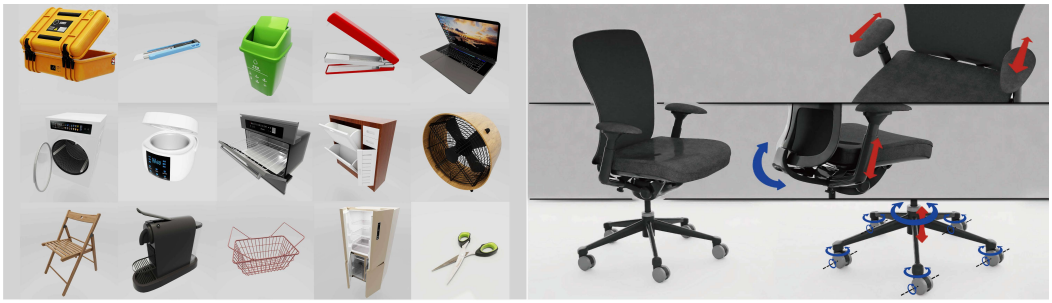
ARTVIP: ARTICULATED DIGITAL ASSETS OF VISUAL REALISM, MODULAR INTERACTION, AND PHYSICAL FIDELITY FOR ROBOT LEARNING

Anonymous authors
Paper under double-blind review



(a) Examples of articulated objects (b) Details of visual realism and physical fidelity

(c) Scene instances with articulated objects (d) Learning applications and annotations

Figure 1: Overview of ArtVIP.

ABSTRACT

Robot learning increasingly relies on simulation to advance complex ability such as dexterous manipulations and precise interactions, necessitating high-quality digital assets to bridge the sim-to-real gap. However, existing open-source articulated-object datasets for simulation are limited by insufficient visual realism and low physical fidelity, which hinder their utility for training models mastering robotic tasks in real world. To address these challenges, we introduce ArtVIP, a comprehensive open-source dataset comprising high-quality digital-twin articulated objects, accompanied by indoor-scene assets. Crafted by professional 3D modelers adhering to unified standards, ArtVIP ensures visual realism through precise geometric meshes and high-resolution textures, while physical fidelity is achieved via fine-tuned dynamic parameters. Meanwhile, the dataset pioneers embedded modular interaction behaviors within assets and pixel-level affordance annotations. Feature-map visualization and optical motion capture are employed to quantitatively demonstrate ArtVIP's visual and physical fidelity, with its applicability validated across imitation learning and reinforcement learning experiments. Provided in USD format with detailed production guidelines, ArtVIP is fully open-source, benefiting the research community and advancing robot learning research.

1 INTRODUCTION

Embodied AI is catalyzing the transformation of robotic systems from constrained laboratory settings (Billard & Kragic, 2019; Spong et al., 2020) to complex, unstructured real-world environments (Brohan et al., 2023b; Zhao et al., 2023; Brohan et al., 2023a). The emergence of large-scale pretrained models (Zhang & Yan, 2023; Kim et al., 2024; Wang et al., 2024b) and novel learning

paradigms (Team, 2025; Intelligence, 2025) has ushered in a data-centric era. In this new era, the availability of high-quality data is a critical bottleneck for developing scalable and generalizable embodied intelligence.

While collecting data and deploy robots in real-world is resource-intensive and challenging to scale, simulation provides an efficient alternative to enhance robot learning. Simulation supports imitation learning by collecting unlimited and low-cost training data (Wu et al., 2024) and reinforcement learning by providing virtual environments (Makoviychuk et al., 2021; Torne et al., 2024). Meanwhile, simulations enable rapid deployment and standardized test (Ramasubramanian et al., 2022; Do et al., 2025) of algorithms without concerns about hardware damage or safety issues. Overall, simulation facilitates the exploration of innovative strategies for robot learning.

High-quality digital assets are vital to simulation for robot learning. Simulation platforms (Koenig & Howard, 2004; Todorov et al., 2012; Kolve et al., 2017; Makoviychuk et al., 2021; Puig et al., 2023) depend on digital assets to accurately represent the real world digitally and to simulate its physical characteristics (Choi et al., 2021). High-quality digital assets can effectively reduce the sim-to-real gap, thereby enhancing the performance of robot learning algorithms. For instance, digital-twin assets, which are virtual replicas created via reverse-modeling techniques, can benefit pre-deployment validation and optimization of robotic systems (Straub et al., 2019; Ramakrishnan et al., 2021). Moreover, high-quality digital assets can serve as training data or seed models for synthetic-asset methods such as 3D reconstruction (Liu et al., 2023a; Li et al., 2020; Sun et al., 2023; Liu et al., 2023b) and domain-randomization (Dai et al., 2024; Ge et al., 2024; Torne et al., 2024) techniques, enhancing the data distribution and providing limitless diversity of objects and environments. Conversely, utilizing poor-quality data for synthetic-data generation exacerbates the sim-to-real gap and impair robot learning models (Schraml & Notni, 2024; Osvaldová et al., 2024; Kim et al., 2022).

As robot learning turn form mastering simple tasks such as pick and grasp to dexterous manipulation and interaction tasks, high quality articulated-object assets is of great demand. Current open-source articulated-object datasets fail to meet the needs of robot learning. For instance, PartNet-Mobility (Xiang et al., 2020) suffers from limited visual realism and insufficient physical fidelity of dynamic joints. BEHAVIOR-1K (Li et al., 2024a) offers better visual fidelity, but it is locked into the OmniGibson simulator (Li et al., 2024a) and its physical parameters have not been fine-tuned. Moreover, both datasets are largely sourced from internet-searchable 3D model repositories (Inc., 2024a;b) without adhering to consistent modeling standards, leading to inconsistency in quality. Apart from using existing datasets, people attempts to obtain simulation assets in other ways, facing further challenges. Retrieval-based methods (Liu et al., 2024b;a) and reconstruction techniques (Chen et al., 2024; Eppner et al., 2024) often inherit stylistic biases from their training data and have limited geometric generalization. More recent pipelines (Qiu et al., 2025; Mandi et al., 2024; Le et al., 2025) introduce promising directions, yet face challenges such as mesh quality variance, segmentation noise, and lack of robust joint parameter tuning.

The main bottleneck for articulated object datasets lies in asset quality rather than quantity; to this end, we identify four key aspects that require careful consideration.

- **Visual Realism.** Assets should be constructed with precise geometric meshes and high-resolution textures to ensure a photorealistic appearance. The amount of triangular faces should be optimized to guarantee real-time simulation performance.
- **Modular Interaction.** Assets should support interactivity (e.g., toggling a switch to turn on a light). These interactions should be modular to enable reuse across scenarios.
- **Physical Fidelity.** Accurate collision geometry and joint dynamics (stiffness, damping, friction) of articulated assets are essential for simulated motion to faithfully reproduce real-world kinematics and dynamics.
- **Simulation Friendliness.** Information expanding simulation usages such as pixel-level affordance annotations and accompanied scenes are encouraged. Meanwhile, open-source assets compatible to various simulation platforms and replicable asset creation process should be provided.

To meet the mentioned requirements, we introduce ArtVIP, a high-quality and readily deployable suite of ***Art****iculated-object digital assets with* ***V****isual realism, modular* ***I****nteraction*, and ***P****hysical fidelity*, designed to facilitate the learning and evaluation of diverse manipulation skills such as rotating, clicking, pulling, and pressing. As illustrated in Fig. 1, ArtVIP encompasses both articulated

object models and complementary indoor-scene assets, all meticulously authored by professional 3D modelers under a unified asset specification to ensure consistent visual quality and realism. Physical properties are precisely tuned to reproduce real-world dynamics, thereby enhancing the physical fidelity. Furthermore, ArtVIP provides pixel-level affordance annotations and uniquely embeds interaction semantics directly into the assets, enabling modular reuse and scalable behavior modeling.

In conclusion, ArtVIP offers the following contributions:

- We release a collection comprising 9 categories, 37 subcategories, and 992 high-quality digital-twin articulated objects. All assets exhibit both visual realism and physical fidelity, supported by quantitative evaluations.
- We provide digital-twin scene assets and configured scenarios integrating articulated objects within scene for immediate use. Extensive experiments on imitation learning, reinforcement learning, and 3D construction algorithms demonstrate the broader applicability of the assets.
- All assets are provided in the modern USD format and remain compatible with established robotics workflows via conversion to legacy formats such as URDF or MJCF. The detailed production process offer comprehensive guidance to facilitate community adoption and replication.

2 Related Works

Simulation Platforms. A typical simulation planform integrates a physics engine (Smith et al., 2005; Todorov et al., 2012; Coumans & Bai, 2016; Corporation, 2025; Tasora et al., 2016) and a rendering engine (Matl, 2019; Chociej et al., 2019; Rojtberg, Pavel and Rogers, David and Streeting, Steve and others, 2001 – 2024). Game engines (Technologies, 2025.05.14; Games, 2025) offer similar features but do not natively support ROS (Quigley et al., 2009; Macenski et al., 2022) for robotics. MuJoCo (Todorov et al., 2012) and Webots (Webots, 2018) excel in simulating rigid body and multi-joint dynamics but prioritize computational efficiency over high-fidelity rendering. Gazebo (Koenig & Howard, 2004), despite its large community and robust integration with ROS, provides outdated rendering performance and exhibits lower accuracy in physical simulation. Frameworks like AI2THOR (Kolve et al., 2017), Habitat (Savva et al., 2019; Szot et al., 2021; Puig et al., 2023) and ALFRED (Shridhar et al., 2020) are designed for mobile manipulation and instruction-following, fail to deliver precise physical interactions. In contrast, Isaac Sim (Nvidia, 2025.05.14) offers the highest-fidelity visual rendering and leverages powerful GPU-parallel physics computation, making it well-suited for robot learning. Other platforms, such as RoboCasa (Nasiriany et al., 2024) (built upon MuJoCo) and OmniGibson (Li et al., 2024a) (built upon Isaac Sim), have become challenging to maintain. Consequently, we developed ArtVIP specifically for Isaac Sim to capitalize on its superior rendering and physics capabilities.

Datasets for Robot Simulation. Many datasets provide digital assets suitable for robot simulation. Indoor-scene assets (Straub et al., 2019; Shen et al., 2021; Ramakrishnan et al., 2021; Li et al., 2022) contribute significantly to robot navigation tasks but lacking support for graphical user interface (GUI)-based editing. Object digital assets includes ShapeNet (Chang et al., 2015), Objaverse (Deitke et al., 2023) and other digital-twin datasets (Kuang et al., 2023; Dong et al., 2025). However, these assets can only function as rigid bodies in simulations, preventing robots from performing articulated manipulation tasks with them. Limited studies addressed articulated object assets. PartNet-Mobility (Xiang et al., 2020) provides 2,346 articulated-object assets across 46 categories, with many assets suffering from unsmoothed geometric surfaces, low rendering quality, and imprecise dynamic joint. RoboCasa (Nasiriany et al., 2024) offers 2,508 digital assets, but only 24 are articulated objects. BEHAVIOR-1K (Li et al., 2024a) includes 543 articulated-object assets with improved visual fidelity, yet all assets are encrypted and accessible only through OmniGibson. These limitations underscore the need for a high-quality, open-source articulated-object dataset.

Articulated Objects Construction and Generation Methods. Construction methods (Liu et al., 2024a; Chen et al., 2024; Su et al., 2024; Xue et al., 2021; Wang et al., 2024a) can generate articulated objects from images and reduce the labor cost. However, these methods perform reliably only on objects with simple joints, such as cabinets and desks, and produce assets with compromised visual realism. Generative methods (Yang et al., 2022; Liu et al., 2023c; Long et al., 2023; Xu et al., 2023; Koo et al., 2024), are currently limited to static rigid-body objects. These assets often exhibit

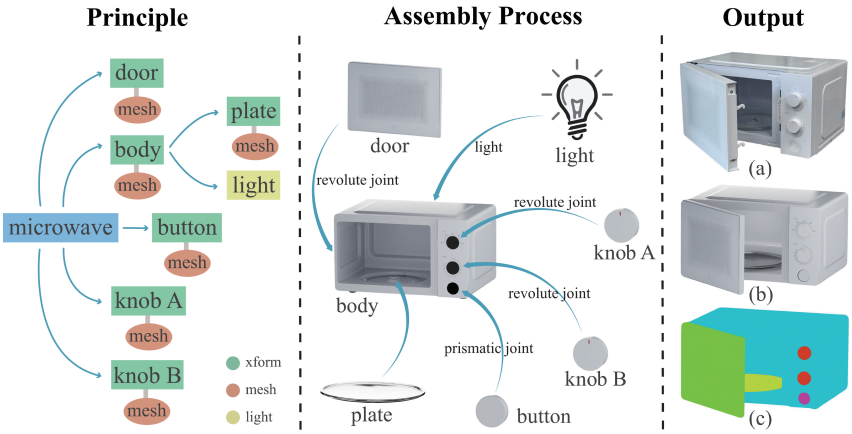


Figure 2: An asset in ArtVIP. **Left:** Top-down assembly principle. **Middle:** Assembly process. **Right:** Comparison between the real object (a) with its digital-twin (b), and annotations (c).

distorted and unreasonable meshes, coupled with poor rendering quality. The absence of support for articulated objects in generative methods further limits their applicability to robot learning tasks.

3 ArtVIP Collection and Methodology

3.1 Overview

Existing datasets are largely sourced from pre-made models from public repositories. This leads to inconsistent modeling quality, disorganized part hierarchies, and non-standardized coordinate systems, all of which typically require manual preprocessing for simulation use. While current generative and reconstruction methods can easily scale up, they are still not mature enough to ensure quality. Given these constraints, we opted to prioritize fidelity over scale at this stage. ArtVIP emphasizes both visual realism and physical fidelity across a comprehensive collection of articulated objects. It covers 9 categories and 37 subcategories, encompassing 992 articulated assets (see Appendix Sec. A). Complementary sim-ready scenes (see Appendix Sec. B) and pixel-level annotations (see Appendix Sec. C) are also provided.

3.2 Visual Realism

To ensure visual realism, professional 3D modelers follow unified modeling and assembly guidelines when manually crafting articulated objects. As shown in Fig. 2, we adopt a top-down mechanical modeling approach that decomposes each articulated object into three hierarchical levels: assembly, module, and mesh. An assembly constitutes the complete functional unit, encompassing multiple modules and meshes. Modelers first establish the assembly's base coordinate frame at the geometric center of the object's bottom surface. Guided by the assembly's affordances, functionality, and joint locations, they partition it into rigid-body modules of the Xform type, which expose dynamic information such as transforms, velocities, and world coordinates. Each rigid-body module contains mesh parts that provide geometric detail, visual appearance, and static physical properties, including collision shapes and mass. Modelers follow strict rules regarding meshes, textures, and materials (see Appendix Sec. D) to ensure visual realism. After modeling individual meshes, they assemble them bottom-up—mesh, module, assembly—and integrate dynamic motion by connecting modules with joints (middle panel of Fig. 2), ensuring the asset preserves intended affordances and appearance. Finally, for the finished asset (right panel of Fig. 2), each module is annotated with pixel-level labels to enable precise identification of interaction affordances.

3.3 Physical Fidelity

In addition to visual realism, physical fidelity plays a critical role in reducing the sim-to-real gap. Optimized collision modeling ensures accurate rigid-body contact, improving precision in tasks

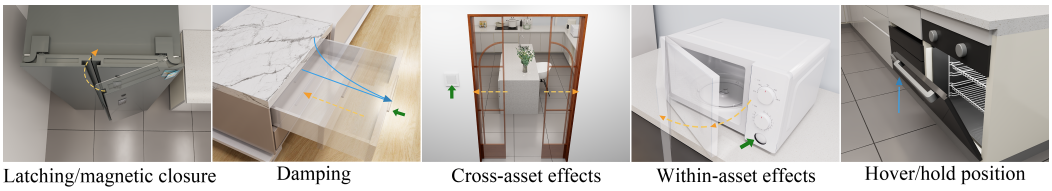


Figure 3: Green arrows denote applied force, yellow dashed lines indicate object motion, and blue arrows show damping. From left to right: (i) latching/magnetic closure — the door automatically closes when near shut; (ii) damping — the damping magnitude increases as the drawer is pushed in; (iii) cross-asset effects — triggering the switch opens the door; (iv) within-asset effects — pressing the microwave button opens the door and turns on the interior light; (v) hover/hold position — the oven door can hold at any angle.

such as grasping handles and other force-mediated contact scenarios. Similarly, joint optimization yields precise joint dynamics, improving the fidelity of articulated components' motion trajectories during fine-grained operations (e.g., opening cabinet doors or pressing switches). ArtVIP adopts the following processes.

Collision. To strike a balance between physical fidelity, interaction consistency, and computational efficiency, ArtVIP represents each mesh's collision shape using a mix of convex hulls, convex decomposition, and fine-tuned collision meshes. For relatively regular or simple geometry, ArtVIP relies on Isaac Sim's default convex hull generation. When a complex mesh can be decomposed without sacrificing its affordance, 3D modelers split its collision volume into multiple primitive meshes (e.g., cubes, cylinders). If neither a convex hull nor fine-tuned collision suffices, ArtVIP employs Isaac Sim's built-in convex decomposition tool, which leverages mesh normals and related methods to produce accurate collision geometry.

Joints. To achieve physical fidelity of dynamic joints and simulate variable joint motions in the real world, we enhance the joint drive equation (NVIDIA, 2025) originally provided by Isaac Sim:

$$\tau = K(q) \cdot (q - q_{\text{target}}(q)) + D \cdot (\dot{q} - \dot{q}_{\text{target}}(q)) \tag{1}$$

where τ denotes the generalized force or torque applied to drive the joint; q and $\dot{q}$ are the joint position and velocity, respectively; D denotes damping; and K denotes stiffness. While this equation models basic joint motions, it does not fully capture complex joint dynamics observed in the real world. For complex joints such as door closers and light switches, τ may vary with q and $\dot{q}$. To accommodate these cases, we parameterize K and the target terms as functions of q, and allow dependence on $\dot{q}$ when needed. The details are described in the Appendix Sec. E.

3.4 Modular Interaction

A key innovation of this work is embedding customizable behaviors directly within each asset to enable interactive functionality without writing additional code.

Reproducing complex interactions. We abstract five canonical behavior primitives for articulated objects and instantiate them across ArtVIP, covering 394 assets and more than 900 joints.

- *Latching/magnetic closure:* Simulates automatic self-closing when the articulation enters a capture angle range, driven by magnetic attraction or mechanical spring/closer assemblies; once captured, a closing torque is applied until fully latched. Examples include refrigerator doors (self-closing hinge with magnetic gasket) and doors equipped with overhead closers.
- *Damping:* Simulates sliding components and rotational hinges whose effective damping peaks near the closed position and varies smoothly along the motion, enabling gentle starts and stops. Examples include nightstand drawers, dishwashers, and cabinets.
- *Cross-asset effects:* Simulates trigger-based coupling between distinct objects, allowing one object's state or event to drive another's behavior. Examples include button-triggered door opening and light switching.
- *Within-asset effects:* Simulates instantaneous, mechanism-internal triggers. For example, pressing a microwave button pops the door open; similar behaviors occur in foot-pedal trash bins and height-adjustable desks.

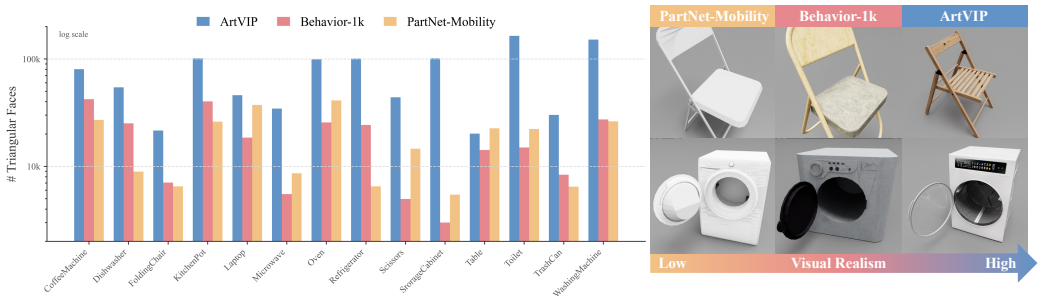


Figure 4: **Left:** Comparison of triangle count. **Right:** Rendering comparison.

- *Hover/hold position:* Simulates static-friction-mediated holding in sliding or rotational joints so that, once external forces are removed, the mechanism can remain at any intermediate pose. Examples include oven doors and drawers.

Improving asset reusability. Enhancing simulation development efficiency hinges on modularizing digital assets and maximizing their reusability. Our approach binds behaviors to assets at design time: researchers and artists can simply import the USD file and instantly obtain interaction affordances. This modular, reusable design reduces development overhead and accelerates algorithm iteration, allowing researchers to focus on advancing embodied AI rather than asset programming.

4 Evaluation

We evaluate ArtVIP along two axes: visual realism and physical fidelity, using quantitative comparisons in simulation and the real world.

4.1 Visual Realism Evaluations

A comparative analysis is conducted among ArtVIP, BEHAVIOR-1K, and PartNet-Mobility(detailed chart in the Appendix Sec. F). As shown on the right of Fig. 4, both BEHAVIOR-1K and PartNet-Mobility exhibit distorted geometry and implausible appearance. In addition, we quantify geometric detail via triangle count, evaluate reconstruction performance, and visualize feature distributions to assess visual realism.

Geometric Detail. Meshes built from densely triangular faces preserve the core geometric detail. A high count of triangular faces improves surface smoothness and minimizes faceting. The left of Fig. 4 illustrates the comparison results on object categories that appear in all three datasets, demonstrating the rich geometric detail in ArtVIP. More analysis and relative profiling are in the Appendix Sec. G.

Reconstruction Performance Evaluation. To assess differences in reconstruction quality across data assets, we conducted experiments using VGGT (Wang et al., 2025), a widely adopted method that has demonstrated strong generalization in real-world reconstruction tasks. Using identical multi-view sampling strategies on the OmniGibson and ArtVIP assets, we generated reconstruction inputs, with results shown on the left portion of Fig. 5. Reconstructions from ArtVIP assets exhibit higher structural fidelity and finer detail preservation compared to those from OmniGibson. This suggests that ArtVIP's more realistic geometry and material representation enhance the quality and compatibility of sampled images for reconstruction tasks. The results underscore the role of high-fidelity assets in supporting viewpoint diversity and accurate structure recovery.

Feature Distribution Visualization Analysis. To verify the visual realism of ArtVIP assets, we randomly sampled 100 3D models and selected corresponding or semantically similar objects from OmniGibson and the real world for comparison. Real-world images were captured using three devices (an Android phone, an iPhone, and an Intel RealSense D435) under multi-view settings. In Isaac Sim, we rendered samples of the ArtVIP and OmniGibson assets using matched camera viewpoints to ensure consistency across domains. We applied t-SNE (Van der Maaten & Hinton, 2008) to visualize the extracted CLIP (Radford et al., 2021) features. As shown on the right portion

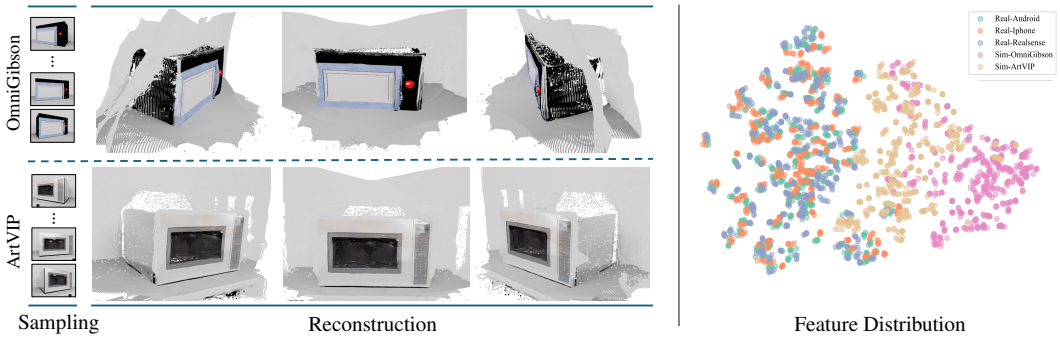


Figure 5: **Left:** Reconstruction of a microwave. OmniGibson yields poor results due to weak visual appearance, while ArtVIP enables better reconstruction via more realistic details. **Right:** CLIP-based (Radford et al., 2021) feature distribution. Each color denotes a data source and ArtVIP features align more closely with real-world data.

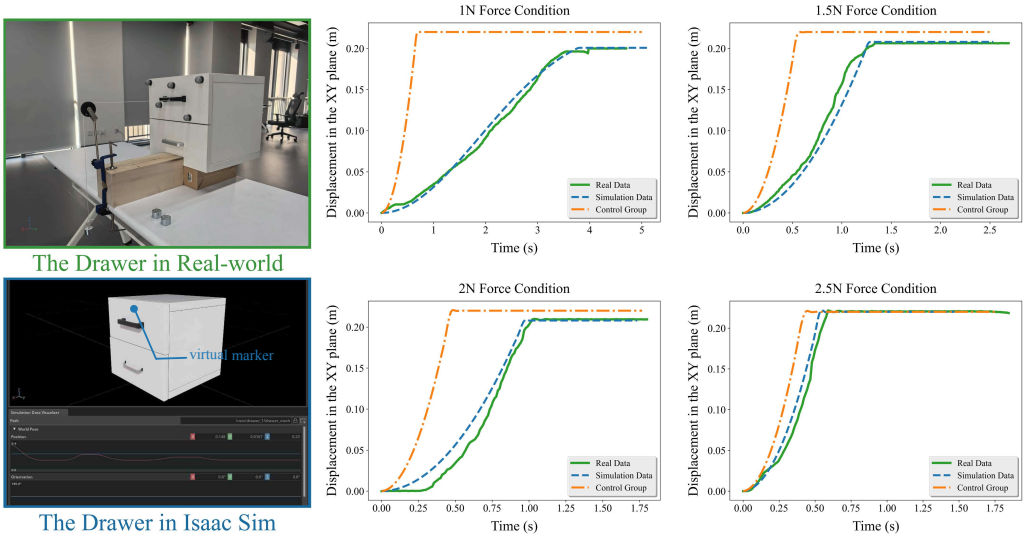


Figure 6: **Left:** Digital-twin asset examples in real-world and simulation. **Right:** Analysis of the drawer's displacement driven by different forces.

of Fig. 5, ArtVIP features align more closely with real-world data, indicating higher consistency in visual semantics, texture, and material. This fidelity enhances the value of ArtVIP for simulation-to-reality transfer in downstream tasks.

4.2 Physical Fidelity and Interaction Evaluations

To demonstrate the physical fidelity of joint motion within articulated objects, we employed an optical tracking system (0.1 mm spatial resolution and 90 Hz sampling rate) to record motion trajectories of joints on real-world objects. These recordings were compared with the joint motions of their corresponding digital-twin articulated objects in simulation to evaluate the discrepancy between simulated and real-world joint behavior. We test in a common scenario where joint motion triggered by external force. More setting descriptions and evaluation results are described in the Appendix Sec. H.

As shown in Fig. 6, in the real-world experiment, horizontal pulling forces of 1 N, 1.5 N, 2 N, and 2.5 N were applied to the drawer by suspending calibrated weights from the end of the fixed pulley system, ensuring consistent force direction. The drawer's displacement in the XY plane was recorded in real time. In the simulation environment, two configurations were evaluated: one with

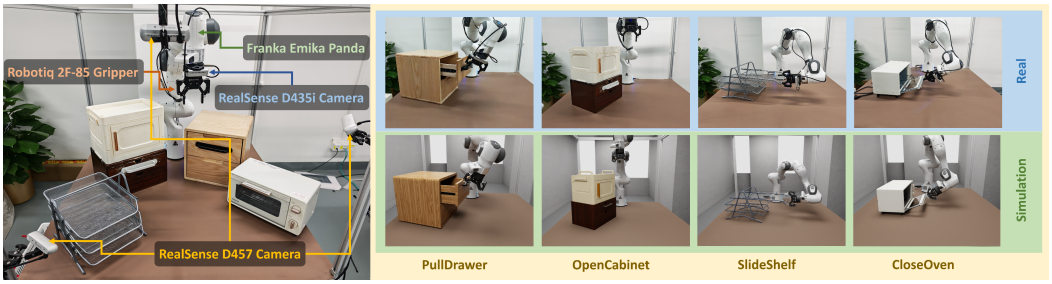


Figure 7: Experimental Setup. We conducted 4 real-world tasks for imitation learning.

default joint parameters and the other with optimized parameters. Both were subjected to the same force configuration as the real-world setup, and the spatial trajectories of the drawer's keypoints were tracked. The close agreement between the displacement obtained from simulation and real-world experiments, as shown in the right of Fig. 6, demonstrates the physical fidelity of the joints in ArtVIP.

5 Applications

To further verify the capability of ArtVIP in supporting downstream robotic learning tasks, we conducted extensive experiments in both the real-world and simulated environments following two primary paradigms in robotic learning: Imitation Learning and Reinforcement Learning.

5.1 Imitation Learning in Real World Environments

Experimental Setup. As illustrated in Fig. 7, we used a Franka robotic arm equipped with a Robotiq 2F-85 gripper and four RealSense cameras to create the real-world experimental environment. These cameras include three external RealSense D457 cameras (placed on the left, right, and top of the table) and one hand-eye RealSense D435i camera mounted at the wrist of the robotic arm. For simulation, we used Isaac Sim and replicated this real-world setup, including the Franka robotic arm, the operating table, camera settings, and the manipulated objects from ArtVIP. We constructed the simulated scene to match the real-world experiment environment as closely as possible.

Task Design and Data Collection. As shown in Fig. 7, we design four challenging articulated-object manipulation tasks: (1) **PullDrawer**, (2) **OpenCabinet**, (3) **SlideShelf**, and (4) **CloseOven**. These tasks demand precise and flexible motions, including rotation, angled pushing, and horizontal translation (see Appendix Sec. I). Data was collected via teleoperation in both real and simulated environments, where articulated objects were randomly placed within a predefined workspace and human operators completed each task. For each task, we gathered 100 successful trajectories in the real world and 100 in simulation. Each trajectory includes RGB streams from four camera viewpoints and full proprioceptive robot states (e.g., joint positions) throughout execution.

Imitation Learning Algorithm. We used two canonical imitation learning baselines, Action Chunking Transformer (ACT) (Zhao et al., 2023) and Diffusion Policy (DP) (Chi et al., 2023), to train the robotic policies for the articulated object manipulation task (more details in Appendix Sec. I).

Experimental Results on Imitation Learning. For each of the four articulated-object manipulation tasks, we trained ACT and DP under the following dataset settings: **(1) Real-Only (RO)**: 100 real-world trajectories; **(2) Sim-Only (SO)**: 100 simulated trajectories; **(3) Real–Sim–Mixed (RSM100+10/20/50/100)**: 100 real-world + 10, 20, 50, 100 simulated trajectories. For each experiment, we trained ACT and DP for 50k gradient descent iterations with three different random seeds, and evaluated the final checkpoint from each run with 60 rollouts to compute per-task success rates.

Tab. 1 summarizes success rates for ACT and DP under three dataset settings (RO, SO, RSM). We highlight three findings: (1) **Simulation-trained models achieve zero-shot success in the real world** (e.g., ACT 39% on CloserOven), reflecting ArtVIP's high-fidelity visuals and physics that reduce the sim-to-real gap. (2) With equal data volume, **real-world training outperforms simula-**

Table 1: Success rates of ACT and DP across dataset settings: RO (real-only), SO (sim-only), and RSM variants for all tasks.

Method	Dataset	PullDrawer	OpenCabinet	SlideShelf	CloseOven
ACT (Zhao et al., 2023)	RO	64%	34%	27%	58%
	SO	39%	12%	13%	23%
	RSM100+10	64%	36%	26%	59%
	RSM100+20	68%	38%	27%	60%
	RSM100+50	78%	44%	32%	66%
	RSM100+100	81%	46%	36%	68%
DP (Chi et al., 2023)	RO	66%	49%	44%	66%
	SO	20%	10%	18%	28%
	RSM100+10	65%	53%	47%	67%
	RSM100+20	69%	58%	53%	70%
	RSM100+50	73%	62%	56%	73%
	RSM100+100	79%	66%	59%	78%

tion (e.g., DP 49% vs. 10% on OpenCabinet), underscoring persistent sim-to-real challenges. (3) **Mixing real and simulated data boosts performance** (e.g., SlideShelf: DP from 44% to 59%), indicating that articulated assets in ArtVIP align well with real-world data distributions.

5.2 Reinforcement Learning in High-Fidelity Simulators

Reinforcement learning (RL) requires training environments that mirror real-world physical and perceptual complexity. To validate the quality of articulated assets in ArtVIP, we designed a CloseTrashcan task with a Franka robotic arm and trained a two-stage agent with the state-of-the-art visual RL framework EAGLE (Zhao et al., 2025) in Isaac Sim (see Appendix Sec. J).

Table 2: Sim vs. real performance across training checkpoints (k iterations).

Checkpoint (k)	500	450	400	350	300
Sim	0.98	0.91	0.85	0.81	0.73
Real	0.93	0.86	0.80	0.70	0.60

EAGLE enables efficient training of visuomotor policies. In Stage 1, we train a PPO expert (Schulman et al., 2017) with low-level state inputs. In Stage 2, we distill this expert into a visuomotor policy, applying EAGLE's self-supervised attention masks and control-aware augmentation. RandomConv (Lee et al., 2019) is used to diversify control-irrelevant backgrounds. To bridge the background visual gap between simulation and the real world, we adopt the Visual Matching technique introduced in (Li et al., 2024b).

We evaluate success rates at five checkpoints between 300k and 500k training iterations, using 100 trials in simulation and 30 in the real world with varied initial object poses. As shown in Tab. 2, the policy exhibits strong sim-to-real consistency, achieving a Pearson correlation of 0.9886. This highlights the high physical fidelity and visual realism of the assets in ArtVIP.

6 Limitation and Conclusion

We introduced ArtVIP, a high-quality dataset of articulated objects for robotic manipulation, featuring visual realism, accurate physical properties, and modular interaction capabilities. We validated its quality via diverse evaluations and demonstrated effectiveness in both imitation learning and reinforcement learning. Scaling remains bottlenecked by intensive human labor for asset modeling; future work will explore generative methods to automate synthesis, reduce manual effort, and broaden object diversity.

REFERENCES

Aude Billard and Danica Kragic. Trends and challenges in robot manipulation. Science, 364: eaat8414, 2019.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023a.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale, 2023b.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012, 2015.

Zoey Chen, Aaron Walsman, Marius Memmel, Kaichun Mo, Alex Fang, Karthikeya Vemuri, Alan Wu, Dieter Fox, and Abhishek Gupta. Urdformer: A pipeline for constructing articulated simulation environments from real-world images. arXiv preprint arXiv:2405.11656, 2024.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. The International Journal of Robotics Research, pp. 02783649241273668, 2023.

Maciek Chociej, Peter Welinder, and Lilian Weng. Orrb – openai remote rendering backend, 2019. URL https://arxiv.org/abs/1906.11633.

HeeSun Choi, Cindy Crump, Christian Duriez, Asher Elmquist, Gregory Hager, David Han, Frank Hearl, Jessica Hodgins, Abhinandan Jain, Frederick Leve, et al. On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. Proceedings of the National Academy of Sciences, 118:e1907856118, 2021.

NVIDIA Corporation. Nvidia physx sdk, 2025. URL https://developer.nvidia.com/physx-sdk.

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

Tianyuan Dai, Josiah Wong, Yunfan Jiang, Chen Wang, Cem Gokmen, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Automated creation of digital cousins for robust policy learning. In 8th Annual Conference on Robot Learning, 2024.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13142–13153, 2023.

Tan-Dzung Do, Nandiraju Gireesh, Jilong Wang, and He Wang. Watch less, feel more: Sim-to-real rl for generalizable articulated object manipulation via motion adaptation and impedance control. arXiv preprint arXiv:2502.14457, 2025.

Zhao Dong, Ka Chen, Zhaoyang Lv, Hong-Xing Yu, Yunzhi Zhang, et al. Digital twin catalog: A large-scale photorealistic 3d object digital twin dataset, 2025.

Jonathan Embley-Riches, Jianwei Liu, Simon Julier, and Dimitrios Kanoulas. Unreal robotics lab: A high-fidelity robotics simulator with advanced physics and rendering, 2025.

Clemens Eppner, Adithyavairavan Murali, Caelan Garrett, Rowland O'Flaherty, Tucker Hermans, Wei Yang, and Dieter Fox. scene_synthesizer: A python library for procedural scene generation in robot manipulation. Journal of Open Source Software, 2024.

Epic Games. Unreal engine, 2025. URL https://www.unrealengine.com.

Yunhao Ge, Yihe Tang, Jiashu Xu, Cem Gokmen, Chengshu Li, Wensi Ai, Benjamin Jose Martinez, Arman Aydin, Mona Anvari, Ayush K Chakravarthy, et al. Behavior vision suite: Customizable dataset generation via simulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22401–22412, 2024.

Xiaoshen Han, Minghuan Liu, Yilun Chen, Junqiu Yu, Xiaoyang Lyu, Yang Tian, Bolun Wang, Weinan Zhang, and Jiangmiao Pang. Re^{3}sim: Generating high-fidelity simulation data via 3d-photorealistic real-to-sim for robotic manipulation, 2025.

Trimble Inc. 3d warehouse, 2024a. URL https://3dwarehouse.sketchup.com/. Online 3D model repository for SketchUp.

TurboSquid Inc. Turbosquid, 2024b. URL https://www.turbosquid.com/. Professional 3D model marketplace.

Physical Intelligence. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025.

Alexander Kim, Kyuhyup Lee, Seojoon Lee, Jinwoo Song, Soonwook Kwon, and Suwan Chung. Synthetic data and computer-vision-based automated quality inspection system for reused scaffolding. Applied Sciences, 12(19), 2022. ISSN 2076-3417. doi: 10.3390/app121910097. URL https://www.mdpi.com/2076-3417/12/19/10097.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, et al. Openvla: An open-source vision-language-action model, 2024.

Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In IEEE/RSJ International Conference on Intelligent Robots and Systems, volume 3, pp. 2149–2154, 2004.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. arXiv, 2017.

Juil Koo, Seungwoo Yoo, Minh Hieu Nguyen, and Minhyuk Sung. Salad: Part-level latent diffusion for 3d shape generation and manipulation, 2024. URL https://arxiv.org/abs/2303.12236.

Zhengfei Kuang, Yunzhi Zhang, Hong-Xing Yu, Samir Agarwala, Elliott Wu, Jiajun Wu, et al. Stanford-orb: a real-world 3d object inverse rendering benchmark. Advances in Neural Information Processing Systems, 36:46938–46957, 2023.

Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model, 2025. URL https://arxiv.org/abs/2410.13882.

Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. Network randomization: A simple technique for generalization in deep reinforcement learning. International Conference on Learning Representations, 2019.

Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Elliott Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. In 5th Annual Conference on Robot Learning, 2022.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation, 2024a. URL https://arxiv.org/abs/2403.09227.

Xiaolong Li, He Wang, Li Yi, Leonidas Guibas, A. Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation, 2020. URL https://arxiv.org/abs/1912.11913.

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, 2024b. URL https://arxiv.org/abs/2405.05941.

Jiayi Liu, Denys Iliash, Angel X Chang, Manolis Savva, and Ali Mahdavi-Amiri. Singapo: Single image controlled generation of articulated parts in objects. arXiv preprint arXiv:2410.16499, 2024a.

Jiayi Liu, Hou In Ivan Tam, Ali Mahdavi-Amiri, and Manolis Savva. Cage: Controllable articulation generation, 2024b. URL https://arxiv.org/abs/2312.09570.

Xueyi Liu, Bin Wang, He Wang, and Li Yi. Few-shot physically-aware articulated mesh generation via hierarchical deformation, 2023a.

Xueyi Liu, Ji Zhang, Ruizhen Hu, Haibin Huang, He Wang, and Li Yi. Self-supervised category-level articulated object pose estimation with part-level se(3) equivariance, 2023b. URL https://arxiv.org/abs/2302.14268.

Zhen Liu, Yao Feng, Michael J Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. arXiv preprint arXiv:2303.08133, 2023c.

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. arXiv preprint arXiv:2310.15008, 2023.

Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot operating system 2: Design, architecture, and uses in the wild. Science Robotics, 7:eabm6074, 2022.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. arXiv preprint arXiv:2108.10470, 2021.

Zhao Mandi, Yijia Weng, Dominik Bauer, and Shuran Song. Real2code: Reconstruct articulated objects via code generation, 2024. URL https://arxiv.org/abs/2406.08474.

Matthew Matl. Pyrender. https://github.com/mmatl/pyrender, 2019.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In Robotics: Science and Systems, 2024.

Federico Nesti, Gianluca D'Amico, Mauro Marinoni, and Giorgio Buttazzo. Simprive: a simulation framework for physical robot interaction with virtual environments, 2025.

NVIDIA. Joint tuning — Isaac Sim documentation, 2025. URL https://docs.isaacsim.omniverse.nvidia.com/latest/robot_setup/joint_tuning.html#gain-tuning.

Nvidia. Understanding physically-based rendering, 2025a. URL https://docs.omniverse.nvidia.com/simready/latest/simready-asset-creation/material-best-practices.html.

Nvidia. Omniverse rtx renderer, 2025b. URL https://docs.omniverse.nvidia.com/materials-and-rendering/latest/rtx-renderer.html.

Nvidia. Nvidia isaac sim, 2025.05.14. URL https://developer.nvidia.com/isaac/sim. Isaac Sim.

Katarína Osvaldová, Lukáš Gajdošech, Viktor Kocur, and Martin Madaras. Enhancement of 3d camera synthetic training data with noise models. arXiv preprint arXiv:2402.16514, 2024.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots, 2023.

Xiaowen Qiu, Jincheng Yang, Yian Wang, Zhehuan Chen, Yufei Wang, Tsun-Hsuan Wang, Zhou Xian, and Chuang Gan. Articulate anymesh: Open-vocabulary 3d articulated objects modeling, 2025. URL https://arxiv.org/abs/2502.02590.

Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In ICRA workshop on open source software, volume 3, pp. 5, 2009.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In International Conference on Machine Learning, pp. 8748–8763, 2021.

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2021. URL https://arxiv.org/abs/2109.08238.

Aswin K Ramasubramanian, Robins Mathew, Matthew Kelly, Vincent Hargaden, and Nikolaos Papakostas. Digital twin for human–robot collaboration in manufacturing: Review and outlook. Applied Sciences, 12:4811, 2022.

Rojtberg, Pavel and Rogers, David and Streeting, Steve and others. Ogre scene-oriented, flexible 3d engine. https://www.ogre3d.org/, 2001 – 2024.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In Proceedings of the IEEE/CVF international conference on computer vision, pp. 9339–9347, 2019.

Dominik Schraml and Gunther Notni. Synthetic training data in ai-driven quality inspection: The significance of camera, lighting, and noise parameters. Sensors, 24(2), 2024. ISSN 1424-8220. doi: 10.3390/s24020649. URL https://www.mdpi.com/1424-8220/24/2/649.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, et al. igibson 1.0: A simulation environment for interactive tasks in large realistic scenes. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 7520–7527, 2021.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10740–10749, 2020.

Russell Smith et al. Open dynamics engine, 2005.

Mark W Spong, Seth Hutchinson, and M Vidyasagar. Robot modeling and control. John Wiley &amp, 2020.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797, 2019.

Jiayi Su, Youhe Feng, Zheng Li, Jinhua Song, Yangfan He, Botao Ren, and Botian Xu. Artformer: Controllable generation of diverse 3d articulated objects. arXiv preprint arXiv:2412.07237, 2024.

Xiaohao Sun, Hanxiao Jiang, Manolis Savva, and Angel Xuan Chang. Opdmulti: Openable part detection for multiple objects, 2023. URL https://arxiv.org/abs/2303.14087.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. Advances in Neural Information Processing Systems, 34:251–266, 2021.

Alessandro Tasora, Radu Serban, Hammad Mazhar, Arman Pazouki, Daniel Melanz, Jonathan Fleischmann, Michael Taylor, Hiroyuki Sugiyama, and Dan Negrut. Chrono: An open source multi-physics dynamics engine. In High Performance Computing in Science and Engineering: Second International Conference, HPCSE 2015, Soláň, Czech Republic, May 25-28, 2015, Revised Selected Papers 2, pp. 19–49, 2016.

Gemini Robotics Team. Gemini robotics: Bringing ai into the physical world, 2025.

Unity Technologies. Unity, 2025.05.14. URL https://unity.com/. Game development platform.

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

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. Arxiv, 2024.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of machine learning research, 9(11), 2008.

Haowen Wang, Zhen Zhao, Zhao Jin, Zhengping Che, Liang Qiao, Yakun Huang, Zhipeng Fan, Xiuquan Qiao, and Jian Tang. Sm 3: Self-supervised multi-task modeling with multi-view 2d images for articulated objects. In International Conference on Robotics and Automation, pp. 12492–12498, 2024a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2025.

Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers, 2024b.

Webots. http://www.cyberbotics.com, 2018. URL http://www.cyberbotics.com. Open-source Mobile Robot Simulation Software.

Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. arXiv preprint arXiv:2412.13877, 2024.

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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11097–11107, 2020.

Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, and Kai Zhang. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model, 2023. URL https://arxiv.org/abs/2311.09217.

Han Xue, Liu Liu, Wenqiang Xu, Haoyuan Fu, and Cewu Lu. Omad: Object model with articulated deformations for pose estimation and retrieval, 2021.

Jie Yang, Kaichun Mo, Yu-Kun Lai, Leonidas J. Guibas, and Lin Gao. Dsg-net: Learning disentangled structure and geometry for 3d shape generation, 2022. URL https://arxiv.org/abs/2008.05440.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In International Conference on Learning Representations, 2023.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023.

Yinuo Zhao, Kun Wu, Tianjiao Yi, Zhiyuan Xu, Zhengping Che, Chi Harold Liu, and Jian Tang. Efficient training of generalizable visuomotor policies via control-aware augmentation. In Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, 2025.

(a) Kitchen (b) Small livingroom

Figure 8: Scenes: all articulated joints in the open state.

A Articulated Objects

ArtVIP comprises 992 articulated objects, encompassing 9 categories and 37 subcategories, with a total of 2156 prismatic joints and 1809 revolute joints. The detailed breakdown, including approximate human labor time, is presented in Tab. 3.

B Scenes

We provide sim-ready complex, dynamic environments—six in total: childrenroom, diningroom, kitchen, kitchen with parlor, large livingroom, and small livingroom (see Fig. 8 for two example scenes). Every object in these environments, including fixed furniture, supports physical interaction. This includes switches, small appliances, plush toys, laptops, books, spice jars, and more. For example, the kitchen environment contains a total of 65 joints, and all objects can be used just like their real-world counterparts. Robots can operate the light switch on the wall, open the refrigerator door, place items on shelves, or challenge their motion capabilities by crouching to open drawers beneath the stove top. Additionally, users can freely place the 992 articulated objects provided in ArtVIP into any of these environments via the Isaac Sim GUI, enabling the creation of rich robot interaction scenarios such as grasping, pulling, pressing, and placing. Users can also utilize open-source tools like mjcf2usd and urdf2usd to convert assets from other datasets into the USD format, allowing seamless integration with ArtVIP assets. This kind of sim-ready, complex environment is currently unique to ArtVIP. Moreover, the ability to edit and save assets directly through a GUI reflects an open-source spirit that is not yet common in other datasets.

C Annotations

Annotations in ArtVIP provide objective descriptions of object parts, thereby supporting robots' ability to infer task-appropriate interaction behaviors. We further argue that annotations are most meaningful when aligned with consistent modeling standards. For example, for a desk, modelers often merge the legs and tabletop into a single mesh, which limits part-level annotation based on distinct interaction functions. To address this, we highlight functional components in Tab. 4 that frequently participate in interactions yet are commonly overlooked during mesh segmentation. An example segmentation is shown in Fig. 9.

D Modeling Standards

In simulation systems, the use of high-quality meshes, textures, and materials confers several advantages. High-fidelity visuals reduce the disparity between simulation and reality (Nesti et al., 2025), thereby narrowing the sim-to-real gap and enabling robotic policies to be deployed in real-world environments with minimal or even zero-shot adaptation (Han et al., 2025; Embley-Riches et al., 2025). Photorealistic simulation data can be employed to train and validate visual perception algorithms, such as object detection, semantic segmentation, and SLAM. Moreover, realistic models not

Table 3: Detailed breakdown of object categories, modeling time (each), physics tuning time (each), and count.

category	subcategories	modeling time	physics tuning time	count
furniture	chair	2h	0.3h	23
	table	1.5h	0.2h	131
	cabinet	3.1h	0.4h	183
	cupboard	15h	2h	11
	bed	3h	0.3h	28
	home decor	2.5h	0.3h	30
kitchenware	cookware	1.9h	0.3h	81
kitchen appliances	coffee machine	3h	0.3h	14
	built-in oven	5h	0.3h	14
	microwave	3h	0.3h	8
	oven	4h	0.3h	11
	dishwasher	5h	0.4h	19
	water dispenser	3h	0.3h	6
	rice cooker	3h	0.3h	14
	fridge	6h	0.5h	22
	juicer	4h	0.3h	6
fixtures	faucet	2h	0.2h	14
	toilet	4h	0.4h	14
	door	2h	0.3h	10
appliances	computer	2.5h	0.3h	13
	fan	1.8h	0.3h	34
	air conditioner	4h	0.3h	3
	washing machine	5.7h	0.5h	30
	speaker	1.5h	0.3h	14
	floor lamp	1h	0.3h	28
cleaning tools	mop	2h	0.3h	8
	pump bottle	2h	0.3h	14
	trash can	2h	0.3h	18
stationery	scissors	1h	0.2h	28
	stapler	1.5h	0.2h	11
	utility knife	1h	0.2h	19
	folder	0.5h	0.2h	8
storage	storage box	2h	0.3h	25
	toolbox	2.5h	0.3h	22
	cardboard box	1.5h	0.2h	28
Mechanical equipment	electrical equipment	3h	0.3h	17
	non-electrical equipment	3h	0.3h	33

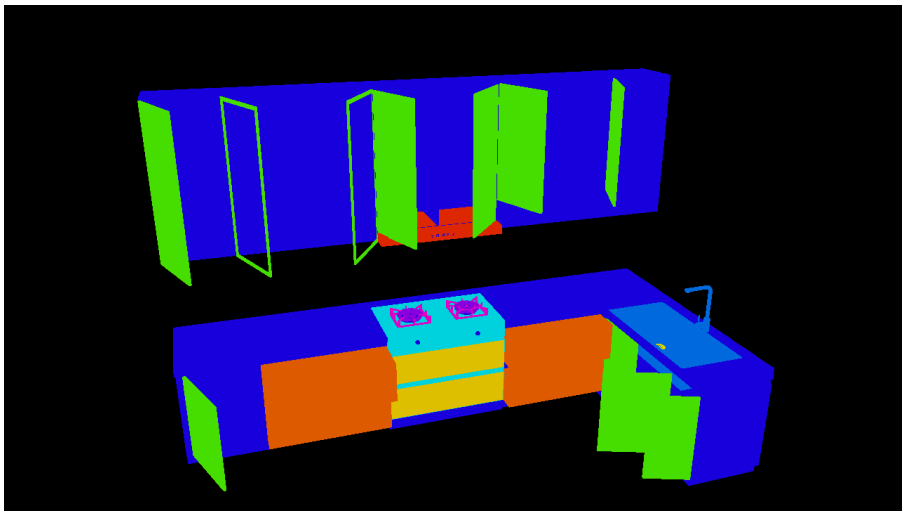


Figure 9: Segmentation result of the kitchen scene.

only enhance visual fidelity but also improve interaction effects within simulations. When robots perform actions such as grasping, collision, or force-based interactions, accurate geometry ensures stable and reliable feedback. To achieve photorealistic appearance and minimize the sim-to-real visual gap, we addressed the following standards:

Mesh. Manifold meshes form the core geometric foundation of each asset, defining the object's overall contour and spatial occupancy. These meshes are critical for generating collision bodies that maintain accuracy in physical interactions. ArtVIP ensures that mesh details produce smooth surfaces and lifelike contours, avoiding jagged or blocky appearances. Additionally, through normal vector optimization algorithms, redundant vertices are merged, reducing geometric data volume and thereby alleviating computational burdens in simulation.

Texture. Textures are mapped onto mesh surfaces via UV coordinates to provide visual details. ArtVIP employs high-resolution textures to capture fine surface characteristics, such as the metallic sheen of a refrigerator or the subtle grain of wood on a chair. Furthermore, textures are meticulously aligned with the UV map to prevent stretching, distortion, or visible seams.

Material. A material is a collection of rendering parameters, including references to textures, that defines how an object's surface responds to light. ArtVIP leverages RTX Renderer (Nvidia, 2025b) in Isaac Sim and adopts Physically Based Rendering (PBR) (Nvidia, 2025a) to accurately simulate diffuse and specular reflections, enabling rendering effects such as roughness and emissive properties. This approach allows for the realistic representation of diverse materials, achieving true-to-life visual fidelity.

E Physical Fidelity of Joints

To achieve physical fidelity of dynamic joint and simulate variable joints motions in the real world, we enhance the joint drive equation originally provided by Isaac Sim:

$$\tau = K(q) \cdot (q - q_{\text{target}}(q)) + D \cdot (\dot{q} - \dot{q}_{\text{target}}(q)) \tag{2}$$

where τ represents the force(F) and torque(T) applied to drive the joint, q and $\dot{q}$ are the joint position and velocity, respectively, D donates damping, and K donates stiffness. While this equation can model basic joint motions, it fails to fully replicate complex dynamic joint motions in the real world. For complex joints such as door closers and light switches, τ may vary with q and $\dot{q}$. To accommodate the above situations, we design functions of q and $\dot{q}$.

Impact from $\dot{q}$. Friction must be accounted for in simulation and cannot be modeled as a constant. It imposes resistance to the force generated by the joint drive τ, and we propose the following equation

Table 4: Annotation labels and descriptions in ArtVIP.

Label	Description
armrest	Chair armrest
backrest	Chair backrest
ball_handle	Handle for lifting the main body, such as the handle of a toolbox
blade	Blade of a utility knife, scissors, or fan blades
body	Parts that need labeling excluding base and lid
button	Applies to all push-button switch components of models
door	Door of cabinets, refrigerators, ovens, etc.
drawer	Drawer of cabinets, refrigerators, toolboxes, etc.
front_cover	Cover of a folder
fun_guard	Fan protective cover
handle	Any handles
headrest	Chair headrest
jaw	Head of pliers, the part that contacts the gripped item
keyboard	Computer keyboard
knob	Applies to all rotary switch components of models
lid	Such as cardboard box lid, electric steamer lid, trash can lid
light	All types of lights
mop_head	Mop head
pedal	Foot pedal, such as on a step-on trash can
pipe	Water pipe part of faucet
plate	All types of plates
pole	Rod-shaped component
portafilter	A handle holds the coffee grounds
pot	Inner pot of rice cookers, steamers, etc.
rack	Rack in an oven, refrigerator door shelf
roller	Washing machine drum
screen	Electronic product screen
seat	Chair seat
shelf	Shelf part of cabinets, refrigerators, etc.
spout	Spout of a pump bottle, water dispenser, etc.
stapler_magazine	Staple compartment of a stapler
tabletop	Top surface of a table
toilet_seat	Toilet seat
touch_pad	Computer touchpad
wheel	Chair wheels

with three different conditions:

$$F_{\text{friction}}(\dot{q}) = \begin{cases} -F_{\text{ext}} & \dot{q} = 0 \text{ and } |F_{\text{ext}}| \leq \mu_s \cdot (|F| + |T|) \quad (3a) \\ -\mu_s \cdot (|F| + |T|) \cdot \text{sign}(F_{\text{ext}}) & \dot{q} = 0 \text{ and } |F_{\text{ext}}| > \mu_s \cdot (|F| + |T|) \quad (3b) \\ -D \cdot \dot{q} \cdot \text{sign}(\dot{q}) & \dot{q} \neq 0 \quad (3c) \end{cases}$$

We illustrate the friction from static friction, to maximum static friction, and finally to dynamic friction, corresponding to conditions from Eqn. equation 3a through Eqn. equation 3c. F_{ext} denotes the static friction. The coefficient u_s denotes the static friction coefficient, which can be configured in Isaac Sim via the `Joint Friction` parameter. The *sign* function ensures that the frictional force is applied in the correct direction.

Impact from q. The latch release mechanism exemplifies the position-dependent joint drive, we analyze a button-actuated trash bin lid mechanism. When the button is depressed, it triggers a linkage to retract the spring-loaded latch, enabling the lid to freely rotate under torsional spring torque to $q_{\text{upper\_bound}}$.

$$q_{\text{target}}(q) = \begin{cases} q_{\text{upper\_bound}} & \text{if } q > q_{\text{threshold}} \text{ and } S_{\text{open}} = 1 \quad (4a) \\ q_{\text{lower\_bound}} & \text{if } q < q_{\text{threshold}} \text{ and } S_{\text{open}} = 0 \quad (4b) \end{cases}$$

We further investigate joint motion with abrupt stiffness variations, exemplified by refrigerator door closers and magnetic latching mechanisms. To maintain static equilibrium in the stationary state, a high stiffness value k_{high} is employed. When $S_{\text{open}} = 1$ (door opening phase), the stiffness progressively decreases with increasing q. Upon exceeding the critical position $q_{\text{threshold}}$, the stiffness reaches its minimum k_{low}, and the joint target position switches to $q_{\text{upper\_bound}}$. During door closure, as q approaches $q_{\text{threshold}}$ from above, the target position abruptly transitions to $q_{\text{lower\_bound}}$, accompanied by an exponential stiffness surge to rapidly complete closure, emulating commercial door closer dynamics. This behavior is formalized as:

$$K(q) = \begin{cases} k_{\text{high}}, & q = q_{\text{lower\_bound}} & \text{(5a)} \\ k_{\text{high}} - \alpha q, & \text{if } q_{\text{lower\_bound}} < q \leq q_{\text{threshold}} \text{ and } S_{\text{open}} = 1 & \text{(5b)} \\ k_{\text{low}} + k_{\text{max}} e^{-\lambda q}, & \text{if } q_{\text{lower\_bound}} < q \leq q_{\text{threshold}} \text{ and } S_{\text{open}} = 0 & \text{(5c)} \\ k_{\text{low}}, & q_{\text{threshold}} < q < q_{\text{upper\_bound}} & \text{(5d)} \end{cases}$$

F Chart Comparison with Existing Datasets

We present a detailed comparison of ArtVIP with existing articulated-object datasets in Tab. 5

Table 5: Detailed comparison with existing articulated-object datasets.

	ArticulatedAssets	Prismatic joints	Revolute joints	Visual realism	Physical fidelity	Modular interaction
ArtVIP	992	2156	1809	high	high	394
BEHAVIOR-1K	545	318	819	medium	low	None
PartNet-Mobility	2347	7659	4312	low	low	None

G Visual Realism Comparison

We present further comparative analysis in Fig. 10. PartNet-Mobility employs the URDF format, with meshes stored in OBJ format and material information defined in MTL files. Although the OBJ files are manually crafted, they frequently exhibit distorted meshes, significantly compromising visual quality. The MTL material format inherently lacks the capability to model physically accurate light reflection, resulting in a lack of environmental realism across all PartNet-Mobility assets. Our analysis reveals that many materials in PartNet-Mobility rely solely on base color for rendering, and the absence of textures substantially degrades the overall rendering quality. Although BEHAVIOR-1K adopts the USD format, which supports physically based rendering (PBR), it still suffers from issues related to distorted meshes and poor texture quality.

To mitigate issues such as distorted meshes and angular surfaces, we employed a high number of triangular faces to ensure smooth surfaces and enhanced geometric detail. For categories such as toilets and refrigerators, ArtVIP significantly surpasses BEHAVIOR-1K and PartNet-Mobility in the number of triangular faces utilized. However, this approach entails a trade-off, as it reduces the simulation frame rate. To address this, we conducted profiling analysis to optimize the simulation frame rate for each object. In our experiments, we selected the kitchen, which contains the highest number of articulated objects, and the living room, which features the most extensive texture rendering, as testing environments. Each asset from ArtVIP was individually placed within these scenes, ensuring that the overall rendering frame rate consistently exceeds 60 Hz (i7-13700, Nvidia 4090, 64 GB).

H Physical Fidelity and Interaction Evaluations

Motion Triggered by Latch Release. To validate the modular interaction within assets, we compared the triggered joint in both real-world and virtual microwave. We conducted button-press experiments in each environment to initiate the door-opening action and recorded the resulting door motion trajectories. In the real-world tests we tracked a marker on the door using the optical tracking system to capture its spatial motion after the button pressed. In the simulation we set a virtual

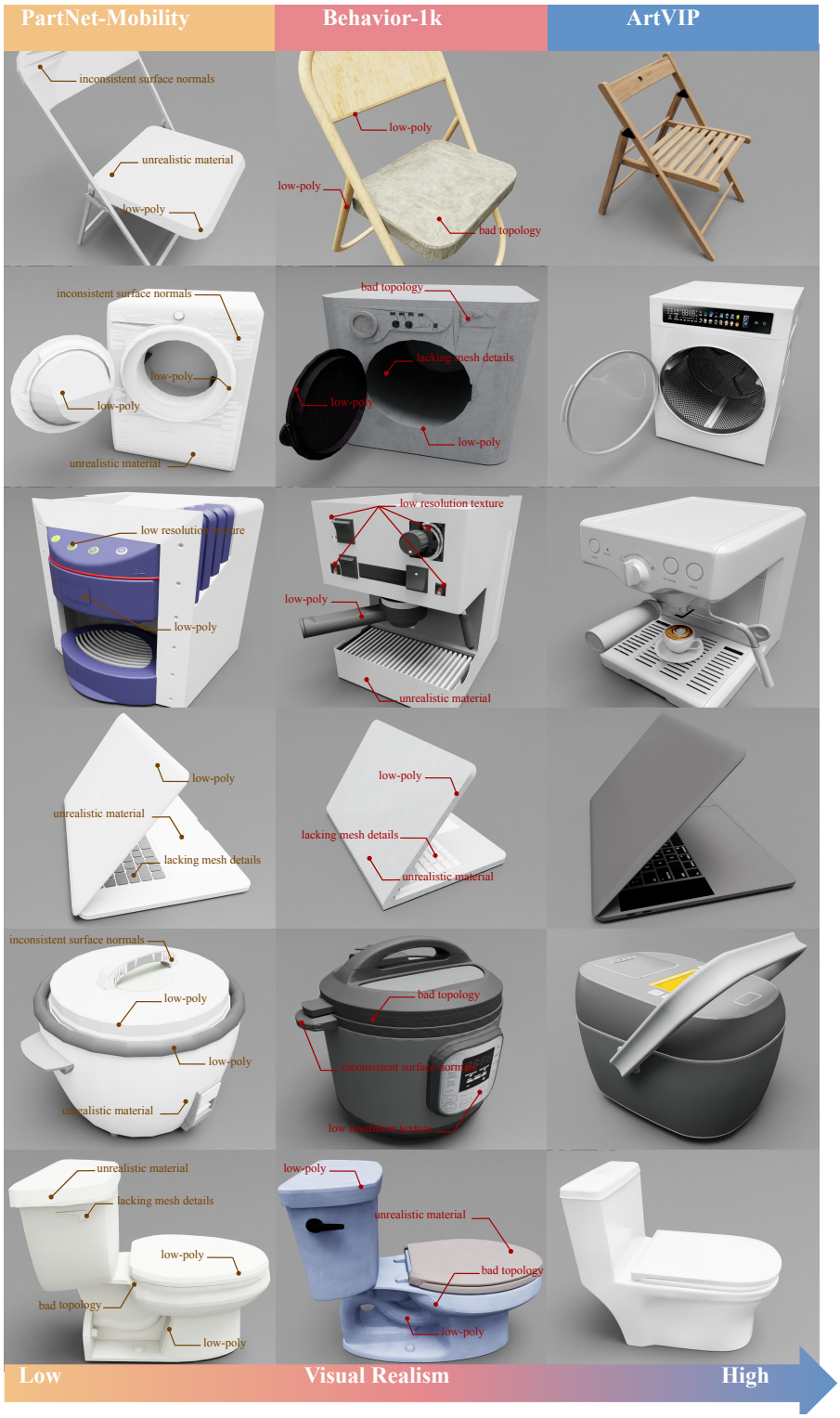


Figure 10: Comparisons of ArtVIP, BEHAVIOR-1K, and PartNet-Mobility.

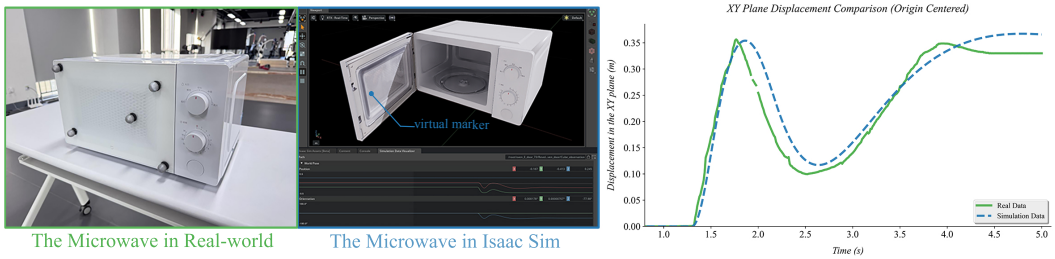


Figure 11: **Left and Middle:** Digital-twin asset examples in real-world and simulation. **Right:** Analysis of the Microwave's displacement.

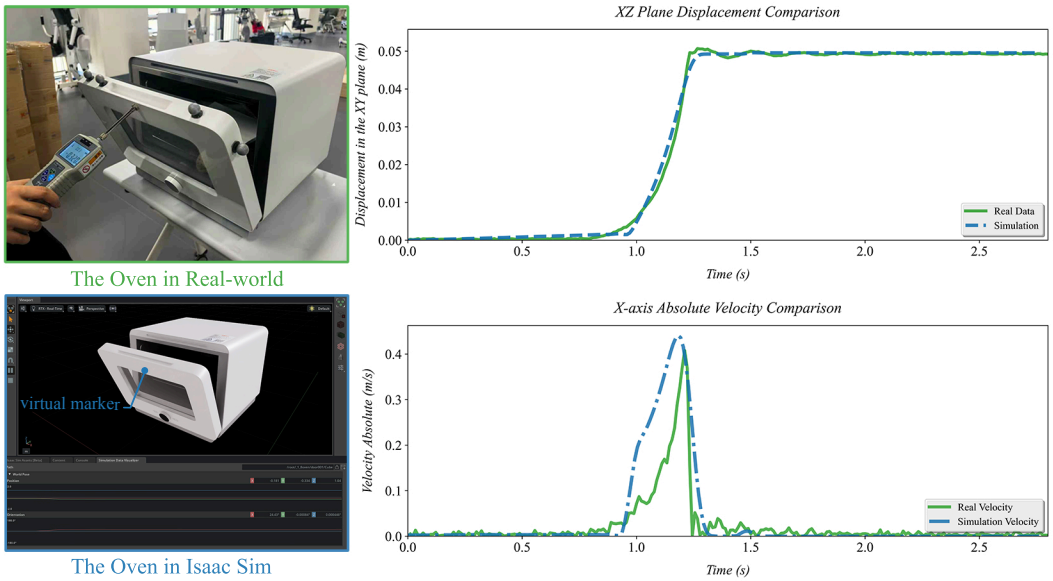


Figure 12: **Left:** Digital-twin asset examples in real-world and simulation. **Right:** Analysis of the oven's displacement.

marker at the same position as the real-world marker on the door, and we triggered the door opening via pressing the button as well (for which the activation configured in modular interaction) and logged the virtual marker's trajectories. We performed ten trials in each environment and computed the average spatial trajectory as Fig. 11 shown.

Motion Triggered by Joint Position Threshold. Appliances equipped with door closers typically exhibit a dynamic change in motion once the door reaches a certain angle during closing. After arriving at a certain angle, the door closer causes the door to accelerate and snap shut against the appliance body. To evaluate how well the simulation captures this physical transition, we focus on analyzing the door's linear and angular velocities during the transition from the threshold state to full closure. In both the simulation and real-world experiments, a force of no more than 1.0 N is applied when the door is within the threshold range to trigger the door closer mechanism. We then record the kinematic behavior following the activation of the door closer. In the real-world setup, the optical motion capture system is used to track the spatial displacement of markers on the door. Both the simulation and real-world experiments are repeated ten times, and we compute the average spatial trajectories and changes in velocity along the X-axis for quantitative comparison (Fig. 12).

I Imitation Learning Application

Task Summary. As shown in Fig. 13, we design four challenging articulated-object manipulation tasks: (1) **PullDrawer**, (2) **OpenCabinet**, (3) **SlideShelf**, and (4) **CloseOven**. These tasks demand

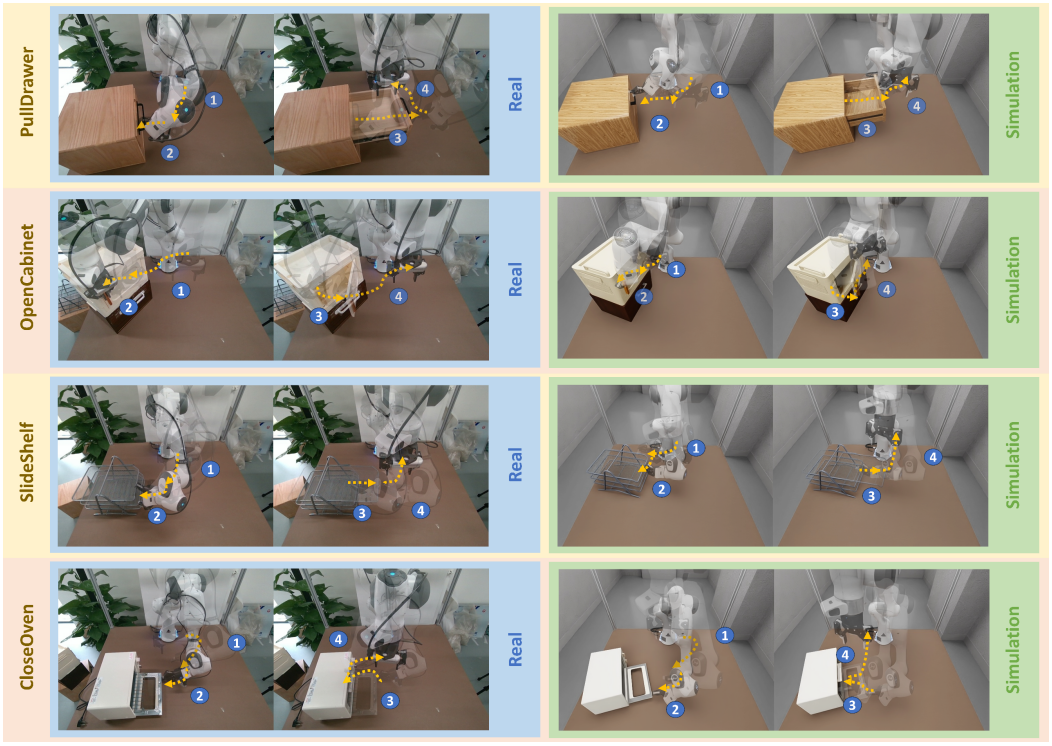


Figure 13: The four articulated-object manipulation tasks conducted for imitation learning.

precise and flexible motions, including rotation, angled pushing, and horizontal translation. We define these tasks as follows:

- **PullDrawer**. his task requires the robot to insert the gripper into the handle of the drawer, securely press the handle, and gradually pull the drawer out along a linear trajectory using a smooth and consistent motion.
- **OpenCabinet**. For this task, the robotic arm needs to precisely locate the thin vertical handle of the cabinet door. The gripper has to align vertically, firmly grip the handle, and pull the door outward along a curved path while maintaining a stable trajectory.
- **SlideShelf**. This task involves horizontal manipulation of the shelf. First, the gripper needs to rotate around 90 degrees to align parallel to the shelf's direction. It then grips the base of the shelf and moves horizontally, pulling the shelf out along its guide rails in a stable and controlled manner.
- **CloseOven**. To complete this task, the robotic arm needs to close its gripper to push against the bottom edge of the oven door. The arm then rotates and lifts under the door, applying a curved upward force to close the door.

Imitation Learning Algorithm. The input to the imitation learning models consists of RGB image data from multiple camera views and the robot's proprioceptive states. The output is the robot control signals, such as joint positions, enabling end-to-end task execution. We used two state-of-the-art imitation learning methods, Action Chunking Transformer (ACT) (Zhao et al., 2023) and Diffusion Policy (DP) (Chi et al., 2023), to train the robotic policies for the articulated object manipulation task. Hyperparameters of both methods are demonstrated in Tab. 6 and Tab. 7.

- **Action Chunking Transformer (ACT)** (Zhao et al., 2023): ACT is built on the transformer network architecture and leverages temporal ensemble techniques to produce fluid and precise action sequences.
- **Diffusion Policy (DP)** (Chi et al., 2023): DP employs a diffusion-based generative model that captures multi-modal action distributions, offering robustness and high success rates for complex robotic tasks.

	Hyperparameter	Value		Hyperparameter	Value
Training	Batch size	48	Network Architectures	Encoder layer	4
	Learning rate	1e-4		Decoder layer	7
	Optimizer	AdamW		Forward dim	3200
	KL weight	10		Heads num	8
	Action sequence	50		Transformer hidden dim	512
	Training step	50k		Backbone	ResNet50

Table 6: Implementation details of Action Chunking Transformer (ACT).

	Hyperparameter	Value		Hyperparameter	Value
Training	Batch size	48	Network Architectures	Diffuion Network	Unet1D
	Learning rate	1e-4		Pooling	SpatialSoftmax
	Optimizer	AdamW		Noise scheduler	DDIM
	EMA power	0.75		EMA model	True
	Action sequence	16		Noise schedule	SquaredcosCap
	Training step	50k		Backbone	ResNet50

Table 7: Implementation details of Diffusion Policy (DP).

Detailed Experiment Results. The Full experiment results are presented in Tab. 8.

J Reinforcement Learning Application

Training Details. We extend the visual RL framework EAGLE (Zhao et al., 2025) to articulated-object tasks in ArtVIP. Fig. 14a shows the CloseTrashcan task, where the robot arm is required to close the trashcan within a given time limit. EAGLE is a two-stage visual RL framework designed for efficiency and generalization. In Stage 1, the teacher policy receives low-level states, including the robot arm's proprioceptive input, the lid's joint value, and the 3D relative position between the trashbin and the gripper. In Stage 2, the student policy is provided only with the wrist camera image and the robot's proprioceptive state—no object-related states are available. Fig. 14b presents the training curves in Stage 2.

For implementation details, in Stage 1, we replace EAGLE's original RL agent with PPO; In Stage 2, a privileged-state teacher is distilled into a visuomotor student while a self-supervised attention mask learned as follows:

$$\mathcal{L}_{att} = \mathcal{L}_{rec} + \mathcal{L}_{ae} + \beta\mathcal{L}_{ctl} + \lambda\mathcal{L}_{sps}, \tag{6}$$

where $\mathcal{L}_{rec}$ and $\mathcal{L}_{ae}$ are reconstruction losses, $\mathcal{L}_{ctl}$ predicts dynamics, and $\mathcal{L}_{sps}$ enforces mask sparsity. Hyper-parameters β and λ weight auxiliary losses.

The student policy is trained with the distillation loss:

$$\hat{\mathcal{L}}(\pi_\theta) = \mathbb{E}_{(\mathbf{o},\mathbf{s})\sim\mathcal{D}}\left[\|\pi_\theta(\mathbf{o}_{\text{aug}}) - \pi_e(\mathbf{s})\|_2^2\right], \tag{7}$$

where $\mathbf{s}$ contains privileged states and $\mathbf{o}_{\text{aug}}$ are images augmented by the learned mask with Eqn. equation 6. Hyper-parameters used in EAGLE are listed in Tab. 9.

Reward Functions. The **CloseTrashcan** task is a long-horizon challenge requiring the robot to first approach the trashcan lid and then close it smoothly. To facilitate efficient RL training, we design a multi-objective reward function as follows:

$$r_t(\boldsymbol{s}_t, \boldsymbol{a}_t) = \lambda_1 r_{dst}(\boldsymbol{s}_t) + \lambda_2 r_{dir}(\boldsymbol{s}_t) + \lambda_3 r_{cls}(\boldsymbol{s}_t) + \lambda_4 r_{smth}(\boldsymbol{a}_t), \tag{8}$$

where r_{dst} rewards proximity between the gripper and the lid, r_{dir} encourages alignment toward the lid, r_{cls} measures lid closure progress, and r_{smth} promotes smooth actions. The reward weights are set as: $\lambda_1 = 0.5$, $\lambda_2 = 0.125$, $\lambda_3 = 10$, $\lambda_4 = -0.01$.

Baseline Comparison. To put EAGLE's performance in context, we compare it with a vision-based PPO method. As shown in Tab. 10, due to the high computation complexity and low data diversity,

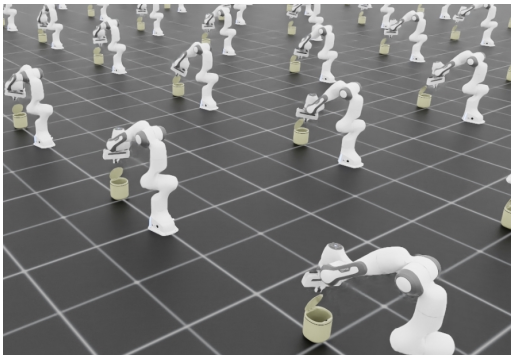
Table 8: Performance results (scheme A): per-seed scores and mean ± 90% CI

Task	Method	Strategy	seed1	seed2	seed3	Mean ± CI90
PullDrawer	ACT	RO	0.567	0.767	0.600	0.644 ± 0.059
		SO	0.433	0.433	0.300	0.389 ± 0.060
		RSM100+10	0.500	0.667	0.767	0.640 ± 0.059
		RSM100+20	0.667	0.600	0.767	0.678 ± 0.057
		RSM100+50	0.833	0.767	0.733	0.778 ± 0.051
		RSM100+100	0.767	0.867	0.800	0.811 ± 0.048
	DP	RO	0.600	0.733	0.633	0.656 ± 0.058
		SO	0.133	0.233	0.233	0.200 ± 0.049
		RSM100+10	0.600	0.650	0.700	0.650 ± 0.057
		RSM100+20	0.650	0.700	0.733	0.694 ± 0.056
		RSM100+50	0.700	0.733	0.750	0.728 ± 0.055
		RSM100+100	0.733	0.767	0.867	0.789 ± 0.050
OpenCabinet	ACT	RO	0.300	0.400	0.333	0.344 ± 0.058
		SO	0.167	0.100	0.100	0.122 ± 0.040
		RSM100+10	0.333	0.367	0.367	0.356 ± 0.059
		RSM100+20	0.367	0.400	0.367	0.378 ± 0.059
		RSM100+50	0.433	0.500	0.400	0.444 ± 0.061
		RSM100+100	0.567	0.367	0.433	0.456 ± 0.061
	DP	RO	0.467	0.500	0.500	0.489 ± 0.061
		SO	0.133	0.033	0.133	0.100 ± 0.037
		RSM100+10	0.500	0.533	0.567	0.533 ± 0.058
		RSM100+20	0.550	0.583	0.600	0.578 ± 0.057
		RSM100+50	0.600	0.617	0.633	0.617 ± 0.057
		RSM100+100	0.667	0.700	0.600	0.656 ± 0.058
SlideShelf	ACT	RO	0.233	0.233	0.333	0.267 ± 0.054
		SO	0.100	0.167	0.133	0.133 ± 0.042
		RSM100+10	0.200	0.267	0.300	0.256 ± 0.053
		RSM100+20	0.233	0.300	0.267	0.267 ± 0.054
		RSM100+50	0.300	0.367	0.300	0.322 ± 0.057
		RSM100+100	0.333	0.333	0.400	0.356 ± 0.059
	DP	RO	0.467	0.433	0.433	0.444 ± 0.061
		SO	0.167	0.167	0.200	0.178 ± 0.047
		RSM100+10	0.433	0.467	0.500	0.467 ± 0.058
		RSM100+20	0.500	0.533	0.550	0.528 ± 0.057
		RSM100+50	0.533	0.567	0.583	0.561 ± 0.056
		RSM100+100	0.567	0.600	0.600	0.589 ± 0.060
CloseOven	ACT	RO	0.500	0.633	0.600	0.578 ± 0.061
		SO	0.267	0.267	0.167	0.233 ± 0.052
		RSM100+10	0.500	0.600	0.667	0.589 ± 0.060
		RSM100+20	0.533	0.633	0.633	0.600 ± 0.060
		RSM100+50	0.733	0.533	0.700	0.656 ± 0.058
		RSM100+100	0.667	0.800	0.567	0.678 ± 0.057
	DP	RO	0.600	0.700	0.667	0.656 ± 0.058
		SO	0.267	0.233	0.333	0.278 ± 0.055
		RSM100+10	0.633	0.667	0.700	0.667 ± 0.057
		RSM100+20	0.667	0.700	0.733	0.700 ± 0.056
		RSM100+50	0.700	0.733	0.750	0.728 ± 0.055
		RSM100+100	0.767	0.733	0.833	0.778 ± 0.051

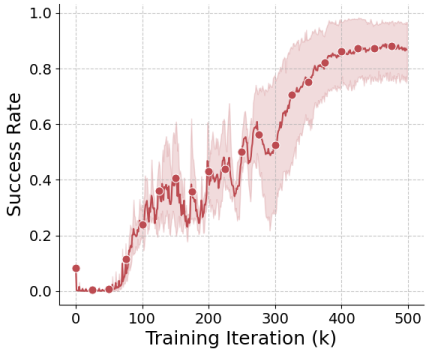
the baseline performs poorly on the CloseTrashcan task, while EAGLE achieves a 98% success rate after 500k training iterations.

K THE USE OF LARGE LANGUAGE MODELS.

A large language model (LLM) was used strictly as a writing aid for language polishing (grammar, clarity, and style). All ideas, methodological designs, datasets, code, analyses, and results are original and solely produced by the authors.



(a) Training task.



(b) Training curve over five random seeds.

Figure 14: RL-based training of visuomotor policy with ArtVIP.

	Hyperparameter	Value
Teacher (Stage 1)	Learning rate for all net	5e-4
	Optimizer	Adam
	Batch size	12×4096
	Discount factor	0.99
	Clip ratio	0.2
	Rollout size	96×4096
Student (Stage 2)	Observation	128×128
	Learning rate for all net	1e-4
	Optimizer	Adam
	Batch size	256
	Frame stack	1
	Replay buffer size	100k
	λ	0.01
	β	0.5
	α in *random overlay*	linear schedule from 0.4 to 0.9

Table 9: Hyperparamters for EAGLE.

Table 10: EAGLE vs. vision-based PPO: success rate across training checkpoints (k).

Checkpoint (k)	100	200	300	400	500
EAGLE	0.23	0.28	0.73	0.85	0.98
Vision-based PPO	0.16	0.19	0.21	0.22	0.24