Articulated Object Estimation in the Wild

Abdelrhman Werby[1,2*], Martin Büchner[1*], Adrian Röfer[1*], Chenguang Huang[3], Wolfram Burgard[3], Abhinav Valada[1]

[1]University of Freiburg [2]University of Stuttgart [3]University of Technology Nuremberg

Abstract: Understanding the 3D motion of articulated objects is essential in robotic scene understanding, mobile manipulation, and motion planning. Prior methods for articulation estimation have primarily focused on controlled settings, assuming either fixed camera viewpoints or direct observations of various object states, which tend to fail in more realistic unconstrained environments. In contrast, humans effortlessly infer articulation by watching others manipulate objects. Inspired by this, we introduce *ArtiPoint*, a novel estimation framework that can infer articulated object models under dynamic camera motion and partial observability. By combining deep point tracking with a factor graph optimization framework, *ArtiPoint* robustly estimates articulated part trajectories and articulation axes directly from raw RGB-D videos. To foster future research in this domain, we introduce *Arti4D*, the first ego-centric in-the-wild dataset that captures articulated object interactions at a scene level, accompanied by articulation labels and ground-truth camera poses. We benchmark *ArtiPoint* against a range of classical and learning-based baselines, demonstrating its superior performance on *Arti4D*. We make code and *Arti4D* publicly available at `https://artipoint.cs.uni-freiburg.de`.

Keywords: Articulated Object Estimation, 3D Scene Understanding, Interactive Perception

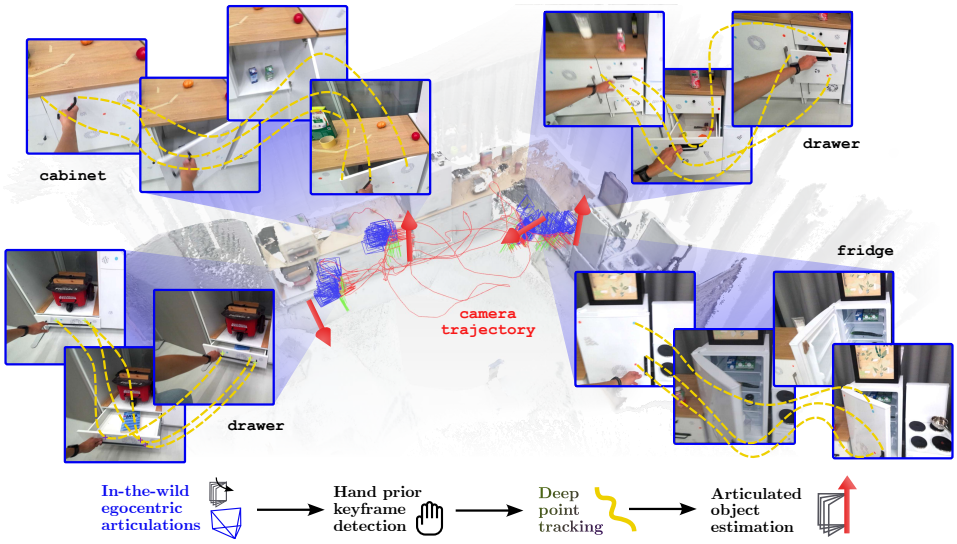


Figure 1: We present *ArtiPoint*, our novel approach to articulated object estimation in the wild. ArtiPoint makes use of deep point tracking and factor graph optimization. It does not rely on costly scene-wise optimization or deep models that are prone to overfit. We evaluate our approach on the Arti4D dataset, a novel scene-level articulated object dataset that is recorded with a moving, ego-centric camera, compared to previous benchmarks that are limited to isolated objects and static camera poses.

[*]These three authors contributed equally to this work. Work done while at University of Freiburg.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

1 Introduction

Robotic manipulation is concerned with moving external objects to achieve a task-relevant goal. While recent approaches have made tremendous progress in manipulating unconstrained objects [1, 2, 3], manipulating constrained *articulated* objects remains challenging in model-free settings that rely on affordances [4, 5]. On the other hand, model-based approaches have thoroughly investigated manipulating articulated objects, demonstrating both manipulation success and transferability to novel objects [6, 7, 8, 9, 10]. However, the aforementioned approaches focus on offline learning of predictive models in curated settings, where only the articulated object itself moves [11, 12]. If we consider deploying a robot in a novel environment, these requirements are not met [4, 13, 14]. As a consequence, most approaches for estimating articulated objects and estimating motion models fail when operating *in-the-wild*, i. e., when faced with dynamic camera poses, occlusions, and significant clutter, where objects are not isolated and interactions are less constrained. While humans acquire manipulation skills through observing others interacting with various entities such as articulated objects [15], endowing robotic systems with those capabilities remains an open problem. Ultimately, this limits the transfer of insights from articulated object estimation [6, 9] to the problem of human-to-robot imitation in everyday robotics.

In this work, we propose to address this limitation by exploiting interaction priors. As humans manipulate environments with their hands, we aim to observe those interactions. First, we extract interaction segments containing object interactions. Secondly, we obtain point trajectories within regions close to the detected hand, which describe both the static and dynamic parts of the observed scene. We leverage recent state-of-the-art models in any-point tracking [16, 17], which yields point trajectories throughout whole interaction segments, including an estimated visibility score. Third, we lift those point trajectories into 3D using depth measurements and compensate for camera motion using accurate camera odometry. We separate all static 3D trajectories with a negligible length from dynamic trajectories, which potentially represent the motion of dynamic objects. This enables us to estimate the motion of the moving part triggered through hand interaction. Finally, given the acquired point trajectories, we estimate the underlying articulation model, classify the joint type (prismatic or revolute), and globally register the model in the observed scene.

As we address the novel task of estimating articulated object motion *in-the-wild*, we introduce Arti4D, the first ego-centric in-the-wild dataset for scene-level articulated object manipulation from human demonstrations. It comprises 45 RGB-D sequences across four diverse scenes, capturing 414 human-object interactions under dynamic camera motion, occlusions, and uncurated environments ranging from kitchen scenes to robot lab rooms. The dataset was recorded by a human operator holding an RGB-D camera while actively exploring scenes and interacting with articulated objects. Unlike prior datasets, Arti4D includes $\mathfrak{se}(3)$-labeled ground-truth articulation axes, interaction windows, difficulty annotations, reconstructed scenes, and accurate camera pose ground truth. It offers a challenging benchmark not only for articulated object estimation but also for visual odometry and SLAM research in complex, real-world settings.

Concretely, we make the following contributions:

1. We present a novel articulated object estimation framework called *ArtiPoint* that operates on deep point trajectories, enabling robust deployment in uncontrolled settings.

2. We introduce *Arti4D*, the first real-world *in-the-wild* dataset of articulated object interactions providing human exploration and ego-centric articulation demonstrations on a scene-level. The dataset includes articulation axes labels and 3D ground truth camera poses.

3. We compare our method against a set of classical and deep articulation model estimation pipelines and present extensive ablations.

4. We publish code, data, and model predictions of our method at `https://artipoint.cs.uni-freiburg.de`.

2 Related Work

In the following, we review previous work in articulated object estimation and review recent developments in any-point tracking.

Probabilistic Methods: Early work in articulation model estimation often employed probabilistic formulations to infer the relationships among articulated object parts. Sturm *et al.* [18, 6] proposed a probabilistic framework that infers the articulation model from 6D trajectories of object parts. While the initial work used marker-based tracking to obtain these trajectories, subsequent work [19, 20] extended these frameworks to handle sparse (markerless) objects by extracting visual features, such as SURF [21] or by detecting rectangles in dense-depth images [22].

Deep-Learning-based Methods: Recent advancements in deep learning diversify the articulation estimation methods. By consuming a sequence of depth images with a fixed viewing pose, ScrewNet [23] enables category-agnostic articulation motion prediction in an end-to-end manner. A follow-up work, DUST-net [24] further estimates the uncertainty of the motion to introduce interpretability. While these works take a depth image sequence as input, ANCSH [25] predicts articulation joint parameters and state for unseen instances using a depth image, assuming prior knowledge of the instance category. Other works utilize neural networks as a part of their pipelines for articulation motion prediction. While Heppert *et al.* [26] formulate a two-stage pipeline consisting of a learning-based part tracking stage and a factor graph optimization stage, Buchanan *et al.* [9] combine the learning-based method with interactive perception, leveraging a neural network to propose an initial estimation of the articulation motion and interactive perception for prediction refinement. FormNet [27] predicted the motion field residual and the part connectedness prior to post-processing the final prediction.

3D Reconstruction of Articulated Objects: Reconstructing articulated objects in 3D is more challenging than reconstructing rigid objects, as it requires modeling the articulation between their constituent parts [28]. Early works [29, 30] leverage structure-from-motion to reconstruct the articulated object at different configurations, followed by post-processing techniques to estimate joint axes and types. More recent works inspired by Gaussian splatting offer a smooth and continuous representation for complex shapes. For instance, ArtGS [12] utilizes 3D Gaussians to jointly optimize for canonical Gaussians and the articulation model. Ditto [31] is a neural network-based method for estimating the articulation model and the 3D geometry of an articulated object, given point cloud observations before and after interaction. PARIS [11] takes multiview images at different articulation states and jointly estimates the articulation parameters and the 3D geometry of the object.

Tracking Any Point (TAP): Tracking points is a key problem in various computer vision and robotics tasks, such as 3D reconstruction, video analysis, and object tracking. Early works relied on sparse feature extractors, such as SIFT [32], SURF [21], and GFTT [33], as well as optical flow. Tracking any point as a task was introduced in PIP [34], where the goal was to track pixel locations across entire sequences. TAP-Vid [35] enhances the problem formulation by providing a benchmark and introducing TAP-Net. Subsequent works [16, 36, 37, 38] improved performance while Karaev *et al.* [17, 39] introduce joint tracking of points to account for their dependencies.

In this work, we address the novel problem of estimating articulations from mobile observations *in-the-wild*. We connect deep any-point tracking [39] to the estimation approach of Buchanan *et al.* [9] to form a holistic approach to inferring the parameters of multiple articulated objects in extensive, interactive environments.

3 Approach

The goal of *ArtiPoint* is to estimate the underlying motion models of articulated objects given posed RGB-D observations of human interactions with such objects. To support this goal, we exploit as a prior that humans typically use their hands to interact with objects. Our approach consists of four stages, which we visualize in Fig. 2: First, we extract segments that contain hand-object interactions. We then sample 2D keypoints near the hand and track them throughout the segment to capture the object in motion. To obtain 3D motion, we lift the 2D tracks to 3D, compensate for camera motion,

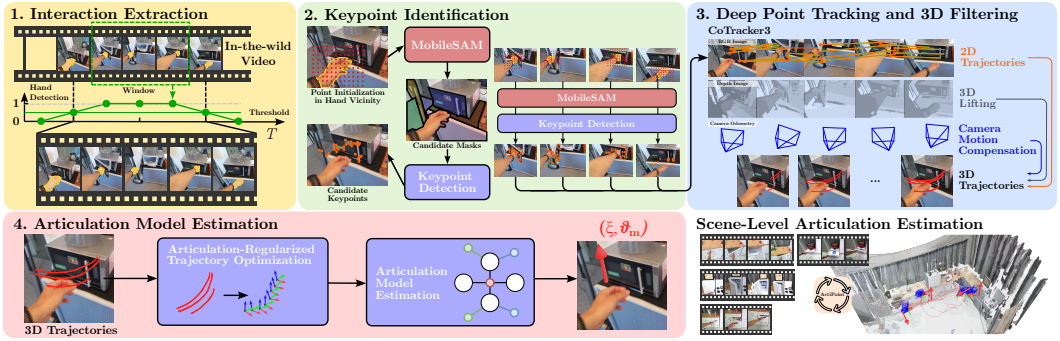


Figure 2: Overview of our method: We take an ego-centric RGB-D video as input and employ hand tracking as a trigger signal to identify interaction segments (top left). We uniformly sample points around the hand masks and prompt a class-agnostic instance segmentation model (MobileSAM [40]), which yields object masks in the immediate vicinity that may be undergoing articulation. Given those masks, we detect stable keypoints (bottom left) that are fed into an any-point tracking model (CoTracker3 [39]) in order to obtain point trajectories throughout each entire articulation segment (top right). Finally, we estimate the underlying articulation model of the object through a factor graph formulation that operates on the obtained point trajectories (bottom right).

filter out static points in 3D, and apply trajectory smoothing in our third stage to mitigate noise. Finally, we input refined 3D tracks into a factor graph framework to jointly estimate the trajectory of the object parts and the parameters of the articulation motion model.

3.1 Extraction of Interaction Intervals

Given a scene level ego-centric RGB-D sequence $\{I_t, D_t\}_{t=1}^{T}$ of a human operating and interacting with articulated objects in the scene, we extract interaction segments $S = \left\{(t_{start}^{(n)}, t_{end}^{(n)})\right\}_{n=1}^{N}$ containing frames of interactions. We use a preexisting hand segmentation model [41] to extract a hand mask H_t from each RGB frame I_t, and assign a binary label $d_t \in \{0, 1\}$ to the frame based on the hand visibility. To avoid irregular false positives/negatives, we compute the moving average $\bar{d}_t$ for the last w_h frames as $\bar{d}_t = \frac{1}{w_h}\sum_{i=t-w_h}^{t} d_i$. A frame is identified if the smoothed detection $\bar{d}_t$ exceeds τ_h, initiating a segment s_n at time $t_{start} = t$. The segment continues as long as $\bar{d}_t \geq \tau_h$, and terminates at $t_{end} = t$ when $\bar{d}_t$ drops below this threshold. Finally, the extracted segments S are filtered based on their duration, ensuring that $T_{min} \leq t_{end}^{(i)} - t_{start}^{(i)} \leq T_{max}$, where T_{min} and T_{max} represent the minimum and maximum allowed segment lengths, respectively.

3.2 Deep Point Tracking

For each obtained segment $s_n \in S$ and hand masks $\{H_t\}_{t=t_{start}^{(n)}}^{t_{end}^{(n)}}$ associated with a frame within the segment, we uniformly sample points around the hand. We use the sampled points as prompts for the class-agnostic instance segmentation model [40] to identify masks of potential articulated objects near the hand for each frame. Given the resulting K objects' 2D masks $\{O_k^t\}_{k=1}^{K}$ in frame t, our goal is to select F query points $q_t \in \mathbb{R}^{F\times 2}$ that lie on the aforementioned masks and allow tracking throughout the entire segment. We achieve this by extracting GFTT keypoints via the lightweight Shi-Tomasi method [33] within each object mask in frame t. By concatenating all selected query points for a segment s_n, we obtain a set of query points $Q_i \in \mathbb{R}^{K\times F\times 2}$ where K represents the number of keyframes selected from the segment s_n. Subsequently, we employ CoTracker3 [39] to track keypoints $\mathcal{X}_n \in \mathbb{R}^{T_n\times F\times 2}$ and obtain point visibilites $\mathcal{V}_i \in \{0, 1\}^{T_n\times F}$ of the query points across all T_n frames of each segment s_n.

3.3 3D Track Estimation and Filtering

Given the resulting 2D trajectory of query points X_n for segment s_n, first, we lift the trajectory to 3D using the corresponding depth frame D_t and camera intrinsics $\mathbf{K}$, and we filter out points

with invalid depth values. This results in a set of 3D tracks $\mathcal{P}_n \in \mathbb{R}^{T_n \times F \times 3}$ and a corresponding visibility mask $\mathcal{V}_n \in \{0,1\}^{T_n \times F}$ indicating the validity of each 3D point at each time step. Second, to compensate for camera motion and obtain 3D trajectories in global coordinates, we transform all the 3D point tracks within each segment to the global frame, assuming accurate camera odometry. Third, since some points may lie on static objects, we filter out these static points by computing the positional variance of each 3D track within the segment. Tracks with a variance below a certain percentile threshold σ_{static} are considered static and are discarded. Fourth, points may undergo occlusion during the interaction, potentially leading to unreliable point tracks. Therefore, we filter out tracks that are occluded for more than a specified percentage $\sigma_{reliable}$ of the frames within each segment. Lifting the 2D trajectory to 3D introduces high-frequency temporal jitter in the resulting 3D trajectory. We employ trajectory smoothing via minimizing a cost function (Eq. (1)) penalizing spikes in acceleration and high velocities while preserving fidelity

$$E(\mathbf{p}) = \sum_{t=1}^{T} v_t ||\mathbf{p}_t - \hat{\mathbf{p}}_t||^2 + \lambda_{vel} ||\mathbf{p}_t - \mathbf{p}_{t-1}||^2 + \lambda_{jerk} ||\mathbf{p}_t - 3\mathbf{p}_{t-1} + 3\mathbf{p}_{t-2} - \mathbf{p}_{t-3}||^2, \quad (1)$$

where the first term promotes closeness between smoothed points $\hat{p}_t$ and observed points p_t weighted by the point visibility v_t. The second term regularizes significant velocity changes weighted by λ_{vel} whereas the third term minimizes sudden changes in acceleration weighted by λ_{jerk}.

3.4 Exploiting the Articulation Prior

The most recent works examining articulation model estimation [23, 26, 9] opt for representing 1-DoF articulations as an articulation parameter $\xi = \langle \omega, v \rangle \in \mathfrak{se}(3) \subseteq \mathbb{R}^6$. Scaling an element $\xi \in \mathfrak{se}(3)$ with a configurational position θ is sufficient to represent any rigid-body transformation which can be decomposed into a proportional linear translation and rotation. Such a transformation $\xi\theta$ can be converted to the equivalent, non-linear $SE(3)$ rigid-body transform using the *exponential map* $\exp : \mathfrak{se}(3) \to SE(3)$, while the inverse of the exponential map is referred to as the *log map* $\log : SE(3) \to \mathfrak{se}(3)$ [42]. The advantage of this representation is its ability to represent diverse articulations, i. e. prismatic, revolute, and screw joints, while also being differentiable which makes them suitable for optimization.

In this work, we employ the factor graph formulation of Buchanan *et al.* [9] to estimate the articulation model which best fits an extracted point trajectory. However, this estimator operates on a sequence of pose observations between two moving parts, not directly on points of these parts. Thus, we extract poses as follows: Given our extracted points $\mathcal{P}_n$ and visibility masks $\mathcal{V}_n$, we interpret these as observation $\mathcal{Z}$ as

$$\mathcal{Z} = \left\{ P_m = \left\{ \left\langle p_f^t, p_f^{t+\epsilon} \right\rangle \;\middle|\; \mathcal{V}_{n,f}^t \wedge \mathcal{V}_{n,f}^{t+\epsilon} \right\} \right\}_{m=1}^{M}, \quad (2)$$

where $p_f^t, p_f^{t+\epsilon}$ are two observations of point f at times t and $t+\epsilon$ taken from $\mathcal{P}_n$. Extracting pose trajectories from these observations is equivalent to identifying the transformations ${}^t\mathbf{T}_{t+\epsilon,m}$ minimizing

$${}^t\mathbf{T}^*{}_{t+\epsilon,m} = \min_{{}^t\mathbf{T}_{t+\epsilon,m}} \sum_{j=1}^{|P_m|} \left\| p_j^{t+\epsilon} - {}^t\mathbf{T}_{t+\epsilon,m} \cdot p_j^t \right\|, \quad (3)$$

where $\langle p_j^t, p_j^{t+\epsilon} \rangle \in P_m$. These local transformations can be integrated to yield poses ${}^W\mathbf{T}_m$ in the global frame of reference as by recursion: We define ${}^W\mathbf{T}_0 = {}^W\mathbf{T}_A$ and build on this root with ${}^W\mathbf{T}_m = {}^W\mathbf{T}_{m-1} \cdot {}^{m-1}\mathbf{T}_m$ where ${}^W\mathbf{T}_A$ describes a global offset of the local trajectory. This local offset can reasonably be obtained as the mean of $\mathcal{P}_n^1$. In our approach, we propose exploiting the context of our problem, an extend the estimation in Eq. (3) to be *articulation-regularized* across the entire motion. We do so, by introducing a shared base twist $\hat{\xi}$ for the entire path as

$$\hat{\xi}^*, \theta_1^*, \ldots, \theta_M^* = \min_{\hat{\xi}, \theta_1, \ldots, \theta_M} \sum_{m=1}^{M} \sum_{j=1}^{|P_m|} \left\| p_j^{t+\epsilon} - \exp(\hat{\xi}\theta_m) \cdot p_j^t \right\|, \quad (4)$$

where ${}^{t}\mathbf{T}_{t+\epsilon}$ can finally be extracted as ${}^{t}\mathbf{T}_{t+\epsilon,m} = \exp(\hat{\xi}^{*}\theta_{m}^{*})$. As this formulation can be interpreted as a factor graph, we use GTSAM [43] to solve for it.

Finally, we extract the relative poses required by the articulation estimator [9], as ${}^{A}\mathbf{T}_{m} = {}^{W}\mathbf{T}_{A}{}^{-1} \cdot {}^{W}\mathbf{T}_{m}$ which is similar to the technique described in [9], where the authors use the initial frame of the articulation as the static frame. From this data the estimator then determines ξ_{n}^{*} for segment s_{n}.

4 Arti4D Dataset

We present the *Arti4D* dataset, which, to the best of our knowledge, is the first ego-centric *in-the-wild* human demonstration dataset capturing scene-level articulated object manipulation. With this dataset, we aim to create a foundation for studying the problem of *circumstantial articulated object estimation from human demonstrations*. In this problem, a human collects videos of interactions with multiple objects. The aim is to detect all interactions and estimate the articulation parameters of all manipulated objects. Arti4D consists of 45 egocentric RGB-D sequences across four distinct scenes (`RR080`, `DR080`, `RH078`, and `RH201`), featuring 414 human-object interactions recorded *in-the-wild*.

Unlike previous datasets such as *PARIS* [11], *DTA-Multi* [44], or *ArtGS-Multi* [12], the in-the-wild character presents novel challenges: the camera poses are dynamically changing as the scene is explored, the articulated objects are partially occluded throughout the interaction, and the objects are not isolated, i. e., are an integrated part of the larger environment. We present exemplary interactions and the associated camera trajectory in Fig. 1. In this manner, our dataset markedly differentiates itself from existing datasets in the domains of articulated object estimation [11, 45, 46] and 3D scene understanding [47, 48, 49, 50, 51].

In tandem with the sequences, our dataset provides ground-truth axis labels for all articulations in the scenes as $\mathfrak{se}(3)$ points, ground-truth temporal interaction segments, as well as difficulty ratings of all interactions ranked as either `EASY` or `HARD`. This is based on the level of hand visibility throughout each interaction, whether an object exhibits a reasonable number of depth measurements, whether the articulating hand is fully retrieved in between articulations, and whether large extents of the object are occluded. Furthermore, we provide cm-accurate ground-truth camera poses, obtained via external tracking, to ease the task of estimating articulations from dynamic camera observations. In addition, we believe that Arti4D constitutes a challenging benchmark for the tasks of visual odometry and simultaneous localization and mapping. Given that the articulations cover significant extents of the field of view, finding stable correspondences becomes difficult [52], ultimately complicating odometry estimation.

5 Experimental Results

In the following, we present our experimental findings by comparing against two sets of capable baselines, demonstrating the performance of ArtiPoint on various splits of Arti4D, and lastly, ablating key components of our proposed pipeline. As introduced in Sec. 4, we employ the Arti4D dataset to evaluate the performance of estimating articulation models *in-the-wild*. The reported results are averaged across all object interactions recorded across 45 RGB-D sequences stemming from four diverse environments: `RR080`, `DR080`, `RH078`, and `RH201`.

Baselines: We consider two sets of baselines. The first set consists of the deep learning-based approach Ditto [53] and the Gaussian splatting-based method ArtGS [12]. Both methods are tailored towards semi-static variations of articulation states of isolated objects. As this is somewhat different from the in-the-wild paradigm introduced as part of the Arti4D dataset, we adapted these methods by masking out hands and selecting a representative number of frames from before, during, and after the interaction (Sec. 3.1) from which to reconstruct the motion model. However, both Ditto and ArtGS are prone to occlusions and variations in camera poses as they aim to identify correspondences among various articulation states.

Table 1: Overall comparison of classical and deep articulation estimation methods on the *Arti4D* dataset. We report the axis-angle errors θ_{err}, the positional errors d_{L2} of revolute joints, and the accuracy of joint type detection. We note that *ArtiPoint* achieves the lowest error in estimating articulation parameters.

Method	Prismatic joints		Revolute joints		Type accuracy [%]	
	θ_{err}[deg]	d_{L2}[m]	θ_{err}[deg]	d_{L2}[m]	Prismatic	Revolute
ArtGS	52.29	–	56.82	0.25	**1.00**	0.00
Ditto	55.03	–	60.89	0.29	0.00	**1.00**
Sturm *et al.* [18] w/ bbox	63.15	–	57.98	1.34	0.00	1.00
Buchanan [9] w/ bbox	49.67	–	58.82	0.22	0.00	1.00
ArtiPoint w/ Sturm *et al.* [18]	26.85	–	18.32	1.40	0.70	0.96
ArtiPoint w/ indep. transforms	15.60	–	18.61	0.15	0.65	0.96
ArtiPoint (ours)	**14.54**	–	**17.14**	**0.07**	0.68	0.98

Our second set of baselines consists of two factor graph-based articulation estimation pipelines. As Sturm *et al.* [18, 6, 22] only provides a back-end for estimating the articulation model, we employ the ArtiPoint frontend but fit oriented bounding boxes to all detected objects based on the masks extracted by MobileSAM [40]. Using the 3D bounding boxes of the objects, we derive part poses across all frames and feed these into the estimation framework of Sturm *et al.* [18]. Lastly, we also employ our front-end pipeline to provide object poses ${}^{W}\mathbf{T}_{A}$ from any-point tracking for Sturm *et al.* [18].

Metrics: In order to evaluate estimated articulation models, we match predicted interaction segments against ground truth segments under an $IoU > 0.5$. Given this matching, we employ the following metrics to quantify articulation estimation performance: 1. The angular error $\theta_{err} = \arccos(|\hat{\mathbf{a}} \cdot \mathbf{a}|)$ evaluated between unit-length ground truth and predicted axes, and 2., the Euclidean distance d_{L2} between predicted and ground truth axes (revolute joints only), and the accuracy of predicted joint types.

Implementation Details: For our interaction extraction module (Sec. 3.1) we use $w_h = 6$, $T_{min} = 30$, and $T_{max} = 90$. For our 3D track estimation and filtering (Sec. 3.3) we choose a track reliability threshold $\sigma_{reliable} = 0.5$, and for the optimization-based smoothing we choose $\lambda_{vel} = 0.5$, $\lambda_{jerk} = 5$. Our approach requires 20 GB of RAM and 16GB of VRAM.

5.1 Quantitative Results on Arti4D

We report an overall comparison on Arti4D in Tab. 1. We note that ArtGS [12] and Ditto [31] perform poorly in both estimating and classifying joints, which is primarily due to partial observability throughout interactions. In addition, we observe that the non-isolatedness of objects complicates estimation accuracy in the case of ArtGS and Ditto. Using the estimator proposed by Sturm *et al.* [18] with a bounding box-based front-end leads to worse results compared to ArtGS [12] and Ditto [31]. Furthermore, we find that the estimator of Buchanan *et al.* [9] outperforms Sturm *et al.* [18], making use of a bounding box-based frontend approach as proposed in [54]. Ultimately, we find that the pose trajectories obtained using the independent-transform estimator operating on a frame-pair basis (Eq. (3)) yield higher rotational and translational errors compared to ArtiPoint, Thus, the proposed articulation-regularized estimator (Eq. (4)) increases articulation model estimation accuracy. We note that our trajectory estimator also benefits the estimation approach of Sturm *et al.*. We also study the performance of the different approaches across easy and difficult interactions in Tab. 2. We find that ArtiPoint performs best in both cases in parameter estimation.

5.2 Qualitative Results on Arti4D

In addition to the quantitative comparisons, we depict predictions on two drawers (prismatic) and a revolute joint of a storage case in Fig. 3. The two drawers are observed under a small number of keypoints whereas the case exhibits frequently missing depth measurements. Nonetheless, ArtiPoint is able to estimate the underlying motion model robustly.

Table 2: Difficulty-level comparison of estimation methods on *Arti4D*.

Diff.	Method	Prismatic joints θ_{err}[deg]	Prismatic joints d_{L2}[m]	Revolute joints θ_{err}[deg]	Revolute joints d_{L2}[m]	Type accuracy [%] Prismatic	Type accuracy [%] Revolute
EASY	ArtGS	53.65	-	54.06	0.30	**1.00**	0.00
	Ditto	54.46	-	51.83	0.37	0.00	**1.00**
	ArtiPoint w/ Sturm *et al.* [18]	20.24	-	13.11	0.96	0.82	0.95
	ArtiPoint	**12.63**	-	**11.66**	**0.05**	0.75	0.95
HARD	ArtGS	50.12	-	59.48	0.20	**1.00**	0.00
	Ditto	56.09	-	67.11	0.23	0.00	**1.00**
	ArtiPoint w/ Sturm *et al.* [18]	38.94	-	22.10	1.80	0.48	0.96
	ArtiPoint	**18.03**	-	**21.10**	**0.87**	0.54	1.00

Table 3: Ablations of key components of our approach on the *Arti4D* dataset. As before, we report the same set of metrics as in Tab. 1.

	Prismatic joints θ_{err} [deg]	Prismatic joints d_{L2} [cm]	Revolute joints θ_{err} [deg]	Revolute joints d_{L2} [cm]	Type Accuracy [%] Prismatic	Type Accuracy [%] Revolute
w/ ORB keypoints	19.22	-	25.61	0.10	0.55	0.92
w/ keyframe stride 4	14.67	-	20.48	0.08	0.67	0.94
w/o trajectory smoothing	16.24	-	19.38	0.11	0.58	0.92
w/ unreliable tracks	14.61	-	18.86	0.09	0.67	0.93
ArtiPoint (ours)	**14.54**	–	**17.14**	**0.07**	0.68	0.98

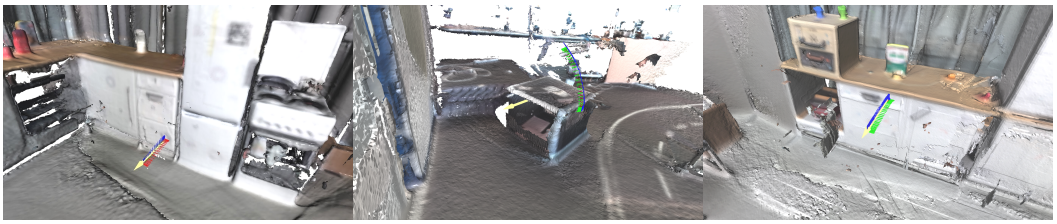


Figure 3: Qualitative results on Arti4D: Estimated joint axis and pose trajectory of a drawer in a kitchen scene (RH201), a revolute joint of a storage case (RH078), and another drawer in RH201.

5.3 Ablation Study

To further understand the contribution of different components of the ArtiPoint pipeline, we conducted an ablation study on the Arti4D dataset. From Tab. 3, we can conclude that using ORB keypoints [55] leads to worse results across all metrics. Increasing the keyframe stride to 4 in Sec. 3.2 reduces the computation overhead but also reduces estimation performance. Furthermore, deactivating the trajectory smoothing impacts the accuracy of both estimated revolute and rotational joints, as well as reducing the overall joint type prediction accuracy. Finally, incorporating unreliable tracks or highly occluded tracks slightly decreases the predicted axis quality, proving the robustness of our proposed estimator.

6 Conclusion

We presented ArtiPoint, a novel framework for estimating articulation motion models from human object interactions *in-the-wild*. Unlike prior works operating in controlled settings involving fixed camera poses, isolated objects, and full observability, we employ deep any-point tracking and factor graph optimization to infer articulation models from ego-centric human demonstrations. As part of our work, we introduced Arti4D, the first real-world, ego-centric articulation demonstration dataset that includes interaction segments, axis labels, and ground truth camera odometry. ArtiPoint demonstrates robust performance while outperforming both classical and previous deep and object-wise rendering methods in articulated object estimation. To foster further research in articulation understanding in realistic, unconstrained settings, we make code, model predictions, and dataset publicly available.

7 Limitations

Our proposed ArtiPoint framework relies on key point tracking, which inherently requires the presence of somewhat distinct features that allow tracking. State-of-the-art any-point tracking methods as discussed in Sec. 2 show strong robustness even under feature-sparse conditions. However, when objects do not yield dependable depth estimates, due to properties such as color, surface paint, or reflections, tracking their motion becomes particularly challenging. To address this limitation, future work will explore the integration of inpainting methods informed by monocular depth estimation. Additionally, our system is currently limited to relatively simple articulation models that involve only two-body kinematics, making it infeasible to estimate more complex articulations. Another limitation of our approach is the requirement for a trigger signal to temporally segment the RGB-D sequence. In our implementation, this trigger takes the form of human hand detection, which is chosen due to its reliability in indicating articulations. Alternative methods in action recognition, whether in egocentric or third-person views, often lack robustness or require extensive inference times.

Acknowledgments

This work was funded by the BrainLinks-BrainTools center of the University of Freiburg and an academic grant from NVIDIA.

References

[1] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. *Conference on Robot Learning*, 2024.

[2] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *International Journal of Robotics Research*, 2024.

[3] E. Chisari, N. Heppert, M. Argus, T. Welschehold, T. Brox, and A. Valada. Learning robotic manipulation policies from point clouds with conditional flow matching. *Conference on Robot Learning*, 2024.

[4] A. Gupta, M. Zhang, R. Sathua, and S. Gupta. Opening cabinets and drawers in the real world using a commodity mobile manipulator. *Robotics: Science and Systems*, 2025.

[5] I. Nematollahi, E. Rosete-Beas, A. Röfer, T. Welschehold, A. Valada, and W. Burgard. Robot skill adaptation via soft actor-critic gaussian mixture models. In *IEEE International Conference on Robotics and Automation*, pages 8651–8657. IEEE, 2022.

[6] J. Sturm, C. Plagemann, and W. Burgard. Unsupervised body scheme learning through self-perception. In *IEEE International Conference on Robotics and Automation*, pages 3328–3333, 2008.

[7] H. Zhang, B. Eisner, and D. Held. Flowbot++: Learning generalized articulated objects manipulation via articulation projection. *Conference on Robot Learning*, 2023.

[8] D. Honerkamp, T. Welschehold, and A. Valada. N^2 m^2: Learning navigation for arbitrary mobile manipulation motions in unseen and dynamic environments. *IEEE Transactions on Robotics*, 39 (5):3601–3619, 2023.

[9] R. Buchanan, A. Röfer, J. Moura, A. Valada, and S. Vijayakumar. Online estimation of articulated objects with factor graphs using vision and proprioceptive sensing. In *IEEE International Conference on Robotics and Automation*, pages 16111–16117, 2024.

[10] F. Schmalstieg, D. Honerkamp, T. Welschehold, and A. Valada. Learning hierarchical interactive multi-object search for mobile manipulation. *IEEE Robotics and Automation Letters*, 8(12): 8549–8556, 2023.

[11] J. Liu, A. Mahdavi-Amiri, and M. Savva. Paris: Part-level reconstruction and motion analysis for articulated objects. In *International Conference on Computer Vision*, pages 352–363, 2023.

[12] Y. Liu, B. Jia, R. Lu, J. Ni, S.-C. Zhu, and S. Huang. Building interactable replicas of complex articulated objects via gaussian splatting. In *International Conf. on Learning Representations*, 2025.

[13] D. Honerkamp, M. Büchner, F. Despinoy, T. Welschehold, and A. Valada. Language-grounded dynamic scene graphs for interactive object search with mobile manipulation. *IEEE Robotics and Automation Letters*, 2024.

[14] M. Mohammadi, D. Honerkamp, M. Büchner, M. Cassinelli, T. Welschehold, F. Despinoy, I. Gilitschenski, and A. Valada. More: Mobile manipulation rearrangement through grounded language reasoning. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2025.

[15] A. Bandura and R. H. Walters. *Social learning theory*, volume 1. Prentice hall Englewood Cliffs, NJ, 1977.

[16] C. Doersch, Y. Yang, M. Vecerik, D. Gokay, A. Gupta, Y. Aytar, J. Carreira, and A. Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *International Conference on Computer Vision*, pages 10061–10072, 2023.

[17] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht. Cotracker: It is better to track together. In *European Conference on Computer Vision*, pages 18–35, 2024.

[18] J. Sturm, C. Stachniss, and W. Burgard. A probabilistic framework for learning kinematic models of articulated objects, 2011.

[19] S. Pillai, M. Walter, and S. Teller. Learning articulated motions from visual demonstration. In *Robotics: Science and Systems*, Berkeley, USA, July 2014. doi:10.15607/RSS.2014.X.050.

[20] D. Katz, A. Orthey, and O. Brock. Interactive perception of articulated objects. In *International Symposium on Experimental Robotics*, pages 301–315. Springer, 2014.

[21] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417, 2006.

[22] J. Sturm, K. Konolige, C. Stachniss, and W. Burgard. 3d pose estimation, tracking and model learning of articulated objects from dense depth video using projected texture stereo. In *RGB-D: Advanced Reasoning with Depth Cameras Workshop, RSS*, 2010.

[23] A. Jain, R. Lioutikov, C. Chuck, and S. Niekum. Screwnet: Category-independent articulation model estimation from depth images using screw theory. In *IEEE International Conference on Robotics and Automation*, pages 13670–13677, 2021.

[24] A. Jain, S. Giguere, R. Lioutikov, and S. Niekum. Distributional depth-based estimation of object articulation models. In *Conference on Robot Learning*, pages 1611–1621, 2022.

[25] X. Li, H. Wang, L. Yi, L. J. Guibas, A. L. Abbott, and S. Song. Category-level articulated object pose estimation. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 3706–3715, 2020.

[26] N. Heppert, T. Migimatsu, B. Yi, C. Chen, and J. Bohg. Category-independent articulated object tracking with factor graphs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3800–3807, 2022.

[27] V. Zeng, T. E. Lee, J. Liang, and O. Kroemer. Visual identification of articulated object parts. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2443–2450, 2021.

[28] N. Heppert, M. Z. Irshad, S. Zakharov, K. Liu, R. A. Ambrus, J. Bohg, A. Valada, and T. Kollar. Carto: Category and joint agnostic reconstruction of articulated objects. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 21201–21210, 2023.

[29] X. Huang, I. Walker, and S. Birchfield. Occlusion-aware reconstruction and manipulation of 3d articulated objects. In *IEEE International Conference on Robotics and Automation*, pages 1365–1371, 2012.

[30] R. Martín-Martín, S. Höfer, and O. Brock. An integrated approach to visual perception of articulated objects. In *IEEE International Conference on Robotics and Automation*, pages 5091–5097. IEEE, 2016.

[31] Z. Jiang, C.-C. Hsu, and Y. Zhu. Ditto: Building digital twins of articulated objects from interaction. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 5616–5626, 2022.

[32] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[33] J. Shi et al. Good features to track. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[34] A. W. Harley, Z. Fang, and K. Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision*, pages 59–75, 2022.

[35] C. Doersch, A. Gupta, L. Markeeva, A. Recasens, L. Smaira, Y. Aytar, J. Carreira, A. Zisserman, and Y. Yang. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022.

[36] Y. Zheng, A. W. Harley, B. Shen, G. Wetzstein, and L. J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *International Conference on Computer Vision*, pages 19855–19865, 2023.

[37] S. Cho, J. Huang, J. Nam, H. An, S. Kim, and J.-Y. Lee. Local all-pair correspondence for point tracking. In *European Conference on Computer Vision*, pages 306–325, 2024.

[38] C. Doersch, P. Luc, Y. Yang, D. Gokay, S. Koppula, A. Gupta, J. Heyward, I. Rocco, R. Goroshin, J. Carreira, et al. Bootstap: Bootstrapped training for tracking-any-point. In *Proceedings of the Asian Conference on Computer Vision*, pages 3257–3274, 2024.

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

[40] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, and C. S. Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023.

[41] G. Camporese. Hands segmentation is all you need. *https://github.com/guglielmocamporese*, 2021. URL https://github.com/guglielmocamporese/hands-segmentation-pytorch.

[42] T. D. Barfoot. *State estimation for robotics*. Cambridge University Press, 2024.

[43] F. Dellaert and G. Contributors. borglab/gtsam, May 2022. URL https://github.com/borglab/gtsam).

[44] Y. Weng, B. Wen, J. Tremblay, V. Blukis, D. Fox, L. Guibas, and S. Birchfield. Neural implicit representation for building digital twins of unknown articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3141–3150, 2024.

[45] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, et al. Sapien: A simulated part-based interactive environment. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*.

[46] R. Martín-Martín, C. Eppner, and O. Brock. The rbo dataset of articulated objects and interactions. *International Journal of Robotics Research*, 38(9):1013–1019, 2019.

[47] D. Rotondi, F. Scaparro, H. Blum, and K. O. Arras. Fungraph: Functionality aware 3d scene graphs for language-prompted scene interaction. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

[48] C. Zhang, A. Delitzas, F. Wang, R. Zhang, X. Ji, M. Pollefeys, and F. Engelmann. Open-vocabulary functional 3d scene graphs for real-world indoor spaces. *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2025.

[49] C. Kassab, S. Morin, M. Büchner, M. Mattamala, K. Gupta, A. Valada, L. Paull, and M. Fallon. Openlex3d: A new evaluation benchmark for open-vocabulary 3d scene representations. *arXiv preprint arXiv:2503.19764*, 2025.

[50] A. Delitzas, A. Takmaz, F. Tombari, R. Sumner, M. Pollefeys, and F. Engelmann. SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2024.

[51] A. Werby, C. Huang, M. Büchner, A. Valada, and W. Burgard. Hierarchical Open-Vocabulary 3D Scene Graphs for Language-Grounded Robot Navigation. In *Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi:10.15607/RSS.2024.XX.077.

[52] E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[53] C.-C. Hsu, Z. Jiang, and Y. Zhu. Ditto in the house: Building articulation models of indoor scenes through interactive perception. In *IEEE International Conference on Robotics and Automation*, pages 3933–3939, 2023.

[54] J. Sturm, V. Pradeep, C. Stachniss, C. Plagemann, K. Konolige, and W. Burgard. Learning kinematic models for articulated objects. In *International Joint Conferences on Artificial Intelligence*, 2009.

[55] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *International Conference on Computer Vision*, pages 2564–2571, 2011.

[56] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[57] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang. OpenVINS: A research platform for visual-inertial estimation. In *IEEE International Conference on Robotics and Automation*, 2020.

[58] R. Murai, E. Dexheimer, and A. J. Davison. MASt3R-SLAM: Real-time dense SLAM with 3D reconstruction priors. *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2025.

[59] Z. Teed and J. Deng. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in Neural Information Processing Systems*, 2021.

[60] H. Lim, D. Kim, G. Shin, J. Shi, I. Vizzo, H. Myung, J. Park, and L. Carlone. KISS-Matcher: Fast and Robust Point Cloud Registration Revisited. In *IEEE International Conference on Robotics and Automation*, 2025.

Articulated Object Estimation in the Wild: Supplementary Material

In this supplementary material, we provide additional insights on the evaluation protocol and utilized metrics in Sec. S.1.1, shed light on the necessity of ground truth camera poses to make ArtiPoint work in real-world deployment in Sec. S.1.2, evaluate to what degree ground truth interaction segments benefit the prediction performance in Sec. S.1.3 compared to predicted ones, and provide additional ablations regarding the hand detection as well as the point tracking in Sec. S.1.4. Furthermore, we provide qualitative insights in Sec. S.1.6 and in-depth explanations regarding the introduced Arti4D dataset in Sec. S.2.

S.1 Experimental Evaluation

In this section, we present additional details on our experiments, the metrics we employ, and the additional ablation study.

S.1.1 Evaluation Protocol & Metrics

In Sec. 5, we report quantitative results of our approach and the baselines. In the following, we detail the prediction to ground truth association procedure as well as the definition of the metrics that we employ to quantify performance.

In order to account for the fact that multiple interactions with the same object instance are likely throughout a single sequence, we match each obtained axis prediction against the corresponding ground truth axes based on the underlying interaction windows. This entails computing the 1-D intersection-over-union (IoU) between all predicted interaction segments and all ground truth interaction segments, as labeled as part of Arti4D. We consider a match whenever an $IoU > 0.5$ is exceeded.

As stated in the main manuscript, our metrics consist of positional and angular error. We compute the positional error d_i for the placement of an articulation's rotation axis $\hat{\mathbf{a}}$ and the ground truth axis $\mathbf{a}_{gt}$ using the help of supporting points $\hat{\mathbf{p}}, \mathbf{p}_{gt}$ as

$$d_i = \begin{cases} \frac{(\hat{\mathbf{p}}_i - \mathbf{p}_{gt})^\top (\hat{\mathbf{a}}_i \times \mathbf{a}_{gt})}{\|\hat{\mathbf{a}}_i \times \mathbf{a}_{gt}\|} & \text{if } \|\hat{\mathbf{a}}_i \times \mathbf{a}_{gt}\| > \epsilon \\ \|(\hat{\mathbf{p}}_i - \mathbf{p}_{gt}) \times \mathbf{a}_{gt}\| & \text{else} \end{cases}, \tag{1}$$

where the first case covers the case in which the axes are not parallel with $\epsilon = 10^{-4}$. If a model does not provide a point on the axis directly, but a twist, we compute $\hat{\mathbf{p}}_i = \frac{\omega_i \times v_i}{\|\omega_i\|^2}$. The angular error of a prediction, we compute simply by using the normalized dot-product of the axes

$$\Theta_{err,i} = \cos^{-1}\left(\frac{\mathbf{a}_{gt}^\top \hat{\mathbf{a}}_i}{\|\mathbf{a}_{gt}\| \, \|\hat{\mathbf{a}}_i\|}\right). \tag{2}$$

We report only angular errors for prismatic joints, as the location of the axis does not have any effect on the motion of the parts of the articulated object.

S.1.2 Performance Under Estimated Camera Poses

The experimental results reported in the main manuscript rely on ground truth camera odometry from the Arti4D dataset. However, in order to account for direct real-world deployment of our method, we have conducted additional experiments that do not require access to those. To do so, we have evaluated a number of prominent RGB-D SLAM approaches that produce metric map estimates: ORB-SLAM3 [56], Open-VINS [57], MAST3R-SLAM [58], and DROID-SLAM [59]. While all deep learning-based methods provide globally consistent maps, all traditional approaches fail due to

loss of camera tracking. Given their non-static character, the interaction segments pose the greatest challenge. As our method requires camera odometry instead of only keyframe-level pose estimates, we employ DROID-SLAM over MAST3R-SLAM.

First, we note that the maps produced by DROID-SLAM are registered against the ground truth point cloud using KISS-Matcher [60] in order to provide grounds for evaluation, which, in turn, may induce translational and rotational errors. When utilizing DROID-SLAM camera poses, we achieve reasonable results with only slightly increased angular and translational errors for both prismatic and revolute joints. Nonetheless, we observe a slight increase in prismatic type prediction accuracy when employing DROID-SLAM poses.

Table S.1: Comparison of estimated object axis under estimated and ground truth camera poses

Method	Prismatic joints		Revolute joints		Type accuracy [%]	
	θ_{err}[deg]	d_{L2}[m]	θ_{err}[deg]	d_{L2}[m]	Prismatic	Revolute
w/ DROID-SLAM poses	14.67	–	18.12	0.10	0.71	0.94
w/ Arti4D odometry	14.54	–	17.14	0.07	0.68	0.98

S.1.3 Evaluation Using Ground-Truth Interaction Segments

In addition to the results relying on predicted interaction segments using the windowed hand detection scheme, we provide an evaluation based on ground truth interaction segments that are labeled as part of Arti4D. In general, we would expect lower angular and translational errors given that the method is not affected by hand occlusions (under large opening angles), motion blur, or non-detected hands. However, we find that our articulation estimation is robust wrt. to the interaction segmentation as we observe smaller angular errors on prismatic objects but larger errors for revolute-jointed objects, thus not allowing a clear interpretation.

Table S.2: Evaluation under ground-truth interaction segments

Method	Prismatic joints		Revolute joints		Type accuracy [%]	
	θ_{err}[deg]	d_{L2}[m]	θ_{err}[deg]	d_{L2}[m]	Prismatic	Revolute
w/ GT segments	**11.99**	–	20.24	0.09	**0.74**	0.97
w/ pred. segments	14.54	–	**17.14**	**0.07**	0.68	**0.98**

S.1.4 Additional Ablation Study

In the following sections, we share additional ablations on hyperparameters of ArtiPoint and scene-respective results.

S.1.4.1 Ablation on Hand Detection

Extracting the interaction intervals, as described in Sec. 3.1 and illustrated in Fig. S.1, is a critical component of the ArtiPoint pipeline as it directly affects the number of articulated objects detected. As such, it requires careful parameter tuning, as small values of w_h or T_{min} increase the number of false positives. To better understand the impact of these parameters on the final results, we conduct a detailed ablation study with its findings presented in Tab. S.3. Decreasing T_{min} leads to a small degradation in both angular error (θ_{err}) and joint type classification accuracy. Increasing w_h leads to over-smoothing of the raw hand detection signal, causing the segments to contain elongated idle phases at the start and the end, thus inducing unfavorable noise, and resulting in a noticeable degradation in both angular error (θ_{err}) and joint type classification accuracy for both joint types. Furthermore, we also list the effect of not filtering outlier tracks, not using backward tracking, using CoTracker2 over CoTracker3, and performing static trajectory using three-dimensional trajectory data instead of two-dimensional data that is less subject to noisy depth.

Table S.3: Ablations of key parameters for extracting the interaction intervals component Sec. 3.1 on the Arti4D dataset. As before, we report the same set of metrics as in Tab. 1. Default values: $T_{max} = 90, T_{min} = 30, w_h = 6$.

Method	Prismatic joints		Revolute joints		Type accuracy [%]	
	θ_{err}[deg]	d_{L2}[m]	θ_{err}[deg]	d_{L2}[m]	Prismatic	Revolute
w/o filtering outlier tracks	13.17	–	22.80	0.14	0.64	0.84
w/o backward tracking	13.92	–	20.67	0.13	0.71	0.94
CoTracker2	15.78	–	21.04	0.10	0.66	0.94
static traj. filter in 3D	15.18	–	18.23	0.10	0.65	0.92
$w_h = 12$	15.58	–	19.88	0.08	0.66	0.97
$T_{max} = 120$	14.57	–	18.13	0.07	0.68	0.96
$T_{min} = 15$	14.57	–	17.88	0.07	0.68	0.96
ArtiPoint	14.54	–	17.14	0.07	0.68	0.98

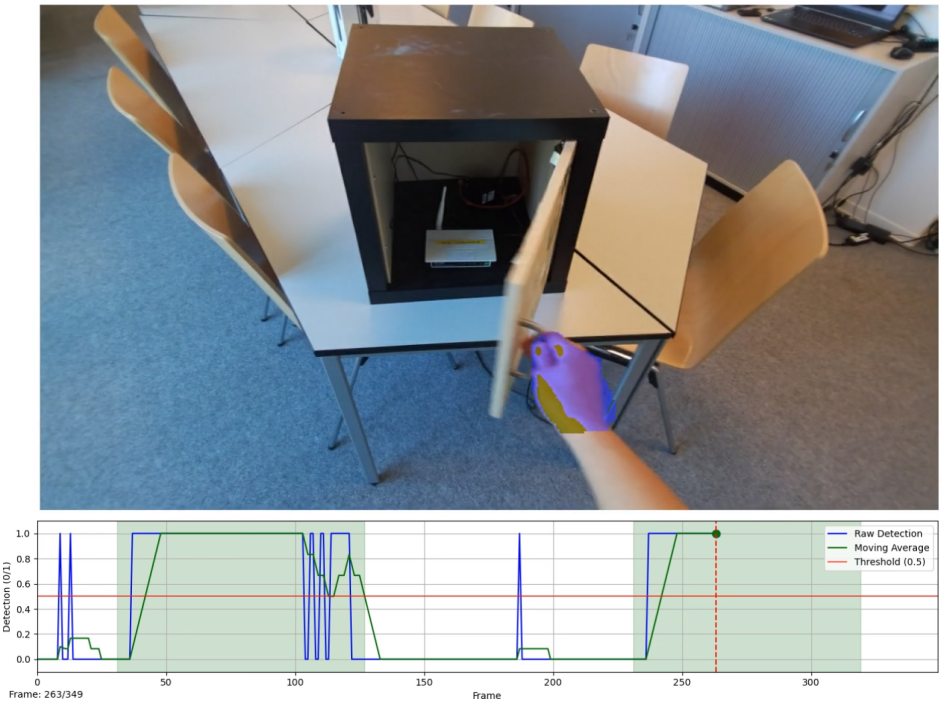
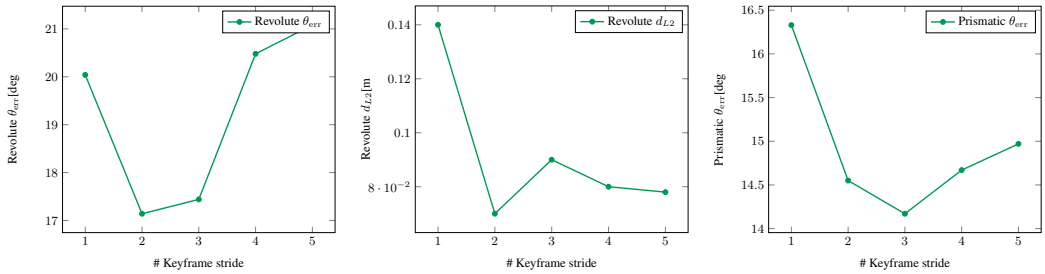


Figure S.1: Hand detection and interaction extraction: We visualize a live frame of an interaction, including a hand mask marked violet (top). In addition, we visualize the frequency of raw hand detections over time up to the live frame as well as its moving average (bottom). The horizontal red line indicates the threshold at which an interaction segment is created, given the moving average signal. The vertical dashed red line indicates the current frame.

S.1.4.2 Ablation on Point Tracking

In this section, we present ablation results on the any-point tracking component as illustrated in Fig. S.2. In particular, we evaluate to which degree different keyframe strides used as input to the point tracking component affect the downstream axis prediction performance (see Sec. 3.2). Choosing the keyframe stride is vital hyperparameter of the point tracking stage. While a smaller stride leads to an increase in the number of detected points to be tracked by Cotracker3 [39], larger strides reduce the number of detected points to be tracked, thereby lowering the computational load. Retaining a sufficient number of points is necessary for estimating an object's 3D trajectory over time. Overall, we observe the lowest prediction errors for a keyframe stride of 2 to 3 based on the angular errors. However, in the case of the translational errors, we are not able to derive a distinct statement.



(a) Revolute θ_{err} (b) Revolute d_{L2} (c) Prismatic θ_{err}

Figure S.2: Impact of keyframe stride on tracking accuracy and error. (a) The revolute joint angular error θ_{err}[deg] exhibits a minimum at stride 2. (b) The revolute joint positional error d_{L2}[m] is lowest at a stride of 2, with higher errors observed for both smaller and larger strides. (c) Similarly, the prismatic joint angular error θ_{err} exhibits its minimum at stride of 2. We conclude that a stride of 2 is optimal in terms of point density and computational efficiency.

S.1.5 Scene-Respective Results

In addition, to the `EASY`/`HARD` differentiation evaluated in Tab. 2, we show scene-respective results in Tab. S.4. This involves averaging the predictions of all object interactions contained in a scene split. We list the number of sequences per scene as well as the number of labeled objects per sequence in Sec. S.2. As reported in Tab. S.4, `RH078` constitutes the most difficult split of the Arti4D dataset. In comparison, `DR080` and `RH201` seem to represent simpler environments.

We attribute worse results on `RH078` to a number of non-separable interactions as the hand is occasionally not fully retrieved between interactions. As a consequence, the proposed interaction extraction baseline potentially fails at differentiating two different interactions. We have mentioned the hand trigger limitation in Sec. 7 and leave improvements on that front to feature work. In addition to that, we observe that there are comparably more revolute joints in scene `RH078` whose associated objects are rather textureless and of metallic character, thus hindering consistent depth observations.

Table S.4: Scene-respective results: We report the scene-wise results using the established set of metrics. We find that `RH078` constitutes the most difficult split of the Arti4D dataset.

Method	Prismatic joints		Revolute joints		Type accuracy	
	θ_{err}[deg]	d_{L2}[m]	θ_{err}[deg]	d_{L2}[m]	Prismatic	Revolute
`RH078`	35.11	–	10.37	0.05	0.55	1.00
`RR080`	15.11	–	9.41	0.10	0.56	1.00
`DR080`	7.86	–	17.97	0.12	0.83	1.00
`RH201`	8.73	–	20.88	0.04	0.84	0.95
Overall	**14.55**	–	**17.14**	**0.07**	**0.68**	**0.98**

S.1.6 Qualitative Results

In the following, we provide additional qualitative results. In Fig. S.3, we visualize the output of our proposed interaction extraction, any-point tracking and track filtering components. We observe a sufficient number of point trajectories even under partially missing depth measurements or feature-sparse textures. In addition to that, we visualize a full scene-level output of ArtiPoint on `DR080` scene in Fig. S.4. Overall, ArtiPoint detects the majority of interactions and produces reliable estimates considering the in-the-wild character of the recorded Arti4D sequences.

S.2 Arti4D Dataset

In the following, we provide additional insight on the in-the-wild object articulation dataset Arti4D. We provide 45 sequences across four distinct environments as listed in Sec. S.2 and visualized in Fig. S.7. In addition to the sequence IDs and the recording names of the produced sequences, we

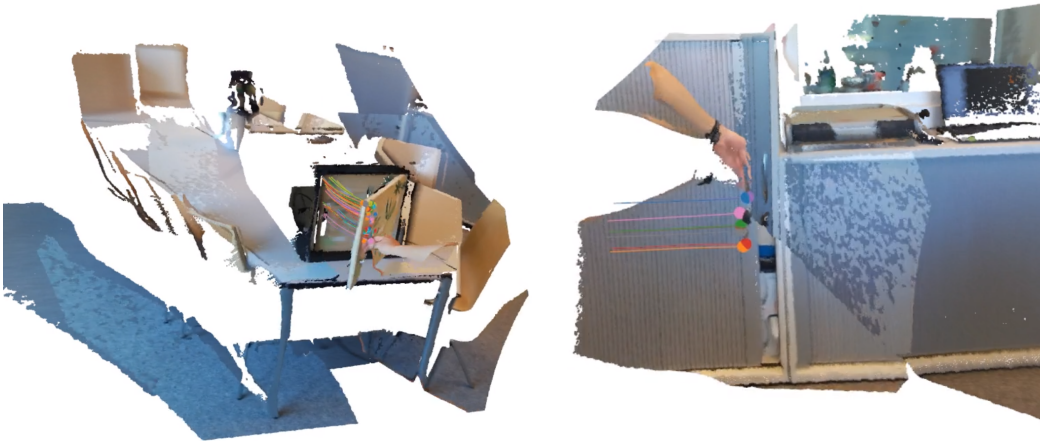


Figure S.3: Smoothed point trajectories: We visualize a cabinet (revolute) and its tracked keypoints (left) as well as a linear slider shelf (prismatic) on the right. Each keypoint trajectory is represented with a unique color. Both sets of point trajectories visualized constitute the output of our track filtering introduced in Sec. 3.3.

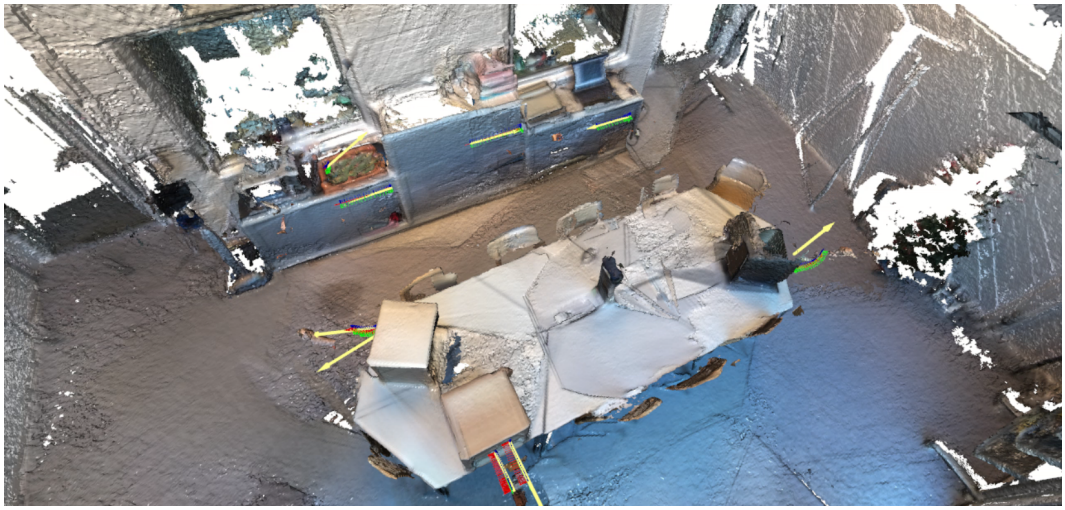


Figure S.4: Scene-level prediction: We depict a full scene-level output of the ArtiPoint framework on sequence `scene_2025-04-11-11-44-32` of the `DR080` scene. Yellow arrows denote axes of motion of predicted object interactions while coordinate frames represent the estimate part poses throughout articulation based on the proposed estimation framework (see Sec. 3.4).

report the number of labeled objects for each sequence, the ratio between prismatic and revolute joints as well as the ratio between easy and hard objects. First, note that the number of objects is not equal to the number of object interactions per sequence, as several sequences contain repeated interactions with the same single object instance. This constitutes a corner case in terms of articulated object interaction as it requires prediction methods to fuse, e.g., two predictions belonging to a single object. While most interactions are separable by detecting the absence of a hand mask, especially the `RH078` split which contains a number of hard-to-separate interactions. This is due to the fact that the interacting hand was not always fully retrieved in-between interactions of two distinct objects. Similar to the repeated interactions case mentioned before, this represents another corner case requiring advanced action recognition.

As part of the dataset, we make both rosbags and processed raw data public. While the rosbags include TF data at a higher frequency, the raw data includes aligned RGB, depth, and camera poses at 15 Hz. We employed an Azure Kinect RGB-D camera that was handheld throughout all interactions. In terms of ground truth camera pose retrieval, we employed external tracking using

HTC Vive trackers, which provide cm-level accuracy. In case of sudden odometry glitches induced by considerable occlusions or reflections on glass or metal, we have removed those sequences from the dataset. We found that running classical structure-from-motion approaches to reconstruct the underlying sequence fails as significant parts of the camera field of view cover articulations. In turn, the contained articulations break assumptions towards mostly *static* visual correspondence made in structure-from-motion methods. Thus, we leverage the ground truth camera poses and perform TSDF fusion to produce scene reconstructions. We depict four TSDF-reconstructed sequences stemming from each of the splits in Fig. S.5. The reconstructions reflect minimal ground truth odometry drift and enable precise anchoring of object axes.

The ground truth object axes were labeled based on the reconstructed sequences using Blender, exported as JSONs, and verified by a second reviewer. Furthermore, we provide metadata on difficulty levels and temporal interaction segment ground truth labels. We provide four exemplary depth-masked RGB frames covering an interaction of a microwave featuring semi-transparent glass and metal surfaces in Fig. S.6. As depicted, a considerable part of the object does not produce depth estimates. Objects of that kind are labeled using the `HARD` category.

Finally, we depict several RGB frames drawn from the four distinct scene splits in Fig. S.7, underlining the variety of objects and challenging conditions of the proposed dataset. We also make the sequence reconstructions in the form of meshes and point clouds available.

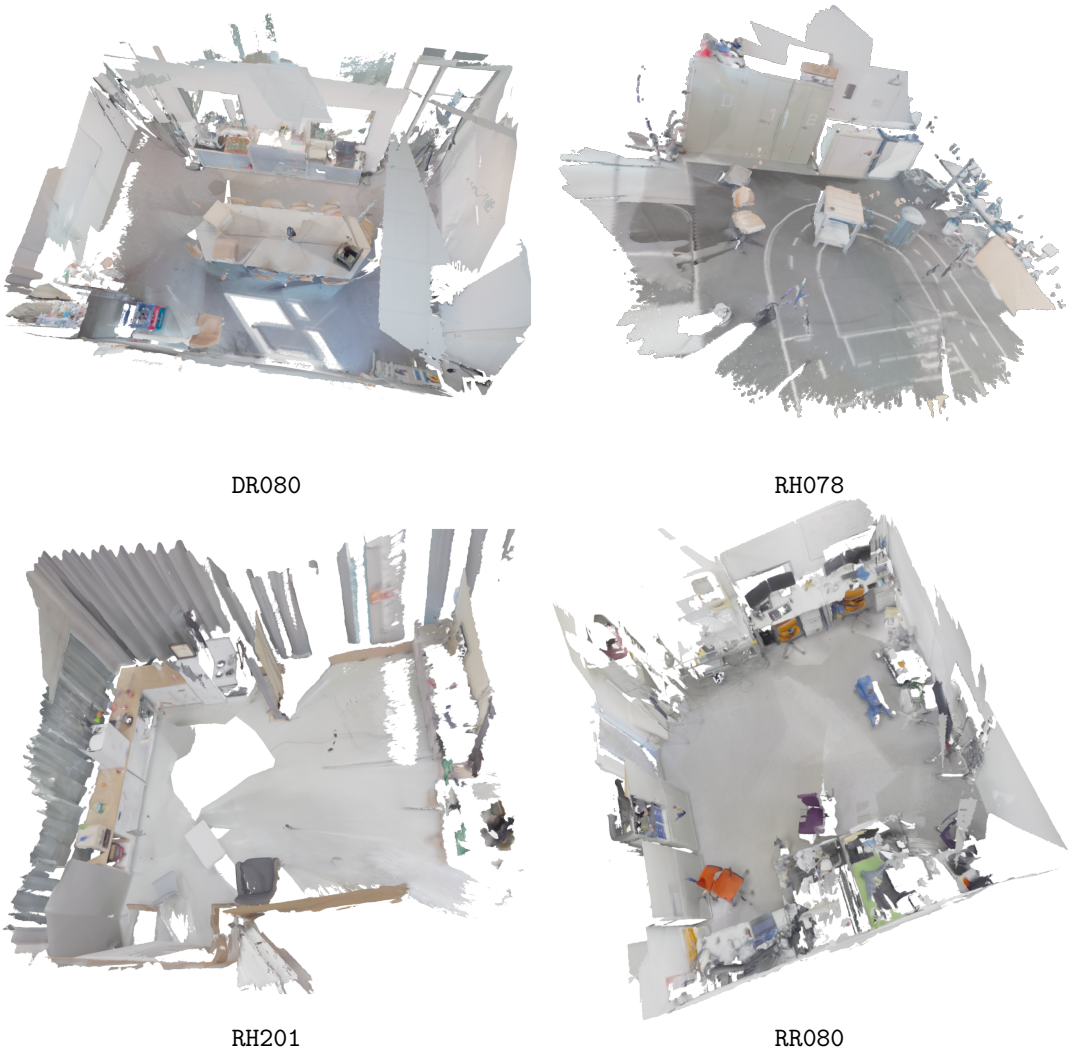


Figure S.5: We visualize the reconstructed scenes of the four Arti4D environments: DR080, RH078, RH201, and RR080.

Figure S.6: We visualize an exemplary interaction with the depth projected onto the RGB observation in red wherever a depth reading was unavailable for the particular image coordinate. As depicted, metallic, semi-transparent, or dark-colored objects do not produce reliable depth estimates, ultimately complicating the lifting of 3D point trajectories used to estimate the underlying articulation.

Table S.5: Overview of all Arti4D demonstration sequences: We assign sequence identifiers to each recording and list the number of objects interacted with per sequence as well as the distribution of prismatic vs. revolute joints (PRISM / REV) and objects classified as either EASY or HARD. The number of objects corresponds to the number of annotated axes per sequence, whereas the number of interactions denotes the number of performed articulations, thus including objects that are articulated multiple times per sequence.

Scene	*Sequence ID*	*Recording*	*# Objects*	*# Interactions*	# PRISM / REV	# EASY / HARD
RH078	RH078-00	scene_2025-04-04-19-14-38	7	7	3 / 4	7 / 0
	RH078-01	scene_2025-04-04-19-18-54	6	7	2 / 4	4 / 2
	RH078-02	scene_2025-04-07-11-39-17	7	7	3 / 4	7 / 0
	RH078-03	scene_2025-04-07-11-41-52	8	9	3 / 5	6 / 2
	RH078-04	scene_2025-04-07-11-48-40	7	7	3 / 4	7 / 0
	RH078-05	scene_2025-04-09-10-30-11	8	8	5 / 3	3 / 5
	RH078-06	scene_2025-04-09-10-32-52	6	6	5 / 1	3 / 3
	RH078-07	scene_2025-04-09-10-35-47	7	7	6 / 1	2 / 5
	RH078-08	scene_2025-04-09-10-38-38	7	7	5 / 2	3 / 4
	RH078-09	scene_2025-04-09-10-46-48	7	8	7 / 0	4 / 3
	RH078-10	scene_2025-04-09-10-49-20	7	7	6 / 1	2 / 5
RR080	RR080-00	scene_2025-04-10-13-11-16	17	17	14 / 3	9 / 8
	RR080-01	scene_2025-04-10-16-05-09	14	14	13 / 1	8 / 6
	RR080-02	scene_2025-04-17-15-25-14	11	12	11 / 0	7 / 4
	RR080-03	scene_2025-04-17-15-33-44	9	9	9 / 0	7 / 2
	RR080-04	scene_2025-04-22-09-53-49	8	8	8 / 0	5 / 3
	RR080-05	scene_2025-04-22-09-56-24	10	10	9 / 1	9 / 1
	RR080-06	scene_2025-04-22-09-58-49	7	8	7 / 0	4 / 3
	RR080-07	scene_2025-04-22-11-45-15	9	9	8 / 1	7 / 2
	RR080-08	scene_2025-04-22-11-48-01	9	9	8 / 1	7 / 2
	RR080-09	scene_2025-04-22-11-50-40	8	8	7 / 1	6 / 2
DR080	DR080-00	scene_2025-04-11-11-44-32	11	11	7 / 4	5 / 6
	DR080-01	scene_2025-04-11-12-58-58	10	10	6 / 4	5 / 5
	DR080-02	scene_2025-04-11-13-01-59	9	9	5 / 4	4 / 5
	DR080-03	scene_2025-04-11-13-18-00	9	10	4 / 5	3 / 6
	DR080-04	scene_2025-04-11-13-43-03	11	11	7 / 4	5 / 6
	DR080-05	scene_2025-04-11-14-01-06	11	11	7 / 4	5 / 6
	DR080-06	scene_2025-04-11-15-43-24	11	12	7 / 4	5 / 6
	DR080-07	scene_2025-04-11-15-46-48	11	11	6 / 5	4 / 7
RH201	RH201-00	scene_2025-04-24-17-52-21	11	11	5 / 6	6 / 5
	RH201-01	scene_2025-04-24-17-54-13	9	9	5 / 4	5 / 4
	RH201-02	scene_2025-04-24-19-18-42	11	11	5 / 6	7 / 4
	RH201-03	scene_2025-04-24-19-21-50	8	8	2 / 6	5 / 3
	RH201-04	scene_2025-04-24-19-24-09	8	8	5 / 3	5 / 3
	RH201-05	scene_2025-04-25-10-36-37	9	9	5 / 4	7 / 2
	RH201-06	scene_2025-04-25-10-53-40	8	8	4 / 4	3 / 5
	RH201-07	scene_2025-04-25-10-56-33	8	8	3 / 5	3 / 5
	RH201-08	scene_2025-04-25-11-11-47	15	16	6 / 9	6 / 9
	RH201-09	scene_2025-04-25-11-15-47	7	7	5 / 2	4 / 3
	RH201-10	scene_2025-04-25-14-58-42	9	9	6 / 3	4 / 5
	RH201-11	scene_2025-04-25-15-02-14	7	7	3 / 4	4 / 3
	RH201-12	scene_2025-04-25-15-04-48	7	7	4 / 3	3 / 4
	RH201-13	scene_2025-04-25-15-16-29	9	9	5 / 4	3 / 6
	RH201-14	scene_2025-04-25-15-19-22	10	10	5 / 5	5 / 5
	RH201-15	scene_2025-04-25-15-22-54	8	8	4 / 4	3 / 5

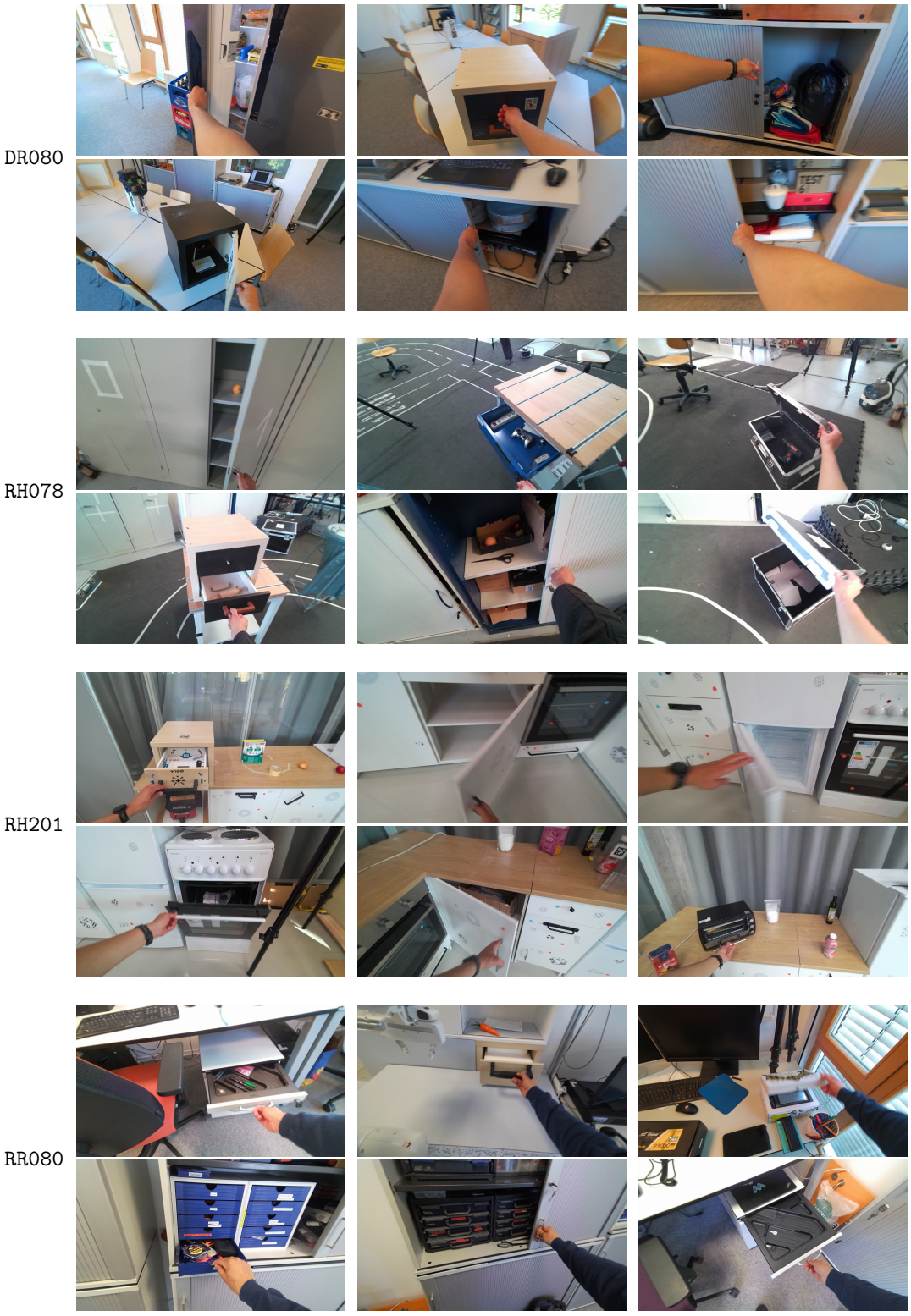


Figure S.7: We visualize several object interactions across the four different environments (DR080, RH078, RH201, RR080) captured as part of the Arti4D dataset.