

AOMGen: Photoreal, Physics-Consistent Demonstration Generation for Articulated Object Manipulation

Yulu Wu¹, Jiujun Cheng^{1,†}, Haowen Wang^{2,†}, Dengyang Suo¹, Pei Ren³, Qichao Mao¹, Shangce Gao⁴, Yakun Huang⁵

¹Tongji University ²Anhui University ³Beijing Innovation Center of Humanoid Robotics
⁴University of Toyama ⁵Beijing University of Posts and Telecommunications

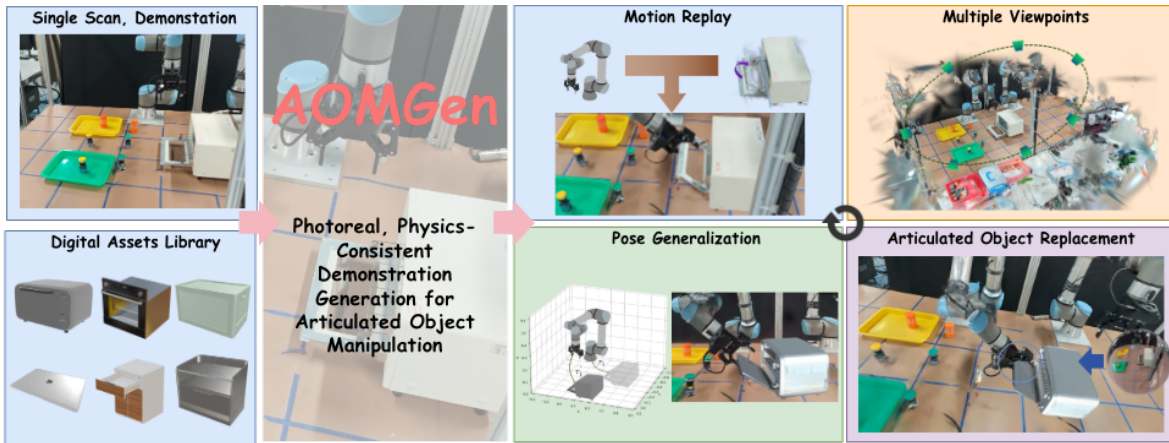


Figure 1. As a powerful articulated object manipulation data generator, the proposed AOMGen generates visually realistic and interaction-accurate data for any object of the same category within a unified framework. At the same time, the generated data provides effective assistance in improving the model’s performance.

Abstract

Recent advances in Vision-Language-Action (VLA) and world-model methods have improved generalization in tasks such as robotic manipulation and object interaction. However, successful execution of such tasks depends on large, costly collections of real demonstrations, especially for fine-grained manipulation of articulated objects. To address this, we present AOMGen, a scalable data generation framework for articulated manipulation which is instantiated from a single real scan, demonstration and a library of readily available digital assets, yielding photoreal training data with verified physical states. The framework synthesizes synchronized multi-view RGB temporally aligned with action commands and state annotations for joints and contacts, and systematically varies camera viewpoints, object styles, and object poses to expand a single execution into a diverse corpus. Experimental results demonstrate that fine-tuning VLA policies on AOMGen data increases the success rate from 0% to 88.7%, and the policies are tested on un-

seen objects and layouts.

1. Introduction

Robotic manipulation models based on Vision-Language-Action (VLA), which integrate visual perception, language understanding, and action execution, have demonstrated remarkable generalization capabilities across various tasks [9, 16, 24]. Achieving such capabilities typically requires substantial amounts of high-quality robot manipulation demonstration data collected from real-world interactions [5, 14, 33]. In particular, fine-grained manipulation tasks involving articulated objects rely more heavily on diverse and precise data compared to other tasks, as their complex kinematic constraints require richer demonstrations for learning effective control policies [15, 36]. Nevertheless, collecting sufficient high-quality demonstration data remains expensive, labor-intensive, and inherently lim-

[†] Corresponding authors: Jiujun Cheng (chengjj@tongji.edu.cn), Haowen Wang (wanghaowen@ahu.edu.cn).

ited in scenario coverage, making it challenging to obtain adequate data for these tasks.

To overcome the limitations posed by the inefficient collection of real-world robot manipulation data, existing research has primarily explored physics-based simulation and video-driven world modeling as two mainstream strategies. Physics-based simulation platforms [1, 22, 32] efficiently generate large, state-action aligned demonstrations, reducing manual annotation and hardware costs [17, 20], which have enabled several methods to achieve promising results by leveraging simulated data. However, simulation environments still exhibit significant gaps in visual realism compared to real-world scenes, posing substantial challenges for Sim-to-Real transfer [11, 18]. Recent video world model approaches enhance data realism by directly learning from large-scale real-world videos [10, 38, 41, 43]. Compared with purely simulation-based methods, they offer higher visual fidelity and diversity while reducing reliance on hand-crafted simulation assets. However, these models often provide insufficient supervision over physical realism and action executability, leading to physically inconsistent interactions. Hence, for generating visually realistic and physically stable demonstration data, DemoGen [36] and R2RGen [35] synthesize additional robot manipulation demonstrations from limited real-world examples, including spatially augmented end-effector trajectories and 3D visual observations. However, they have several critical limitations: (1) limited to simple grasping and placement, unable to handle fine-grained articulated manipulation; (2) fixed object appearances and geometries, restricting generalization to novel objects or poses; and (3) single-view inputs, reducing visual realism in multi-view observations. These limitations underscore the need for methods capable of generating more visually realistic and physically consistent demonstrations for articulated object manipulation tasks.

To address the above limitations, we propose AOMGen, a novel framework that synthesizes photorealistic and physically consistent demonstration data for category-level articulated object manipulation from a single real-world scan and demonstration (see Fig. 1). Unlike prior methods, AOMGen generalizes across spatial variations and enables object replacement within the same category, enabling significant scalability. By exploiting the shared kinematic and motion structures among category instances, AOMGen transfers manipulation behaviors to novel objects while maintaining visual realism and physical plausibility, without relying on physics simulators. AOMGen consists of two core modules: (1) *Scene Reconstruction and Motion Recovery*, which accurately reconstructs real-world manipulation scenes using 3D Gaussian Splatting (3DGS). In this module, we first segment Gaussian points from raw observations and align the 3DGS reconstruction with the real-world coordinate frame. Using real manipulation trajec-

tries as physical priors, we then recover accurate and physically consistent articulated motions, ensuring high-fidelity geometric alignment. (2) *Articulated Object Replacement with Pose Generalization*, which enables replacing original objects with new instances from the same category and simulating their corresponding interactions. Specifically, we establish mappings between the original and new object models to ensure correct articulation parameters, including joint configurations, sizes, and initial poses. We further enhance realism by transferring scene lighting and materials from the original scene to the new objects, producing visually realistic and physically plausible demonstrations for training manipulation policies.

In summary, our main contributions include:

- Using a single static scan video of an articulated object, AOMGen can generate manipulation data for other object in the same category.
- AOMGen promotes precise physical interactions and high visual realism in all the synthesized data.
- The architecture supports arbitrary adjustments to the target object’s pose, greatly expanding the diversity of configurations in the generated data and pushing the boundaries of generalization.
- The synthetic data produced by AOMGen has proven effective for VLA training, resulting in improved model performance.

2. Related Work

2.1. Robot Manipulation Data Generation

The efficient generation of high-quality robotic manipulation data is widely studied by researchers, due to the importance of visual data for policy learning.

MimicGen and its extensions [7, 12, 23] reconfigure the collected demonstration data to synthesize corresponding manipulation data. This provides a good generalization solution for seen objects; however, it fails to transfer to manipulation cognition for unseen objects.

Another line of work leverages LLM and VLM to generate manipulation data from a single image [4, 10, 38, 41, 43] or generalize existing observation data [42]. These methods have been effective in visual authenticity and efficiency, but due to the uncontrollability of the generative models, the physical authenticity of the generated data cannot be guaranteed. Therefore, AOMGen reconstructs and generalizes a real-world collected demonstration to ensure both its visual and physical performance are excellent.

2.2. Gaussian Splatting Reconstruction & Editing

3D Gaussian Splatting, as an explicit radiance field representation, possesses real-time rendering and interpretable editing capabilities. Some Real-to-Sim-Real pipelines leverage 3DGS to reconstruct scenes and reduce visual dis-

crepancy with the real world [6, 21, 29, 34]. Furthermore, to achieve physically realistic interactions, some works directly edit Gaussian points to reconstruct manipulation trajectories consistent with real-world demonstrations from humans [15, 39] or robots [26, 37]. Specifically, Sizhe Yang et al. [37] propose RobotSplat that leverages 3DGS to generate novel demonstrations. But RobotSplat fails to generalize the manipulation data for articulated objects.

Unlike the robot arm trajectory, the motion trajectory of the target object has no real recorded data for direct reference. Justin Kerr et al. [15] provide a part-level motion recovery method based on DINO features, but how to accurately transfer it to an unseen object remains an open problem. Therefore, we propose a motion transfer method that leverages the real robot arm trajectories to supervise Gaussian editing, thereby generating physically realistic and generalized robotic manipulation data.

On the other hand, Gaussian Editing [3] and Inpainting [8] are the key factors that determine the visual quality of the generated data. How to correctly segment the Gaussian points of the object to be replaced and enhance the material appearance of the new object to make it more consistent with the real-world environment will determine the rendering realism after Gaussian editing.

2.3. Articulated Objects Modeling

Compared to rigid objects, the motion of articulated objects is constrained by the joints, making their motion patterns more complex than simple transformations in the $SO(3)$ space. Recent works [19, 31] model articulated objects by using multi-view RGB images captured before and after interaction, thereby obtaining the geometric structure of articulated objects (such as the pose of the joints and different parts). However, the above methods fail in the absence of enough visual observations after interaction. Therefore, we propose a training-free articulated object modeling approach that explicitly captures a series of features by analyzing its geometric structure and incorporating robotic arm manipulation data.

3. Problem Definition

Given real articulated object manipulation data, which includes static scene scans V_{static} , dynamic manipulation video $V_{dynamic}$ and robotic arm joint states $A = \{A_i | i = 1, 2, \dots, T\}$, where T represents the total number of time frames in the process, along with several simulation 3D assets of the same category, the goal is to reconstruct the motion of the real manipulation scene and quickly replace the target object in the scene, as well as generalize its pose in any way to generate new observation-action pair manipulation data. We define articulated objects of the same category as objects that have similar relative joint positions within

the overall object (e.g., all joints are positioned below the object) and exhibit the same joint motion patterns.

4. Method

AOMGen requires only a single demonstration to generate large-scale articulated object manipulation data. An overview of our pipeline is shown in Fig. 2. In this section, we describe AOMGen in detail. First, we prepare for the reconstruction and pre-processing of the static scene based on 3DGS in Sec. 4.1. After preparing the static scene for Gaussian reconstruction, we segment and model the following in Sec. 4.2: 1) the target articulated object; 2) the robot arm. Using the known trajectory of the robotic arm, we obtain the motion patterns of the articulated object, thereby generating the dynamic scene represented by Gaussian points. Finally, in Sec. 4.3, we build a mapping between the original articulated object and the replacement object, and perform a series of optimizations at the visual level, to generate the generalized demonstration data.

4.1. Scene Reconstruction

To obtain a high-fidelity reconstruction of the scene, we capture a variety of perspectives of the manipulation scene. Once the images are ready, we use COLMAP to obtain sparse scene reconstruction and camera pose estimation.

4.1.1. Part-level Segmented 3DGS

Fine-grained segmentation is a necessary condition for editable GS scenes. Based on SAGA [2], each Gaussian point is attached with a feature f^D of dimensions D , which is learned from multi-view 2D masks extracted by SAM2 [27]. By adjusting the segmentation granularity, part-level image masks can be obtained.

Based on the segmented GS model G , we can obtain the different parts of the articulated object $Part_i \in G$. According to the position of the end-effector at different times, distinguish the movable part $Part_{move}$ and static parts $Part_{static}$ from $Part_i$.

4.1.2. Coordinate Alignment

The current GS coordinate system should be aligned with the real-world coordinate frame to ensure consistency in real-world, simulation and GS scenes. Given the robot URDF in the real-world coordinate frame, we can sample point clouds P_{urdf} from the mesh surface. Using Iterative Closest Point (ICP) [40] on P_{urdf} and P_{robot} , which are the robot points segmented from G , we can get the transformation matrix $T_{gs \rightarrow real}$ and the transformed 3DGS scene G' .

4.2. Motion Recovery

Motion recovery involves simulating two aspects: the robotic arm and the articulated object.

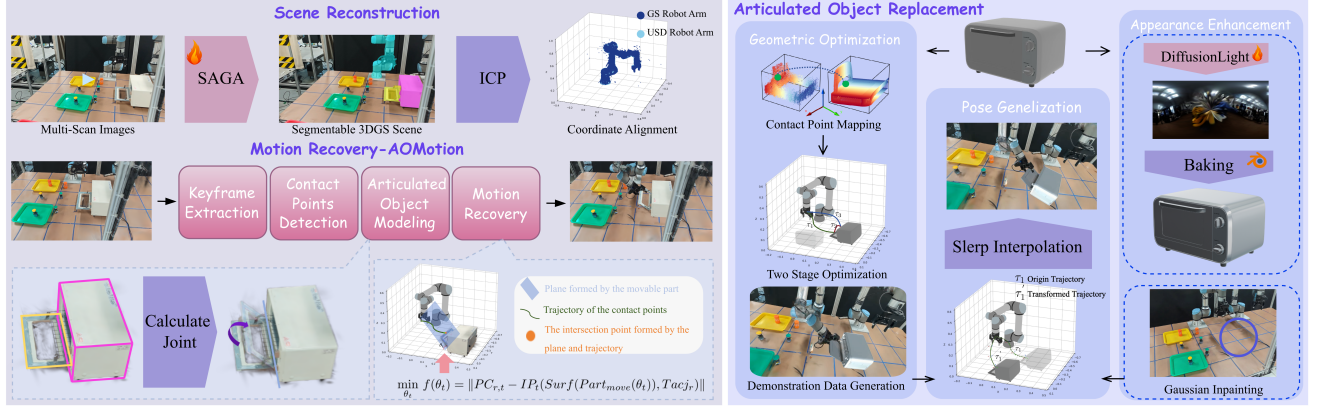


Figure 2. Pipeline of the proposed AOMGen, where a rotational joint object is used as an example to illustrate the complete pipeline, while a prismatic object can be handled in the same manner.

To recover the motion of the robotic arm, the Gaussian-splatting representation of the robotic arm is segmented using the bounding boxes of different links in the robot URDF, defined as $P_{robot}^l = T_{gs \rightarrow real} P_{robot}^l$, where P_{robot}^l denotes the points belonging to the l -th link of the robot arm. Given the collected demonstration, including the joint states, the transformation matrix $T_t^l(A_0, A_t)$ for link l can be calculated by Forward Kinematics, where A_0 and A_t represent the initial and joint states at time t .

On the other hand, to recover the articulated object’s motion without value recording, we design a supervised method **AOMotion**, based on the real-world interaction. AOMotion consists of four submodules: 1. keyframe extraction; 2. contact point detection; 3. articulated object modeling; 4. movable part motion recovery. These modules are described in the following.

4.2.1. Keyframe Extraction

Leveraging the dynamic manipulation video $V_{dynamic}$, we use SAM2 to generate the mask of the movable part $M_{dynamic}^t$ and the robot arm M_{robot}^t at frame t . We aim to determine whether the movable part starts or stops interacting by comparing the change between $M_{dynamic}^t$ and $M_{dynamic}^{t+1}$. However, since M_{robot}^t affects the shape of $M_{dynamic}^t$, the mask needs to be processed as follows during the comparison:

$$\begin{aligned} M^t &= M_{dynamic}^t - M_{robot}^{t+1} \\ M^{t+1} &= M_{dynamic}^{t+1} - M_{robot}^t, \end{aligned} \quad (1)$$

where M^t represents the processed mask. Then, we define the following *MotionScore* to measure whether the current

frame has started or finished moving:

$$\begin{aligned} d_t(p) &= \begin{cases} 0, & \text{if } p \text{ in } M^t \& M^{t+1} \\ 1, & \text{otherwise} \end{cases} \\ MotionScore_t &= \frac{\sum_{p \in M^t} d_t(p)}{\sum_{p \in M^t} 1}, \end{aligned} \quad (2)$$

where p represents the pixel in the mask, $d_t(p)$ represents whether the pixels within the mask are changing.

Moreover, due to the slight discrepancies in the SAM2 segmentation across different frames, we apply the Savitzky-Golay filter to smooth the motion scores and reduce the noise, resulting in the smoothed motion scores $SmoothScore_t$. A dynamic threshold is used to distinguish between motion and stillness. The baseline B for the motion scores is calculated from the 20th quantile of the motion scores. The noise standard deviation σ_{noise} is calculated based on the standard deviation of the motion scores. The final threshold is $\mu = B + 3\sigma_{noise}$.

Based on the smoothed motion scores and dynamic thresholds, we generate a motion label for each frame and get the first and the last motion frames, denoted as *StartFrame* and *EndFrame*, respectively:

$$\begin{aligned} motion_frames_t &= \begin{cases} 1, & \text{if } SmoothScore_t > \mu \\ 0, & \text{otherwise} \end{cases} \\ StartFrame &= \min\{t \mid motion_frames_t = 1\} \\ EndFrame &= \max\{t \mid motion_frames_t = 1\}, \end{aligned} \quad (3)$$

where $motion_frames_t$ indicates whether frame t is in motion. Fig. 3 presents the computation results of the Motion Score.

4.2.2. Contact Point Detection

Obtaining the contact point at the beginning of the interaction between the robotic arm and the movable part will

help us translate the motion of the articulated object into a rotational transformation at the contact point.

The pose of the end-effector at *StartFrame*, which is given by Eq. 3, is defined as $\mathbf{v}_{ee}$. We can obtain the contact point information on the end-effector and the movable part as follows:

$$PC_r = \arg \min_{PC_r \in P'_{r,t}} \left(\min_{PC_m \in Part'_{move}} ||PC_r - PC_m|| \right), \quad (4)$$

where PC_r represents the contact point on the robot arm, and $Part'_{move} = T_{gs \rightarrow real} Part_{move}$. The contact point on the movable part PC_m is computed in the same method.

4.2.3. Articulated Object Modeling

Given the movable part and static parts, obtaining the accurate direction and center of the joint is the key. First, construct the bounding boxes of the two parts, $B_{move} = \{e_{move,i} | i \in [1, 12]\}$ and $B_{static} = \{e_{static,i} | i \in [1, 12]\}$, where e_i represents the edge of the box.

The center and direction of the joint are typically closely related to the geometric position and orientation of the edges to which it is connected. By iterating over all pairs, we design the following scoring scheme to determine the paired edges adjacent to the joint:

$$EdgeScore = 0.8(1 - \text{parallelism}) + 0.2\text{distance} \quad (5)$$

$$\text{parallelism} = \left| \frac{\mathbf{e}_{static,i}}{||\mathbf{e}_{static,i}||} \cdot \frac{\mathbf{e}_{move,j}}{||\mathbf{e}_{move,j}||} \right|,$$

where *distance* is calculated by sampling points on both edges and summing the distances between the points.

However, when the two parts are in close contact, issues still arise. Assuming that the interaction location between the end-effector and the movable part is typically on the opposite side of the joint, we introduce the following criterion based on the above discussion:

$$EdgePair = \arg \min_{i,j} (\lambda_1 EdgeScore \pm \lambda_2 |\mathbf{e}_{move,j} \cdot \mathbf{v}_{ee}| \pm \lambda_3 \text{dist}(PC_m, \mathbf{e}_{move,j})), \quad (6)$$

where $\pm$ corresponds to rotational joints and prismatic joints, respectively. Then the joint direction is calculated by:

$$\text{joint_direction} = \frac{\mathbf{e}_{move, EdgePair[1]}}{||\mathbf{e}_{move, EdgePair[1]}||} \quad (7)$$

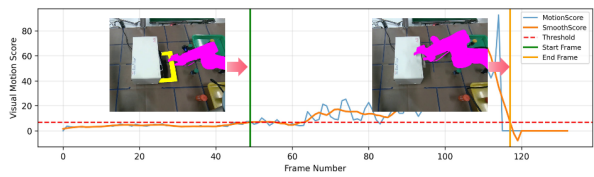


Figure 3. Computation of the Motion Score.

We obtain the joint center through nearest points matching. In detail, for $P_1 \in Part_{move}$, find the closest point $P_2 \in Part_{static}$ to establish a one-to-one mapping. We compute the joint center as the average of the midpoints of the K closest point pairs near the *EdgePart*:

$$\text{joint_center} = \frac{1}{K} \sum_{k=1}^K \left(\frac{P_{1,k} + P_{2,k}}{2} \right), \quad (8)$$

where:

$$\begin{aligned} [P_{1,k}, P_{2,k}] &= \text{Sort}(|P_1 - P_2|, \text{for} \\ P_1 &\in Part_{move} \& \text{dist}(P_1, \mathbf{e}_{move, EdgePair[1]}) < \epsilon, \quad (9) \\ P_2 &\in Part_{static} \& \text{dist}(P_2, \mathbf{e}_{static, EdgePair[0]}) < \epsilon. \end{aligned}$$

4.2.4. Movable Part Motion Recovery

For articulated objects, the motion of the movable part can be described as the physical parameters. We use the trajectory of the contact points $Traj_r$ to supervise the movable part:

$$\min_{\theta_t} f(\theta_t) = ||PC_{r,t} - IP_t(\text{Surf}(Part_{move}(\theta_t)), Traj_r)||, \quad (10)$$

where $IP_t(\text{Surf}(Part_{move}(\theta_t)), Traj_r)$ calculates the intersection point between the plane formed by the surface of the movable part, which rotated by θ_t degrees or moved a distance of θ_t , at time t and the trajectory $Traj_r$.

4.3. Articulated Object Replacement

Providing the same category of articulated object AO_{new} , we propose a novel method to process the geometry and appearance of AO_{new} , as illustrated in the right part of Fig. 2. After processing the given articulated object asset, we convert the USD file into a 3DGS representation [28]. Then, the motion simulation of the new movable part can be obtained using the method from Sec. 4.2.4. Moreover, to expand the generalization boundary, the pose of the replacement object can be arbitrarily modified.

4.3.1. Physical Interaction Adaptation

Given the replacement digital asset for the articulated object AO_{new} , we design a two-stage optimization approach using contact points for supervision to perform geometric processing, ensuring the physical interaction between the new object and the original robotic arm trajectory is consistent and reasonable.

To get the accurate contact point of AO_{new} , a contact point mapping method inspired by NOCS [30] is proposed. Specifically, the point clouds $Part_{move}$ and $Part_{move}^{new}$ (obtained by sampling points from the mesh surface) are normalized to a unit cube space. Then, the relative position of PC_{move} in the normalized space is projected onto the surface of the movable part of the newly normalized articulated

object. Finally, by applying denormalization, the contact point information of the new articulated object PC_{map} is obtained.

(1) For Stage 1, the following optimization function is used to obtain a rough estimate of the geometric features of the new articulated object:

$$\min_{\mathbf{g}} f_1(\mathbf{g}) = \sum_{t=StartFrame}^{EndFrame} \|PC_{r,t} - R_t(PC_{map}|\mathbf{g})\|, \quad (11)$$

where $\mathbf{g} = [s, r_{init}, offset_{\{x,y\}}]$ represents the scaling, initial motion parameter of the movable part, and the XY-axis offset in the original object's coordinate system with respect to AO_{new} . Assuming the movable part moves at a constant speed, $R_t(PC_{map}|\mathbf{g})$ represents the contact point position of PC_{map} after being transformed by $(r_{init}/(end_frame - start_frame)) * t$ in the AO_{new} , following the application of the $\mathbf{g}$ parameters. It is worth noting that $R_t(\cdot)$ corresponds to rotational motion for rotational joints and translational motion for prismatic joints.

(2) However, there are two issues in real-world operations that lead to optimization errors: 1. The movement of the movable part is not uniform. 2. There is some sliding between the end-effector contact point and the surface of the movable part, causing the trajectory of PC_r denoted as $Traj_r$ to not be an arc. Therefore, we add a stage two optimization to minimize this error as much as possible. Like Eq. 10, the specific optimization objective is as follows:

$$\min_{\mathbf{g}} f_2(\mathbf{g}) = \sum_{t=StartFrame}^{EndFrame} \|PC_{r,t} - IP_t(Surf(Part_{move}(\mathbf{g})), Traj_r)\|, \quad (12)$$

where $Part_{move}(\mathbf{g})$ stands for the point cloud of the movable part after the $\mathbf{g}$ transformation. Thus, we obtain the optimized geometric parameters $\mathbf{g}^*$ and apply them to the USD AO_{new} .

4.3.2. Visual Enhancement

At the same time, the visual performance of the replaced object is also crucial for the reality of the generated data. We apply the lighting changes from the real scene to the USD. First, we use DiffusionLight [25] to extract the lighting from the real environment. Then, the ambient lighting is baked onto the object's material in Blender. Moreover, Gaussian Inpainting is applied to handle the Gaussian holes caused by object replacement.

4.3.3. Pose Generalization

To further expand the generalization boundary, the pose of the articulated object can be arbitrarily transformed, denoted as T_{ao} . Begin with the existing trajectory of the end effector $\tau \in R^{7 \times T}$, and divide it into three stages $\tau_{1,2,3}$

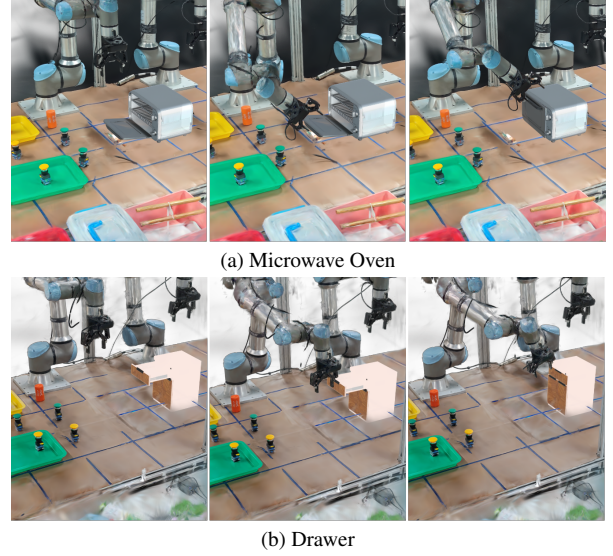


Figure 4. The Gaussian field visualizations of the data generated from AOMGen.

based on the $StartFrame$ and $EndFrame$. Since τ_2 is relatively invariant with respect to the object, the new trajectory for the second stage is $\tau'_2 = T_{ao}\tau_2$. Given the trajectory $\tau_1 = p_{start}, \dots, p_{end}$, the new trajectory has the same starting pose p_{start} but a new ending pose p_{end}^* . The translation of the new trajectory at different frames is obtained using linear interpolation, while the rotation angles are obtained through spherical linear interpolation. After obtaining the new trajectory, the corresponding joint angle transformations can be obtained through inverse kinematics.

5. Experiments

We conduct a series of experiments to validate the effectiveness of the data generated by AOMGen, for rotational joints and prismatic joints. Figure 4 shows the Gaussian field visualizations after replacing the rotational and prismatic joint objects. Specifically, the experimental results address the following three questions:

- Does the data generated by AOMGen align with real physical interactions?
- Can the manipulation data generated from AOMGen be directly used for VLA training?
- Can the generalization strategy enhance the robustness of the VLA model?

Next, detailed experimental settings and result analysis will be provided to answer the above questions.

5.1. Experimental Setup

We collect the real demonstration data using a Universal Robot UR5e equipped with a 2F85 gripper. A mobile device

Replacement Object	Microwave Oven	Tool Box	Computer	Drawer	Cabinet	Average
Success Rate	98%	96%	96%	100%	100%	98%

Table 1. Success Rate in simulation replay for different articulated objects, including rotational joint and prismatic joint.

is responsible for scanning the static scene, while a fixed camera records RGB images of the manipulation process.

To demonstrate the robustness of AOMGen in handling a variety of articulated objects, we select three different simulation assets with rotational joint, including microwave ovens, tool boxes and computers, and two assets with prismatic joint, including drawer and cabinet from ArtVIP [13] for substitution. During pose generalization, the translation range is $[-0.05m, 0.3m] * [-0.05m, 0.05m]$, and the rotation range is $[-45^\circ, 45^\circ]$.

The model $\pi_{0.5}$ [9] is fine-tuned using LoRA, with batch size 16 and learning rate 5×10^{-5} , employing cosine decay with a 10K step warm-up. The training consists of 30,000 steps. The action prediction horizon is set to 5 steps, and precision is configured with bfloat16 (frozen) and float32 (trainable). The model OpenVLA [16] is fine-tuned by LoRA using 50,000 steps, enabled with a rank of 32. Moreover, we use IsaacSim as our simulation platform.

5.2. Simulator Replay

To address Q1, we replay the robotic arm trajectory and the target articulated object’s pose generated from the data to validate the physical interaction realism in the simulator. Specifically, we import the optimized USD assets and their corresponding poses in Sec. 4.3 into the simulator, while replaying the robot arm’s joint states at different time steps, and observe whether the corresponding tasks are completed. We generate 50 data for each replacement object entries with only pose changes, and their Success Rate (SR) in simulation is calculated in Table 1. The results of the simulator replay demonstrate that the generated data adheres to basic physical laws, providing a fundamental guarantee for the effectiveness of subsequent model training. AOMGen can generate physically plausible demonstration data with various replacement objects of the same category. Fig. 5 shows the replay process of a data sample.

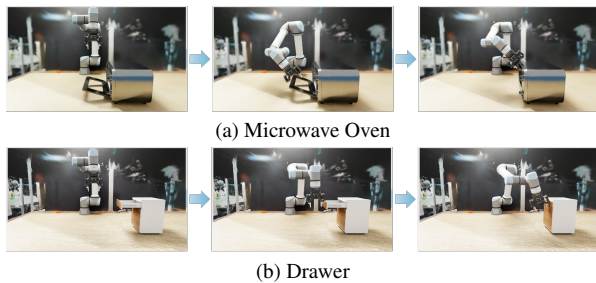


Figure 5. Replay of generated data in the simulator.

5.3. VLA Training

To answer Q2, we compare three policies that are respectively trained on demonstrations generated for each replacement object:

1. The original model, without any data fine-tuning.
2. Fine-tuned using 50 samples generated by AOMGen.
3. Fine-tuned using 150 samples generated by AOMGen.

For each replacement object, during the inference phase, we import the simulation assets of the robotic arm, the articulated object, and the table. To reduce the gap between the simulation environment and the real world, we add a background wall in the simulation environment and apply 3DGS render images, which have already undergone coordinate frame alignment, as textures on the background wall to simulate the real operating scene. To ensure the validity of the experimental results, we conduct experiments on two different models: $\pi_{0.5}$ [9] and OpenVLA [16]. In addition, each evaluation is conducted with 30 trials.

As shown in Fig. 6, for each model, as the number of generated demonstrations increases, a remarkable improvement in success rate is observed. Specifically, $\pi_{0.5}$ achieves a performance of 88.66, while OpenVLA achieves 81.34, which reflects that after fine-tuning with AOMGen-generated articulated object manipulation data, the model has acquired the ability to manipulate the corresponding object.

5.4. Robustness Analysis

Due to the differences between 3DGS rendered images and the simulator environment, the experimental results in Fig. 6 already reflect, to some extent, the model’s robustness to the environment. At the same time, the models are tested under different initial object position configurations, demonstrating their robustness to pose variations. Moreover, it validates the fundamental differences between the data generated by AOMGen and the data generated by the simulator. In this section, we describe how the powerful generalization capability of AOMGen is reflected in the model’s robustness. Illustrative examples of the VLA robustness experiments in various aspects are shown in Fig. 7.

For rotational joints and translational joints, we follow the fine-tuning method described in Sec. 5.3 to obtain the models of OpenVLA and $\pi_{0.5}$ fine-tuned with 150 data samples, with different sizes of the object. First, we perform scale generalization on the optimized USD to test the fine-tuned model’s robustness. We select the microwave oven and drawer for the experiment, and its optimized USD asset

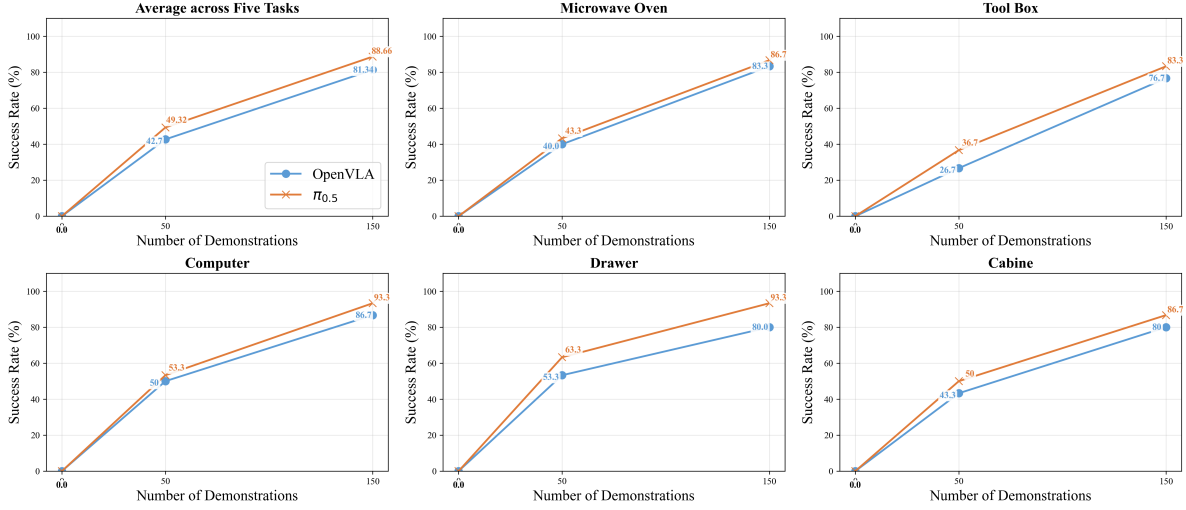


Figure 6. Success rate of the fine-tuned VLA model under different object replacements.



(a) Pose Generalization



(b) Scale Generalization



(c) Articulated Object Generalization

Figure 7. Demonstrations of robustness experiments for different articulated objects.

is progressively scaled between 0.6 and 0.9 to test the VLA model fine-tuned with data generated by AOMGen in the simulator. Each configuration is repeated 20 times to test the success rate. The results are listed in Table 2. As shown in the table, simple processing of the data enables the model to handle the same object at different scales.

Object	VLA Model	Success Rate			
		0.6	0.7	0.8	0.9
Microwave Oven	OpenVLA	55	65	65	70
	$\pi_{0.5}$	65	75	80	85
Drawer	OpenVLA	60	70	80	80
	$\pi_{0.5}$	80	90	90	90

Table 2. Performance when changing the scale of given replacement articulated object.

Then, we evaluate whether the model can generalize to unseen objects after being trained on mixed data with different replaced articulated objects. Specifically, for rotational joints, we fine-tune the VLA model on either single-object data or mixed data, and then test it on an unseen object with 20 trials. The results in Table 3 show that the model fine-

tuned on single data still lacks generalization to unseen objects, whereas the mixed data generated by object replacement enables successful handling of unseen objects within the same category. This further demonstrates the advantage of AOMGen in arbitrarily replacing same-category objects and generating effective data for downstream tasks.

Train Data	Unseen Object
Single Data	15%
Mixed Data	65%

Table 3. Success Rate with model trained by different dataset, to verify the model’s task success rate on unseen objects.

6. Conclusion

In this paper, we propose AOMGen, a powerful articulated object manipulation data generator that includes both rotational joint and prismatic joint. A series of novel designs enhance the physical interaction accuracy of the generated data and the visual realism of the rendered images. The generated data is verified to be applicable for fine-tuning the VLA model, further enhancing the model’s robustness.

Acknowledgments

This work was supported in part by the NSFC under grant 62272344, in part by the Natural Science Foundation of Anhui under Grant No. 2508085QF241, in part by the National Natural Science Foundation of China under Grant 62571055.

References

- [1] Genesis Authors. Genesis: A generative and universal physics engine for robotics and beyond, 2024. 2
- [2] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians. *arXiv preprint arXiv:2312.00860*, 2023. 3
- [3] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting, 2023. 3
- [4] Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023. 2
- [5] Open X-Embodiment Collaboration, Abby O’Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. <https://arxiv.org/abs/2310.08864>, 2023. 1
- [6] Xiaoshen Han, Minghuan Liu, Yilun Chen, Junqiu Yu, Xiaoyang Lyu, Yang Tian, Bolun Wang, Weinan Zhang, and Jiangmiao Pang. Re³sim: Generating high-fidelity simulation data via 3d-photorealistic real-to-sim for robotic manipulation, 2025. 3
- [7] Ryan Hoque, Ajay Mandlekar, Caelan Garrett, Ken Goldberg, and Dieter Fox. Intervengen: Interventional data generation for robust and data-efficient robot imitation learning, 2024. 2
- [8] Sheng-Yu Huang, Zi-Ting Chou, and Yu-Chiang Frank Wang. 3d gaussian inpainting with depth-guided cross-view consistency. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26704–26713, 2025. 3
- [9] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. 1, 7
- [10] Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through video world models. In *CoRL*, 2025. 2
- [11] Yufei Jia, Guangyu Wang, Yuhang Dong, Junzhe Wu, Yupei Zeng, Haonan Lin, Zifan Wang, Haizhou Ge, Weibin Gu, Kairui Ding, Zike Yan, Yunjie Cheng, Yue Li, Ziming Wang, Chuxuan Li, Wei Sui, Lu Shi, Guanzhong Tian, Ruqi Huang, and Guyue Zhou. Discoverse: Efficient robot simulation in complex high-fidelity environments, 2025. 2
- [12] Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning, 2025. 2
- [13] Zhao Jin, Zhengping Che, Zhen Zhao, Kun Wu, Yuheng Zhang, Yinu Zhao, Zehui Liu, Qiang Zhang, Xiaozhu Ju, Jing Tian, Yousong Xue, and Jian Tang. Artvip: Articulated digital assets of visual realism, modular interaction, and physical fidelity for robot learning, 2025. 7
- [14] Joshua Jones, Oier Mees, Carmelo Sferrazza, Kyle Stachowicz, Pieter Abbeel, and Sergey Levine. Beyond sight: Fine-tuning generalist robot policies with heterogeneous sensors via language grounding. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, USA, 2025. 1
- [15] Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. In *8th Annual Conference on Robot Learning*, 2024. 1, 3
- [16] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 1, 7
- [17] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024. 2
- [18] Xinhai Li, Jialin Li, Ziheng Zhang, Rui Zhang, Fan Jia, Tiancai Wang, Haoqiang Fan, Kuo-Kun Tseng, and Ruiping Wang. Robosim: A real2sim2real robotic gaussian splatting simulator, 2024. 2
- [19] Yu Liu, Baoxiong Jia, Ruijie Lu, Chuyue Gan, Huayu Chen, Junfeng Ni, Song-Chun Zhu, and Siyuan Huang. Videoartgs: Building digital twins of articulated objects from monocular video, 2025. 3
- [20] Xiaoxiao Long, Qingrui Zhao, Kaiwen Zhang, Zihao Zhang, Dingrui Wang, Yumeng Liu, Zhengjie Shu, Yi Lu, Shouzheng Wang, Xinzhe Wei, Wei Li, Wei Yin, Yao Yao, Jia Pan, Qiu Shen, Ruigang Yang, Xun Cao, and Qionghai Dai. A survey: Learning embodied intelligence from physical simulators and world models, 2025. 2
- [21] Haozhe Lou, Yurong Liu, Yike Pan, Yiran Geng, Jianteng Chen, Wenlong Ma, Chenglong Li, Lin Wang, Hengzhen Feng, Lu Shi, Liyi Luo, and Yongliang Shi. Robo-gs: A physics consistent spatial-temporal model for robotic arm with hybrid representation, 2024. 3
- [22] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021. 2
- [23] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations, 2023. 2
- [24] NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Chereniadev, Xingye Da, Runyu Ding, Linxi ”Jim” Fan, et al.

- GR00T N1: An open foundation model for generalist humanoid robots. In *ArXiv Preprint*, 2025. 1
- [25] Pakkapon Phongthawee, Worameth Chinchuthakun, Nontaphat Sinsunthithet, Amit Raj, Varun Jampani, Pramook Khungurn, and Supasorn Suwajanakorn. Diffusionlight: Light probes for free by painting a chrome ball. In *ArXiv*, 2023. 6
- [26] Mohammad Nomaan Qureshi, Sparsh Garg, Francisco Yandun, David Held, George Kantor, and Abhishesh Silwal. Splatsim: Zero-shot sim2real transfer of rgb manipulation policies using gaussian splatting, 2024. 3
- [27] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3
- [28] Stefano Scolari. Mesh2splat: Fast mesh to 3d gaussian splat conversion. <https://github.com/electronicarts/mesh2splat>, 2025. Extended and updated version of the author’s Master’s thesis at KTH. 5
- [29] Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation, 2024. 3
- [30] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5
- [31] Haowen Wang, Zhen Zhao, Zhao Jin, Zhengping Che, Liang Qiao, Yakun Huang, Zhipeng Fan, Xiuquan Qiao, and Jian Tang. Sm3: Self-supervised multi-task modeling with multi-view 2d images for articulated objects. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12492–12498, 2024. 3
- [32] Zifan Wang, Ziqing Chen, Junyu Chen, Jilong Wang, Yuxin Yang, Yunze Liu, Xueyi Liu, He Wang, and Li Yi. Mobileh2r: Learning generalizable human to mobile robot handover exclusively from scalable and diverse synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17315–17325, 2025. 2
- [33] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinu Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. In *Robotics: Science and Systems (RSS) 2025*. Robotics: Science and Systems Foundation, 2025. 1
- [34] Yuxuan Wu, Lei Pan, Wenhua Wu, Guangming Wang, Yanzi Miao, Fan Xu, and Hesheng Wang. RL-gsbridge: 3d gaussian splatting based real2sim2real method for robotic manipulation learning, 2025. 3
- [35] Xiuwei Xu, Angyuan Ma, Hankun Li, Bingyao Yu, Zheng Zhu, Jie Zhou, and Jiwen Lu. R2rgen: Real-to-real 3d data generation for spatially generalized manipulation, 2025. 2
- [36] Zhengrong Xue, Shuying Deng, Zhenyang Chen, Yixuan Wang, Zhecheng Yuan, and Huazhe Xu. Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning. *arXiv preprint arXiv:2502.16932*, 2025. 1, 2
- [37] Sizhe Yang, Wenye Yu, Jia Zeng, Jun Lv, Kerui Ren, Cewu Lu, Dahua Lin, and Jiangmiao Pang. Novel demonstration generation with gaussian splatting enables robust one-shot manipulation. *arXiv preprint arXiv:2504.13175*, 2025. 3
- [38] Xiuyu Yang, Bohan Li, Shaocong Xu, Nan Wang, Chongjie Ye, Chen Zhaoxi, Minghan Qin, Ding Yikang, Xin Jin, Hang Zhao, and Hao Zhao. Orv: 4d occupancy-centric robot video generation. *arXiv preprint arXiv:2506.03079*, 2025. 2
- [39] Justin Yu, Letian Fu, Huang Huang, Karim El-Refai, Rares Andrei Ambrus, Richard Cheng, Muhammad Zubair Irshad, and Ken Goldberg. Real2render2real: Scaling robot data without dynamics simulation or robot hardware, 2025. 3
- [40] Juyong Zhang, Yuxin Yao, and Bailin Deng. Fast and robust iterative closest point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3450–3466, 2022. 3
- [41] Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination, 2024. 2
- [42] Zhiyuan Zhou, Pranav Atreya, Abraham Lee, Homer Walke, Oier Mees, and Sergey Levine. Autonomous improvement of instruction following skills via foundation models. *arXiv preprint arXiv:407.20635*, 2024. 2
- [43] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025. 2