Interactive Anomaly Detection for Articulated Objects via Motion Anticipation

Ankan Bhunia Changjian Li Hakan Bilen
University of Edinburgh
https://groups.inf.ed.ac.uk/vico/research/interactiveAD/

Abstract

This paper presents a novel problem, interactive anomaly detection (AD) for articulated objects, and introduces a tailored solution that detects functional anomalies by integrating vision, interaction, and anticipation. Unlike traditional AD methods that rely on passive visual observations, our approach actively manipulates objects to reveal anomalies that would otherwise remain hidden. Our method learns to generate a sequence of actions to interact exclusively with normal objects and to anticipate the resulting normal motion. During inference, the model applies predicted actions to the object and compares the observed motion with the anticipated motion to detect anomalies. Additionally, we introduce a new benchmark, *PartNet-IAD*, for interactive AD, which includes articulated objects with realistic functional anomalies. Experiments show strong generalization to detect anomalies in both seen and unseen object categories.

1 Introduction

Humans possess a remarkable ability to interact effortlessly with a wide range of objects in everyday life. This capability stems from our intuitive understanding of how objects function–we assess their potential uses, apply appropriate forces, and anticipate the outcomes of our actions. Importantly, we adjust our behavior when the actual response of an object deviates from our expectations. These discrepancies often serve as key feedback signals, especially when an object fails to operate as intended–what we refer to as functional anomalies. This work aims to develop a perception system that mirrors this human capability by detecting functional anomalies in articulated objects through a combination of vision, physical interaction, and motion anticipation.

Detecting functional anomalies in articulated objects is essential for quality inspection in manufacturing (*e.g.*, drawers, washing machines, laptops) and for robotic manipulation tasks. Unlike standard anomaly detection (AD) benchmarks that focus on identifying visually apparent defects from static observations [2, 46, 15, 5, 8, 3], many functional anomalies in articulated objects are not visually observable and only become evident during physical interaction. As such, detecting these anomalies requires an *active approach*–manipulating objects, anticipating their normal behavior, and identifying deviations from expected motion (see Fig. 1).

This paper presents a novel perspective on AD by proposing an *interactive AD framework* that addresses two central challenges: (1) learning how to interact with objects to reveal hidden anomalies, and (2) anticipating their normal motion response during interaction. Unlike the prior work [28, 13, 40] which focuses on interaction with articulated objects without modeling expected outcome, our approach predicts anomalies by comparing observed object motion after manipulation with the motion expected under normal conditions. Given an RGBD image or partial point cloud and an atomic action (*e.g.*, pushing or pulling), our model predicts: (i) the 3D segmentation and motion of the target part, and (ii) a motion confidence score, and an interaction end state indicating whether the part has reached an interaction end state (*e.g.*, a door fully opened). Importantly, the model is trained exclusively on

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

 https://doi.org/10.52202/085713-3920

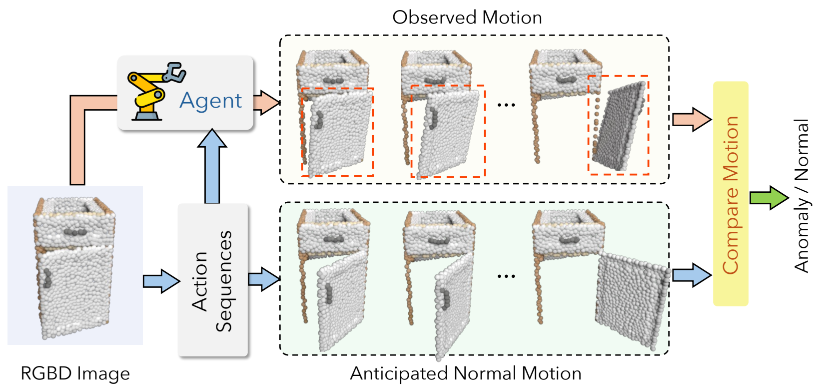

Figure 1: In the proposed interactive anomaly detection setting, the algorithm learns to manipulate articulated objects without any anomalies through generating a sequence of actions while predicting their normal motion. During testing, the algorithm first generates a sequence of actions along with their anticipated normal motion without any interaction (bottom). Then the agent (*e.g.*, a robotic arm) executes the generated actions on the object resulting in the observed motion (top). By comparing the anticipated and observed motions, the algorithm can detect potential functional anomalies.

interactions with normal objects, ensuring that its predicted motion aligns with normal behavior. At inference time, the model generates a predicted motion trajectory by sequentially applying learned atomic actions without physical interaction. The same action sequence is executed on the actual object by an agent (*e.g.*, a robotic arm), and the observed trajectory is recorded. Comparing the anticipated and observed motions enables AD based on motion discrepancies.

To support this task, we introduce *PartNet-IAD*, a new benchmark for interactive AD. Built upon the PartNet-Mobility dataset [39], we inject realistic functional anomalies into articulated objects and simulate interaction environments where a robotic arm can push and pull object parts. Results show that our method generalizes well to unseen objects and categories, effectively detecting functional anomalies through interaction.

In summary, our contributions are: (1) we formulate the novel task of detecting functional anomalies through interaction with 3D articulated objects; (2) we propose a novel model that learns to manipulate objects and detect anomalies by comparing expected and observed motion; (3) we introduce *PartNet-IAD*, the first benchmark for interactive AD, enabling evaluation of AD in articulated objects.

2 Related Work

AD benchmarks. Previous AD benchmarks focus on detecting anomalies from passive observations, such as images [2, 46, 44, 15, 5], videos [8, 32, 42], or point clouds [4, 23, 3]. These approaches make predictions based on single observations captured by cameras and do not actively interact with the object or iteratively update their understanding of the object through a feedback mechanism. While such approaches are effective for rigid objects, they are often insufficient for articulated objects, where passive observation alone may not reliably reveal anomalies. To address this limitation, we propose a learnable framework in which an agent actively interacts with articulated objects to uncover motion anomalies that may not be apparent otherwise. A previous work [14], FixIT, proposes a solution to diagnose and fix malfunctioning articulated objects, however, without learning to actively interact with the object, and instead assumes that interaction videos are directly available as input. In contrast, our framework simultaneously learns to interact and discover anomalies.

Interactive perception for articulated objects. Early research in object manipulation has primarily focused on learning task-specific action primitives, such as manipulating doors and drawers [21, 20]. These approaches often rely on hand-tuned actions to generate informative motion for downstream perception tasks [18, 34]. However, these methods typically assume prior knowledge of the object's structure, which is used to design heuristic rules. Additionally, action primitives tailored for one task (*e.g.*, opening doors) may lack the generality needed for other tasks or objects (*e.g.*, pushing buttons). Recent studies have highlighted the effectiveness of exploiting interactions in simulated environments for learning perception models [41, 30, 27, 28, 13, 36, 29, 40], demonstrating promising

https://doi.org/10.52202/085713-3920

generalization to real-world scenarios [16, 6, 31]. Among these studies, Where2Act [28] presents a learnable framework to estimate dense action affordance maps on articulated objects from a single RGB image or point cloud, while UMPnet [40] extends this method to handle long-horizon action trajectories for goal-conditioned manipulation tasks. AtP [13] proposes an iterative method for interacting with articulated objects to discover and segment their components. In contrast, our work focuses on discovering motion anomalies by learning to interact with articulated objects. Unlike previous methods [28, 40, 13] that predict only action parameters, our framework jointly predicts both the action and anticipated motion, enabling anomaly detection through comparison with observed motion. Our framework enhances robustness through confidence-based motion estimation and supports long-term, temporally consistent action exploration.

Rigid motion analysis. The pivotal component of our framework is a learnable motion prior that estimates normal motion behavior at any instance. Several prior works [11, 1, 24] have presented learning-based rigid motion estimation from point cloud scenes. For articulated objects, however, research on motion estimation has been limited. Recent works estimate object articulation and predict motion flow and segmentation masks from a pair of point cloud frames [33, 43] or multi-view images [25, 10]. In contrast, our formulation of the motion prior is action-conditioned, estimating the rigid motion of articulated object parts for any arbitrary action parameters. Recently, DragAPart [22] proposed a diffusion-based generation pipeline that predicts deformation for a given drag force using images as input. Unlike this approach, our motion prior operates on unstructured 3D point clouds.

3 Method

3.1 Problem Formulation and Overview

Given an observation of a 3D object with articulated parts (*e.g.*, cupboard with a drawer and door), our goal is to classify whether each movable part r exhibits a functional anomaly (1 if anomalous, 0 otherwise). Since such anomalies are often not detectable through passive observations alone, we allow a robot agent to interact with the object over a limited number of timesteps T_{max}. At each timestep t, the robot executes an action $\boldsymbol{a}_t$, parameterized by its end-effector's position $\boldsymbol{p}_t \in \mathbb{R}^3$ and movement direction $\boldsymbol{u}_t \in \mathbb{R}^3$. The scene is captured as a point cloud $\boldsymbol{Q}_t$ at timestep t with $\boldsymbol{Q}_0$ denoting the initial observation prior to any interaction.

An overview of our method is shown in Fig. 2. The process begins with the agent observing the initial state of the object $\boldsymbol{Q}_0$ and planning a sequence of atomic actions (*e.g.*, push, pull). These actions are simulated to anticipate the resulting object motions. At each timestep t, the agent predicts an action $\boldsymbol{a}_t$ and the corresponding motion $\boldsymbol{M}_t$ for a manipulated part r. Note that we omit the part index in both $\boldsymbol{a}$ and $\boldsymbol{M}$ for brevity. The agent initializes its internal object state as $\boldsymbol{S}_0 = \boldsymbol{Q}_0$ and updates this state $\boldsymbol{S}_t$ over time by applying the predicted motion $\boldsymbol{M}_t$ to the relevant part, producing a simulated future state $\boldsymbol{S}_{t+1}$ without physical interaction. This simulation process iterates over T steps, forming a sequence of atomic action-conditioned motion priors (see Sec. 3.2). Because isolated atomic actions may not sufficiently capture the full range of part motions or cover all parts, we incorporate a long-term normal motion estimation strategy. This ensures comprehensive exploration of the action space with temporal consistency and robustness to noise (see Sec. 3.3). The result is a sequence of predicted action-motion pairs $\{(\boldsymbol{a}_t, \boldsymbol{M}_t)\}_{t=0}^{T-1}$ for each part.

The agent then executes the same actions on the real object and collects the resulting observations $\{\boldsymbol{Q}_t\}_{t=1}^{T}$, from which the actual motions are computed $\{\hat{\boldsymbol{M}}_t\}_{t=0}^{T-1}$. A part is classified as anomalous if the observed motion sequence deviates from the predicted one by more than a threshold ϵ (see Sec. 3.4).

3.2 Atomic Action-Conditioned Motion Prior

Given an internal 3D object state $\boldsymbol{S}$ and an action $\boldsymbol{a}$, we learn a function Ψ that predicts the expected normal rigid motion in 3D $\boldsymbol{M}_a$, a confidence score $\boldsymbol{c}_a \in (0, 1)$ for the predicted motion, interaction end-state indicator $\boldsymbol{e}_a \in (0, 1)$ (*e.g.*, the fully open/closed door) and a binary mask $\boldsymbol{m}_p \in \{0, 1\}^N$ indicating points that move coherently with the interaction point $\boldsymbol{p}$ (see Fig. 2(a)). Formally:

$$(\boldsymbol{M}_a, \boldsymbol{c}_a, \boldsymbol{e}_a, \boldsymbol{m}_p) = \Psi(\boldsymbol{S}, \boldsymbol{a}). \tag{1}$$

 https://doi.org/10.52202/085713-3920

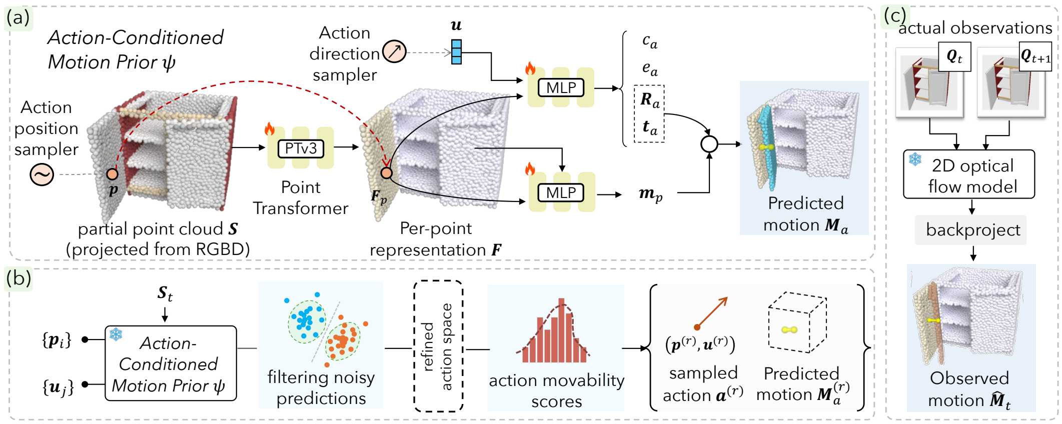

Figure 2: Overview of our approach. (a) We first train an action-conditioned motion prior on a set of articulated objects without anomalies (Sec. 3.2). At testing time, given the input RGBD image, we (b) exploit the learned prior to infer action-motion pairs, while predicting the normal motion (Sec. 3.3). At the same time, (c) an agent interacts with the object based on our inferred actions to obtain the observed motion, which is compared against the anticipated motion to reveal anomalies (Sec. 3.4).

The subscripts a and p indicate the conditioning on action and end-effector's position respectively. We omit timestep t since predictions are atomic–made per single timestep. The predicted motion $\boldsymbol{M}_a$ is a rigid transformation $\boldsymbol{T}_a \in \mathrm{SE}(3)$, comprising a rotation matrix $\boldsymbol{R}_a \in \mathrm{SO}(3)$ and a translation vector $\boldsymbol{t}_a \in \mathbb{R}^3$. We implement Ψ as a deep neural network comprising a backbone feature encoder and two decoders for motion estimation and segmentation respectively.

Backbone feature encoder. The initial point cloud observation $\boldsymbol{S}_0$ is derived from the RGBD image $\boldsymbol{Q}_0 \in \mathbb{R}^{H\times W}$. For each point, the RGB values and surface normals (estimated via KD-tree search [40]) are concatenated, resulting in an input feature set of size $N \times 6$ where $N = H \times W$. All the predicted $\boldsymbol{S}$ and observed states $\boldsymbol{Q}$ are represented in this format. A Point Transformer3 (PTv3) [37] processes the point cloud, extracting per-point features $\boldsymbol{F} \in \mathbb{R}^{N\times d}$, where d denotes the feature dimension.

Motion estimation head. Estimating rigid motion in the camera frame is ambiguous because the transformation matrix depends on the object's position in the scene, in other words, it is not translation-invariant. Following [1], we instead estimate rigid motion in a local coordinate frame centered at the action position $\boldsymbol{p}$, with axes aligned to the global frame. To this end, we concatenate the local feature $\boldsymbol{F}_p$ for $\boldsymbol{p}$ with $\boldsymbol{u}$, and process it through an MLP mapping network to produce a 11-dimensional vector which is further processed by using Gram-Schmidt process [45] to obtain a local 6D rotation $\boldsymbol{R}_a^L$ and 3D translation vector $\boldsymbol{t}_a^L$ respectively relative to $\boldsymbol{u}$. For optimizing the motion estimation, we use the following loss function:

$$\ell_M(\boldsymbol{a}) = \ell_{geo}(\boldsymbol{R}_a^L, \bar{\boldsymbol{R}}_a^L) + \lambda_{L2}\ell_{L2}(\boldsymbol{t}_a^L, \bar{\boldsymbol{t}}_a^L), \tag{2}$$

where ℓ_{geo} computes the geodesic distance [45] in the rotational space and ℓ_{L2} is the L2 distance. $\bar{\boldsymbol{R}}_a^L$ and $\bar{\boldsymbol{t}}_a^L$ are the groundtruth parameters of $\boldsymbol{R}_a^L$ and $\boldsymbol{t}_a^L$ respectively. λ_{L2} is a loss constant. The local rigid matrix is simply transformed into the global coordinate frame as follows:

$$\boldsymbol{R}_a = \boldsymbol{R}_a^L, \quad \boldsymbol{t}_a = (I - \boldsymbol{R}_a^L)\boldsymbol{p} + \boldsymbol{t}_a^L. \tag{3}$$

The end-state $\boldsymbol{e}_a$ is supervised using a binary cross-entropy loss denoted as $\ell_e(\boldsymbol{a}) = \ell_{\mathrm{bc}}(\boldsymbol{e}_a, \bar{\boldsymbol{e}}_a)$, where $\bar{\boldsymbol{e}}_a$ is the groundtruth for the end state.

Segmentation head. Here our goal is to predict what points in $\boldsymbol{S}$ will move together with the point $\boldsymbol{p}$ when force is applied on it. We use a binary classification head which takes in the local feature $\boldsymbol{F}_p$ after getting replicated N times along the sample dimension and concatenated with the point cloud representation $\boldsymbol{F}$ along the feature dimension, to provide global context. The resulting $N \times 2d$ dimensional feature is processed through an MLP mapping network to predict the mask $\boldsymbol{m} \in \{0, 1\}^{N\times 1}$. For optimizing the segmentation head, we use a binary cross-entropy loss ℓ_{bce} for each point $\boldsymbol{p}$ as follows:

$$\ell_m(\boldsymbol{p}) = \ell_{bce}(\boldsymbol{m}_p, \bar{\boldsymbol{m}}_p), \tag{4}$$

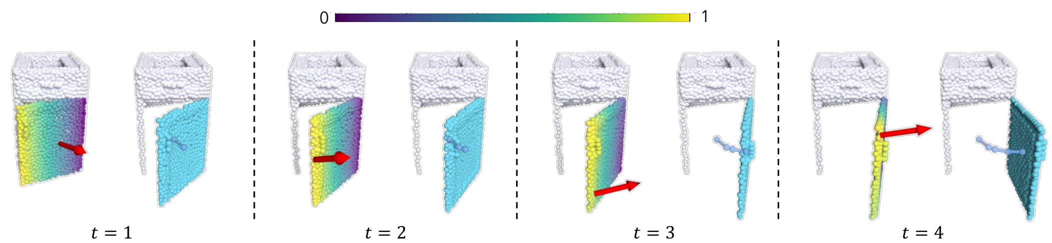

Figure 3: Anticipated trajectory visualization: for each timestep t, we show the movability map along with selected action, and predicted next state. The trajectory is shown for a single point. Movability scores are averaged over 250 sampled action directions and visualized using RGB mapping.

where $\bar{\boldsymbol{m}}_p \in \{0,1\}^N$ is the groundtruth mask.

Finally, the total loss function is a weighted sum of the motion and segmentation losses:

$$\ell_m(\boldsymbol{p}) + \lambda_M \ell_M(\boldsymbol{a}) + \lambda_e \ell_e(\boldsymbol{a}), \tag{5}$$

where λ_M and λ_e are loss constants.

Regressing rigid motion parameters is challenging, particularly under occlusion, where some action-induced motions are harder to predict. To address this, we use the predicted confidence score $\boldsymbol{c}_a$ to modulate the loss function in Eq. (5), following [19]:

$$\ell_m(\boldsymbol{p}) + \boldsymbol{c}_a(\lambda_M \ell_M(\boldsymbol{a}) + \lambda_e \ell_e(\boldsymbol{a})) - \lambda_c \log(\boldsymbol{c}_a), \tag{6}$$

where λ_c is a loss weight. When $\boldsymbol{c}_a$ is predicted to be low for a given action, the corresponding loss is down-weighted. The last logarithmic term is a regularizer and encourages the model to avoid setting low confidence for all samples trivially. We train the model end-to-end on normal objects using the total loss in Eq. (6), enabling it to learn normal motion patterns for arbitrary actions.

3.3 Long-term Normal Motion Anticipation

We describe how the normal motion predictions for individual atomic actions are utilized to estimate the normal trajectory over multiple timesteps (see Fig. 2(b)). Analyzing anomalies over multiple timesteps is crucial, as revealing anomalies typically requires manipulating each part over its full action space (*e.g.*, verifying whether a drawer fully opens without detaching beyond its intended range) and assessing all object parts. To facilitate this, we introduce a part memory to track examined parts and propose a two-stage filtering strategy to select informative actions likely to induce motion in the part.

Part history. For each object, we use two memory masks, $\mathcal{B}$ and $\mathcal{B}^+$, each consisting of N binary elements. $\mathcal{B}$ records previously examined parts, where elements corresponding to the previously manipulated parts are set to 1, while $\mathcal{B}^+$ tracks the part currently being manipulated. Following interaction, the memory is updated as $\mathcal{B} \leftarrow \mathcal{B} \vee \mathcal{B}^+$. Since the mask of the next part is initially unknown, we initialize it as $\mathcal{B}^+ \leftarrow \mathbb{1} - \mathcal{B}$, where $\mathbb{1}$ is N-dimensional vector with all elements set to 1, ensuring already examined parts are excluded. After the first timestep, $\mathcal{B}^+$ is updated based on segmentation head predictions.

Action candidate selection. Directly regressing actions is challenging, so we propose a robust strategy to select informative actions. At each time step, we define the action space for an articulated part r as $\mathcal{A}^{(r)} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{N_r}\} \times \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{N_u}\}$, where $\{\boldsymbol{p}_i\}$ denotes N_r points sampled from the mask $\mathcal{B}^+$ on the part r, and $\{\boldsymbol{u}_j\}$ are N_u directions uniformly sampled in the SO(3) space. We begin by estimating the 'normal' motion $\boldsymbol{M}_{\boldsymbol{a}_{ij}}$ for each pair $\boldsymbol{a}_{ij} = \{\boldsymbol{p}_i, \boldsymbol{u}_j\}$ using Eq. (1).

Given the inherent noise in motion prediction, we implement a two-stage filtering process. First, actions with low confidence scores (<10%) are discarded to eliminate unreliable predictions. Next, we apply a density-based outlier removal strategy by extracting and normalizing the rotation and translation axes from the predicted rigid motion matrices. These unit vectors are then clustered using DBSCAN [12], grouping similar rotations (and translations) while identifying sparse or inconsistent vectors as outliers. For revolute joints, this approach yields two clusters (forward and backward rotations), effectively filtering out noisy predictions.

https://doi.org/10.52202/085713-3920

To ensure motion consistency with the previous timestep, we leverage the clustering results to exclude actions that would induce the opposite motion, refining the action space to $\hat{\mathcal{A}}^{(r)}$. Subsequently, an action is selected based on a movability score $\Omega(\boldsymbol{a})$, which is derived from the predicted rigid transformation matrix $\boldsymbol{T}_a$ and defined as:

$$\Omega(\boldsymbol{a}) = \|\mathbb{I} - \boldsymbol{T}_a\|_F \,, \tag{7}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and $\mathbb{I}$ is the 4×4 identity matrix. Using $\Omega(\boldsymbol{a})$, we define a normalized score $\hat{\Omega}(\boldsymbol{a})$ as:

$$\hat{\Omega}(\boldsymbol{a}) = \frac{\exp\left(\Omega(\boldsymbol{a})/\lambda_T\right)}{\sum_{(\boldsymbol{a})\in\hat{\mathcal{A}}^{(r)}} \exp\left(\Omega(\boldsymbol{a})/\lambda_T\right)}, \tag{8}$$

where λ_T is a temperature parameter. We use $\hat{\Omega}(\boldsymbol{a})$ as a probability distribution to sample action $\boldsymbol{a}^{(r)} = (\boldsymbol{p}^{(r)}, \boldsymbol{u}^{(r)})$ from $\hat{\mathcal{A}}^{(r)}$. We use the predicted motion $\boldsymbol{M}_a^{(r)}$ to derive the next state $\boldsymbol{S}_{t+1}$, and iterate the above process again. The interaction terminates based on the end-state token $\boldsymbol{e}_t$ or when T_{max} is reached. We concatenate the predicted motion over all timesteps to obtain the discrete 'normal' trajectory and denote it as $\boldsymbol{\zeta}_{1:T}^{(r)}$. Fig. 3 shows an example of the anticipated trajectory starting from an initial observation.

3.4 Measuring Observed Trajectory Error

The agent sequentially executes $\{\boldsymbol{a}_t\}_{t=0}^{T-1}$, originally computed for the normal motion anticipation, for each part, generating the actual observation $\{\boldsymbol{Q}_t\}_{t=1}^{T}$. The kinematic change between $\boldsymbol{Q}_t$ and $\boldsymbol{Q}_{t+1}$ is estimated as rigid transformations denoted as $\hat{\boldsymbol{M}}_t$ (see Fig. 2(c)). To compute $\hat{\boldsymbol{M}}_t$, we omit the depth channels of $\boldsymbol{Q}_t$ and $\boldsymbol{Q}_{t+1}$, retaining only their RGB values, and apply an optical flow method [35] to predict 2D motion flow. Given that the camera remains fixed during interactions, we backproject the 2D motion flow into 3D using depth information to obtain 3D motion flow vectors. We compute both forward and backward flows and derive their respective transformation matrices using the Kabsch algorithm [17]. To enhance robustness, we iteratively refine the set of flow vectors using RANSAC while enforcing bidirectional flow consistency. This process yields a reliable set of 3D motion vectors for the observed flow. The transformations $\hat{\boldsymbol{M}}_t$ are accumulated over all timesteps for part r and the observed trajectory $\hat{\boldsymbol{\zeta}}_{1:T}^{(r)}$ is computed.

Finally, the anomaly label y_r for part r is assigned by comparing the expected and observed motion trajectory using the Hausdorff distance D_H [7]:

$$y_r = \begin{cases} 1 & \text{if } \Delta_r > \epsilon, \\ 0 & \text{otherwise,} \end{cases} \text{ where } \Delta_r = D_H(\boldsymbol{\zeta}_{1:T}^{(r)}, \hat{\boldsymbol{\zeta}}_{1:T}^{(r)}). \tag{9}$$

We choose the Hausdorff distance for its sensitivity to the worst-case deviation which is beneficial for identifying anomalies effectively. Importantly, our model is not trained explicitly trained to optimize the prediction rule in Eq. (9).

4 Experiment

4.1 Interactive AD Benchmark

We introduce a new benchmark for interactive AD: the *PartNet-IAD* dataset. To construct the dataset, we use objects from the PartNet-Mobility dataset [38] as normal instances, and modify their physical and kinematic structures to generate anomalous articulations, which are used exclusively for evaluation. More details about dataset statistics, anomaly generation process, and anomaly types are provided in the supplementary. We focus on two evaluation settings: i) evaluate our model on the test set of the training categories to measure its generalization ability to unseen objects within the same categories, and ii) evaluate on the test set of the testing categories to measure the generalization ability to unseen object categories. For training the normal motion anticipation module, we use a mutually exclusive set of 402 normal objects (from the train set of the training categories of PartNet-Mobility).

Table 1: **PartNet-IAD benchmark**. AUROC (%) of our framework compared against baselines.

	Unseen objects in training categories												Testing categories										
												Average											Average
A: No-interaction + Cls	60.3	61.7	58.5	64.0	58.9	60.3	63.5	57.1	60.1	64.0	65.3	61.3	51.7	56.1	52.0	56.7	58.3	56.0	56.2	58.0	59.1	51.3	55.5
B: Random-interaction + Cls	64.1	63.4	63.6	67.1	65.3	63.4	63.1	58.0	61.1	64.6	63.2	63.3	53.6	55.7	52.8	57.8	59.1	56.2	55.6	59.0	61.8	50.1	56.1
C: Where2Act + Cls	70.5	61.5	67.1	70.1	72.0	66.1	69.1	62.9	63.4	63.6	64.0	66.2	58.8	60.5	61.4	61.8	61.0	63.3	62.7	64.3	70.8	55.3	62.0
D: Where2Act + Heuristics	79.4	63.2	76.5	72.8	80.7	75.6	77.7	66.2	73.3	70.3	78.1	73.9	66.7	76.3	71.4	73.7	70.6	76.2	73.9	70.6	63.8	59.5	70.1
E: Where2Act + MotionPrior	87.5	72.2	80.5	82.7	84.1	81.2	86.7	73.9	85.8	70.1	84.4	80.9	74.5	83.5	80.3	81.7	75.0	81.5	79.0	84.6	70.0	63.8	77.3
Our Method	**91.3**	**75.3**	**88.9**	**85.3**	**92.6**	**87.5**	**89.8**	**77.8**	**85.0**	**82.8**	**90.2**	**86.1**	**78.9**	**91.6**	**83.2**	**84.4**	**83.1**	**88.4**	**89.1**	**87.4**	**80.6**	**71.9**	**83.9**

4.2 Implementation Details

Environment and action setting. We create interactive simulated environments by using Pybullet [9]. For the agent, we use a suction-based flying gripper [40] as the robot actuator. The flying gripper can be initialized at any position and orientation, with the gripper either closed or open. The object is observed using an RGBD camera with known intrinsics, positioned 5 units away from the object and facing its center. The camera is placed on the upper hemisphere with a random azimuth $[120°, 270°)$ and altitude $[25°, 40°]$, primarily capturing the front view of the object in the canonical frame of the original PartNet-Mobility objects. To execute an action, the agent first moves its end-effector to the sampled 3D position, aligning its orientation perpendicular to the object surface. In practice, the closest point on the surface is used to place the end-effector. The agent then moves the end-effector 0.18 meters along the action direction. The suction behavior is implemented as a force constraint between the suction cup and the selected 3D position on the object, as described in [40].

Collecting interaction data for training. To create a large set of action-motion interaction pairs, we randomly select an articulated part for each normal object and initialize its pose with a 50% chance at its rest state (*e.g.*, fully closed drawer) or a random pose (*e.g.*, a half-opened drawer). We then sample a 3D position on the part's surface and an action direction uniformly from a spherical distribution. The robot arm executes the action, and we record the resulting motion as a rigid transformation matrix along with its segmentation mask. Additionally, an end-state token is stored if the action leads to an end state (fully closed or fully open). For non-movable parts, we record interaction data for 5% of points on these parts, storing their resulting motion as an identity matrix. The offline dataset consists of 145M interaction pairs, generated over 3-4 days on a single 64-core CPU machine by parallelizing the simulation across multiple CPU cores, to kickstart the training. To complement the offline dataset, online data sampling [28] (see supplementary) is introduced after the 10th epoch, with each batch consisting of 70% offline data and 30% online data, focusing on interactions more likely to induce motion.

Inference details. During inference, we use the current memory mask $\mathcal{B}^+$ to sample $N_p = 1000$ points for the starting interaction of each part and $N_p = 250$ points after the first timestep once the part is identified. We sample $N_u = 128$ action direction to form the total action space. The temperature λ_T is set to 0.3 in Eq. (8). T_{max} is set to 15.

4.3 Baselines and Results

Since we are the first to propose and formalize this task, no prior work exists for direct comparison. To evaluate our method and establish benchmarks for the proposed task, as shown in Tab. 1, we compare against five alternative approaches (A-E) specifically designed for the task:

A: *No-interaction + Classifier* takes an initial observation frame as input and performs anomaly classification without any interaction.

B: *Random-interaction + Classifier* selects random action positions and directions (uniformly sampled from the action space) at each timestep for interaction using the simulator, followed by frame-by-frame anomaly classification on the observed frames. We perform same number of steps (15) as ours and if a single frame during the interaction is flagged as anomaly by the classifier we consider the articulation as anomaly.

C: *Where2Act + Classifier* employs a learned action prediction model (*e.g.*, Where2Act [28]) to predict the action parameters and pass them to the simulator to obtain the observed frames, replacing the random action sampling.

 https://doi.org/10.52202/085713-3920

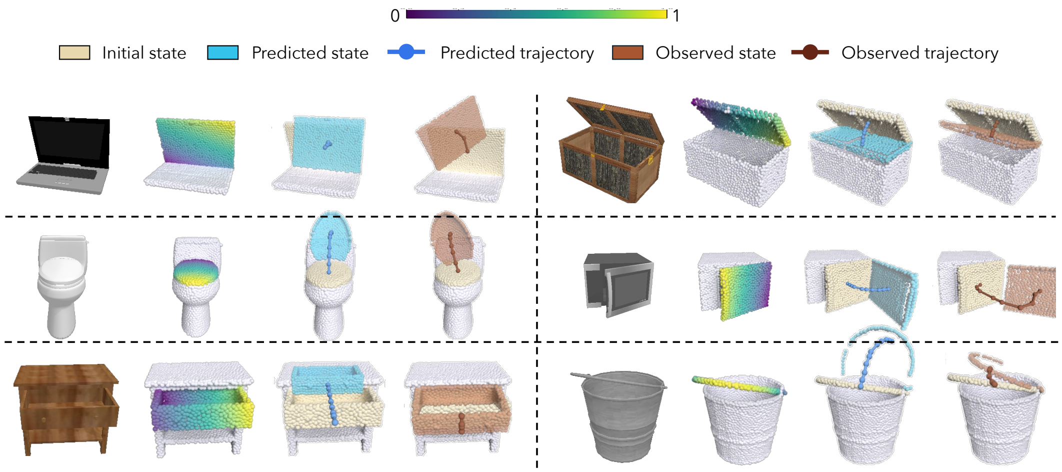

Figure 4: Qualitative results of interactive AD: Each block shows (left to right) the input RGBD, estimated movability map, and predicted vs. observed final trajectories. Top two rows are from training categories; the bottom row is from testing categories. Our model correctly detects anomalies in all cases. Color coding is shown above; full interaction videos are in the supplementary material.

D: *Where2Act + Heuristics* includes a separate model that is trained to infer the joint type and joint axis for each moving part. We then use Where2Act agent to predict action and the simulator executes the action. We then detect anomalies by comparing the observed motion axis with the predicted ones.

E: *Where2Act + MotionPrior* employs the Where2Act agent to predict the action and utilizes our action-conditioned motion prior to estimate normal motion, obtaining anticipated trajectories for the agent's actions, which are compared with the observed trajectories to detect anomalies.

For A-C, we implement a frame-by-frame anomaly classifier using a Point Transformer-based model. They take a single point cloud frame as input and predict an anomaly label. The classifier is trained using normal frames and augmented pseudo-anomaly frames, generated online via random deformations or transformations. For D, we use a Point Transformer for part segmentation, followed by a pooling operation over each segmented part to obtain the joint axis vector and joint type (revolute or prismatic). This model is trained on the normal objects from the training categories. For fair comparison, we train Where2Act using a suction-based gripper. Anomaly classification performance is evaluated using the area under the ROC curve (AUROC), computed per part and averaged across all test objects. Qualitative results of our framework are shown in Fig. 4.

Table 2: We evaluate our method using quantitative metrics for motion estimation (MSE and mIoU) and interaction performance (action success rate [ASR] and part success rate [PSR]). The evaluation is conducted on normal objects from the test categories.

Method	MSE ↓	mIoU ↑	ASR(%) ↑	PSR (%) ↑
w/o noise filtering	0.08	0.65	96.78	92.85
w/ noise filtering	0.05	0.72	98.35	94.12

Importance of interactive part analysis. We first validate our claim that reliable functional anomaly detection requires agent interaction. Baseline A, lacking interaction, misses anomalies not visible in the input frame. Baseline B, relying on random interactions, is ineffective since most actions either cause no motion or fail to reveal anomalies. In contrast, our method, which uses a learned agent for meaningful interactions, achieves significantly better performance.

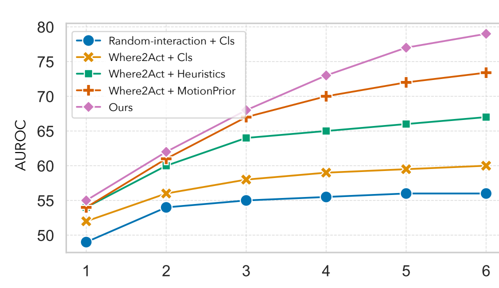

Figure 5: **Interaction steps vs. AUROC.** The results are evaluated on the testing categories.

Importance of our action-dependent framework. Baselines B and C are action-independent—they do not consider which specific action caused the irregular motion, often missing anomalies like a drawer failing to close.

Figure 6: **Real-world visualization of the motion prior network on the AKB-48 [26] dataset.** We used our trained motion prior model in PartNet-Mobility to estimate normal motion for the real-world scanned objects from the AKB-48 dataset. For each row, we show the input point cloud, the movability score map, and four sampled action-motion pairs, respectively. The sampling of action-motion pair is performed based on the normalized scores. Red arrows indicate the input actions, and the anticipated motions are shown in cyan.

Baseline `C` misses action-conditioned temporal cues due to frame-wise classification, while Baseline `D` predicts joint axes but ignores the magnitude of action-induced motion. In contrast, our final model conditions anomaly prediction on the executed action, resulting in significantly higher performance.

Importance of our joint action-motion representation. In Baseline `E`, we naively apply our motion anticipation module to Where2Act-predicted actions, yielding modest gains by estimating normal motion for comparison with the observed motion. However, it struggles with multi-step interactions. Without motion history, it often results in back-and-forth movements without progressing to new states. In contrast, our method uses predicted joint motion matrices to guide actions consistently in one direction—either fully closing or fully opening—enabling more reliable trajectory comparisons.

Interaction efficiency. Next we evaluate by plotting AUROC against interaction steps in Fig. 5. Our method continues improving with more interactions, while other baselines plateau after 2-3 steps. This highlights our pipeline's ability to perform meaningful interactions and effectively reveal hidden anomalies.

Methodological differences with Where2Act [28]. While Where2Act tackles a different problem, we highlight the key differences between our method and its architecture below: 1) Where2Act focuses on predicting which actions are likely to move an articulated part, but it does not model how the part moves. In contrast, our method estimates the rigid motion flow that fully characterizes the part's kinematic response to the action. This is crucial for our task, as it provides the anticipated motion necessary for comparison with the observed one. 2) Our formulation enables long-horizon, temporally consistent action exploration, which Where2Act lacks. 3) By explicitly estimating motion, our method can filter out noisy actions by identifying inconsistencies in predicted motions. 4) Our model leverages rich supervisory signals from simulation during training, including motion mask and rigid motion matrix tied to each input action, whereas Where2Act uses only discrete supervision (motion vs. no motion). This allows to capture the continuous range of articulation dynamics more accurately.

Additional metrics for motion estimation and interaction results. Table 2 reports additional quantitative metrics to assess two components of our framework. For the motion prior network, we evaluate motion prediction error (MSE) and part segmentation accuracy (mIoU). For interaction performance, we report action success rate (ASR) and part success rate (PSR). ASR measures the proportion of actions that successfully move a part–defined as achieving a displacement greater than 0.01 unit-length or 0.5 relative to its total motion range at any timestep. PSR indicates the percentage of parts that reach a fully open or closed state over multiple interaction steps. All evaluations are conducted on normal objects from the testing categories. We compare two variants of

https://doi.org/10.52202/085713-3920

our framework: one with and one without the noise filtering module. Results demonstrate that noise filtering significantly improves motion estimation and boosts interaction success rates.

Real-world visualization of the motion prior network. To evaluate the generalization ability of the learned motion prior, we use real-world articulated objects from the AKB-48 dataset [26]. The textures of the shapes in this dataset are scanned from real-world objects. Fig. 6 visualizes our normal motion estimation model (pretrained on the PartNet-Mobility dataset) applied to three categories from AKB-48: Bucket, Box, and Trashcan. As shown, our model performs favorably even on shapes with significant geometric differences and realistic textures, without any finetuning. For more results please refer to the supplementary.

4.4 Limitations and Failure Cases

We assume articulated parts undergo rigid transformations, meaning each part moves independently without affecting others. While this holds for most real-world objects, it does not apply to multi-link rigid systems or soft-body deformations. Our framework is generalizable to various joint types but the *PartNet-IAD* dataset is limited to 1-DoF prismatic and revolute joints. Additionally, it does not handle articulation requiring composite actions, such as locking a door (*e.g.*, turning a key and pushing a latch). Our method uses a suction-based end-effector for robust grasping across diverse objects, aligning with real-world robotics. However, it is unsuitable for tasks needing precise grasping, handling small objects, or gripping uneven surfaces. While our model generalizes across categories – *e.g.*, learning to open a safe from experience with microwaves due to similar revolute joints – it cannot handle entirely novel articulations (*e.g.*, different joint types). Finally, although our method is designed with real-world generalization in mind, our experiments are primarily conducted in simulation without using a real robot.

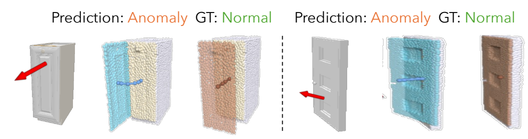

Figure 7: Failure cases. We show the input RGBD, anticipated motion and the observed motion. The applied action in first timestep is shown in red arrow. We use the same color coding as in Fig. 4

Fig. 7 highlights two failure cases due to ambiguity in their joint type and motion direction respectively. The left one is a false positive where the model misinterprets normal motion by assuming a revolute joint. In the right one, the door is designed to be pushed to open, while our model infers the wrong opening direction (*i.e.*, pulling out) for this ambiguous case, leading to inaccurate action and motion predictions.

5 Conclusion and Future Work

In this work, we introduced a novel problem, interactive AD for detecting functional anomalies in articulated objects through learned agent interactions. Unlike existing techniques that either lack interaction or rely on random actions, our approach actively explores the object's motion space to uncover hidden anomalies. By integrating a learned motion anticipation and action selection, our method achieves significantly higher performance across multiple evaluation metrics. Our findings demonstrate the importance of intelligent interaction in anomaly detection, paving the way for more advanced robotic perception systems capable of understanding complex articulated structures.

Broader Impacts. This work enables robotic agents to actively detect functional anomalies in articulated objects, with potential benefits in quality control, assistive robotics, and autonomous inspection. Those may include furniture (*e.g.* for cabinets) and consumer electronics (*e.g.* for laptops) industry for quality control and durability testing. It may also be used in product development and prototyping to identify design flaws and improve functionality, and, as well as, in service industry for inspecting and repairing articulated objects, and warranty assessments. While this technology advances safe and intelligent interaction, unintended damage from probing actions or generalization issues in real-world settings may pose safety and reliability concerns.

Acknowledgments. The project was funded by Toyota Motor Europe. HB was supported by the EPSRC Visual AI grant EP/T028572/1.

References

[1] Aseem Behl, Despoina Paschalidou, Simon Donné, and Andreas Geiger. Pointflownet: Learning representations for rigid motion estimation from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7962–7971, 2019.

[2] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019.

[3] Paul Bergmann, Xin Jin, David Sattlegger, and Carsten Steger. The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization. *arXiv preprint arXiv:2112.09045*, 2021.

[4] Paul Bergmann and David Sattlegger. Anomaly detection in 3d point clouds using deep geometric descriptors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2613–2623, 2023.

[5] Ankan Bhunia, Changjian Li, and Hakan Bilen. Looking 3d: Anomaly detection with 2d-3d alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17263–17272, 2024.

[6] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.

[7] Lei Chen, M Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *ACM SIGMOD*, 2005.

[8] Yang Cong, Junsong Yuan, and Ji Liu. Sparse reconstruction cost for abnormal event detection. In *CVPR 2011*, pages 3449–3456. IEEE, 2011.

[9] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016.

[10] Jianning Deng, Kartic Subr, and Hakan Bilen. Articulate your nerf: Unsupervised articulated object modeling via conditional view synthesis. In *Advances in Neural Information Processing Systems*, 2025.

[11] Ayush Dewan, Tim Caselitz, Gian Diego Tipaldi, and Wolfram Burgard. Rigid scene flow for 3d lidar scans. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1765–1770. IEEE, 2016.

[12] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. Density-based spatial clustering of applications with noise. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, 1996.

[13] Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. Act the part: Learning interaction strategies for articulated object part discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15752–15761, 2021.

[14] Yining Hong, Kaichun Mo, Li Yi, Leonidas J Guibas, Antonio Torralba, Joshua B Tenenbaum, and Chuang Gan. Fixing malfunctional objects with learned physical simulation and functional prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1413–1423, 2022.

[15] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. Registration based few-shot anomaly detection. In *European conference on computer vision*, pages 303–319. Springer, 2022.

[16] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12627–12637, 2019.

[17] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

[18] Dov Katz, Moslem Kazemi, J Andrew Bagnell, and Anthony Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *2013 IEEE International Conference on Robotics and Automation*, pages 5003–5010. IEEE, 2013.

 https://doi.org/10.52202/085713-3920

[19] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[20] Chad C Kessens, Joseph B Rice, Daniel C Smith, Stephen J Biggs, and Richard Garcia. Utilizing compliance to manipulate doors with unmodeled constraints. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 483–489. IEEE, 2010.

[21] Ellen Klingbeil, Ashutosh Saxena, and Andrew Y Ng. Learning to open new doors. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2751–2757. IEEE, 2010.

[22] Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. Dragapart: Learning a part-level motion prior for articulated objects. In *European Conference on Computer Vision*, pages 165–183. Springer, 2024.

[23] Wenqiao Li, Xiaohao Xu, Yao Gu, Bozhong Zheng, Shenghua Gao, and Yingna Wu. Towards scalable 3d anomaly detection and localization: A benchmark via 3d anomaly synthesis and a self-supervised learning network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22207–22216, 2024.

[24] Yancong Lin and Holger Caesar. Icp-flow: Lidar scene flow estimation with icp. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15501–15511, 2024.

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

[26] Liu Liu, Wenqiang Xu, Haoyuan Fu, Sucheng Qian, Qiaojun Yu, Yang Han, and Cewu Lu. Akb-48: A real-world articulated object knowledge base. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14809–14818, 2022.

[27] Martin Lohmann, Jordi Salvador, Aniruddha Kembhavi, and Roozbeh Mottaghi. Learning about objects by learning to interact with them. In *Advances in Neural Information Processing Systems*, volume 33, pages 3930–3941, 2020.

[28] Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6813–6823, 2021.

[29] Chuanruo Ning, Ruihai Wu, Haoran Lu, Kaichun Mo, and Hao Dong. Where2explore: Few-shot affordance learning for unseen novel categories of articulated objects. In *Advances in Neural Information Processing Systems*, 2023.

[30] Santhosh K Ramakrishnan, Dinesh Jayaraman, and Kristen Grauman. An exploration of embodied visual exploration. *International Journal of Computer Vision*, 129(5):1616–1649, 2021.

[31] Kanishka Rao, Chris Harris, Alex Irpan, Sergey Levine, Julian Ibarz, and Mohi Khansari. Rl-cyclegan: Reinforcement learning aware simulation-to-real. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11157–11166, 2020.

[32] Nicolae-C Ristea, Florinel-Alin Croitoru, Radu Tudor Ionescu, Marius Popescu, Fahad Shahbaz Khan, Mubarak Shah, et al. Self-distilled masked auto-encoders are efficient video anomaly detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15984–15995, 2024.

[33] Ziyang Song and Bo Yang. Ogc: Unsupervised 3d object segmentation from rigid dynamics of point clouds. In *Advances in Neural Information Processing Systems*, volume 35, pages 30798–30812, 2022.

[34] 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.

[35] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[36] Ruihai Wu, Yan Zhao, Kaichun Mo, Zizheng Guo, Yian Wang, Tianhao Wu, Qingnan Fan, Xuelin Chen, Leonidas Guibas, and Hao Dong. Vat-mart: Learning visual action trajectory proposals for manipulating 3d articulated objects. *arXiv preprint arXiv:2106.14440*, 2021.

[37] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024.

[38] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11097–11107, 2020.

[39] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[40] Zhenjia Xu, Zhanpeng He, and Shuran Song. Universal manipulation policy network for articulated objects. *IEEE robotics and automation letters*, 7(2):2447–2454, 2022.

[41] Zhenjia Xu, Jiajun Wu, Andy Zeng, Joshua B Tenenbaum, and Shuran Song. Densephysnet: Learning dense physical object representations via multi-step dynamic interactions. *arXiv preprint arXiv:1906.03853*, 2019.

[42] Yiru Zhao, Bing Deng, Chen Shen, Yao Liu, Hongtao Lu, and Xian-Sheng Hua. Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1933–1941, 2017.

[43] Jia-Xing Zhong, Ta-Ying Cheng, Yuhang He, Kai Lu, Kaichen Zhou, Andrew Markham, and Niki Trigoni. Multi-body se (3) equivariance for unsupervised rigid segmentation and motion estimation. In *Advances in Neural Information Processing Systems*, volume 36, pages 76085–76097, 2023.

[44] Qiang Zhou, Weize Li, Lihan Jiang, Guoliang Wang, Guyue Zhou, Shanghang Zhang, and Hao Zhao. Pad: A dataset and benchmark for pose-agnostic anomaly detection. In *Advances in Neural Information Processing Systems*, 2023.

[45] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.

[46] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision*, pages 392–408, 2022.

 https://doi.org/10.52202/085713-3920

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Unlike traditional anomaly detection methods that rely on passive visual observations, our proposed anomaly detection task requires agent interaction to detect functional anomalies that would otherwise remain hidden. Experiments shown in Table 1 of our proposed PartNet-IAD benchmark demonstrate strong generalization in detecting anomalies across both seen and unseen object categories.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations are discussed in Section 4.4.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The paper introduces a new benchmark and a new model. The dataset is derived from existing public articulated shape dataset and the generation procedure is explained in detail in the main text and supplementary material. Similarly, we provide the required details to construct, train, and test the models.

Guidelines:

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

https://doi.org/10.52202/085713-3920

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: We will release the dataset and model, the source code that is used to generate the data and to train the models upon publication.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide these details under Section 4.2, Appendix A.1 and Appendix A.2.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We discussed the statistical significance of the experiments in Appendix A.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We have provided detailed computing resources for training in Appendix A.2.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: There are no human subjects or participants in this research and there are no data-related concerns either.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We have included a paragraph in Section 5 to discuss both positive and negative potential impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

https://doi.org/10.52202/085713-3920

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper poses no such risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All the existing assets are properly cited and their licenses are explicitly mentioned in Appendix A.1.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: Our new assets including the network training and testing codes, and the new dataset will be released under the Attribution-NonCommercial-ShareAlike 4.0 International license upon publication.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

https://doi.org/10.52202/085713-3920

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs.

    Guidelines:

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