ARTMODELING: AN INSTANCE-MATCHED DATASET AND BENCHMARK FOR ARTICULATED-OBJECT PERCEPTION AND DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Simulation assets are used to evaluate articulated-object perception and to model interaction dynamics, but performance measured in simulation does not establish accuracy on physical objects. Controlled comparison requires real observations, detailed digital assets, and recorded motion linked to the same physical instances. We introduce ArtModeling, an instance-matched dataset and benchmark pairing 497 physical objects across 43 categories with articulated digital assets, within a collection of 1,500 simulation-ready models. Each paired instance combines calibrated multi-state RGB-D observations, camera poses, part-level masks, an articulated digital asset, and recorded interaction motion. The assets are manually modeled from independent scans and physical measurements, capturing detailed part geometry and occluded structures. Registration to the metric frame of the real observations enables evaluation on real and rendered inputs with matched viewpoints and joint states against a shared geometric reference. Released motion recordings support physical-parameter calibration and the reassessment of alternative parameter sets against the same real motion, while SolverAdapt supports refinement in the target simulator. The benchmark evaluates articulation estimation from real observations, performance differences under matched real and rendered inputs, and physical consistency on held-out interactions. Among the evaluated reconstruction methods, geometric rankings remain stable across input domains while joint-estimation rankings change, with larger discrepancies for methods using RGB-D inputs. Motion-based calibration improves trajectory agreement on held-out interactions. ArtModeling provides a reusable basis for assessing how findings in simulation extend to real objects, enabling controlled evaluation of articulated perception and dynamics against shared physical references.

1 INTRODUCTION

Articulated digital assets provide rendered observations and reference annotations for perception benchmarking (Xiang et al., 2020; Liu et al., 2023), and serve as interactive models for digital-twin simulation and robot training (Weng et al., 2024; Jin et al., 2026). These applications assume that method rankings from synthetic evaluation are preserved on real observations and that simulated dynamics reproduce the behavior of the corresponding physical objects (Jin et al., 2026; Iliash et al., 2026). However, real RGB-D observations contain sensor noise, missing depth values, and appearance variations that rendering does not fully reproduce. These discrepancies can affect methods differently and alter their relative performance. When real and synthetic evaluations use different objects and reference annotations (Liu et al., 2023; Weng et al., 2024), the effects of input differences cannot be isolated from differences in geometry, articulation, and evaluation references.

Recent datasets have expanded the coverage of articulated objects, but real observations paired with simulation-ready reference models remain limited. PartNet-Mobility, Shape2Motion, and Artiverse provide articulated models without paired real captures, preventing direct comparison of synthetic and sensed inputs for the same objects (Xiang et al., 2020; Wang et al., 2019; Iliash et al., 2026). RBO, AKB-48, and MultiScan provide real observations or object models with articulation annotations (Martín-Martín et al., 2019; Liu et al., 2022; Mao et al., 2022), but incomplete surface coverage

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

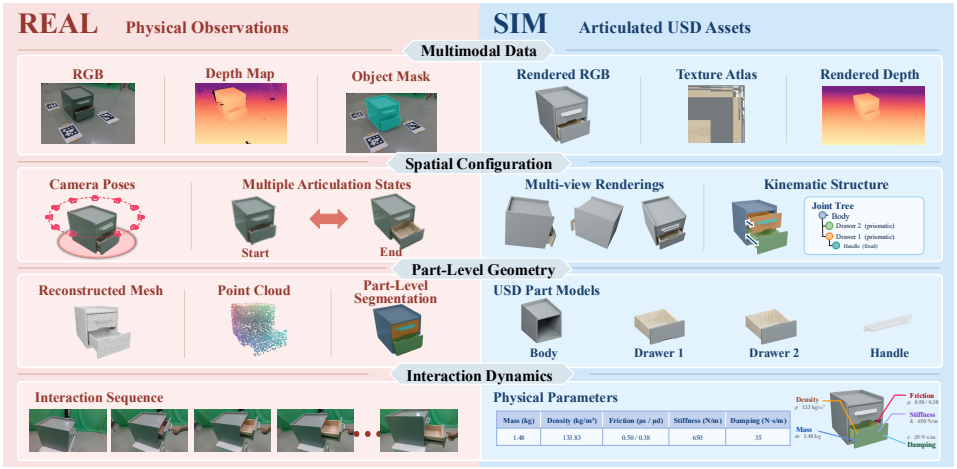


Figure 1: **Overview of ArtModeling.** ArtModeling establishes instance-level correspondence between **real observations** and **detailed articulated USD assets** of the same physical objects. The assets provide independently modeled part geometry, including fine details and occluded structures, together with textures, kinematic annotations, and physical parameters. Calibrated camera poses and matched joint states support controlled comparisons between real and rendered observations, while recorded interactions provide reusable references for calibrating and reassessing asset dynamics.

and reconstruction artifacts limit the use of scan-derived geometry as high-accuracy references for simulation and evaluation. When the reference is reconstructed from the evaluated capture, it inherits the same sensing errors. ArtVIP constructs detailed digital assets from real-world objects but does not provide paired multimodal observations of those objects (Jin et al., 2026). Motion measurements from the corresponding real object are needed to evaluate and recalibrate physical parameters such as joint stiffness and damping. An instance-matched collection linking detailed reference assets to real observations and motion for evaluating perception and dynamics remains missing.

In this paper, we introduce **ArtModeling**, an articulated-object dataset grounded in real-world instances, as illustrated in Fig. 1. The collection contains **497** real objects across **43** categories, each paired with a detailed, simulation-ready USD model. An additional **1,003** assets without paired captures bring the asset collection to **1,500** models. Each paired object comes with calibrated multi-view RGB-D observations captured at start and end articulation states, supplementary views from a different placement, object masks, and an interaction sequence. Professional artists model the real objects to create reference geometry independent of the evaluated captures and their sensing errors. Manual modeling captures fine-grained part geometry, including structures occluded in the captures, such as drawer interiors and hinge regions. Part-level physical parameters are initialized by a multimodal foundation model under measured mass constraints, verified by annotators, and refined using recorded interaction trajectories. These trajectories are also released to support parameter evaluation and retuning across simulation environments. Detailed product information and purchase links enable researchers to acquire the same product models for further experiments.

We further benchmark articulation estimation from real observations and its comparison with pose-matched renderings on 153 paired objects, together with a repeated-interaction subset for physical consistency. The evaluations cover per-object optimization, zero-shot prediction with synthetic-trained models, and the comparison of initial physical annotations with motion-refined parameters. Replacing real inputs with renderings leaves the geometric comparison between the optimization methods largely intact while changing it for joint estimation, and the change is larger for the methods that consume depth. Two synthetic-trained part pose estimators likewise exchange their relative accuracy between the two domains. Motion-based calibration improves agreement on a separate interaction of the same joint. These findings motivate articulation estimation that is robust to real sensing, and motion-based validation of physical parameters.

Our contributions are summarized as follows.

- We introduce ArtModeling, an instance-matched dataset whose articulated digital assets closely reproduce the geometry and kinematic structure of their physical counterparts, including fine part details and occluded structures. Paired real observations and interaction recordings connect these detailed assets to shared physical references for evaluating perception and dynamics.
- We develop a construction and annotation pipeline that combines independent geometric modeling, human-reviewed physical-property initialization, and calibration against real motion, together with SolverAdapt for subsequent parameter refinement.
- We establish a benchmark for articulation estimation, paired real–synthetic comparison, and physical consistency, revealing performance gaps between real and synthetic observations with matched viewpoints, joint states, and evaluation references.

2 Related Work

Articulated Object Datasets and Assets. Articulated-object datasets differ in how they establish geometric references and connect them to real observations. **Articulated Object Datasets and Assets.** Real-world collections such as RBO, AKB-48, and MultiScan pair observations with scanned or reconstructed geometry (Martín-Martín et al., 2019; Liu et al., 2022; Mao et al., 2022), whose incomplete surface coverage and reconstruction artifacts limit its accuracy as a simulation reference, and a reference reconstructed from the evaluated capture carries its sensing errors into both the inputs and the reference. Synthetic collections such as Shape2Motion, PartNet-Mobility, and Artiverse provide articulated assets with part geometry and kinematic annotations, but without paired real observations of the modeled objects (Wang et al., 2019; Xiang et al., 2020; Iliash et al., 2026). Combining the two sources, as in OPD and GAPartNet, does not yield matched real and rendered views of the same physical instance (Jiang et al., 2022a; Geng et al., 2023). ArtVIP constructs detailed digital assets from real objects and releases the assets rather than the captures behind them (Jin et al., 2026). ArtModeling instead models each asset from scans and measurements taken independently of the evaluated captures and registers it to the metric frame of those captures, so that real and rendered observations of the same instance can be compared against a shared reference.

Articulated Reconstruction and Motion Estimation. Existing work primarily focuses on recovering part geometry, part poses, and joint parameters from visual observations. Per-object optimization methods fit part geometry and joint parameters to observations of a single object, using neural implicit fields on multi-view RGB in PARIS (Liu et al., 2023), RGB-D with cross-state correspondences in DigitalTwinArt (Weng et al., 2024), 3D Gaussian representations at two joint states in ArtGS and SplArt (Liu et al., 2025b; Lin et al., 2025), and monocular interaction video in VideoArtGS (Liu et al., 2025a). Learned methods instead predict part structure, pose, or joint parameters from a single observation, through canonical part representations (Li et al., 2020; Weng et al., 2021), point clouds in GAPartNet (Geng et al., 2023), and RGB-D features in CAP-Net (Huang et al., 2025), with further variants operating on before-and-after observations, stereo images, code generation, and monocular video (Jiang et al., 2022b; Heppert et al., 2023; Zhao et al., 2025; Artykov et al., 2025). Both families are compared quantitatively on synthetic benchmarks, and their real-world evaluations use object instances different from those evaluated synthetically. We evaluate representative methods of both families on real captures and on matched renderings, holding object identity, viewpoints, joint states, and evaluation references fixed.

Physical Property Estimation and Calibration. Visual-semantic approaches learn physical concepts from appearance or associate language-based material estimates with 3D features (Gao et al., 2024; Zhai et al., 2024), and asset collections apply the same priors under human review, with PhysX-3D verifying VLM proposals (Cao et al., 2025) and Artiverse deriving part masses from geometric volumes and material-dependent densities (Iliash et al., 2026). Motion-based approaches instead fit simulated behavior to video evidence, estimating elastic material fields from generated videos in PhysDreamer (Zhang et al., 2024), spring-mass models from recorded RGB-D interactions in PhysTwin (Jiang et al., 2025), and articulated dynamics from the observed motion of physical counterparts in ArtVIP (Jin et al., 2026). Priors alone do not determine the dynamic response of a specific object, and motion-fitted parameters hold for the excitation and simulator configuration used during calibration. We combine human-reviewed VLM annotation under measured mass constraints with calibration against recorded joint trajectories, and release those trajectories with SolverAdapt so that a parameter set can be reassessed against the same motion references.

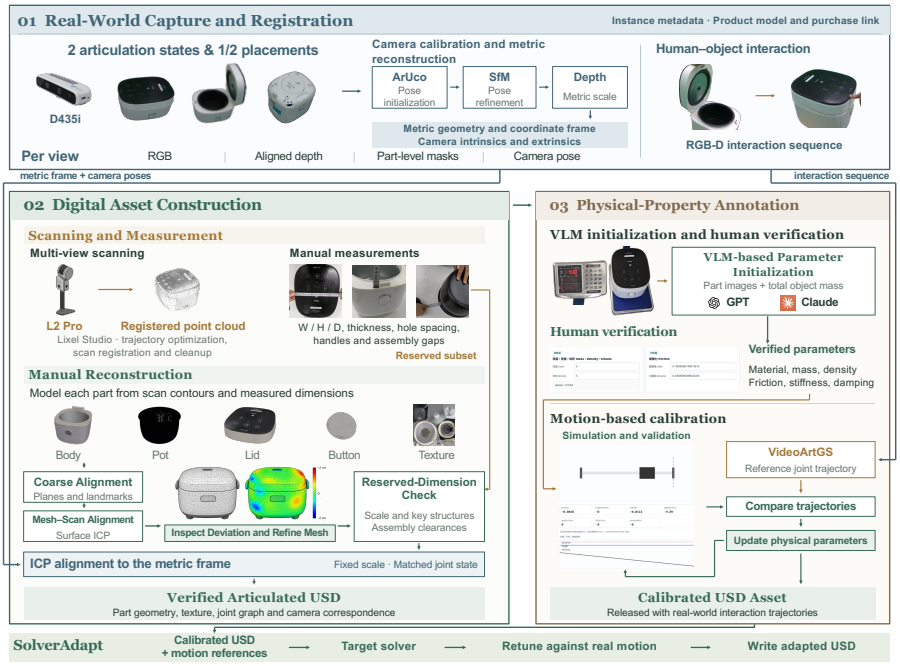


Figure 2: **Overview of the ArtModeling construction pipeline.** (1) Real-world capture and registration. (2) Independent scanning, manual asset modeling, and geometric validation. (3) Human-reviewed VLM annotation and motion-based physical calibration. SolverAdapt supports subsequent parameter refinement.

3 ArtModeling Construction and Analysis

We construct each paired instance through real-world capture and registration, digital asset construction, and physical-property annotation with motion-based calibration, as shown in Fig. 2. The asset geometry is modeled from separate 3D scans and direct measurements, and the asset is registered to the RGB-D observations in a common metric coordinate frame. This instance-level correspondence supports perception evaluation with independently modeled geometry and evaluation of physical parameters against motion recorded from the same object.

3.1 Real-World Capture and Registration

We acquire RGB-D observations of each physical object at two configurations of a selected joint and in two placements using an Intel RealSense D435i (Keselman et al., 2017), with all other joints held fixed. The second placement exposes surfaces occluded by ground contact, and it is omitted for the joint configurations an object cannot hold in that orientation. Each capture setting is covered by approximately 256 views distributed over a hemisphere centered on the object.

ArUco markers (Garrido-Jurado et al., 2014) initialize camera poses, which are refined through multi-view structure-from-motion (Schonberger & Frahm, 2016), while depth measurements establish metric scale. We reconstruct each placement separately and register the resulting geometry and camera poses to a common object coordinate frame w.

Registration of the paired asset to this frame enables rendering from the observed viewpoints. Let $K^{(v)}$ denote the camera intrinsics for view v, $T_{wc}^{(v)}$ its camera-to-frame transform, and T_{wa} the transform from the asset frame to w. The corresponding synthetic observation is

$$I_{\text{syn}}^{(v)} = \Pi\Big(\mathcal{A}\big(q^{(v)}\big); K^{(v)}, \big(T_{wc}^{(v)}\big)^{-1} T_{wa}\Big), \tag{1}$$

where Π denotes the renderer and $\mathcal{A}(q^{(v)})$ denotes the asset in its local coordinate frame at the joint configuration matched to view v. The real and synthetic observations correspond to the same

Table 1: **Scale and data coverage of articulated-object datasets.** Statistics are taken from papers or public releases. Paired instances count distinct physical objects with released real sensor observations and a corresponding articulated digital model. Artist-modeled denotes part-level geometry constructed by professional 3D modelers. Physical annotation excludes generic simulation settings.

Dataset	Scale			Real data		Digital assets		
	Objects / models	Cats.	Paired instances	RGB-D	Real interaction	Geometry source	Physical annotation	Motion calibration
RBO (Martín-Martín et al., 2019)	14	–	14	✓	✓	Mixed	–	–
Shape2Motion (Wang et al., 2019)	2,440	45	–	–	–	CAD	–	–
PartNet-Mobility (Xiang et al., 2020)	2,346	46	–	–	–	CAD	–	–
AKB-48 (Liu et al., 2022)	2,037	48	–	✓	–	Scan/CAD	✓	–
MultiScan (Mao et al., 2022)	982	–	–	✓	–	Scan	–	–
OPDReal (Jiang et al., 2022a)	284	8	284	✓	–	Scan	–	–
OPDSynth (Jiang et al., 2022a)	683	11	–	–	–	CAD	–	–
GAPartNet (Geng et al., 2023)	1,166	27	–	–	–	CAD/scan	–	–
ArtVIP (Jin et al., 2026)	992	37	–	–	–	Artist-modeled	✓	✓
Artiverse (Iliash et al., 2026)	5,402	88	–	–	–	CAD	✓	–
ArtModeling	1,500	43	497	✓	✓	Artist-modeled	✓	✓

object instance with matched camera intrinsics, viewpoints, and joint states, supporting the paired evaluation in Sec. 4.3.

We also record an RGB-D interaction sequence as an operator manually moves the selected part between the two captured joint configurations. The extracted joint trajectory provides a reference for physical-parameter calibration. Each object is associated with its product model and purchase link, enabling acquisition of additional objects of the same product model for further experiments. Capture protocols and coordinate conventions are detailed in Appendix A.



Figure 3: **Examples of real articulated objects and their instance-matched digital assets in ArtModeling.** (a) Real-world objects collected from daily environments. (b) Corresponding simulation-ready USD assets reconstructed from the same physical instances.

3.2 Digital Asset Construction

We manually construct each digital counterpart as an articulated Universal Scene Description (USD) asset from dedicated 3D scans and direct measurements of the physical object, following the pipeline in Fig. 2. Geometry is captured using a Lixel L2 Pro handheld LiDAR scanner and processed in Lixel Studio for scan registration (XGRIDS), background removal, and noise filtering. Annotators record overall dimensions, panel thickness, hole spacing, handle dimensions, and assembly gaps, reserving a subset of these measurements for final geometric validation.

Individual part meshes are constructed from scan contours and measured dimensions. We refine their geometry using mesh-to-scan surface deviations and dimensional discrepancies, then validate the resulting meshes against the reserved measurements for overall scale, key structural dimensions,

and assembly clearances. The meshes are organized into semantic parts and rigid bodies within a parent–child articulation hierarchy. Joint definitions specify the type, axis, origin, orientation, and motion limits of each connection, and material and texture assignment follows the appearance captured in Sec. 3.1.

The completed asset is registered to the RGB-D reconstruction in the metric frame defined in Sec. 3.1. We estimate its rigid pose using ICP (Besl & McKay, 1992), with scale and the matched joint configuration held fixed. The resulting transform T_{wa} establishes the camera correspondence in Eq. 1, preserving the independently modeled geometry. Fig. 3 presents representative physical objects and their paired assets, with additional paired real–digital examples provided in Appendix D.4. Detailed modeling procedures, USD conventions, and validation protocols are provided in Appendix B.

3.3 Physical-Property Annotation

We initialize part-level physical properties with two vision-language models, GPT-5.4 and Claude Opus 4.6 (OpenAI; Anthropic). Both models receive the same part images, part list, and measured mass of the assembled object. Each returns material assignments and estimates of part mass, density, static and dynamic friction coefficients, and joint-drive stiffness and damping. These predictions provide candidate annotations for human review. Annotators resolve disagreements in material assignments and inspect the numerical estimates for physical plausibility. We take the midpoint of the two numeric predictions as the starting value and the interval they span, widened to the range admitted by the adjudicated material, as the search bounds for the calibration below. Part-mass allocations are checked against the measured total mass and adjusted where necessary, and implausible entries are corrected before calibration.

The interaction sequence provides the motion reference for calibrating the asset's contact and joint-drive parameters. We extract the joint's trajectory q_t^{real} using VideoArtGS (Liu et al., 2025a) and jointly optimize the dynamics parameters $\boldsymbol{\theta}$ and an effective excitation sequence $\mathbf{u}$ by minimizing

$$\min_{\boldsymbol{\theta},\mathbf{u}} \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(q_t^{\mathrm{sim}}(\boldsymbol{\theta},\mathbf{u};s)-q_t^{\mathrm{real}}\right)^2}+\mathcal{R}(\boldsymbol{\theta},\mathbf{u}), \tag{2}$$

where T is the number of trajectory samples and s denotes the fixed simulator configuration. The optimized parameters $\boldsymbol{\theta}$ comprise friction coefficients and joint-drive stiffness and damping, while geometry, mass, density, and joint definitions remain fixed. The effective excitation is inferred from the observed motion under the simulation model and represented by a constant force or torque over each interval between successive samples. The regularizer $\mathcal{R}$ constrains the parameter and excitation search, including penalties on excitation magnitude and temporal variation.

Our sensitivity analysis shows that perturbing joint-drive parameters can substantially alter simulated trajectories even when the applied force or torque remains unchanged (Appendix E.2). This sensitivity motivates evaluating and refining parameter settings through their motion response. We provide SolverAdapt for interactive inspection and adjustment of the asset's physical configuration using the recorded trajectories as references. The interface exposes part properties and joint-drive settings, with motion previews and response curves guiding adjustment before validation in the target simulator. Edited parameters are saved back to USD. The calibrated assets are released together with the interaction trajectories to support subsequent evaluation and refinement. Appendix C details initialization, human review, parameter bounds, excitation representation, optimization settings, and the SolverAdapt interface.

3.4 Dataset Comparison and Coverage

Tab. 1 highlights the distinction between asset scale and correspondence to observed physical objects. PartNet-Mobility and Artiverse provide large asset collections without paired real captures (Xiang et al., 2020; Iliash et al., 2026), while ArtVIP constructs detailed models of real objects but does not provide their paired multimodal observations (Jin et al., 2026). Compared with RBO and OPDReal (Martín-Martín et al., 2019; Jiang et al., 2022a), ArtModeling provides a larger paired subset of 497 physical instances, each linked one-to-one to an articulated digital counterpart and ac-

companied by real interaction recordings. These pairs belong to a collection of 1,500 assets spanning 43 categories.

The one-to-one correspondence enables real–synthetic comparisons with object identity, viewpoint, and joint state held fixed. The paired assets add itionally provide fine part geometry and occluded structures modeled independently of the evaluated RGB-D captures, with dimensional accuracy checked against reserved measurements. Their physical annotations are accompanied by motion from the same objects, allowing parameter consistency to be reassessed and refined after release. ArtModeling thus connects controlled perception evaluation with motion-based physical evaluation through the same physical counterparts. Detailed statistics and analyses are provided in Appendix D.

4 EXPERIMENTS

4.1 BENCHMARK SETUP

Evaluation sets. We conduct three complementary evaluations. **Real-input accuracy** compares predicted geometry and joint parameters against the paired reference assets, measuring reconstruction and joint-estimation errors. The optimization benchmark uses 153 of the 497 paired objects for which all four methods produce valid outputs, comprising 105 revolute and 48 prismatic objects. GAPartNet and CAP-Net are evaluated on 174 objects and 385 part instances from categories supported by both models. **Real–synthetic comparison** evaluates error gaps and changes in method rankings between real captures and renderings, with object instances, viewpoints, articulation states, and evaluation references held fixed. **Physical consistency** compares parameter configurations under the same independently obtained excitation, measuring discrepancies between simulated and held-out real motion. This evaluation uses 24 prismatic and 12 revolute objects, each with two repetitions of the same joint motion in the same direction. **Evaluation objects are excluded from dataset-level training and model selection**. Per-object fitting uses the designated observations or calibration interaction.

Paired observations. We render each paired asset at the camera poses, resolution, and joint states of its real captures, with no added depth noise. The asset is rigidly aligned without rescaling to the metric frame of the reconstructed mesh and supplies the reference for both input domains. Instance identity, viewpoints, articulation states, and references are therefore shared, while the inputs differ in sensing, appearance, and the residual modeling error assessed in Appendix D.3.

Methods. PARIS (Liu et al., 2023), DigitalTwinArt (Weng et al., 2024), ArtGS (Liu et al., 2025b), and SplArt (Lin et al., 2025) receive the same views at two joint states, converted to their input formats. PARIS uses RGB; the other three use RGB-D. For learned part pose estimation, we use the released synthetic-trained GAPartNet (Geng et al., 2023) and CAP-Net (Huang et al., 2025) checkpoints without adaptation, with depth point clouds and masked RGB-D crops, respectively. Physical evaluation compares initialized, refined, and intermediate parameter configurations under a shared excitation, as described in Sec. 4.4.

Metrics. Geometry is measured by Chamfer distance for the static part (CD-s), movable part (CD-m), and whole object (CD-w), scaled by 10^3; Mean CD averages these three errors. Joint metrics measure axis-angle error, revolute axis-line distance, and motion-magnitude error. Part pose metrics comprise rotation, translation, and scale errors, 3D bounding-box mIoU, and accuracies A_5 and A_{10} at the joint rotation/translation thresholds 5°/5 cm and 10°/10 cm. Physical consistency is measured by mean per-frame absolute trajectory residual. To compare domains, we report the relative reduction in error under synthetic inputs,

$$\Delta_E = \frac{\bar{E}_{\text{real}} - \bar{E}_{\text{syn}}}{\bar{E}_{\text{real}}} \times 100\%, \tag{3}$$

where both means use the same objects within a joint type. Positive values indicate lower error on synthetic inputs.

Table 2: **Reconstruction errors on real inputs.** CD is scaled by 10^3. Axis and revolute motion errors are in degrees. Pivot and prismatic motion errors are in meters.

Joint	Method	Input	Geometry			Kinematics		
			CD-s ↓	CD-m ↓	CD-w ↓	Axis ↓	Pivot ↓	Motion ↓
Revolute	PARIS	RGB	7.05	15.14	5.72	21.54	**0.07**	17.91
	DigitalTwinArt	RGB-D	4.05	15.69	2.49	35.04	0.38	28.93
	ArtGS	RGB-D	**0.61**	**5.27**	**0.61**	24.33	0.15	21.10
	SplArt	RGB-D	10.95	22.69	11.35	**9.35**	0.17	**10.87**
Prismatic	PARIS	RGB	9.34	35.88	8.86	11.49	—	0.31
	DigitalTwinArt	RGB-D	1.27	12.46	1.47	11.62	—	**0.08**
	ArtGS	RGB-D	**0.84**	**5.56**	**1.16**	14.20	—	0.11
	SplArt	RGB-D	18.50	12.56	16.89	**7.99**	—	0.95

Table 3: **Relative error reductions on synthetic inputs (%).**

Input	Method	Revolute				Prismatic			Joint mean
		Mean CD	Axis	Pivot	Motion	Mean CD	Axis	Motion	
RGB	PARIS	17.05	4.20	5.82	−0.02	50.63	22.21	5.15	**7.47**
RGB-D	DigitalTwinArt	80.37	69.92	52.75	64.29	77.62	15.76	65.78	**53.70**
	ArtGS	41.13	34.43	75.73	24.13	62.03	49.63	85.87	**53.96**
	SplArt	59.33	60.47	18.19	68.60	45.76	38.63	82.32	**53.64**
RGB-D mean		60.28	54.94	48.89	52.34	61.80	34.67	77.99	**53.77**

4.2 Articulation Estimation from Real Observations

Tab. 2 reports geometric and kinematic errors on real observations, averaged over objects within each joint group. ArtGS (Liu et al., 2025b) achieves the lowest geometric errors for both joint types, yet has the second-largest revolute axis error and the largest prismatic axis error. SplArt (Lin et al., 2025) achieves the lowest axis errors for both joint types, but has the largest static-part and whole-object geometric errors in both groups. Movable-part CD also exceeds twice the static-part CD in seven of the eight method–joint-type groups. These results show that geometric accuracy alone is insufficient to assess kinematic accuracy and motivate examining whether method rankings remain consistent between real and synthetic inputs.

4.3 Real versus Synthetic Inputs

4.3.1 Per-Object Optimization Methods

Cross-domain ranking changes are more pronounced in joint estimation than in geometry. For Mean CD, ArtGS (Liu et al., 2025b) and DigitalTwinArt (Weng et al., 2024) remain first and second for both joint types. In contrast, DigitalTwinArt drops from second on synthetic inputs to fourth on real inputs for both revolute axis and motion errors. Synthetic inputs yield lower errors in 27 of the 28 method–metric comparisons, as shown in Tab. 3. Averaged equally over the five joint-estimation metrics, the error reduction is 7.47% for RGB-only PARIS (Liu et al., 2023) and 53.77% across the three RGB-D methods, DigitalTwinArt, ArtGS, and SplArt (Lin et al., 2025). The larger gap among RGB-D methods motivates further investigation of sensitivity to sensed depth, although differences between methods prevent attributing this gap to depth alone.

4.3.2 Models Trained on Synthetic Data

Under the paired-input protocol, GAPartNet (Geng et al., 2023) and CAP-Net (Huang et al., 2025) perform worse on real observations across all six metrics (Tab. 4). Their ordering reverses at the A_{10} threshold, with GAPartNet leading by 5.35 percentage points on synthetic inputs and CAP-Net leading by 7.96 percentage points on real inputs. Mean 3D bounding-box IoU decreases by 22.08 percentage points for GAPartNet and 8.30 percentage points for CAP-Net. Fig. 4 provides

qualitative comparisons on matched objects and views. These results extend the need for real-input evaluation from per-object optimization to synthetic-trained part-pose predictors.

Table 4: Zero-shot performance on synthetic and real inputs. mIoU and accuracies are percentages.

Input	Method	$\mathcal{R}_e \downarrow$	$T_e \downarrow$	$S_e \downarrow$	mIoU ↑	$A_5 \uparrow$	$A_{10} \uparrow$
Synthetic	GAPartNet	33.39	0.022	0.031	60.91	13.40	**34.93**
	CAP-Net	**19.84**	**0.016**	**0.023**	**65.39**	**22.91**	29.58
Real	GAPartNet	46.09	0.039	0.040	38.83	13.33	20.00
	CAP-Net	**23.79**	**0.024**	**0.029**	**57.09**	**16.69**	**27.96**

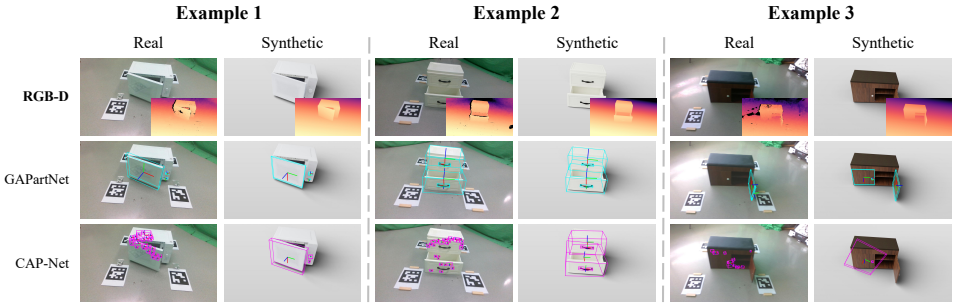


Figure 4: **Part pose predictions on paired inputs.** Each example places real and synthetic inputs side by side at the same viewpoint and joint state. The top row shows RGB with depth insets; the remaining rows show predicted 3D part boxes from GAPartNet and CAP-Net.

4.4 Consistency of Physical Parameters with Recorded Motion

Recorded interactions allow an asset's physical parameters to be evaluated through the motion they produce. For each 36 objects, we extract joint trajectories with VideoArtGS (Liu et al., 2025a), calibrate parameters on the first repetition, and evaluate them on the second. The evaluation excitation is obtained independently of the target trajectory and applied to four parameter configurations: foundation-model initialization, the refined parameters, and the element-wise mean and median of intermediate optimization iterates, excluding both endpoints.

Table 5: **Motion residuals on held-out interactions.**

	Prismatic ($n = 24$)		Revolute ($n = 12$)	
Setting	Residual (cm) ↓	Reduc. (%) ↑	Residual (rad) ↓	Reduc. (%) ↑
Initial	3.748	—	0.332	—
Iter. mean	2.223	40.7	0.290	12.8
Iter. median	2.008	46.4	0.211	36.6
Refined	**1.003**	**73.2**	**0.101**	**69.6**

Tab. 5 summarizes the trajectory residuals before and after calibration and for the intermediate parameter configurations. Calibration improves agreement with the second repetition for all 36 objects. Mean residuals decrease from 3.748 to 1.003 cm for prismatic joints and from 0.332 to 0.101 rad for revolute joints, reductions of 73.2% and 69.6%. The intermediate parameter summaries fall between initialization and refinement in both group means, showing that the parameter configuration affects trajectory agreement under the shared excitation. With excitation fixed, changes to joint-drive parameters can alter simulated trajectories. Appendix E.2 presents the sensitivity analysis.

5 Conclusion

We introduced ArtModeling, an instance-matched dataset and benchmark linking real observations, detailed articulated assets with independently modeled geometry, and recorded motion from the same physical objects. Controlled evaluations reveal changes in joint-estimation rankings between real and rendered inputs, while motion-based calibration improves agreement on held-out interactions. ArtModeling provides shared physical references for reproducible perception benchmarking and continued assessment and refinement of asset dynamics.

AI USE STATEMENT

We used large/vision language models in two roles in this work.

In writing, we used them to check grammar, correct wording, and improve the clarity of text drafted by the authors. All content was written and verified by the authors, and no section of the paper was generated by a model without author review.

In the dataset, we used GPT-5.4 and Claude Opus 4.6 to initialize part-level physical properties of the articulated assets, as described in Section 3.3. Both models received part images, the part list, and the measured mass of the assembled object, and returned material assignments together with estimates of part mass, density, friction coefficients, and joint-drive stiffness and damping. These outputs were treated as candidate annotations rather than final values. Annotators resolved disagreements in material assignment, checked the mass allocation against the measured object mass, and corrected implausible entries, and the dynamics parameters were subsequently calibrated against motion recorded from the corresponding physical object. The model outputs, the reviewed values, and the calibrated values are all released with the dataset.

Large language models were not used to generate research ideas, to design the benchmark or its evaluation protocol, to produce or modify any experimental result, or to write code that produced the reported numbers. The authors take responsibility for the full content of this paper, including the text, the claims, and the released artifacts.

REPRODUCIBILITY STATEMENT

We will publicly release the complete ArtModeling dataset, including the articulated USD assets, paired real observations, annotations, and interaction recordings. All assets correspond to commercially available products, and the release will include product model information and purchase links to support further experiments with physical objects of the same models. We will also open-source selected components of the data acquisition pipeline. Acquisition protocols, asset construction procedures, and physical annotation details are documented in Appendices A, B, and C, respectively. Together, these resources support reuse of the benchmark, reproduction of the data collection procedures, and further evaluation of perception and dynamics on corresponding physical objects.

REFERENCES

Anthropic. Claude opus. `https://www.anthropic.com/claude/opus`. Accessed: 2026-09-26.

Arslan Artykov, Tom Ravaud, Corentin Sautier, and Vincent Lepetit. sim2art: Accurate articulated object modeling from a single video using synthetic training data only. *arXiv preprint*, 2025.

Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pp. 586–606. Spie, 1992.

Ziang Cao, Zhaoxi Chen, Liang Pan, and Ziwei Liu. PhysX-3D: Physical-grounded 3D asset generation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 104070–104083, 2025. doi: 10.52202/085713-3136.

Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. Physically Grounded Vision-Language Models for Robotic Manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12462–12469, 2024. doi: 10.1109/ICRA57147.2024.10610090.

Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. GAPartNet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7081–7091, June 2023.

Nick Heppert, Muhammad Zubair Irshad, Sergey Zakharov, Katherine Liu, Rares Andrei Ambrus, Jeannette Bohg, Abhinav Valada, and Thomas Kollar. CARTO: Category and Joint Agnostic Reconstruction of ARTiculated Objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21201–21210, 2023. doi: 10.1109/CVPR52729.2023.02031.

Jingshun Huang, Haitao Lin, Tianyu Wang, Yanwei Fu, Xiangyang Xue, and Yi Zhu. CAP-Net: A unified network for 6D pose and size estimation of categorical articulated parts from a single RGB-D image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11654–11664, June 2025.

Denys Iliash, Jiayi Liu, Egor Fokin, Qirui Wu, Ali Mahdavi-Amiri, Manolis Savva, and Angel X. Chang. Artiverse: A diverse and physically grounded dataset for articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8932–8942, June 2026.

Hanxiao Jiang, Yongsen Mao, Manolis Savva, and Angel X. Chang. OPD: Single-view 3D openable part detection. In *Computer Vision – ECCV 2022*, pp. 410–426. Springer, 2022a. doi: 10.1007/978-3-031-19842-7_24.

Hanxiao Jiang, Hao-Yu Hsu, Kaifeng Zhang, Hsin-Ni Yu, Shenlong Wang, and Yunzhu Li. PhysTwin: Physics-Informed Reconstruction and Simulation of Deformable Objects from Videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7219–7230, 2025. doi: 10.1109/ICCV51701.2025.00678.

Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building Digital Twins of Articulated Objects From Interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5616–5626, 2022b. doi: 10.1109/CVPR52688.2022.00553.

Zhao Jin, Zhengping Che, Tao Li, 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. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=SqPLEZ66BO.

Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 1–10, 2017.

Xiaolong Li, He Wang, Li Yi, Leonidas J. Guibas, A. Lynn Abbott, and Shuran Song. Category-Level Articulated Object Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3706–3715, 2020. doi: 10.1109/CVPR42600.2020.00376.

Shengjie Lin, Jiading Fang, Muhammad Zubair Irshad, Vitor Campagnolo Guizilini, Rares Andrei Ambrus, Greg Shakhnarovich, and Matthew R. Walter. SplArt: Articulation estimation and part-level reconstruction with 3D gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8841–8851, October 2025.

Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. PARIS: Part-level reconstruction and motion analysis for articulated objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 352–363, October 2023. doi: 10.1109/ICCV51070.2023.00039.

Liu Liu, Wenqiang Xu, Haoyuan Fu, Sucheng Qian, Qiaojun Yu, Yang Han, and Cewu Lu. AKB-48: A real-world articulated object knowledge base. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14809–14818, June 2022. doi: 10.1109/CVPR52688.2022.01439.

Yu Liu, Baoxiong Jia, Ruijie Lu, Chuyue Gan, Huayu Chen, Junfeng Ni, Song-Chun Zhu, and Siyuan Huang. VideoArtGS: Building digital twins of articulated objects from monocular video. *arXiv preprint arXiv:2509.17647*, 2025a. URL https://arxiv.org/abs/2509.17647.

Yu Liu, Baoxiong Jia, Ruijie Lu, Junfeng Ni, Song-Chun Zhu, and Siyuan Huang. Building interactable replicas of complex articulated objects via gaussian splatting. In *International Conference on Learning Representations (ICLR)*, 2025b. URL https://openreview.net/forum?id=ix2yRWarPn.

Yongsen Mao, Yiming Zhang, Hanxiao Jiang, Angel X. Chang, and Manolis Savva. MultiScan: Scalable RGBD scanning for 3D environments with articulated objects. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9058–9071, 2022. doi: 10.52202/068431-0658. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/3b3a83a5d86e1d424daefed43d998079-Abstract-Conference.html.

Roberto Martín-Martín, Clemens Eppner, and Oliver Brock. The RBO dataset of articulated objects and interactions. *The International Journal of Robotics Research*, 38(9):1013–1019, 2019. doi: 10.1177/0278364919844314.

OpenAI. Gpt-5.4 model. https://developers.openai.com/api/docs/models/gpt-5.4. Accessed: 2026-09-26.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 12708–12717. IEEE, 2019.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. Shape2Motion: Joint analysis of motion parts and attributes from 3D shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8876–8884, June 2019. doi: 10.1109/CVPR.2019.00908.

Yijia Weng, He Wang, Qiang Zhou, Yuzhe Qin, Yueqi Duan, Qingnan Fan, Baoquan Chen, Hao Su, and Leonidas J. Guibas. CAPTRA: CAtegory-Level Pose Tracking for Rigid and Articulated Objects From Point Clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13209–13218, 2021. doi: 10.1109/ICCV48922.2021.01296.

Yijia Weng, Bowen Wen, Jonathan Tremblay, Valts Blukis, Dieter Fox, Leonidas J. Guibas, and Stan Birchfield. Neural implicit representation for building digital twins of unknown articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3141–3150, June 2024. doi: 10.1109/CVPR52733.2024.00303.

XGRIDS. Lixel l2 pro: Scanning workflow. https://docs.xgrids.com/en-us/03-lingguang-l/01-lingguang-l2-pro/latest/04-scanning-workflow.html. Accessed: 2026-09-26.

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 (CVPR)*, pp. 11097–11107, June 2020. doi: 10.1109/CVPR42600.2020.01111.

Albert J. Zhai, Yuan Shen, Emily Y. Chen, Gloria X. Wang, Xinlei Wang, Sheng Wang, Kaiyu Guan, and Shenlong Wang. Physical Property Understanding from Language-Embedded Feature Fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28296–28305, 2024. doi: 10.1109/CVPR52733.2024.02673.

Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y. Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T. Freeman. PhysDreamer: Physics-Based Interaction with 3D Objects via Video Generation. In *Computer Vision – ECCV 2024*, volume 15060 of *Lecture*

Notes in Computer Science, pp. 388–406. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72627-9_22.

Mandi Zhao, Yijia Weng, Dominik Bauer, and Shuran Song. Real2code: Reconstruct articulated objects via code generation. In *International Conference on Learning Representations*, 2025.

Appendix

The supplementary material is organized as follows. Appendix A describes RGB-D acquisition, metric reconstruction, and registration between real observations and digital assets. Appendix B details part modeling, articulated USD representation, and quality validation. Appendix C presents physical-property annotation, motion-based calibration, and simulator adaptation with SolverAdapt. Appendix D presents analyses of real–digital reference consistency and physical-property distributions. Appendix E reports additional experimental results. Appendix F documents the dataset structure and file formats, including camera parameters and the selected joint's start and end configurations. Appendix G discusses the limitations of the dataset and its evaluation.

A Real-World Data Acquisition and Calibration

The acquisition protocol records multi-view geometry and interaction motion for each physical instance, as summarized in Fig. 2.

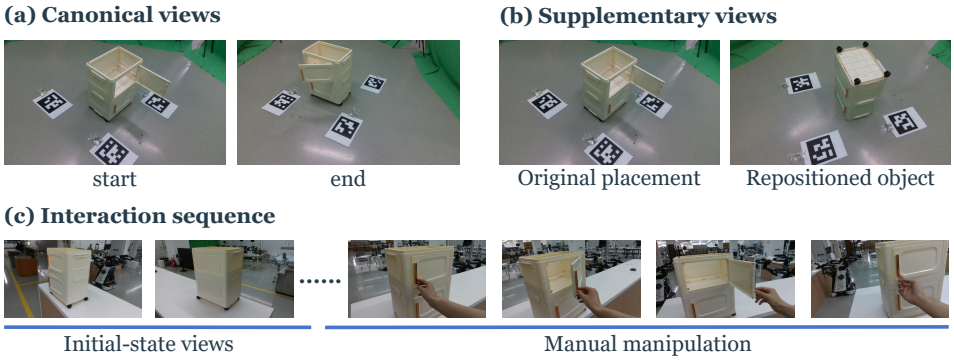


Figure 5: **Acquisition protocol.** Representative RGB frames show the same physical instance. (a) Canonical views capture two configurations of the selected joint. (b) Repositioning exposes the underside hidden by ground contact. (c) Six temporally ordered frames from one interaction sequence show initial-state views followed by drawer manipulation.

A.1 Capture Setup and Protocol

Capture setup. We place each object directly on the ground and capture it with an Intel RealSense D435i RGB-D camera (Keselman et al., 2017). The camera is rigidly mounted on the lid of a laptop carried by the operator. The operator walks around the object along a circular path, adjusting the radius using the live preview to maintain the desired object scale in the image. RGB and depth streams are recorded at 1280×720 and 30 Hz, with RGB-D frames retained at approximately 3 Hz.

Multi-view and articulation-state capture. We capture the selected joint in two configurations, keeping its state fixed throughout each camera pass (Fig. 5(a)). The two passes are stored as `canonical/start` and `canonical/end` (Fig. 14). These labels identify observed configurations rather than numerical joint limits. To expose surfaces hidden by contact with the ground, we reposition the object and acquire supplementary views (Fig. 5(b)). These observations are stored under `supplementary/<state>` (Fig. 14), and the new placement is registered separately before combination with the canonical captures.

Interaction capture. Following the initialization requirement of VideoArtGS (Liu et al., 2025a), we begin each RGB-D interaction sequence by moving the camera around the stationary object.

These initial views support geometric initialization before an operator manipulates the selected part while recording continues (Fig. 5(c)). The ordered frames are stored under `interaction/` (Fig. 14) for joint-trajectory extraction and physical-parameter calibration in Appendix C.

A.2 Camera Calibration and Metric Reconstruction

Part segmentation and manual refinement. We manually provide point and bounding-box prompts for each target part in the RGB frames to guide SAM 2 (Ravi et al., 2025) in generating part masks. We iteratively inspect the results, adjust the prompts, and manually refine the masks until each mask delineates the visible target part and excludes neighboring parts and background. For canonical and supplementary captures, the refined part masks apply to both the RGB frame and its aligned depth map. Fig. 6 shows the prompt-based annotation process, the manual correction of an initial SAM 2 result, and the refined part-level masks across multiple views of the same object.

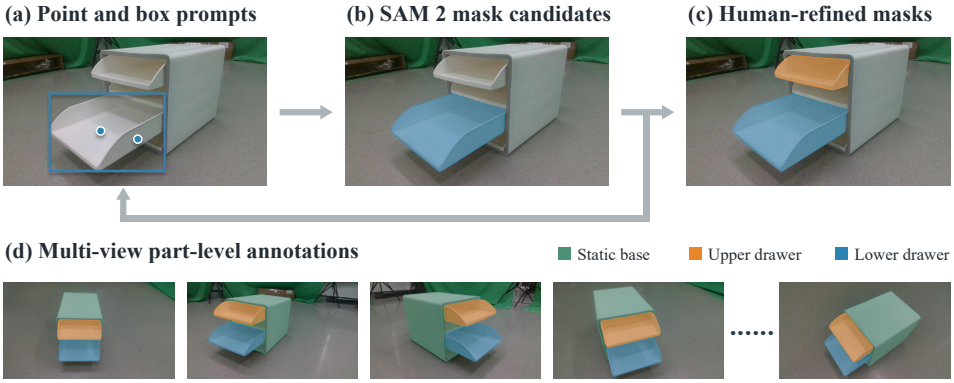


Figure 6: **Part-level mask annotation and multi-view results.** (a) Illustrative point and bounding-box prompts identify the upper and lower drawers. (b) SAM 2 produces initial part-level mask candidates for the prompted view. (c) Manual review corrects missing or misassigned regions. The return arrow indicates prompt adjustment and re-segmentation after inspection. (d) Refined masks for the static base and two movable parts are shown in six views of the same joint state. Green, orange, and blue denote the static base, upper drawer, and lower drawer, respectively.

Metric reconstruction and registration. ArUco (Garrido-Jurado et al., 2014) observations initialize camera poses, followed by HLoc (Sarlin et al., 2019) feature correspondences and global structure-from-motion (Schonberger & Frahm, 2016) refinement. After local hole repair and noise removal, the depth measurements anchor the reconstruction in meters, and each object placement is registered separately. For each canonical capture, we match the selected USD joint to the observed part configuration. We then refine the asset's rigid pose using ICP (Besl & McKay, 1992), keeping the matched joint position and metric scale fixed. The confirmed positions for both captures are saved in `canonical/joint_states.json` (Appendix F).

A.3 Asset Registration and Camera Correspondence

Coordinate transformations. Let $\mathbf{T}_{wc}^{(v)}$ map coordinates from camera v to reconstruction frame w, and let $\mathbf{T}_{wa}$ map asset coordinates into that frame. The asset-to-camera transform is

$$\mathbf{T}_{ca}^{(v)} = \left(\mathbf{T}_{wc}^{(v)}\right)^{-1}\mathbf{T}_{wa}. \tag{4}$$

All translations use meters, and camera-axis conventions are converted to a common convention before composing the transforms. Together with camera intrinsics, image resolution, and the matched joint state, this transform determines the corresponding rendered view.

Stored camera poses. In the capture-side `extrinsics.json`, `image_path` identifies a frame and `object_to_camera_matrix` stores its transform $\mathbf{E}$ from the capture reference frame to the camera frame. Using homogeneous column vectors, the accompanying `transform_matrix`

equals $\mathbf{E}^{-1}\,\mathrm{diag}(1,-1,-1,1)$. It maps camera coordinates back to the capture reference frame after reversing the camera Y and Z axes. These fields therefore require both the stated axis conversion and registration of the capture reference frame to w before use in Eq. 4. Depth PNG values are integer sensor codes and require the capture's depth scale to obtain metric distances. File organization and modality availability are described in Appendix F.

B Asset Modeling and Quality Validation

We combine scans and direct measurements of each physical object with manual part modeling and geometric validation to construct its articulated USD asset.

B.1 Geometry Acquisition and Part Modeling

Scanning and dimensional measurements. We use an L2 Pro scanner to capture the front, sides, back, and accessible interior along a closed scanning trajectory. Movable parts, such as drawers, are scanned separately to expose occluded surfaces. We process the scans in Lixel Studio to optimize trajectories and register repeated captures. Before exporting the point cloud, we remove background points and noise and check for duplicated surfaces and missing regions. We measure overall dimensions, board thickness, hole spacing, handle dimensions, and assembly gaps, reserving a subset of these measurements for final geometric validation.

Part modeling and geometric validation. Using point-cloud outlines and measured dimensions, we manually model each part, including rounded edges, holes, and handles. After matching units and articulation states, we coarsely align the point cloud and mesh using corresponding bottom surfaces, side surfaces, and geometric features. We then refine the alignment using ICP (Besl & McKay, 1992) between the scanned point cloud and points sampled from the mesh surface, keeping the scale fixed. We refine the mesh based on surface deviations and dimensional discrepancies, acquiring additional scans or measurements where geometry is missing. The resulting mesh is checked against the reserved measurements for overall dimensions, structural details, and assembly clearances. If a reserved measurement guides further revision, we treat it as a modeling constraint and exclude it from independent validation.

B.2 Articulated USD Representation

USD structure and part naming. Each asset is loaded through `model.usda`, which composes the binary layer `asset.usdc` with `/root` as the default prim. Rigid parts and joints use instance-local paths `/root/link_N` and `/root/joint_N`. Part geometry uses `Mesh` prims, with visual appearance defined by `Material` and `Shader` networks. A rigid part can contain several meshes, and separately named parts that move together are connected by fixed joints. Tab. 6 summarizes the physics schemas used in the assets.

Table 6: **Physics schemas in the USD assets.** Names follow the serialized USD schema tokens. Parameter estimation and calibration are described in Appendix C.

Schema	Stored information or attachment
`PhysicsRigidBodyAPI`, `PhysicsMassAPI`	Rigid parts, mass, center of mass, inertia, and principal axes.
`PhysicsArticulationRootAPI`	The fixed joint connecting the root rigid body to the world.
`PhysicsFixedJoint`, `PhysicsRevoluteJoint`, `PhysicsPrismaticJoint`	Connected bodies, local joint frames, axes, and limits where applicable.
`PhysicsCollisionAPI`, `PhysicsMeshCollisionAPI`	Collision geometry and its approximation on part meshes.
`PhysicsMaterialAPI`	Contact friction, restitution, and material density.
`PhysicsDriveAPI:angular`, `PhysicsDriveAPI:linear`	Joint-drive targets, stiffness, damping, and force or torque limits.

Joint and coordinate conventions. Fixed, revolute, and prismatic joints connect parent and child bodies through `physics:body0` and `physics:body1`. The root fixed joint has an empty `body0` relationship, anchoring the asset to the simulation world. The fields `localPos0/1` and `localRot0/1` define joint frames relative to the two bodies, with quaternions ordered as (w, x, y, z). Joint zero is defined by these local frames and is independent of the capture-state labels. The `axis` token selects a local X, Y, or Z axis, with positive translation along that axis and positive rotation following the right-hand rule. Finite ranges use `lowerLimit` and `upperLimit`, while continuous revolute joints retain unbounded limits. All assets use right-handed Z-up frames with `metersPerUnit=1` and `upAxis=Z`, with mass in kilograms and time in seconds. Prismatic limits are in meters and revolute limits are in degrees, requiring conversion when an evaluator uses radians. Assets retain their authored root origins, which are not uniformly located at the bottom center. Eq. 4 relates the asset frame to the registered observation cameras.

B.3 Quality Validation and Export

Appearance, collision, and export. We assign each part physically based materials specifying color, roughness, and metallic response, with UV-mapped textures where available. Collision shapes are assigned to the corresponding rigid links using convex hulls, convex decomposition, or signed-distance-field (SDF) representations. Before export, we inspect mesh normals, broken faces, and overlapping surfaces, and verify assembly clearances and joint motion against the observed configurations. The final deliverables include named part meshes, the articulated USD asset, and dimensional validation records. Physical-property annotation and SolverAdapt are detailed in Appendix C.

C Physical-Property Annotation and SolverAdapt

This section describes how physical parameters are initialized, reviewed, calibrated against recorded motion, and retuned for a target simulator.

C.1 Parameter Initialization and Review

Parameter initialization. We initialize part-level physical properties from visual and geometric evidence before motion-based calibration. For each rigid part, the input consists of the part image, semantic part name, and available geometric information. A vision-language model predicts the material category and provides initial estimates of density, friction-related properties, and dynamic parameters associated with the part. The predicted part masses are jointly adjusted under the constraint that their summation equals the measured mass of the complete physical object:

$$\sum_i m_i = M_{\text{obj}}, \tag{5}$$

where m_i denotes the assigned mass of part i and M_{obj} is the measured object mass. This constraint preserves consistency with physical measurement while allowing the model to distribute mass according to part appearance and role. We use GPT-5.4 and Claude Opus 4.6 as the VLM for generating the initial physical-property estimates. The initialization provides starting values and search ranges for subsequent motion-based calibration rather than final simulation parameters.

Human review. Before calibration, annotators review the initialized physical configuration. The review process examines material-category consistency, physically implausible parameter values, and mass allocation across rigid parts. When necessary, annotators correct material labels, adjust abnormal estimates, and revise part-level mass assignments while preserving the measured total mass constraint. The verified configuration is then used as the initialization for motion-based optimization.

C.2 Motion-Based Calibration

Motion-based calibration. Eq. 2 fits the simulated trajectory to the selected joint's trajectory extracted by VideoArtGS (Liu et al., 2025a). For T observations at times $\tau_1, \ldots, \tau_T$, we assign one constant excitation to each interval:

$$u(\tau) = u_i, \qquad \tau_i \le \tau < \tau_{i+1}, \quad i = 1, \ldots, T-1, \tag{6}$$

where $\mathbf{u} = (u_1, \ldots, u_{T-1})$ contains forces along a prismatic axis or torques about a revolute axis. These inputs are inferred from motion under the simulation model rather than measured human-applied forces. An initial excitation search determines the input sign, which remains fixed within an episode while the interval magnitudes and dynamics parameters are jointly optimized within their bounds. Geometry, mass, density, and joint types, axes, pivots, and limits remain fixed.

Regularization and optimization settings. Let $K = T - 1$, $z_i = u_i/U$, and $\Delta z_i = z_{i+1} - z_i$, where U is the excitation-magnitude search bound. The excitation terms in $\mathcal{R}$ are

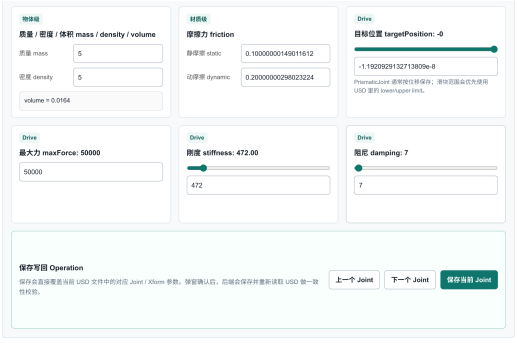
$$\mathcal{R}_u = \frac{0.005}{K} \sum_{i=1}^{K} z_i^2 + \frac{0.02}{K-1} \sum_{i=1}^{K-1} (\Delta z_i)^2 + \frac{0.02}{K-1} \sum_{i=1}^{K-1} [|\Delta z_i| - 0.25]_+^2 , \tag{7}$$

where $[x]_+ = \max(x, 0)$, and the adjacent-interval terms are omitted when $K = 1$.

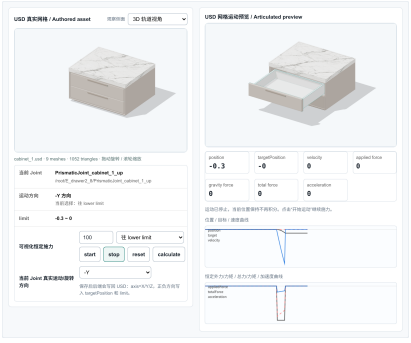
For each material with dynamic friction μ_d exceeding static friction μ_s, the implementation adds $10^3(\mu_\mathrm{d} - \mu_\mathrm{s} + 1)$ to the objective. For parameter j, let b_j be its absolute deviation from initialization divided by the distance to the search bound in that direction. The parameter-bound penalty is $0.005 \sum_j [(b_j - 0.75)/0.25]_+^2$. The trajectory-gap term divides absolute residuals by the authored joint-range scale and penalizes their mean squared excess above 0.10 with weight 0.08. The seven archived runs use bounded differential evolution with a budget of 200 evaluations and seed 20260918, recording initial values, parameter bounds, and refined USD attributes. Motion consistency and sensitivity to stiffness and damping are evaluated in Sec. 4.4.

C.3 Simulator Adaptation with SolverAdapt

SolverAdapt interface. SolverAdapt loads USD assets for interactive inspection of part properties and joint settings (Fig. 7). Users select a joint to inspect and edit the associated part's mass and density, contact friction, and drive parameters. The browser renders the USD meshes and materials, with the selected part driven by the lightweight joint preview. Response curves under a constant force or torque guide adjustment before validation in the target simulator. Stiffness and damping are interpreted according to joint type and drive mode. Saving writes the edited attributes to USD and reloads the asset to verify the stored values.



(a) Parameter editing.



(b) Motion preview.

Figure 7: **SolverAdapt interface.** A cabinet with a prismatic joint is used for this demonstration. (a) The parameter panel exposes part properties, contact friction, and drive settings, with controls for saving changes to USD. (b) The motion panel shows the selected joint's direction and limits alongside a lightweight preview and response curves under a user-specified constant force.

D Detailed Dataset Statistics

This appendix supplements Sec. 3.4 with the full category inventory in Tab. 7, articulation statistics in Fig. 8, reference consistency in Fig. 9, and physical-property distributions in Fig. 10.

Table 7: **Category-level inventory and capture budget.** All assets include paired and additional unpaired models. Parts include separately modeled components connected by fixed joints. Static views count images captured at the start and end articulation states under canonical placement, together with additional images captured under repositioned placements.

Category	All	Paired	Unpaired	Revolute	Prismatic	Parts	Static views
Oven	44	12	32	191	57	293	8,796
Microwave	27	10	17	59	50	138	8,520
Rice cooker	21	18	3	45	36	112	14,760
Air fryer	15	12	3	3	15	54	9,240
Laptop	30	15	15	30	0	60	13,275
Heater	8	8	0	20	0	36	5,696
Refrigerator	60	12	48	104	31	199	10,296
Drawer cabinet	240	40	200	137	693	1,087	32,640
File cabinet	60	15	45	56	171	291	11,670
Letter tray	13	10	3	0	45	67	8,410
Storage cabinet	198	43	155	325	156	699	32,680
Shelf cabinet	28	10	18	64	39	174	8,790
Cubby	25	10	15	17	24	66	7,410
Drawer box	27	15	12	0	61	96	12,495
Toolbox	18	10	8	38	28	85	7,190
Basket	20	12	8	40	0	62	10,368
Lamp	58	18	40	88	21	193	14,256
Trash can	100	25	75	179	24	311	21,150
Stand	16	15	1	55	15	205	11,295
Safe	36	8	28	83	20	243	7,136
First-aid kit	8	8	0	31	4	64	5,816
Jewelry box	12	8	4	4	16	48	6,632
Rice bin	12	8	4	15	12	72	6,168
Sunglasses	62	12	50	124	0	186	10,476
Scissors	21	12	9	21	0	56	8,508
Knife	30	12	18	0	31	62	9,828
Folding phone	11	8	3	12	0	52	6,856
Glasses case	11	8	3	11	0	22	6,104
Folding chair	80	10	70	101	0	181	8,940
Ladder	11	6	5	19	0	55	4,428
Pump	13	6	7	0	13	26	5,010
Pencil case	9	6	3	9	0	18	4,704
Glue	15	5	10	15	15	60	4,385
Compass	9	5	4	9	0	18	3,730
Calculator	9	5	4	9	1	21	4,240
Latch	14	5	9	0	14	28	3,580
Selfie stick	12	5	7	0	36	48	4,075
Hanger	10	5	5	20	0	30	4,500
Telescope	11	5	6	33	11	55	3,655
French press	10	5	5	0	10	30	4,125
Faucet	35	20	15	52	7	100	15,380
Handle	8	5	3	8	0	16	4,305
Bucket	43	10	33	45	0	88	8,070
Total	**1,500**	**497**	**1,003**	**2,072**	**1,656**	**5,807**	**399,588**

D.1 Inventory and Counting Conventions

Tab. 7 separates the full asset collection from the paired subset in each category. An asset is counted once regardless of the number of views, articulated configurations, or mesh components. A paired instance links released real RGB-D observations of one physical object to an articulated digital model of that same object. Each physical object is counted once, regardless of the number of views or captured joint configurations. Static capture counts follow the acquisition protocol described in Sec. 3.1. RGB, aligned depth, and part masks associated with a camera pose constitute one view and are not counted as separate observations. Interaction frames are excluded from these static-view totals. Tab. 1 follows the counting conventions of the original papers or public releases. AKB-48 includes CAD-derived models, whereas MultiScan counts articulated-object instances in scene

scans. PartNet-Mobility uses the SAPIEN 1.2 inventory. ArtVIP and Artiverse report subcategories. Motion calibration denotes refinement of asset dynamics against real-object motion and describes the documented construction or validation procedure.

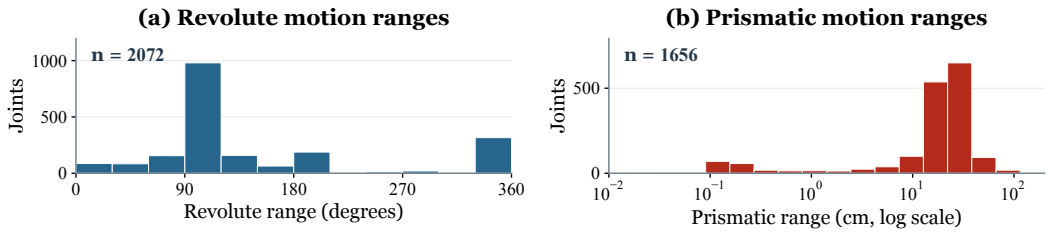


Figure 8: **Joint motion ranges.** (a) Revolute ranges in degrees and (b) prismatic ranges in centimeters, with a logarithmic horizontal axis for prismatic joints.

D.2 Articulation Structure and Motion Ranges

Among the 497 paired assets, 368 contain multiple movable parts and 143 combine revolute and prismatic joints, corresponding to 74.0% and 28.8% of the paired subset. Mixed structures form a subset of the models with multiple movable parts. These statistics describe internal articulation and exclude fixed joints and global object motion. Joint multiplicity describes the full authored asset, while the capture protocol observes a selected joint at two configurations and during an interaction. The captured configurations therefore do not enumerate the full joint-state space.

Fig. 8 shows the distribution of authored joint motion spans. Revolute ranges are concentrated around 90°–120°, with a smaller concentration around 180°–210° (Fig. 8(a)). Prismatic ranges extend across several orders of magnitude, from sub-centimeter travel to displacements of tens of centimeters (Fig. 8(b)). This variation motivates reporting angular and translational errors separately and accounting for motion span when comparing trajectory residuals across joint types, as in Eq. 2. The histograms describe the authored limits of the plotted joints; an interaction may traverse only part of that range.

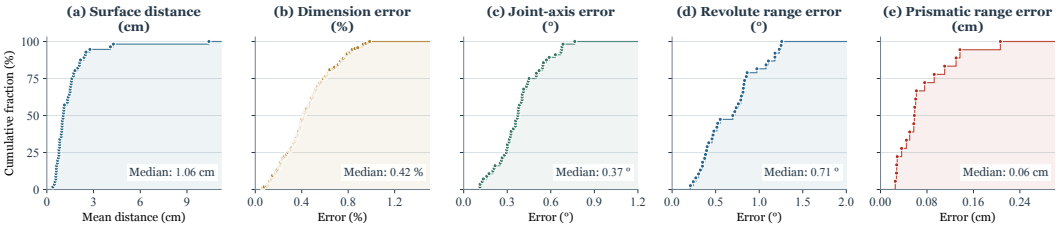


Figure 9: **Real–digital reference consistency.** Cumulative distributions of (a) per-object surface distances between real RGB-D observations and registered digital assets, and (b)–(e) dimensional, joint-axis, revolute-range, and prismatic-range discrepancies.

D.3 Real–Digital Reference Consistency

Fig. 9 summarizes complementary comparisons between physical objects and their registered digital assets. The surface distances in Fig. 9(a) measure agreement with the RGB-D observations under matched joint states and a common metric frame. The displayed median is 1.06 cm. These distances include depth noise, incomplete visibility, registration error, and asset discrepancy, so they do not isolate modeling error. The dimensional, joint-axis, and motion-range comparisons in Fig. 9(b)–(e) address metric and kinematic agreement with the corresponding physical objects. Independent dimensional checks use the reserved measurements described in Appendix B.1; any measurement used to revise the model is excluded from that validation set.

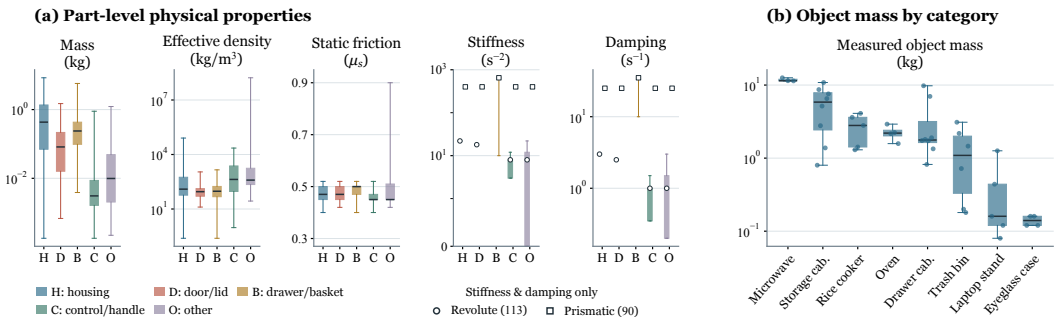


Figure 10: **Physical-property distributions of ArtModeling.** (a) Part-level physical-property distributions across different part roles, including housing, door/lid, drawer/basket, and control/handle. (b) Measured object-mass distributions across selected object categories.

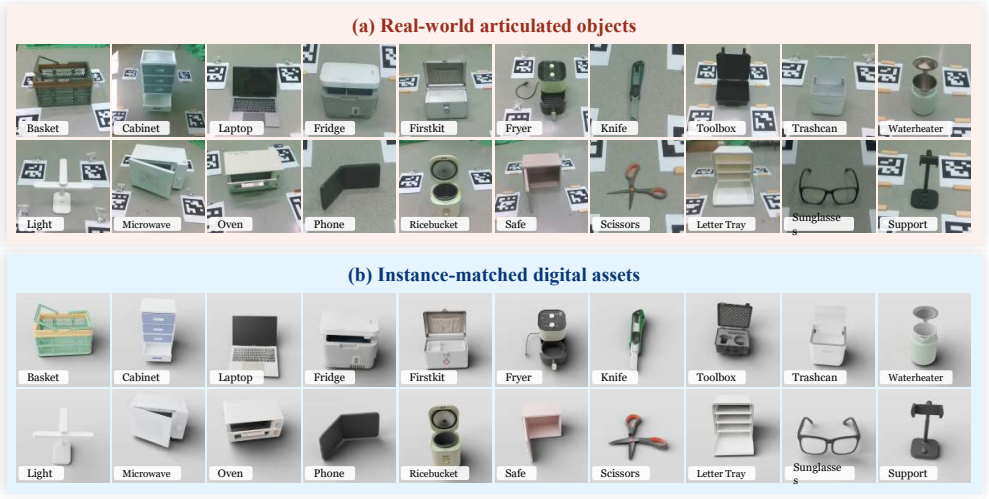


Figure 11: **Additional paired real–digital examples in ArtModeling.** The top block shows real-world RGB-D observations, and the bottom block shows the corresponding simulation-ready USD assets reconstructed from the same physical instances.

D.4 Paired Real–Digital Examples

Fig. 11 shows additional paired examples of real-world articulated objects and their simulation-ready USD assets across object categories. ArtModeling covers visually diverse physical objects, ranging from household appliances to everyday items, and provides corresponding digital assets reconstructed from the same object instances. The paired assets preserve category-specific shape, appearance, and part structure while providing editable, simulation-ready representations.

D.5 Physical Property Distribution

Fig. 10(a) summarizes assigned part mass, effective bulk density, contact friction, and joint-drive stiffness and damping by semantic role. Housing parts generally have greater assigned mass than controls and handles, while drive settings depend on part role and joint type. The relatively concentrated friction values and the broader mass and density distributions describe different components

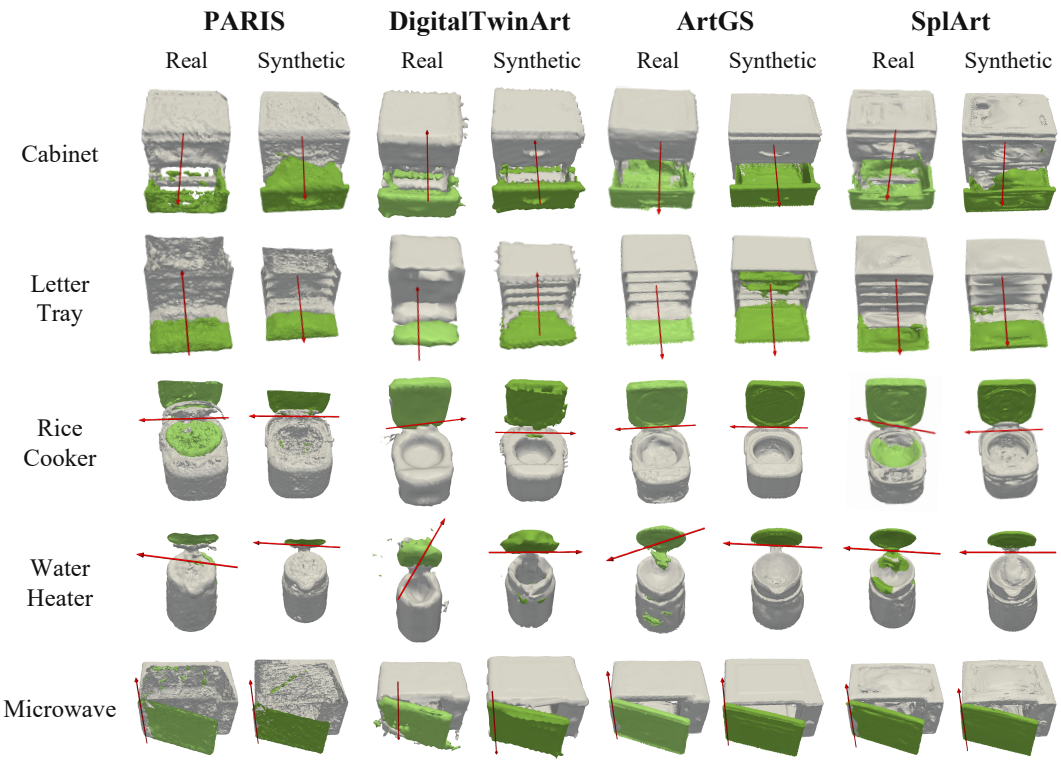


Figure 12: Qualitative comparison of reconstructed meshes and predicted joint axes under real and synthetic observations.

of the physical annotation. Measured object mass also varies substantially within categories such as storage cabinets and trash bins (Fig. 10(b)). This within-category variation motivates instance-specific mass constraints in physical annotation and calibration (Sec. 3.3).

E Additional Experiments

E.1 Qualitative Comparison under Real and Synthetic Inputs

Fig. 12 compares reconstructed meshes and predicted joint axes from PARIS, DigitalTwinArt, ArtGS, and SplArt on representative objects with prismatic and revolute joints. Each method receives real observations and matched renderings of the same object, allowing differences in reconstruction and articulation estimates to be examined without changing object identity.

The geometric differences are visible in both surface regularity and the reconstruction of individual components. ArtGS produces comparatively smooth cabinet and microwave surfaces in both domains, whereas PARIS retains pronounced surface irregularities. DigitalTwinArt exhibits clearer changes between the two input domains. For the letter tray, synthetic inputs yield more distinct horizontal compartments than real observations, while the microwave reconstruction has a less fragmented door boundary. These examples illustrate how the input domain affects local structure and part boundaries even when the overall object shape remains recognizable.

The predicted joint axes exhibit a different pattern of variation. For the water heater, DigitalTwinArt and ArtGS produce visibly different axis orientations under real and synthetic inputs, despite recovering the main body and lid in both cases. The microwave examples show more consistent axis directions across domains, indicating that these changes depend on both the method and the object. Together with the quantitative results in Tab. 2 and Tab. 3, these examples reinforce the distinction between geometric reconstruction and kinematic estimation. A visually coherent mesh does

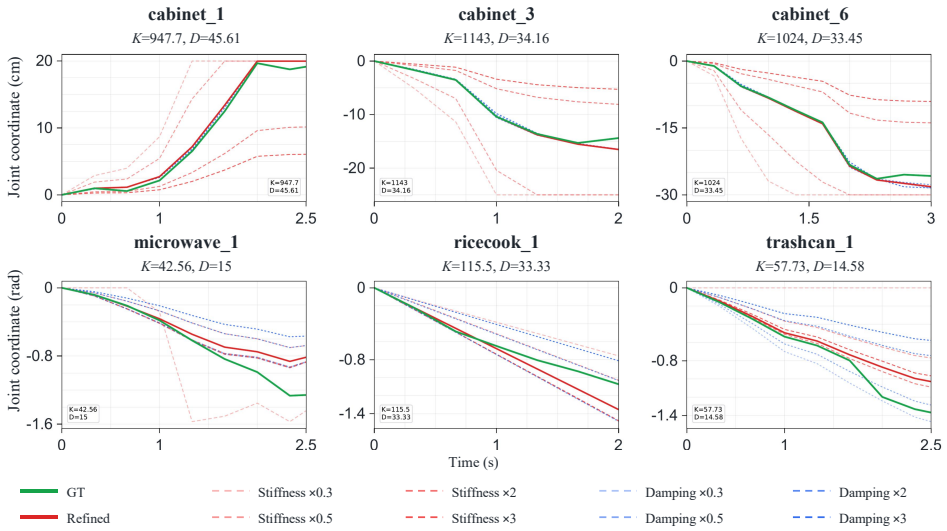


Figure 13: **Sensitivity to joint-drive stiffness and damping.** Each perturbation scales one calibrated parameter by 0.3, 0.5, 2, or 3, with all other parameters and excitation fixed. The top row shows prismatic joints and the bottom row revolute joints. GT denotes the trajectory extracted from the recording; K and D denote the calibrated stiffness and damping.

not by itself establish accurate articulation recovery, and evaluation on matched real and synthetic observations is needed to characterize both aspects.

E.2 Parameter Sensitivity Analysis

The calibrated configuration remains sensitive to parameter perturbations. For the six objects in Fig. 13, we scale stiffness or damping by 0.3, 0.5, 2, or 3, keeping geometry, initial state, excitation, and all other parameters fixed. Stiffness changes substantially affect the three prismatic trajectories, whereas damping changes leave them close to the calibrated result; the three revolute trajectories respond to both parameters. Across eight perturbations per object, an average of 1.00 gives a lower residual than Refined for the prismatic examples and 2.67 for the revolute examples. Calibration on one interaction therefore leaves room for improvement on another, making the recorded motion a reusable evaluation target for alternative parameter settings and subsequent retuning with SolverAdapt.

F Datasheet for ArtModeling

Dataset directory layout. Data are organized under `datasets/<object_id>/`. For paired instances, the same numeric identifier is used for the physical object and its corresponding digital asset. The root `manifest.json` maps each `object_id` to its instance name, capture paths, supplementary state labels, and paired asset when available. Each of the 497 paired instances contains both `real/` and `sim/`, while the 1,003 simulation-only assets contain `sim/` without a corresponding `real/` capture. Fig. 14 shows the directory structure. The root-level `intrinsics.json` is reserved for camera calibration shared by canonical, supplementary, and interaction captures. It specifies the RGB and depth intrinsics, image resolutions, and distortion models for the recording configuration.

```
ArtModeling/
├─ manifest.json                  # instance and path index
├─ intrinsics.json                # shared RGB-D camera calibration
└─ datasets/<object_id>/
   ├─ real/
   │  ├─ canonical/{start,end}/
   │  │  ├─ img/                  # RGB frames
   │  │  ├─ depth/                # depth frames
   │  │  ├─ mask/                 # part-level masks shared by RGB-D
   │  │  └─ extrinsics.json       # per-view poses, where available
   │  ├─ supplementary/<state>/   # same static-capture layout
   │  └─ interaction/
   │     ├─ img/                  # ordered RGB frames
   │     ├─ depth/                # ordered depth frames
   │     └─ video/                # RGB and depth-preview videos
   └─ sim/                        # paired instances only
      ├─ model.usda               # USD entry layer
      ├─ asset.usdc               # binary asset layer
      └─ textures/                # material images, where used
```

Figure 14: **Dataset directory layout for paired instances.** Observations and paired assets are grouped by object ID, with shared camera calibration at the root. Part masks for each static frame are shared by RGB and aligned depth. Simulation-only assets omit `real/` and retain `sim/`.

Canonical joint states. For each paired instance, `canonical/joint_states.json` stores the selected USD joint's start and end positions confirmed during registration. The `joint_path` field identifies the selected USD joint, while `start` and `end` give its matched positions. The `unit` field is `degree` for a revolute joint and `meter` for a prismatic joint. Both positions follow the joint's USD zero and positive direction (Appendix B.2). Each position applies to all views in the corresponding `start/` or `end/` capture.

Observation files and frame correspondence. The `start` and `end` labels identify observed joint configurations rather than numerical joint limits. RGB and depth frames are stored directly in `img/` and `depth/` as 8-bit and 16-bit PNG images, respectively. Depth values are integer sensor codes and require the capture's depth scale to obtain metric distances. For canonical and supplementary captures, `mask/` stores the part masks associated with each RGB-D frame. Each mask delineates the visible region of a target part, separating it from other parts and the background. Depth is aligned to the RGB image grid, so the same part masks apply to both modalities and are stored in a single `mask/` directory for each capture state. Frame identifiers associate RGB images, depth maps, and their part masks, with frames ordered by numeric index within each sequence. The `video/` directory provides RGB playback and depth visualization, while metric processing uses the depth PNGs.

Stored camera poses. In `extrinsics.json`, `image_path` associates each pose record with its corresponding RGB frame. The field `object_to_camera_matrix` stores the transform $\mathbf{E}$ from the capture reference frame to the camera frame. For homogeneous column vectors, `transform_matrix` equals $\mathbf{E}^{-1}\,\mathrm{diag}(1,-1,-1,1)$, mapping camera coordinates back to the capture reference frame with the camera Y and Z axes reversed. Using these fields in Eq. 4 requires accounting for this axis conversion and registering the capture reference frame to reconstruction frame w (Appendix A.3).

Asset files. The USD entry layer `model.usda` composes `asset.usdc` and resolves its material dependencies. Joint types, parent–child connections, local joint frames, axes, and limits are stored in the USD asset. Material dependencies are stored alongside the asset, including textures and local material modules where required. Appendix B.2 describes the USD structure, joint definitions, and unit conventions.

G LIMITATIONS

ArtModeling currently focuses on objects with revolute and prismatic joints. The capture protocol follows one selected joint with the remaining joints held fixed, so the real observations cover a subset of the configurations supported by the assets. Extending the collection to additional joint types and capturing coordinated multi-joint motion would support evaluation of more diverse kinematic structures, part interactions, and occlusion patterns.

Physical calibration is evaluated on a second repetition of the same joint motion under a fixed simulator configuration. Applied forces are inferred from motion, so different combinations of excitation and physical parameters may produce similar trajectories. Additional force measurements and recordings spanning different speeds, loads, and motion directions would better constrain parameter estimation and test the calibrated dynamics over a wider range of interactions.