ARTIST: Articulated Real-To-Interactive-Sim Twin

Anonymous CoRL submission

Paper ID *****

Abstract

*Real-to-Sim-to-Real frameworks enable data-efficient robot learning by leveraging realistic simulations, but existing approaches struggle to reconstruct articulated objects without manual interaction or dense multi-view observations. We present ARTIST (**A**rticulated **R**eal-**T**o-**I**nteractive-**S**im **T**win), a framework that automatically builds digital twins of articulated objects from a single monocular video. ARTIST first reconstructs and decomposes objects into parts by combining monocular 3D reconstruction with open-vocabulary segmentation, and then estimates articulations by adapting an actor–critic vision–language model to operate on reconstructed parts. On the ArtVIP dataset, ARTIST improves both 3D asset reconstruction and articulation estimation for previously unseen real-world objects. Finally, we demonstrate that ARTIST enables Real-to-Sim-to-Real transfer by replaying a single robot demonstration in simulation, highlighting its potential for scalable robot learning with minimal supervision.*

1. Introduction

While recent advances in imitation learning have enabled robots capable of manipulating a variety of objects, they still struggle with long-term manipulation tasks, as these require a prohibitively large number of real-world interactions. Real-to-Sim-to-Real frameworks [1, 10, 21] aim to mitigate data requirements by reconstructing real-world scenes in physics-based simulators, which can then be used for demonstration augmentations [1] or fine-tuning [21]. However, the automatic creation of articulated objects (e.g., doors, drawers, laptops) in such frameworks still requires human intervention [1, 21].

Two main challenges arise when reconstructing articulated objects. The first is estimating the 3D parts of the object, and the second is modeling the correct interactions between these parts. While recent advances in 3D generation enable the rapid creation of realistic meshes [2, 3, 7, 14, 16, 20, 25, 26], most methods are not part-aware [3, 14, 20, 25] or articulation-aware [2, 7, 16, 26]. Specialized approaches for articulated object generation have been proposed [5, 9, 11, 12, 15, 17], but these typically require dense multi-state observations [5, 11, 15] or are limited to narrow domains [9, 12], making it difficult to generalize to diverse object categories. More recently, Articulate-Anything [8] demonstrated that vision-language models (VLMs) can serve as actor-critics to estimate articulations, but the method relies on a precomputed dataset of 3D parts, limiting adaptability to novel objects.

In this work, we address these limitations by proposing a framework for reconstructing articulated objects from a single monocular video. Specifically, we perform 3D part decomposition by combining monocular reconstruction [24] with 2D open-vocabulary segmentation [19]. To estimate interactions between reconstructed parts, we adapt the actor–critic mechanism from [8] to operate on reconstructed assets. We introduce ARTIST (**A**rticulated **R**eal-**T**o-**I**nteractive-**S**im **T**win), a framework that automatically reconstructs articulated objects suitable for Real-to-Sim-to-Real pipelines. We evaluate ARTIST on the ArtVIP dataset and further demonstrate its utility by replaying robot demonstration trajectories in simulation.

In summary, our contributions are:

1. We introduce ARTIST, a novel framework for reconstructing articulated Real-to-Sim-to-Real assets from monocular video.
2. We propose a method for reconstructing 3D object parts by combining monocular reconstruction with open-vocabulary segmentation.
3. We adapt actor-critic VLMs to estimate interactions between reconstructed parts.

2. Method

The objective is to create an articulated digital twin of a target object, represented in URDF file format, including part-object meshes and detailed joint information. ARTIST takes as input a video demonstration of interaction with an articulated object and creates a manipulable digital twin that can be used in simulation. It makes use of state-of-the-art mesh generation, 3D mesh segmentation, articulation estimation, and integrates with robot manipulation methods to

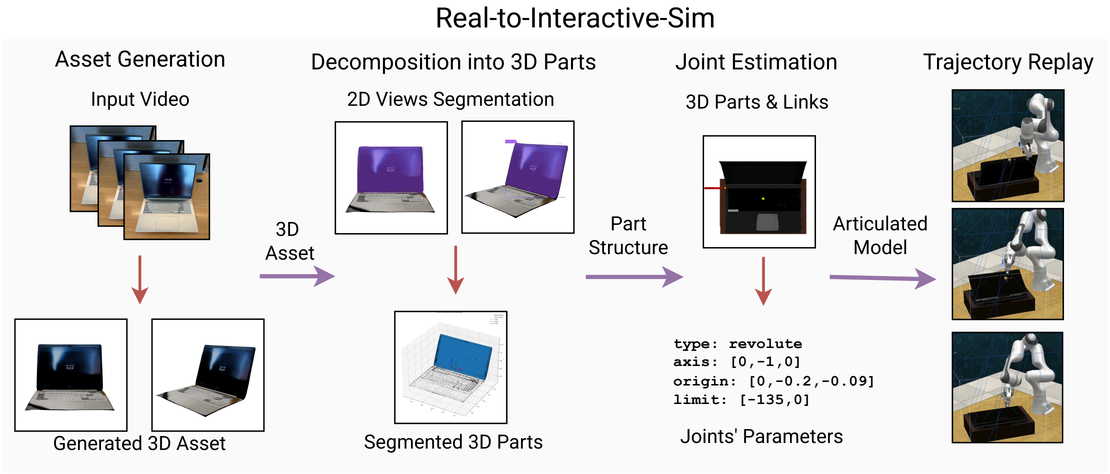


Figure 1. Overview of the four main stages from ARTIST. First, the object is reconstructed using TRELLIS [24] using few views. Second, the 3D asset is decomposed into 3D parts via DINO-X [19] and instance segmentation with SAM [18]. Then, estimation of articulated joints is performed by prompting Gemini 2.5 Flash in an actor-critic way. Finally, the 3D asset can be used for trajectory replay in a physics simulator, suitable for policy learning [1].

create a full pipeline from input views to interactable and realistic simulations for real-world articulated objects (see Figure 1 for details).

2.1. 3D Generation for Real-to-Sim Assets

A key requirement for real-to-sim transfer is the ability to reconstruct target objects with high fidelity from potentially limited observation. For articulated objects, this challenge is amplified by the presence of small movable parts such as dials, buttons, and sliders, as well as unique geometries not covered in existing annotated datasets. Retrieval-based reconstruction [22], even when paired with large-scale resources such as PartNet-Mobility [23], cannot generalize to the wide variety of objects encountered in practice, particularly when unseen geometries or fine-scale components are present. To overcome these limitations, we propose to use generative models for asset generation, enabling high-quality mesh construction from real demonstrations. Given multiple views of an object in its resting state, sampled directly from demonstration videos, we first obtain a target object description via a VLM and pre-process the frames to remove background and other objects potentially present in the scene. Using object-centric images, we reconstruct precise textured meshes that preserve both global structure and fine details necessary for downstream manipulation.

We address the real-to-sim gap by generating physically plausible assets for policy learning. After benchmarking perceptual quality and Chamfer distance on PartNet-Mobility, we adopt TRELLIS [24] for its structurally accurate reconstructions needed for articulation estimation and multi-view conditioning support.

2.2. 3D-aware Part Decomposition

As the generated object meshes represent the entire asset and lack part-awareness, they need to be segmented into individual parts to later estimate their movement. The entire process, from input image to part-wise segmented mesh, is visualized in Figure A6.

2D Part Detection and Segmentation Since it is critical to correctly identify each of the movable parts, our method grounds the segmentation targets on the observed part movement given the input demonstration. A VLM describes each of the articulated parts in the input video and outputs a list of parts which are used as the target in the object detection step. (See Appendix A9 for the prompt).

As a next step, we render the generated asset from multiple views and apply open-vocabulary object detection combined with zero-shot segmentation to obtain 2D segmentation masks. In particular, we use DINO-X [19] to generate bounding boxes and SAM [18] to get instance-wise segmentations for every rendered view. We choose DINO-X as it outperformed Grounding DINO [13] and peers in our zero-shot part detection tests, consistent with LVIS rare-category benchmarks evaluation. As SAM produces semantic-agnostic masks, cross-view correspondence breaks and results vary with viewpoint/lighting; we address this in the next section by fusing 2D masks into aligned 3D part segmentations.

3D Part Decomposition Using the camera parameters and depth images, points are sampled from each segmen-

(a) Chamfer Distance for TRELLIS-generated vs. Articulate-Anything retrieved full assets on ArtVIP.

	ARTIST (Ours)	Articulate-Anything
Category	CD ↓	CD ↓
Household Items	**0.28** (n=29)	0.91 (n=29)
Small Appliances	**0.12** (n=17)	0.56 (n=15)
Major Appliances	**0.39** (n=34)	0.89 (n=18)
Small Furniture	**0.32** (n=21)	0.74 (n=21)
Large Furniture	**0.98** (n=19)	1.31 (n=12)
Overall	**0.41** (n=120)	0.86 (n=95)

(b) 3D Part Reconstruction on ArtVIP [6]. We compare ARTIST to Articulate-Anything [22]. Predicted and ground-truth parts are matched with Hungarian Matching.

	ARTIST (Ours)			Articulate-Anything		
Category	Part Acc. ↑	Mean CD ↓	Recall ↑	Part Acc. ↑	Mean CD ↓	Recall ↑
Household Items	6.9%	**0.60** (n=29)	**43.6%**	**10.3%**	1.72 (n=29)	31.6%
Large Furniture	**10.5%**	1.72 (n=19)	**28.6%**	0.0%	1.77 (n=12)	17.1%
Major Appliances	**2.9%**	**0.79** (n=34)	**12.6%**	0.0%	1.35 (n=18)	9.8%
Small Appliances	0.0%	**0.27** (n=17)	**25.2%**	0.0%	0.97 (n=15)	17.6%
Small Furniture	**4.8%**	**0.56** (n=21)	**33.3%**	0.0%	1.29 (n=21)	22.0%
Overall	**5.0%**	**0.78** (n=120)	**20.9%**	3.2%	1.44 (n=95)	18.5%

Table 1. Evaluation of ARTIST on ArtVIP dataset: (a) full-object reconstruction comparison; (b) 3D object parts reconstruction.

tation mask and reprojected into a shared 3D space. To ameliorate the multi-view segmentation issues, overlapping point clouds of the same label are merged based on point overlap calculated via KD clustering to avoid having multiple clouds of the same instance, ending up with one segmented point cloud per object part.

Since 2D object detectors will often highlight the entire object, segmentation masks may overlap, leading to fragmented output part meshes. To solve this, we remove points from overlapping masks in 2D for the same view favoring fine-grained masks. For example, if the masks for the whole drawer, an individual drawer, and its handle overlap, then the mask of the handle would be erased from the other segmentations (see Fig. A4 for details).

Since some parts might not be detected from a given view and thus overlap exists between separate views, we perform the same step in 3D by clustering the segmented point clouds and removing close enough points of larger masks (see Fig. A3). Finally, the mesh is clustered vertex-wise with each instance point cloud, then faces are assigned by majority vote. In this way, we transform individual 2D parts segmentations into a *consistent decomposition of the 3D assets into parts*. In addition, the texture for each part mesh is preserved by creating a UV texture map and saving the face to UV coordinate assignments.

2.3. Articulation Estimation

To estimate joint articulation, we make use of parts of the Articulate Anything pipeline, which we adapt to use generated meshes instead of part retrieval from the PartNet-Mobility [23] dataset.

To create a full URDF detailing the object with part meshes and joint parameters, we need to first estimate the link placement, infer the joint type and finally predict the joint parameters matching the original movement. Since we are generating assets in the exact state as they were observed, we can directly use the coordinates of each mesh as a fixed link placement step.

For the joint estimation, the VLM actor predicts python code, making use of pre-defined functions and in-context examples. The resulting generated code is compiled into URDF and rendered using Sapien renderer [23]. For each movable joint, a video is rendered in simulation where the joint is moved through its entire range. This predicted video, along with the ground truth manipulation, are given to the critic. The critics job is to give feedback in the form of a rating, failure reasons and possible improvements. It also receives in-context examples. This actor-critic loop concludes once the realism rating assigned by the critic exceeds a score of 9 out of 10.

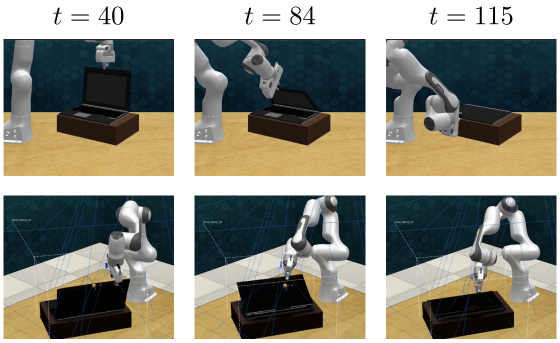


Figure 2. Trajectory replay on Digital Twin. **Top:** Original demonstration on ground truth RLBench [4] object. **Bottom:** Replay of robot trajectory using Digital Twin in the simulator.

2.4. Demonstration Replay

At this point the reconstructed object can be placed in simulation. To verify that it can be used for Real-To-Sim-To-Real we start from a simulated demonstration, reconstruct the object, place it inside the simulation, and replay the same trajectory from the demonstration. With this approach, we can compare the final state of the reconstructed and the original object. If manipulation of the twin leads to task success in the same way as the original across many different variations, it can be considered a good replica. In such case, digital twins become useful for imitation learning from few demonstrations [1].

3. Experiments

We aim to answer the following questions: 1. How well ARTIST can generate and decompose novel objects into parts using only one RGB image. 2. What is the quality of articulation estimation. 3. How to exploit the reconstructed asset in simulation.

3.1. Results for PartNet-Mobility Objects

We used a pre-processed version of the PartNet-Mobility dataset, following the approach of Articulate-Anything. We passed the front-view images to TRELLIS to generate a mesh and the VLM to name the parts. The joint critic compares a rendered video of the predicted URDF with a rendered video of the ground-truth object having their joints moved from lower to upper limit.

ARTIST achieves a 19.1% joint prediction success rate as shown in Figure A8. While lower than the Articulate-Anything baseline, we are generating meshes using TRELLIS [24], in contract to Articulate-Anything that also assumes known parts for retrival. A single front-view image conditioning often results in the generated meshes being either entirely flat, or having flat parts which should be the articulated parts. These generated meshes would not only cause the part decomposition to fail frequently, but also with separated part meshes the predicted joints would be outside of the threshold for correct joint origin predictions, thus resulting in failures. This especially seemed to be a problem for the Camera, Phone, and Remote categories, which made up a decent amount of all objects (together 56 objects).

Accounting for a large portion of failures is the part decomposition, mainly limited by 2D segmentation, since zero-shot open-vocabulary object part detection performance is still inconsistent [19]. In Figure A8, failure cases are broken down. Since our method does not require the link placement step, failures are only divided into joint estimation errors. When comparing the percentages of joint estimation errors between ARTIST and the baseline, joint axis, origin, and limit errors roughly account for the same amount of errors, whereas there is a large difference in joint type errors (7.7% against 22.5%).

3.2. Results for ArtVIP Objects

To test the quality and success rate of ARTIST compared to the current state-of-the-art, we compare our complete method against Articulate-Anything with mesh retrieval on the ArtVIP dataset. Unlike in the previous experiment, where Articulate-Anything worked with already provided meshes, in this case, the adaptability to recreate completely unseen objects can be highlighted, and generalization performance between both approaches can be verified. Since ArtVIP objects are of high visual and annotation quality, this also marks an important experiment for objects that are close to real-life objects and thus have implications for real-to-sim-to-real performance. To evaluate the success rate of part decomposition of ARTIST, we present part-wise results in Table 1. These highlight that ARTIST always predicts better matching parts than Articulate-Anything, as shown by higher recall across all categories. The structural precision and reconstruction quality of part meshes is also lower across all categories for ARTIST compared to the baseline, indicated by lower average Chamfer distance of parts.

3.3. Manipulation of Digital Twin in Simulation

To test whether the recreated objects can be used effectively in simulation by replaying original trajectories on the digital twin, we replayed 100 episodes with a randomized initial state (object placement and rotation). Out of the 100 trajectories recorded by manipulating the original object in RLBench [4], the replayed trajectories led to successful task completions on the digital twin in all 100 cases. For this initial test, we use the *close_laptoplid* task, in which the goal is to close the lid of a laptop positioned on a wooden stand. The goal condition is reached when the revolute joint connecting the lid to the base reaches its closed position. Results of the replay of one demonstration is visualized in Figure 2. This experiment shows that our generated digital twins recreate the original object very closely, and can be used in simulation to safely and efficiently learn manipulation of the original object, thus proving useful for a real-to-sim-to-real setting.

4. Conclusion and Future Work

We proposed ARTIST, a framework for articulated object reconstruction that combines monocular 3D part decomposition with articulation estimation using actor-critic VLM. On the ArtVIP dataset, ARTIST demonstrated improved reconstruction of novel objects. Moreover, the high-quality meshes produced were suitable for replaying robot trajectories in simulation.

Limitations. First, the reprojection of segmentation masks can be inconsistent from certain views. Fine-tuning the reconstruction method to be part-aware could solve the issue. Second, the actor–critic VLM sometimes predicts incorrect articulations. We believe that incorporating geometric supervision (e.g., by exploiting the differentiability of Gaussian splatting in TRELLIS [24]) can mitigate this issue. Finally, while we demonstrated trajectory replay, we plan to show that policies trained through interaction with such reconstructions can transfer back to the real world.

References

[1] Leonardo Barcellona, Andrii Zadaianchuk, Davide Allegro, Samuele Papa, Stefano Ghidoni, and Efstratios Gavves.

Dream to manipulate: Compositional world models empowering robot imitation learning with imagination, 2024. 1, 2, 3

[2] Daoyi Gao, Yawar Siddiqui, Lei Li, and Angela Dai. MeshArt: Generating Articulated Meshes with Structure-guided Transformers, 2024. arXiv:2412.11596 [cs] version: 1. 1

[3] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d, 2024. 1

[4] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment, 2019. 3, 4, 7

[5] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building Digital Twins of Articulated Objects from Interaction, 2022. arXiv:2202.08227 [cs]. 1

[6] Zhao Jin, Zhengping Che, Zhen Zhao, Kun Wu, Yuheng Zhang, Yinuo 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. 3, 6

[7] Juil Koo, Seungwoo Yoo, Minh Hieu Nguyen, and Minhyuk Sung. Salad: Part-level latent diffusion for 3d shape generation and manipulation, 2024. 1

[8] 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, 2024. arXiv:2410.13882 [cs]. 1

[9] Jiahui Lei, Congyue Deng, Bokui Shen, Leonidas Guibas, and Kostas Daniilidis. NAP: Neural 3D Articulation Prior, 2023. arXiv:2305.16315 [cs]. 1

[10] Xinhai Li, Jialin Li, Ziheng Zhang, Rui Zhang, Fan Jia, Tiancai Wang, Haoqiang Fan, Kuo-Kun Tseng, and Ruiping Wang. Robogsim: A real2sim2real robotic gaussian splatting simulator, 2024. 1

[11] Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. PARIS: Part-level Reconstruction and Motion Analysis for Articulated Objects, 2023. arXiv:2308.07391 [cs]. 1

[12] Jiayi Liu, Denys Iliash, Angel X. Chang, Manolis Savva, and Ali Mahdavi-Amiri. SINGAPO: Single Image Controlled Generation of Articulated Parts in Objects, 2024. arXiv:2410.16499 [cs]. 1

[13] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2024. 2

[14] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3d: Single image to 3d using cross-domain diffusion, 2023. 1

[15] Zhao Mandi, Yijia Weng, Dominik Bauer, and Shuran Song. Real2Code: Reconstruct Articulated Objects via Code Generation, 2024. arXiv:2406.08474 [cs]. 1

[16] Kiyohiro Nakayama, Mikaela Angelina Uy, Jiahui Huang, Shi-Min Hu, Ke Li, and Leonidas J Guibas. Difffacto: Controllable part-based 3d point cloud generation with cross diffusion, 2023. 1

[17] Neil Nie, Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. Structure from action: Learning interactions for articulated object 3d structure discovery, 2023. 1

[18] 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, 2024. 2

[19] Tianhe Ren, Yihao Chen, Qing Jiang, Zhaoyang Zeng, Yuda Xiong, Wenlong Liu, Zhengyu Ma, Junyi Shen, Yuan Gao, Xiaoke Jiang, Xingyu Chen, Zhuheng Song, Yuhong Zhang, Hongjie Huang, Han Gao, Shilong Liu, Hao Zhang, Feng Li, Kent Yu, and Lei Zhang. Dino-x: A unified vision model for open-world object detection and understanding, 2024. 1, 2, 4

[20] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model, 2023. 1

[21] 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. 1

[22] Aditya Vora, Sauradip Nag, and Hao Zhang. Articulate That Object Part (ATOP): 3D Part Articulation from Text and Motion Personalization, 2025. arXiv:2502.07278 [cs]. 2, 3

[23] 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, 2020. 2, 3, 7, 11

[24] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation, 2024. 1, 2, 4

[25] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models, 2024. 1

[26] Yunhan Yang, Yukun Huang, Yuan-Chen Guo, Liangjun Lu, Xiaoyang Wu, Edmund Y. Lam, Yan-Pei Cao, and Xihui Liu. Sampart3d: Segment any part in 3d objects, 2024. 1

Appendix

A. Reproduction of Articulate-Anything results

While the authors report a joint prediction success rate of 75%, with 15% of total predictions being failures due to link placement across the entire PartNet-Mobility data set, in my recreation we observed slightly lower success rates, with 59% joint prediction success and 30% of objects failing due to incorrect link placement. The observed difference in joint prediction performance can mainly be explained by the increased failure rate of link placement. One possible reason is the use of Gemini 2.5 Flash, for which we have observed a higher tendency to not follow the prompt instructions precisely, especially when generating the predicted Python function, which leads to the prediction becoming unable to be compiled to URDF and thus rendered.

We also break down the failure cases during joint placement in Figure A1. The joint placement failures (joint axis, type, origin, and limit) exhibit similar percentages to the results reported by Articulate-Anything, also in the order of most to least problematic with the joint axis making up the most errors and the joint limit the least, although the numbers we observed are slightly higher for each category.

Additionally, we highlight the lack of flexibility of mesh retrieval when applied to unseen objects from the ArtVIP dataset [6] in Figure A2. In the first case (left), a completely different object is retrieved (left side of image) since no similar object to the ground truth (right side) exists in the retrieval dataset and the VLM object selector determined both objects as close enough. In the second case (middle & right), a somewhat similar object is retrieved, but the it the ground truth is still very low. The proposed objects contained visually more similar objects, but none of them matched the configuration of the ground truth (doors and shelves).

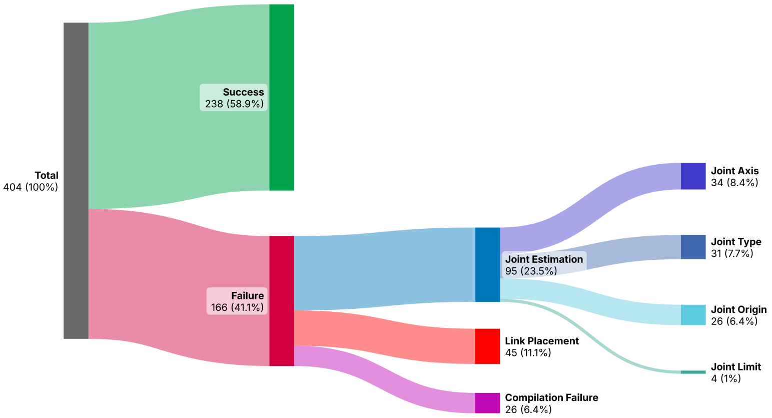


Figure A1. Joint Estimation failure breakdown for Articulate Anything method on the PartNet-Mobility dataset. Since ground truth mesh parts are used, only the actor-critic VLM is evaluated.

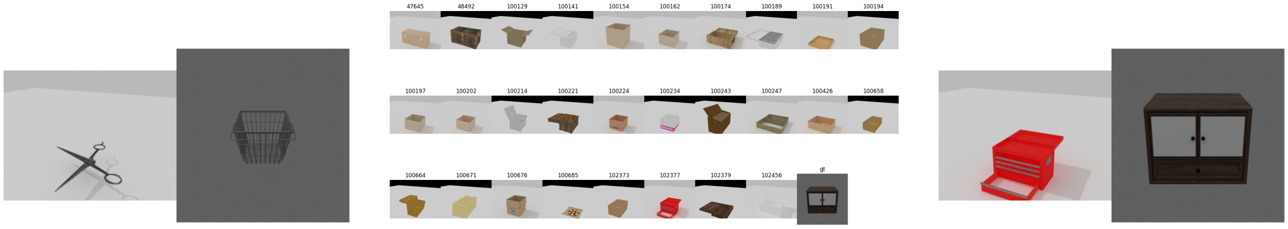


Figure A2. Failure of Mesh Retrieval to generalize to unseen objects.

B. 3D Part Decomposition

Here we provide additional details about the decomposition process of generated 3D assets. In Figure A3, we highlight the main parts of our decomposition method: Given an input image or video demonstration of the original object, use TRELLIS to generate an object mesh and detect movable parts of the object via a VLM. Then, render the generated mesh from multiple views in Blender. For each view, use DINO-X for zero-shot object detection and SAM for segmentation given the predicted bounding boxes. Masks are projected into 3D, merged, and overlapping points are removed. Finally, the mesh vertices are assigned to object part labels based on proximity to labeled points. After assigning mesh faces to labels by majority of vertices, the generated mesh is completely decomposed into parts.

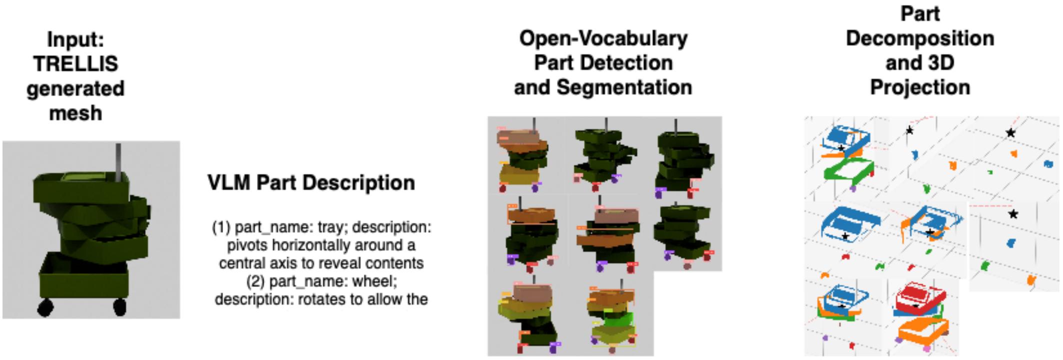
Figure A4 visualizes an important step for merging multiple 2D masks into 3D: Eliminating overlap between separate masks. Since segmented pixels are projected into 3D and used to separate the object mesh by performing nearest-neighbour matching with the mesh vertices, points with different labels overlapping with each other causes artifacts in the final mesh segmentation.

We also highlight some common failure cases for open-vocabulary object part detection in Figure A5. In the first image, segmentation of a scissor failed since it detected the blade and both handles as individual parts whereas it should have instead separated both scissor arms which move against each other. Next, the entire object was labeled as a drawer which semantically is not incorrect, however, the clearly visible doors were not detected. In the following case, some fine-grained parts like caster wheels and the handle are detected, but instead of segmenting the actual doors, the whole object is detected wrongly as a cabinet door. Finally, we show an example of successful segmentation. In this case, (almost) all movable parts were detected in the given view.

In Figure A6, we visualize the segmentation process in more detail, starting from input image of simulated assets of the RLBench dataset [4], which are used to generate 3D assets. These are rendered from multiple views and segmented in each of them, as shown in the next step. Next, we project the 2D masks into 3D segmented point clouds, and finally, load the mesh and partition it vertex-wise according to point label correspondence.

C. Additional Results for ARTIST on simulated objects

In this section, we provide further evaluation of our method against articulated object datasets. We generated a large number of assets given a single render of PartNet-Mobility [23] objects and provide results for the quality of generated meshes in Table A1. Figure A8 breaks down failures of ARTIST on recreating and articulating objects from PartNet-Mobility.


Input: TRELLIS generated mesh
VLM Part Description
Open-Vocabulary Part Detection and Segmentation
Part Decomposition and 3D Projection

(1) part_name: tray; description: pivots horizontally around a central axis to reveal contents (2) part_name: wheel; description: rotates to allow the

Figure A3. 3D Mesh Decomposition Process.

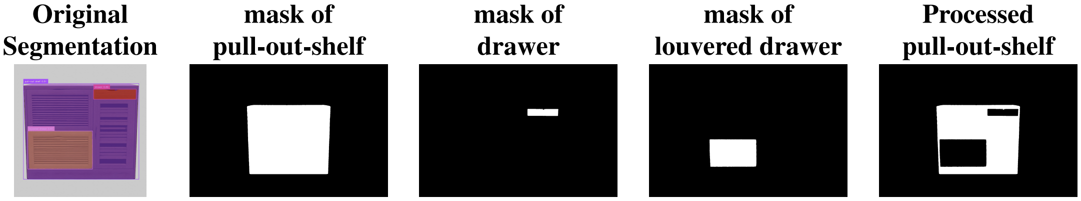


Figure A4. Process of cleaning overlapping masks to avoid 3D mesh segmentation problems.

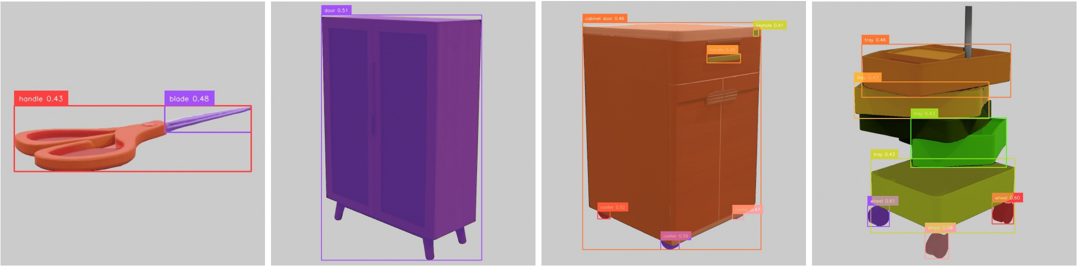


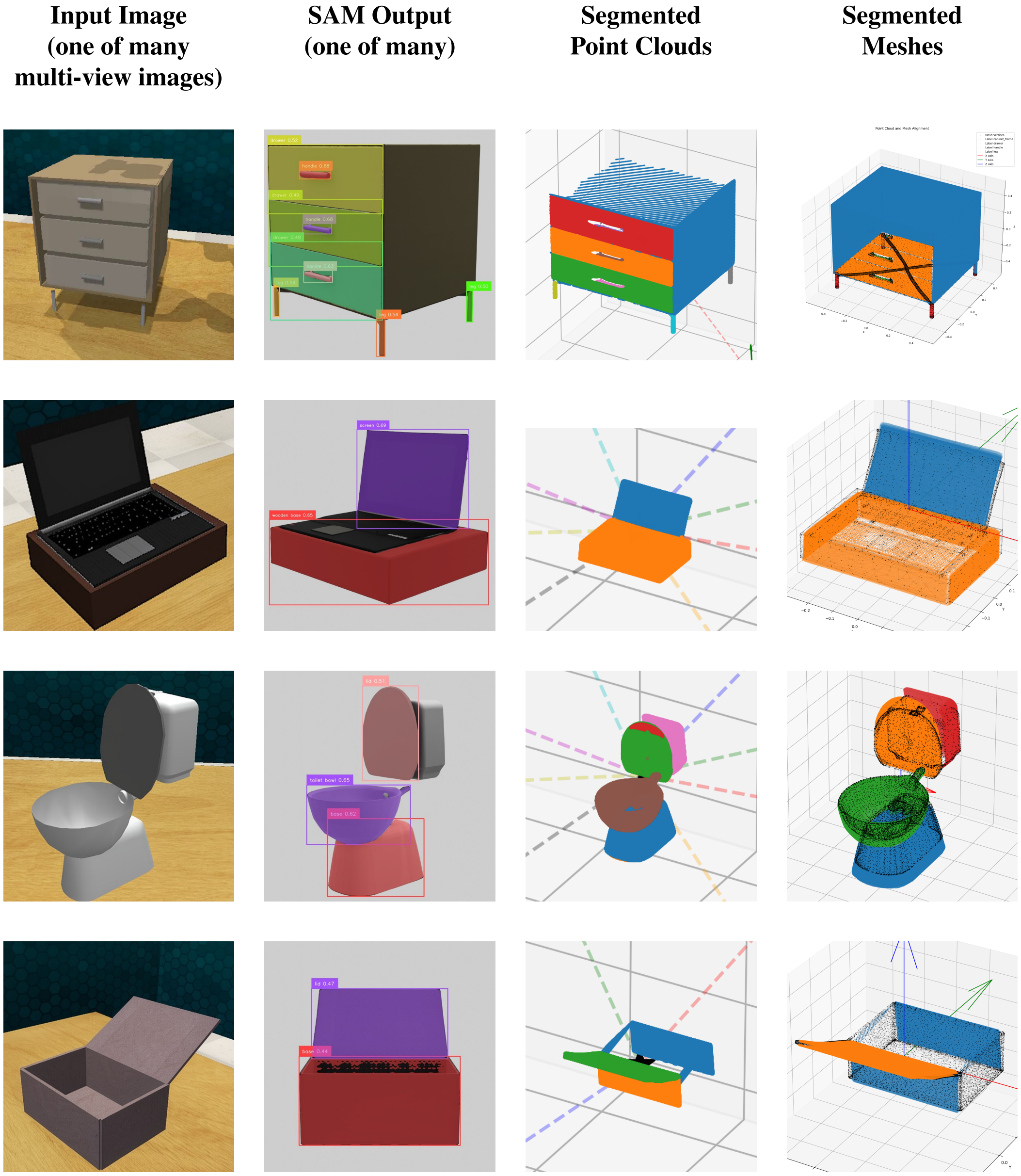
Figure A5. Failure cases of our open-vocabulary 2D part segmentation.

Table A1. Mesh quality evaluation using TRELLIS 3D generation on PartNet-Mobility assets using Chamfer Distance (lower is better).

Category	Mean ↓
Household Items	0.32 (n=17)
• Tools	0.12 (n=7)
• Eyewear	0.31 (n=2)
• Containers	0.46 (n=5)
• Other	0.38 (n=3)
Small Appliances	0.22 (n=11)
• Kitchen	0.22 (n=4)
• Electronics	0.23 (n=7)
Major Appliances	0.33 (n=13)
• Kitchen	0.32 (n=8)
• Bathroom	0.23 (n=3)
• Other	0.62 (n=2)
Furniture	0.39 (n=18)
• Seating	0.36 (n=7)
• Tables	0.42 (n=4)
• Storage	0.39 (n=6)
• Other	0.43 (n=1)
Other Items	0.37 (n=24)
• Fixtures	0.33 (n=9)
• Electronics	0.39 (n=13)
• Misc	0.44 (n=2)
Overall Average	0.33 (n=83)

D. Implementation Details

Here we provide the VLM prompt used for detecting movable parts given the input demonstration video, along with an example response of segmentation targets in Figures A9 and A10, respectively.



Input Image (one of many multi-view images)

SAM Output (one of many)

Segmented Point Clouds

Segmented Meshes

Figure A6. Visualization of the segmentation and 3D reconstruction pipeline: input images (column 1), SAM outputs for multi-view rendering of the object (column 2), segmented point clouds (column 3), and segmented meshes (column 4).

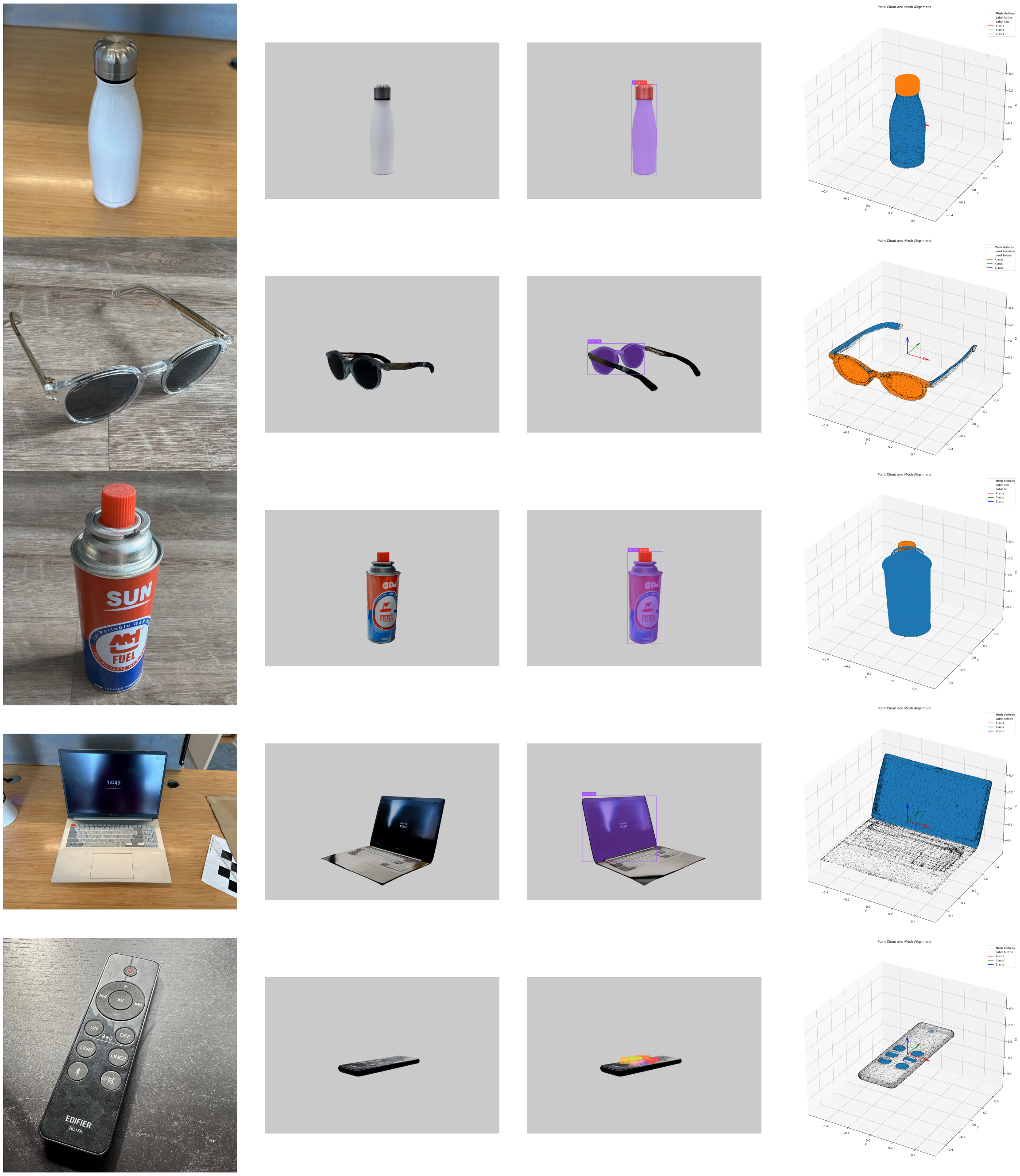


Figure A7. Digital Twin creation and decomposition into 3D parts for real-world objects. **Left:** One of the input images. **Left middle:** Rendered view of generated asset. **Right middle:** 2D Segmentation mask of rendered views. **Right:** Fully decomposed 3D point clouds.

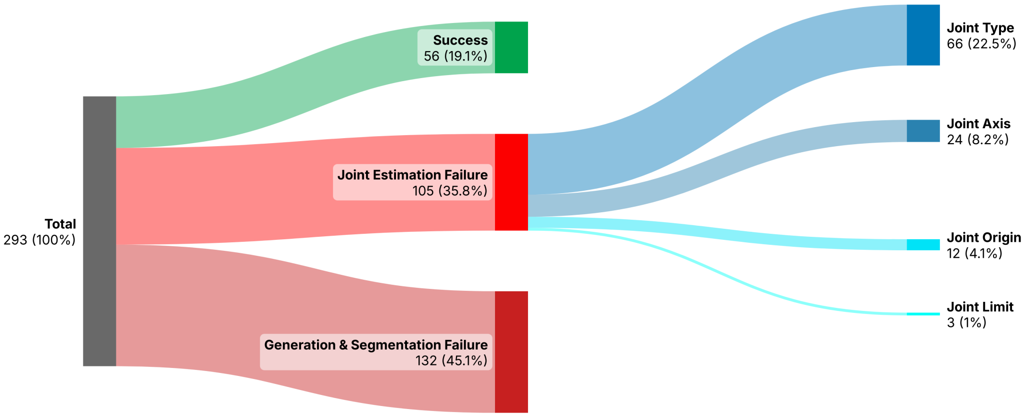


Figure A8. Joint estimation failure reasons for ARTIST on PartNet-Mobility [23] dataset.

Prompt

You have a good understanding of the structure of articulated objects. Your job is to assist the user to analyze the structure of an object. Specifically, the user will give you a video of an articulated object, and your task is to recognize the main parts of that object.

Output format

(1) part_name: name of the part; description: a brief description about the part, and how it moves
(2) part_name: name of the part; description: a brief description about the part, and how it moves

Remember

(1) Do not answer anything not asked.
(2) Your answer should be purely based on the input video, do not imagine anything.
(3) If there are multiple parts with the same semantic, just add one part to the list. For example, if there are four wheels, just add one part whose name is wheel.
(4) Your answer has to be based on the object being manipulated. If there is a robotic or a human arm interacting with the object, ignore it and just describe the object.

Figure A9. Prompt given to VLM to detect articulated parts given the input video of the target object.

(1) part_name: screen; description: The display part of the laptop, which is connected to the base by a hinge and can be opened or closed.
(2) part_name: key; description: One of the buttons on the keyboard that can be pressed down to input characters or commands.
(3) part_name: trackpad; description: A touch-sensitive surface used for controlling the cursor, located below the keyboard.
(4) part_name: power button; description: A button used to turn the laptop on or off, which can be pressed.

Figure A10. Example object part descriptions for a laptop, generated by a VLM prompted with the demonstration input video.