathrm{P}}(\mathbf{x}; q_t) = \mathbf{x} + q_t \mathbf{a}, \tag{2}$$

$$\mathcal{T}_{\mathrm{R}}(\mathbf{x}; q_t) = \mathbf{c} + \exp(q_t [\mathbf{a}]_\times)\,(\mathbf{x} - \mathbf{c}), \tag{3}$$

where $\mathbf{a}$ is a unit axis, $\mathbf{c}$ is a point on the revolute axis, and q_t denotes translation distance or rotation angle. For the prismatic hypothesis, we initialize $\mathbf{a}$ from the dominant direction of the estimated translations. For the revolute hypothesis, we initialize the axis from relative rotation vectors and recover an axis point through

$$(I - R_t)\mathbf{c} \approx \mathbf{d}_t. \tag{4}$$

We remove the axis-point ambiguity by constraining $\mathbf{c}$ to the plane through the reference point-cloud centroid perpendicular to $\mathbf{a}$. Random-subset hypothesis generation and robust residual scoring

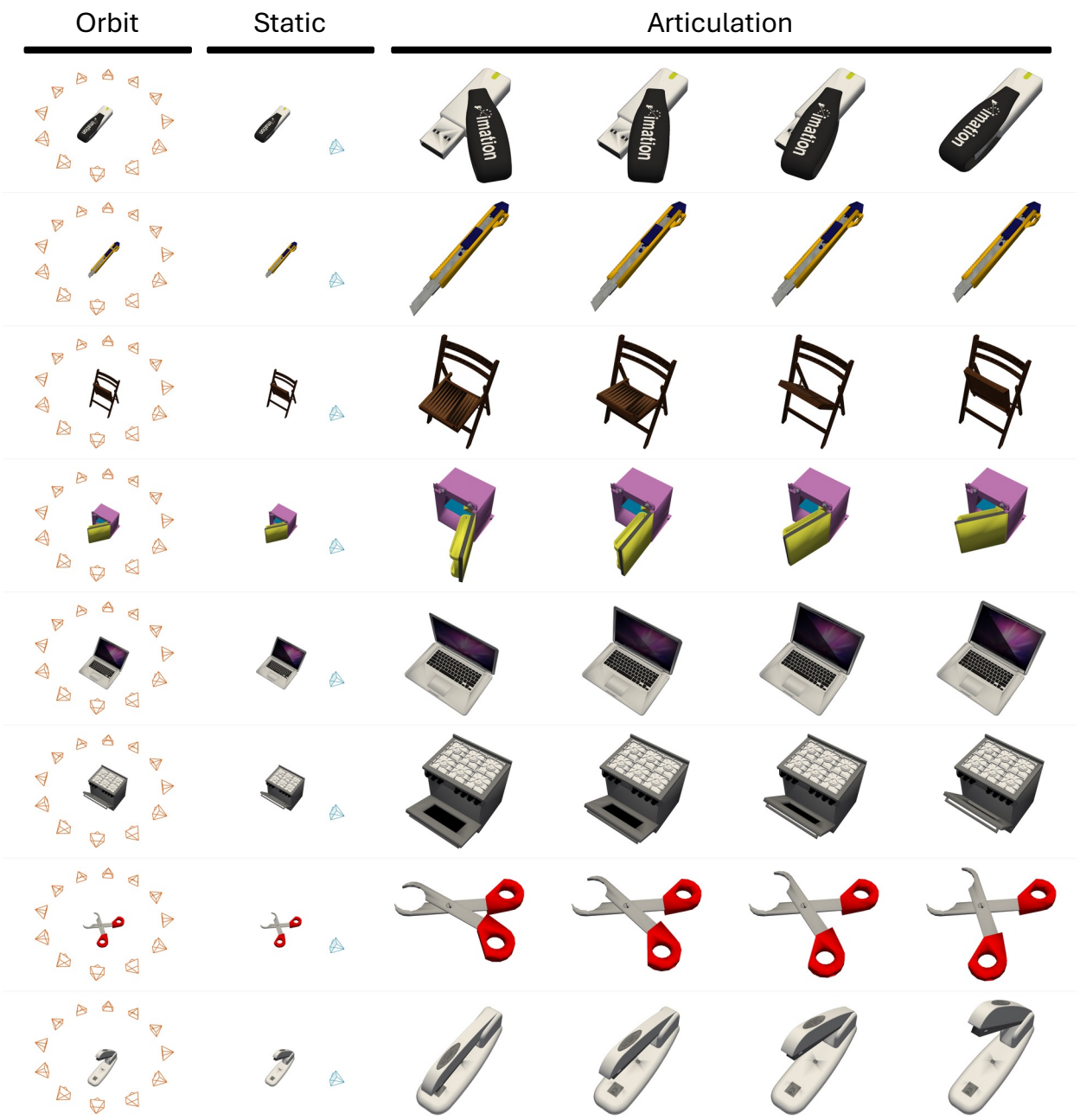

Figure 1: Synthetic dataset for benchmarking the joint estimation method. There are two different camera configurations: orbit camera and static camera. The actual articulation sequences are interpolated between the given initial and final states of the moving part.

identify an initial consensus set. We then refine each hypothesis using confidence-weighted Huber least squares on rotation and translation residuals, normalizing translation by the reference point cloud's bounding-box diagonal.

Image-space refinement. Starting from the provided object pose, we align the reference meshes by optimizing a shared rigid transformation and anisotropic scale against the static-part masks. We extract DINOv2 features, compress them using a shared PCA projection, and attach reference-frame descriptors to visible mesh vertices. These descriptors remain fixed on the mesh during articulation.

For each initialized joint hypothesis, we render the articulated mesh and compare its silhouette and features with the observations:

$$\mathcal{L}^t_{\text{img}} = \lambda_m \mathcal{L}^t_{\text{mask}} + \lambda_b \mathcal{L}^t_{\text{boundary}} + \lambda_f \mathcal{L}^t_{\text{feature}}. \tag{5}$$

The mask term penalizes missing foreground and rendered foreground outside the observed mask, with a reduced penalty on the latter. The boundary term measures distances from rendered silhouette edges to the observed mask boundary. The feature term measures cosine distance between rendered and observed descriptors over their valid foreground overlap.

We optimize the articulation states first, then jointly refine the states and joint parameters. A final stage additionally optimizes a shared rigid transformation and isotropic scale correction of both meshes:

$$\mathcal{L} = \frac{1}{|\mathcal{K}_{\text{tr}}|} \sum_{t \in \mathcal{K}_{\text{tr}}} \left(\mathcal{L}^t_{\text{img,mov}} + \tfrac{1}{2} \mathcal{L}^t_{\text{img,stat}} \right) + \lambda_s \mathcal{L}_{\text{smooth}} + \lambda_{\text{scale}} (\log s)^2, \tag{6}$$

where s is the scale correction. The smoothness term penalizes second differences of the temporally ordered keyframe states:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{K-2} \sum_{i=2}^{K-1} \left(q_{i+1} - 2q_i + q_{i-1} \right)^2. \tag{7}$$

The reference state is fixed to zero. The 3D trajectories are used only for initialization and do not contribute residuals to this image-space refinement.