DIPO: Dual-State Images Controlled Articulated Object Generation Powered by Diverse Data

Ruiqi Wu[1, 2, 3*] **Xinjie Wang**[3] **Liu Liu**[3] **Chunle Guo**[1, 2] **Jiaxiong Qiu**[3] **Chongyi Li**[1, 2†]
Lichao Huang[3] **Zhizhong Su**[3] **Ming-Ming Cheng**[1, 2]
[1]NKIARI, Shenzhen Futian [2]VCIP, CS, Nankai University [3]Horizon Robotics

Abstract

We present **DIPO**, a novel framework for the controllable generation of articulated 3D objects from a pair of images: one depicting the object in a resting state and the other in an articulated state. Compared to the single-image approach, our dual-image input imposes only a modest overhead for data collection, but at the same time provides important motion information, which is a reliable guide for predicting kinematic relationships between parts. Specifically, we propose a dual-image diffusion model that captures relationships between the image pair to generate part layouts and joint parameters. In addition, we introduce a Chain-of-Thought (CoT) based **graph reasoner** that explicitly infers part connectivity relationships. To further improve robustness and generalization on complex articulated objects, we develop a fully automated dataset expansion pipeline, name **LEGO-Art**, that enriches the diversity and complexity of PartNet-Mobility dataset. We propose **PM-X**, a large-scale dataset of complex articulated 3D objects, accompanied by rendered images, URDF annotations, and textual descriptions. Extensive experiments demonstrate that DIPO significantly outperforms existing baselines in both the resting state and the articulated state, while the proposed PM-X dataset further enhances generalization to diverse and structurally complex articulated objects. Our code and dataset are available at https://github.com/RQ-Wu/DIPO.

1 Introduction

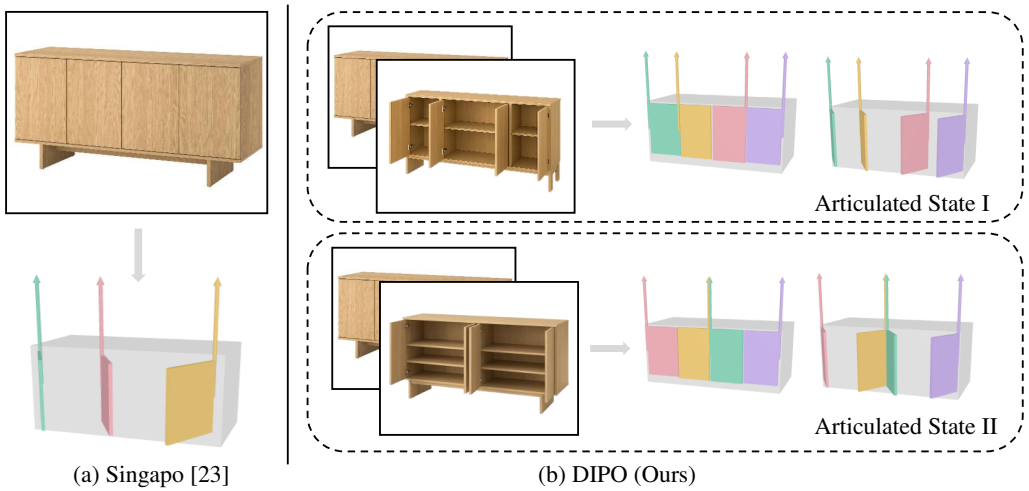
Articulated objects are pervasive in everyday environments. Achieving accurate modeling of articulated structures is the key enabler for building interactive virtual environments. It plays a crucial in simulation [49, 42, 46], animation [48, 5, 33, 21], robot manipulation [11, 9, 28, 32], and embodied AI [19, 36, 31, 20, 16].

However, constructing such models manually is highly labor-intensive and unscalable. As a result, increasing attention has been devoted to developing automatic methods for articulated object modeling [39, 22, 43, 18, 24, 6, 23]. Despite promising progress, existing methods exhibit clear performance degradation when applied to structurally complex or visually ambiguous objects. These limitations stem from two fundamental bottlenecks.

The first issue is **input modality constraints.** Reconstruction-based approaches [39, 22, 43] often rely on multi-view or multi-state images to reconstruct articulation behavior with high accuracy. While effective, these methods demand expensive data acquisition setups, precise camera calibration, and well-aligned temporal input, making them difficult to scale. On the other research line, benefiting from the controllability of diffusion models [12, 37, 35, 30, 45, 51, 50], many generation-based methods [18, 24, 6, 23] are proposed. They utilize minimal input, such as category priors or a

*This work was done while Ruiqi Wu was a Research Intern with Horizon Robotics.
†denotes correspondence author.

Preprint. Under review.



(a) Singapo [23] (b) DIPO (Ours)

Figure 1: **Visual comparison on real-captured data.** (a) SINGAPO struggles with challenging data and fails to model motion relationships due to its reliance on a single input. However, our DIPO (b), which conditioned on dual-state image pairs, effectively generates accurate layouts and enables precise control if part motion across different articulated states.

single RGB image, to synthesize articulated objects directly. However, category priors lack spatial specificity, and single-image inputs lack of explicit articulation information. As a result, these methods can only infer kinematic behaviors in a probabilistic manner. Consequently, neither class of methods offers both control and generalization ability when facing challenging data.

Secondly, **limitations in training data.** Data-driven modeling approaches require large-scale datasets with both articulation diversity and structural complexity. However, most existing datasets fall short in some aspects. For example, PartNet-Mobility (PM) [46] offers a large number of articulated assets, but the object instances are dominated by simple and repetitive layouts with limited variability. In contrast, the Articulated Container Dataset (ACD) [14] contains more realistic and structurally diverse objects, but suffers from small scale, limiting its utility for model training.

To address the first issuse, we propose **DIPO**, a generation framework for 3D articulated objects conditioned on resting (closed) state and articulated (open) state image pairs. The dual-state image pair encodes essential motion cues and connectivity information. Compared to single-image methods, dual-state input resolves ambiguity in part motion and spatial relationships. As for multi-view methods, it is significantly easier to acquire while maintaining sufficient articulation information. DIPO is built on a diffusion transformer architecture [30] and consists of two core components. First, a *Dual-State Injection Module* helps the network to model the relationships between dual-state images. Second, a *Graph Reasoner* based on Chain-of-Thought (CoT) techniques [41, 17] infers part connectivity step by step. Moreover, this module few-shot learns on visual prompts synthesized by GPT-4o [3, 1] to acquire better performance. The proposed method achieves higher controllability and improved performance in articulated 3D object generation.

In response to the second challenge, we propose a new dataset named **PartNet-Mobility-Complex (PM-X)**, which provides diverse and structurally complex articulated objects with rendered images, URDF annotations [34], and language descriptions. PM-X is built by a fully automated data construction pipeline based on an agent system, named **LEGO-Art**. Starting from natural language prompts sampled from a LLM [3], the pipeline first generates coarse part layouts in a discretized 3D space. Then we develop a toolkit to transfer them to annotations with precise coordinates and articulation parameters. Based on retrieval algorithms [24], we can acquire the final 3D object and the rendered images. Finally, a vision-language model (VLM) [1] is used to filter implausible samples.

We collect a resting state image from the Internet and generate corresponding articulated state images by a visual generative model [1]. As illustrated in Figure 1, our method outperforms the state-of-the-art method, i.e. SINGAPO [23]. Our main contributions are summarized as follows:

- We propose a novel dual-state image model for controllable articulated 3D object generation, integrating layout diffusion and CoT-based connectivity reasoning.

- We develop LEGO-Art pipeline to construct structurally diverse articulated objects, and contribute PM-X, a new large-scale dataset with rendered images and physical annotations.
- Extensive experiments demonstrate that DIPO significantly outperforms state-of-the-art methods, and the proposed LEGO-Art and constructed PM-X dataset enhance generalization to complex structures.

2 Related Work

2.1 Articulated Object Creation

Recent progress in articulated object modeling can be broadly categorized into reconstruction-based and generation-based approaches.

Reconstruction methods commonly rely on multi-view or multi-state inputs to reconstruct part-level geometry and articulation parameters. CLA-NeRF [39] reconstructs articulated objects from sparse multi-view RGB images within a known category. PARIS [22] extends this setting to unknown categories with dual-state multi-view RGB images. Weng et al. [43] further incorporate depth information to support richer geometry priors. However, they rely on densely aligned inputs and known part counts, limiting their applicability in real-world settings. In contrast, our approach only conditioned a pair of images, which reduces input complexity while preserving articulation fidelity.

Generative approaches aim to synthesize articulated objects from compact inputs, bypassing the need for dense observations. NAP [18] parses layouts and articulation parameters into graphs and generates articulated 3D objects unconditionally. CAGE [24] achieves a controllable generation from the given articulation graph. Despite these models support efficient sampling, they lack explicit visual guidance to achieve more accurate controllability. URDFormer [6] solves this issue by combining a visual detector [25, 44] to extract spatial layout and a transformer to predict articulation parameters. SINGAPO [23] proposes a diffusion model [12, 37, 35, 30] conditioned on resting state images to generate articulated objects. However, the controllability of current approaches remains limited due to the absence of explicit articulation dynamics. The proposed DIPO effectively addresses this limitation by utilizing the motion information provided by a pair of images captured in the resting and articulated states.

2.2 Synthetic Articulated Object Datasets

The availability of large-scale 3D datasets with part-level structures has significantly facilitated research on articulated object modeling. Early datasets such as those used in [13, 47] are constructed by manually segmenting shapes from ShapeNet [4] and SketchUp [38], and annotating articulation parameters for part pairs. Shape2Motion [40] expands the scale by introducing an annotation tool that supports visual verification through animation. PartNet-Mobility[46] is a large-scale articulated object dataset constructed on PartNet[27]. It offers annotations of part-level articulation along with high-quality rendered images, and is one of the most widely adopted benchmarks. GAPartNet [10] focuses on functional part detection across categories, emphasizing generalizable and actionable parts such as buttons and handles. These datasets have enabled the development of deep learning models for articulation analysis, but are still limited in structural complexity and diversity. To improve articulation diversity and realism, ACD [14] collects complex articulated objects from ABO [7], 3D-Future [8] and HSSD [15]. While the articulation structures in ACD are more intricate, the scale of dataset remains limited. To address both diversity and scalability limitations, we present PM-X, a large-scale, URDF-compatible dataset of procedurally generated articulated objects with high structural complexity.

3 Generate Articulated Objects from Dual-Image Pairs

3.1 Overview

We propose a diffusion network to generate all the parameters of articulated objects conditioned on a pair of dual-state images and a part-level connectivity graph. The overall architecture is illustrated in Figure 2. To support this generation process, we parameterize each part in terms of its spatial location, articulation connectivity, and semantic attributes. The i-th part $\mathbf{p}_i$ is represented by the

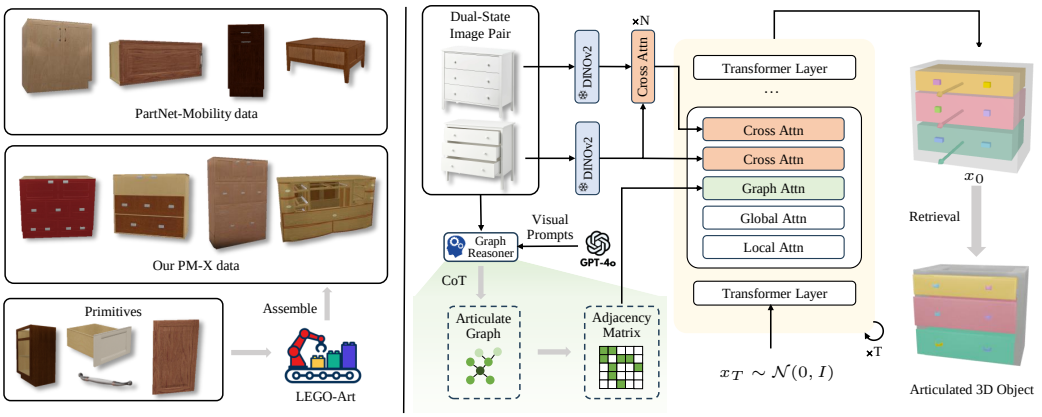


Figure 2: Overview of the proposed DIPO framework. The left part shows the proposed LEGO-Art pipeline assembles the primitives in existing dataset and construct the PM-X dataset, which are more diverse and complex compared to PM dataset. The right part shows that our diffusion model equipped with CoT-based Graph Reasoner for articulate graph inference, and conditioned on resting & articulated image pairs to generate articulated objects.

bounding box coordinates $\mathbf{b}_i \in \mathbb{R}^6$, semantic label l_i, articulation type t_i, joint axis $\mathbf{a}_i \in \mathbb{R}^6$, and motion range $\mathbf{r}_i \in \mathbb{R}^2$. To facilitate unified processing, all attributes are repeated to a 6-dimensional array, resulting in a 5×6 matrix representation for each part.

3.2 Dual-State Image Conditioning

We condition the denoising process on both resting-state and articulated-state images to capture motion-aware cues. Let $\mathcal{F}_R$ and $\mathcal{F}_A$ denote the DINOv2 [29] features from the resting and articulated images, respectively. To integrate these into the diffusion network, we apply a **Dual-State Injection Module** at each layer.

(a) Resting state (b) Articulated state

Figure 3: Dual-state visual prompt used by the *Graph Reasoner*. GPT-4o can produce realistic and structurally complex image pairs.

Given part embeddings X, we first perform cross-attention with resting-state features $\mathcal{F}_R$ to capture static appearance. We then guide articulated features $\mathcal{F}_A$ to attend to $\mathcal{F}_R$, and subsequently inject this context-enhanced signal into X. The overall conditioning update at each diffusion step is defined as:

$$X = X + \mathrm{CA}(X, \mathcal{F}_R) + \mathrm{CA}(X, \mathrm{CA}(\mathcal{F}_A, \mathcal{F}_R)), \tag{1}$$

where $\mathrm{CA}(Q, K)$ denotes a standard cross-attention operation that query Q attends to key-value source K. This design allows the model to generate more accurate part movement and joint behavior by contrasting the two input states.

3.3 Graph Reasoner via Chain-of-Thought Prompting

We introduce the **Graph Reasoner**, a Chain-of-Thought (CoT) based module that predicts the articulated part connectivity graph from dual-state images, serving as a structural prior for the diffusion process. The reasoning follows a step-by-step paradigm. It first identifies candidate parts and estimates their coarse spatial layout, then verifies whether the layout satisfies the given articulation rules, and finally infers attachment relationships to generate the articulation graph. After that, we convert the predicted articulation graph into an adjacency matrix, which serves as an attention mask to guide the self-attention of the diffusion model along valid structural connections.

In addition, we leverage the instruction-following and visual-editing capabilities of GPT-4o to generate dual-state image pairs of structurally diverse objects as Figure 3 shows. These results

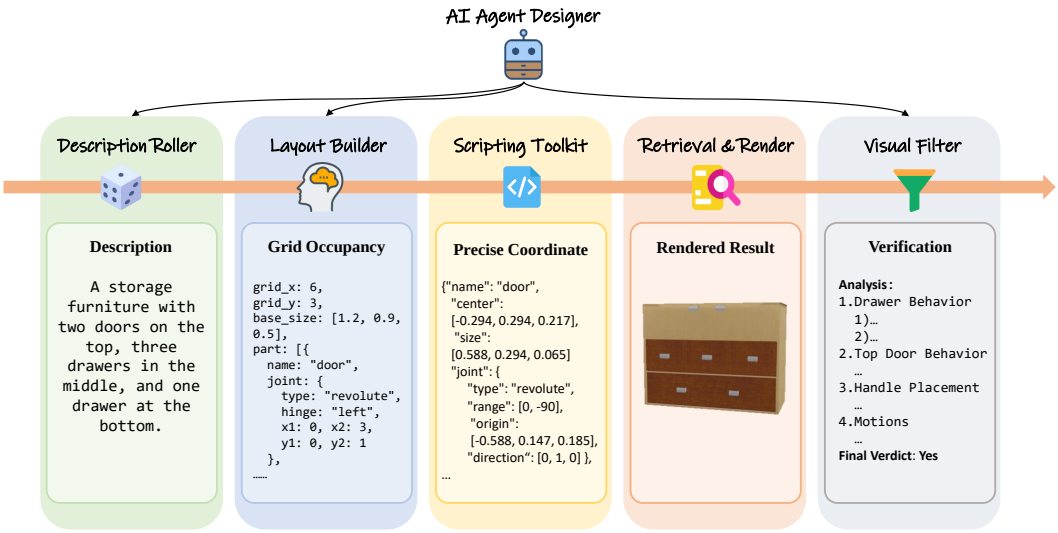


Figure 4: An overview of the fully automated synthesis pipeline for the proposed PM-X dataset. The synthesis pipeline consists of five functional modules executed in sequence: (1) a description roller that uses an LLM to generate natural language descriptions for structured layout, (2) a layout builder to generate part-level grid occupancy and joint configurations, (3) a scripting toolkit to construct precise coordinate from the grid-based layout information, (4) a retrieval and render module to assemble geometry and render dual-state images, and (5) a visual filter that uses a VLM to validate the plausibility of generated samples. In particular, modules (1), (2), and (5) are automatically constructed and managed by the AI Agent Designer.

serve as strong example visual prompts for the Graph Reasoner to achieve a higher stability and generalization of graph prediction.

4 Construct Complex Data from Partnet-Mobility

4.1 LEGO-Art pipeline

To gain favorable performance on challenging data, we require a large-scale 3D dataset with diverse part layouts. However, existing datasets still fall short in complementary ways: PM [46] offers sufficient data but lacks articulation complexity, while ACD [14] includes more realistic kinematic structures but is limited in dataset scale.

To address this issue, we design a **L**anguage-driven **E**ngine via **G**rid **O**rganization for **Art**iculation objects construction (**LEGO-Art**). It is a fully automated synthesis pipeline that generates complex articulated 3D assets by assembling part primitives from existing dataset. Figure 4 shows the overall workflow of the synthesis pipeline. The details of each step are illustrated below.

- **Description Roller.** The pipeline begins by generating a natural language description of an articulated object by a LLM agent (e.g., "a storage furniture with two doors on top, three drawers in the middle, and one drawer at the bottom"). This serves as a high-level blueprint for the object's structure without requiring precise geometry.
- **Layout Builder.** Given this textual input, the second agent translates the description into a part layout and articulation configuration. Instead of predicting exact 3D coordinates, which often introduces hallucination, we discretize the space into a grid and assign parts to grid cells. Each part is associated with joint metadata such as type, axis, and motion direction.
- **Scripting Toolkit.** We develop a scripting toolkit that converts the grid-level spatial layout into precise 3D coordinates and assigns articulation parameters of the axis and direction of joint, motion range and joint type.
- **Retrieval & Render.** We assign geometry to each part by retrieving mesh primitives from PartNet-Mobility by the algorithm proposed by [24]. Parts are scaled and positioned

according to the layout, and connected as specified by the URDF. Then, we render a resting and articulated state image pair of each object by `BLENDER`.

- **Visual Filter.** To ensure data quality, we include a final filtering step. We use a VLM to assess whether each rendered object plausibly matches its description and articulates correctly. Only assets that pass this check are included in our final dataset, PM-X.
- **AI Agent Designer.** To simplify the development of the above components, we adopted a prompt-based agent design process. Specifically, we described our intended system behavior in natural language and used an LLM to co-design the system prompts for the Description Roller, Layout Builder, and Visual Filter agents.

The proposed LEGO-Art enables scalable generation of physically valid, semantically rich, and structurally diverse articulated assets with minimal human effort, and plays an essential role in enabling our DIPO to generalize to more challenging dataset.

4.2 PM-X Dataset

Based on the LEGO-Art, we build a large-scale dataset from the part primitives of the PartNet-Mobility dataset, named **PM-X**. PM-X consists of 600 automatically generated structural-complex articulated objects. For every object, we futher provide correspondence rendered images, URDF files, and natural language descriptions. Due to the experiments settings, we only consider `StorageFurniture` and `Table` objects in the proposed dataset. However, the synthesis pipeline can be extended to a wider category of articulated objects, and the overall dataset size can also be scaled up. Compared to existing datasets, PM-X offers not only significantly greater structural complexity and articulation diversity, but also sufficient scale to serve as a standalone training set for generative models. These characteristics make it particularly effective for improving generalization and robustness in articulated object generation tasks, especially under out-of-distribution settings. Our experiments also demonstrate the superiority of the PM-X dataset. Table 1 illustrates that the PM-X dataset surpasses previous datasets in both object quantity and average part count, highlighting its scalability and structural richness.

Table 1: Comparison of dataset scale and part complexity.

Dataset	# Objects	Avg. # Parts
PM [46]	570	4.94
ACD [14]	135	7.48
PM-X (Ours)	600	19.40

5 Experiments

5.1 Implementation Details

We follow the dataset split way of SINGAPO [23] to build the training and testing set. Specifically, the training set is made up of 493 articulated objects from the PM [46] dataset, combined with 600 samples from our proposed PM-X dataset. Each object is rendered by `BLENDER_EEVEE_NEXT` engine to produce dual-state image pairs from 20 random views. We further introduce a complex data augmentation to enhance the performance of the model, which is detailed in the supplementary materials. For evaluation, we use 77 held-out objects from PM, each rendered from two random views, resulting in 144 dual-state test samples. In addition, we include 135 objects from the ACD dataset [14] to further assess the generalizability of the model to out-of-distribution data.

To accelerate convergence, we initialize our model with the pretrained weights from CAGE [24]. We train our model for 200 epochs with a batch size of 20. The model is optimized by AdamW [26] with $\beta = (0.9, 0.99)$ The learning rate is set to 5×10^{-4} for the image-conditioned module and 5×10^{-5} for the base model. All experiments are conducted on 8 NVIDIA 4090 GPUs.

5.2 Comparisons

5.2.1 Baselines & Metrics

Three representative methods, which are URDFormer [6], NAP [18], and SINGAPO [23], are selected as comparison baselines. Specifically, we finetune the pre-trained URDFormer and retrain the SINGAPO for a fair comparison. For NAP, we follow the experiment setting of SINGAPO that

Table 2: Comparison of reconstruction quality and graph prediction accuracy on **PartNet-Mobility** test set. Lower is better (↓) except for Acc% (↑).

	Reconstruction quality						Graph
	RS-d_{gIoU} ↓	AS-d_{gIoU} ↓	RS-d_{cDist} ↓	AS-d_{cDist} ↓	RS-d_{CD} ↓	AS-d_{CD} ↓	Acc% ↑
URDFormer [6]	1.2327	1.2332	0.2885	0.4403	0.4417	0.6910	6.62
NAP-ICA [18]	0.5706	0.5765	0.0563	0.2547	0.0209	0.3473	25.06
SINGAPO [23]	0.5134	0.5236	0.0487	0.1107	0.0191	0.1270	75.97
DIPO(Ours)	**0.4561**	**0.4683**	**0.0359**	**0.0732**	**0.0132**	**0.0423**	**85.06**

Table 3: Comparison of reconstruction quality and graph prediction accuracy on **ACD** test set. Lower is better (↓) except for Acc% (↑).

	Reconstruction quality						Graph
	RS-d_{gIoU} ↓	AS-d_{gIoU} ↓	RS-d_{cDist} ↓	AS-d_{cDist} ↓	RS-d_{CD} ↓	AS-d_{CD} ↓	Acc% ↑
URDFormer [6]	1.1074	1.1094	0.2868	0.3948	0.6229	0.7608	1.52
NAP-ICA [18]	0.9955	1.0000	0.1713	0.3246	0.1141	0.3061	8.27
SINGAPO [23]	0.9700	0.9728	0.1582	0.2057	0.1047	0.1762	36.67
DIPO (Ours)	**0.9126**	**0.9151**	**0.1253**	**0.1541**	**0.0751**	**0.1085**	**48.15**

insert an image cross attention block into each layer to achieve controllable generation of images, marked as NAP-ICA.

To evaluate reconstruction quality and articulation correctness, we adopt four metrics: (1) d_{gIoU} ↓, the generalized IoU between predicted and ground-truth part bounding boxes; (2) d_{cDist} ↓, the Euclidean distance between part centers; (3) d_{CD} ↓, the Chamfer Distance [2] between predicted and ground-truth meshes; and (4) Acc ↑, the graph prediction accuracy. All metrics are computed over both resting and articulated states. For clarity, we prefix the metric names with `RS-` and `AS-` in the tables to indicate the evaluation state.

5.2.2 Quantitative Comparison

We report quantitative results on the PM and ACD datasets in Table 2 and Table 3, respectively. To reduce the impact of stochastic variation, we evaluate all diffusion-based generative methods five times per test sample and report the averaged metric values.

As shown in Table 2, our method DIPO achieves the best performance in terms of reconstruction quality and accuracy of articulate graph on the PartNet-Mobility test set. Importantly, we observe that the performance drop from RS (rest ing state) to AS (articulated state) is significantly smaller for our method than for all others. It indicates that dual-image conditioning provides effective control signals that help the model maintain accurate articulation predictions.

On the ACD test set (Table 3), which contains more diverse and realistic articulated objects, our method continues to outperform all baselines. DIPO shows consistently superior reconstruction accuracy in both states and delivers the best graph prediction accuracy. The evaluation results on ACD dataset demonstrate that our method performs well on out-of-distribution data.

The above results demonstrate that the proposed DIPO achieves superior quantitative performance with both high accuracy and strong generalization across structurally diverse datasets.

5.2.3 Qualitative Comparison

Figure 5 provides a qualitative comparison between our method and two strong baselines, NAP-ICA [18] and SINGAPO [23]. Each example includes: (1) the input dual-state image pair (closed and open), (2) the predicted articulation graph, (3) the reconstructed part layout and joints in resting state, and (4) the final articulated geometry. The examples cover a wide spectrum of scenarios, including synthetic data from PM and ACD datasets. In addition, the last three rows are real-world examples: we either collect resting-state images from the Internet or directly capture image pairs of nearby objects in both states. This provides a more realistic evaluation of generalization beyond existing datasets. For Internet-collected examples that only provide resting-state images, we employ GPT-4o to generate the articulated counterparts, showcasing the flexibility of our method.

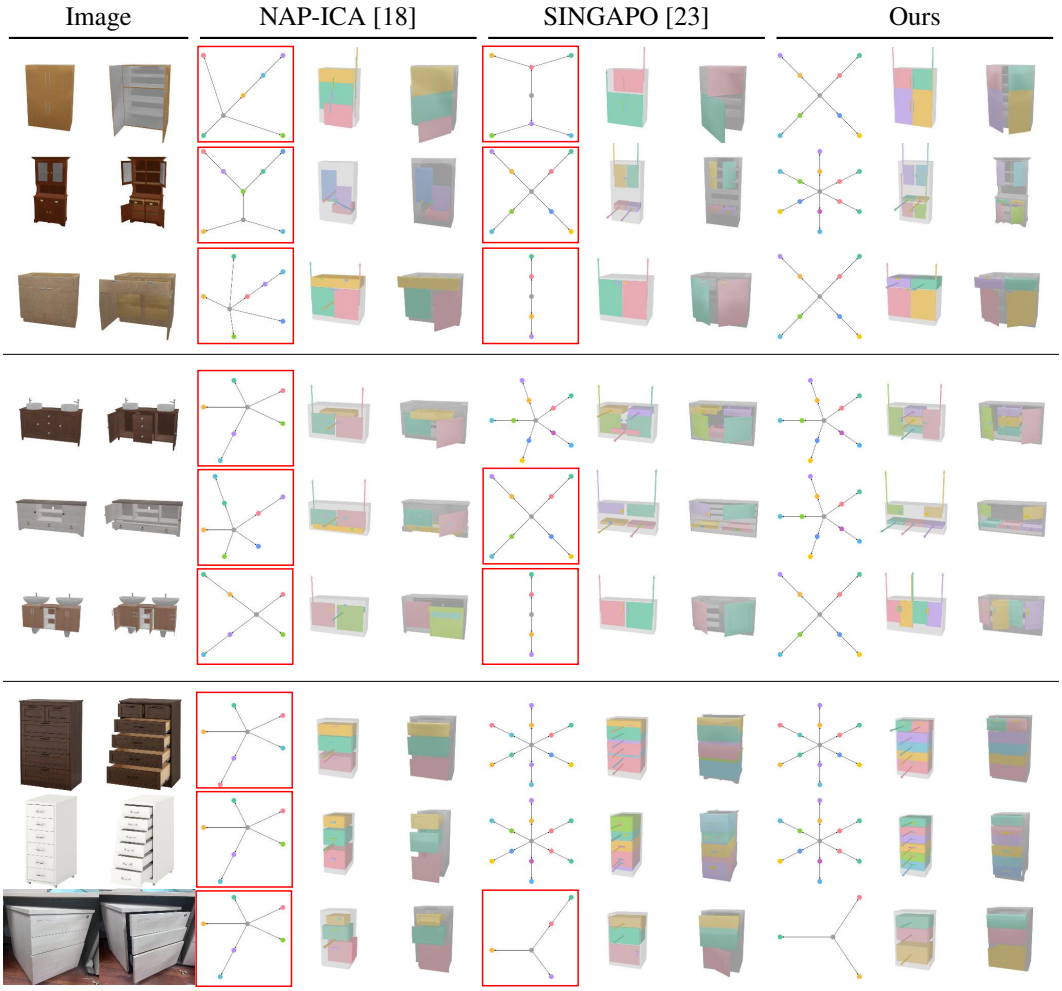


Figure 5: Visual comparison between the proposed DIPO and two baselines. The fist two columns show the dual-state image pairs. The precdiction results of articulate graph, the part layout and joint visualization in resting state, and the final geometry in articulated state are also illustrated. The first three rows are sampled from the PM dataset, the middle three rows are from the ACD dataset, and the last three rows are real-world images. Incorrect parts connections are marked with red box.

Compared to baselines, our method DIPO demonstrates superior visual quality and better accuracy of articulation graph prediction. Thanks to the large-scale structurally diverse training provided by the PM-X dataset, our method shows better robustness when handling complex objects or real-world data. Moreover, cases in which parts are densely arranged and exhibit highly similar textures often confuse single-image baselines, resulting in incorrect articulation inference. In contrast, our method leverages the contrastive cues between resting and articulated states to recognize part boundaries, joint connectivity, and part motions more accurately.

These qualitative results strongly support the effectiveness and generalization ability of the proposed DIPO.

5.3 Ablation Study

We conduct detailed ablation studies to verify the effectiveness of each key component in our framework, including the PM-X dataset, Dual-state Injection Module (DIM), and Graph Reasoner (GR). We construct several variants by selectively altering these components. The quantitative results are summarized in Table 5. In addition, we further analyze the settings of each component in isolation in the following paragraphs.

Table 5: Ablative results of reconstruction quality and graph prediction accuracy on **ACD** test set. Lower is better (↓) except for Acc% (↑).

Settings			Reconstruction quality					
PM-X	DIM	GR	RS-d_{gIoU} ↓	AS-d_{gIoU} ↓	RS-d_{cDist} ↓	AS-d_{cDist} ↓	RS-d_{CD} ↓	AS-d_{CD} ↓
			0.9872	0.9900	0.1608	0.2096	0.1083	0.1792
✓			0.9429	0.9464	0.1389	0.1868	0.0849	0.1538
	✓		0.9565	0.9589	0.1478	0.1819	0.0924	0.1407
		✓	0.9902	0.9931	0.1697	0.2157	0.1208	0.1881
✓	✓		0.9212	0.9233	0.1257	0.1589	0.0752	0.1200
✓		✓	0.9332	0.9368	0.1391	0.1843	0.0844	0.1439
	✓	✓	0.9497	0.9515	0.1500	0.1786	0.0973	0.1317
✓	✓	✓	0.9126	0.9151	0.1253	0.1541	0.0751	0.1085

Impact of PM-X dataset. Table 5 shows that across various settings of ablative experiments, incorporating the PM-X dataset consistently improves reconstruction quality, indicating its broad effectiveness. To further validate this effect, we additionally experiment with using only 25% and 50% of the PM-X data. As shown in Figure 6, IoU scores for both resting and articulated states degrade steadily as the PM-X ratio increases, confirming the importance of PM-X in enhancing structural accuracy and generalization.

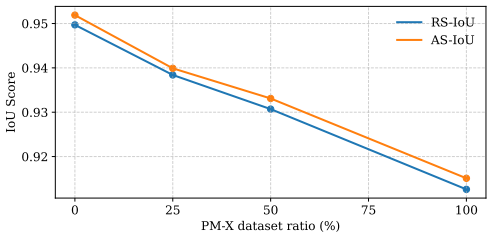


Figure 6: Ablative comparison under different ratios of PM-X data

Effectiveness of Dual-Image Input. We conduct ablation experiments to assess the contribution of the DIM module. As shown in Table 5, adding DIM significantly improves performance across all reconstruction metrics. The effectiveness of DIM is further reflected in Figure 1, 5, where our method accurately identifies the motion direction according to articulated images. It demonstrates that the dual-image design not only enhances articulation prediction, but also imposes the ability of structural reasoning to the model.

Analysis of Graph Reasoner As illustrated in Table 5, the GR module can not consistently improve performance across all settings. This is because while GR enables more accurate prediction, it also tends to produce more complex topologies. For model variants not trained on the PM-X dataset, such complex graphs may become out-of-distribution, leading to suboptimal perform However, when the model is trained with the structurally diverse PM-X dataset, the benefits of GR become more apparent. Moreover, we conduct more detail ablative experiments to verify the effectiveness of each component of GR. The results of prediction accuracy can be seen in Table 4.

Table 4: Ablative results of Graph Reasoner.

Settings	**Acc% ↑**
w/o CoT	39.26
w/o Visual Input	37.77
w/o dual-state input	39.63
Full Model (GR)	48.15

6 Conclusion

We propose DIPO, a framework that advances vision-conditioned articulated object generation under challenging data. We design a diffusion model conditioned on resting and articulated image pairs for articulated 3D object generation, which provides richer part motion information and leads to improved reconstruction accuracy. A Chain-of-Thought graph reasoner is further introduced to enhance part connectivity prediction. In addition, we develop LEGO-Art, an automated pipeline for constructing diverse and complex articulated objects, and contribute PM-X, a large-scale dataset built by the proposed pipeline. Powered by PM-X, our model achieves superior performance and stronger

generalization. Extensive experiments validate the effectiveness of each component and the overall advantage of our approach over existing methods.

Acknowledgments and Disclosure of Funding

Shenzhen Science and Technology Program (JCYJ20240813114237048) “Science and Technology Yongjiang 2035” key technology breakthrough plan project (2024Z120)Chinese government-guided local science and technology development fund projects (scientific and technological achievement transfer and transformation projects) (254Z0102G)

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Harry G Barrow, Jay M Tenenbaum, Robert C Bolles, and Helen C Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings: Image Understanding Workshop*, pages 21–27. Science Applications, Inc, 1977.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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

[5] Xu Chen, Tianjian Jiang, Jie Song, Max Rietmann, Andreas Geiger, Michael J Black, and Otmar Hilliges. Fast-snarf: A fast deformer for articulated neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):11796–11809, 2023.

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

[7] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21126–21136, 2022.

[8] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129:3313–3337, 2021.

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

[10] 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*, pages 7081–7091, 2023.

[11] Karol Hausman, Scott Niekum, Sarah Osentoski, and Gaurav S Sukhatme. Active articulation model estimation through interactive perception. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3305–3312. IEEE, 2015.

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[13] Ruizhen Hu, Wenchao Li, Oliver Van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Learning to predict part mobility from a single static snapshot. *ACM Transactions On Graphics (TOG)*, 36(6):1–13, 2017.

[14] Denys Iliash, Hanxiao Jiang, Yiming Zhang, Manolis Savva, and Angel X Chang. S2o: Static to openable enhancement for articulated 3d objects. *arXiv preprint arXiv:2409.18896*, 2024.

[15] Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16384–16393, 2024.

[16] Jeonghwan Kim, Jisoo Kim, Jeonghyeon Na, and Hanbyul Joo. Parahome: Parameterizing everyday home activities towards 3d generative modeling of human-object interactions. *arXiv preprint arXiv:2401.10232*, 2024.

[17] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[18] Jiahui Lei, Congyue Deng, William B Shen, Leonidas J Guibas, and Kostas Daniilidis. Nap: Neural 3d articulated object prior. *Advances in Neural Information Processing Systems*, 36:31878–31894, 2023.

[19] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. *arXiv preprint arXiv:2108.03272*, 2021.

[20] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR, 2023.

[21] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19711–19722, 2024.

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

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

[24] Jiayi Liu, Hou In Ivan Tam, Ali Mahdavi-Amiri, and Manolis Savva. Cage: controllable articulation generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17880–17889, 2024.

[25] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024.

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[27] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019.

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

[29] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[30] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[31] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023.

[32] Shengyi Qian and David F Fouhey. Understanding 3d object interaction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21753–21763, 2023.

[33] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5020–5030, 2024.

[34] Morgan Quigley, Brian Gerkey, and William D Smart. *Programming Robots with ROS: a practical introduction to the Robot Operating System*. " O'Reilly Media, Inc.", 2015.

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

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

[37] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[38] Trimble Inc. 3d warehouse, 2025. Accessed: 2025-05-14.

[39] Wei-Cheng Tseng, Hung-Ju Liao, Lin Yen-Chen, and Min Sun. Cla-nerf: Category-level articulated neural radiance field. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8454–8460. IEEE, 2022.

[40] 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*, pages 8876–8884, 2019.

[41] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[42] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Photo wake-up: 3d character animation from a single photo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5908–5917, 2019.

[43] Yijia Weng, Bowen Wen, Jonathan Tremblay, Valts Blukis, Dieter Fox, Leonidas 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*, pages 3141–3150, 2024.

[44] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022.

[45] Ruiqi Wu, Liangyu Chen, Tong Yang, Chunle Guo, Chongyi Li, and Xiangyu Zhang. Lamp: Learn a motion pattern for few-shot video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7089–7098, 2024.

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

[47] Zihao Yan, Ruizhen Hu, Xingguang Yan, Luanmin Chen, Oliver Van Kaick, Hao Zhang, and Hui Huang. Rpm-net: recurrent prediction of motion and parts from point cloud. *arXiv preprint arXiv:2006.14865*, 2020.

[48] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2863–2873, 2022.

[49] Ji Yang, Xinxin Zuo, Sen Wang, Zhenbo Yu, Xingyu Li, Bingbing Ni, Minglun Gong, and Li Cheng. Object wake-up: 3d object rigging from a single image. In *European Conference on Computer Vision*, pages 311–327. Springer, 2022.

[50] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

[51] Xuying Zhang, Yupeng Zhou, Kai Wang, Yikai Wang, Zhen Li, Shaohui Jiao, Daquan Zhou, Qibin Hou, and Ming-Ming Cheng. Ar-1-to-3: Single image to consistent 3d object generation via next-view prediction. *arXiv preprint arXiv:2503.12929*, 2025.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: We claim that the main contribution are a dual-state images conditioned model to generate articulated objects, and a pipeline named LEGO-Art construct a large-scale dataset named PM-X which contains diverse complex articulated objects.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

Justification: We discussion our limitations in supplementary due to page limit.

Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We detailed the details of the proposed network and the workflow of dataset construction.

   Guidelines:

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

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

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

   Answer: [Yes]

   Justification: We provide the code of our method and some data samples in supplemental material.

   Guidelines:

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

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide all details of our experiments setting.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We provide detailed quantitative results in our experiments.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide the the computer resources we used (8×4090 GPUs).

   Guidelines:

   - The answer NA means that the paper does not include experiments.

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Yes, the research fully conforms to the NeurIPS Code of Ethics in all respects. No ethical concerns have been identified.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Broder impacts are illustrated in supplemental material.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: Yes, all third-party assets used in the paper are properly credited. Their licenses and terms of use are clearly stated and fully respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [Yes]

Justification: Yes, all newly introduced assets are well documented. The dataset, model, and pipeline will be released with detailed usage instructions and annotations alongside the assets.

Guidelines:

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

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

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

Answer: [NA]

Justification: There are no crowdsourcing experiments and research with human subjects in our paper.

Guidelines:

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

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

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

Answer: [NA]

Justification: There are no research with human subjects in our paper.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: LLMs play an important, original role in LEGO-Art pipeline and Graph Reasoner. Moreover, Multi-Modal LLM can provide generated image pairs for a more flexible input way to our model.

Guidelines:

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

Abstract

Our supplementary materials give more details of the proposed DIPO and the experimental settings, which can be summarized as follows:

- The details of LEGO-Art and Graph Reasoner.
- More visual results of the proposed PM-X dataset.
- The details of data augmentation.
- The code and checkpoint of DIPO for inference.
- A video shows some animated visual examples of complex objects.

A Details of LEGO-Art Pipeline

In our LEGO-Art pipeline, we design a modular LLM-based [3, 1] framework, where each agent specializes in a distinct subtask. These agents collaborate to generate structured, diverse, and physically plausible articulated object layouts. Below, we detail the system prompt of each agent.

Description Roller:

You are an expert in generating clear and realistic natural language descriptions of articulated object structures.
Each object must belong to one of the following categories: **['Storage Furniture', 'Table', 'Refrigerator', 'Dishwasher', 'Oven', 'Washer']**.

Your task is to imagine a plausible object structure from one of these categories and describe only its articulated parts in natural language.
The available part types are: **['base', 'door', 'drawer', 'tray', 'handle', 'knob']**. **Note:** "tray" parts are **only** allowed if the object is a microwave.

Each object must contain exactly one implicit **"base"** part, and any number of other parts, depending on the category.

You will be provided with a complexity level:

- **simple:** 1–5 parts, minimal structure.
- **mid:** 6–10 parts, basic spatial layout.
- **complex:** 11 or more parts with more detailed or hierarchical arrangements.

Your output must:

- Only describe the structure of the object: what parts it has, how many of each, and where they are located (e.g., left, right, middle, top, bottom, inside).
- Use precise but simple language in a single sentence.
- **Exclude** any mention of color, texture, material, appearance, or any decorative details.
- Ensure the description is consistent with both the object category and the specified complexity level.

Example output: *"A storage furniture with two doors in the middle, one drawer at the bottom, and four drawers on the left and right sides."*
Important: Do not include 3D coordinates or structured data. Only output the structural description in plain English.

Figure 7: The system prompt of Descrition Roller.

Layout Builder:

Example Question 1:
The following code is a function that generates a layout from a given object in a grid format.

```
def sample_base(grid_x, grid_y, base_size):
    # generate the base, and return a coordinate list of grids
    ...

def generate_part_in_grid(base, grid_coords, x1, x2, y1, y2,...):
    # generate coordinates and articulation info of a part
    ...

def generate_layout(info):
    # convert grid-level layout into coordinates and articulation parameters
    base, grid_coords = sample_base(...)
    articulate_tree = [base]

    for part in info['part']:
        part = generate_part_in_grid(...)
        articulate_tree.append(part)

        ...
```

Example Answer 1:
You've developed a complete pipeline for procedurally generating articulated object layouts and rendering them visually. The system includes ...
Example Question 2:
I need you to generate the info in a python dict from a natural language description. The dict is the only python code in your output. Note that all [x1, x2, y1, y2] should be an integer. The name of the part can only be one of the [drawer, door, handle, knob] (strictly!)
Example Answer 2:
Got it! You want to input a natural language description like:
"A wide cabinet, approximately 1.5×1.0×0.5 meters in size, contains two left-hinged doors, each with one handle, and two drawers, each with two handles"
and have it automatically generate a structured info dictionary as:

```
# python dict
{
    ...
}
```

Figure 8: The system prompt of Layout Builder. This agent is inspired by the code of scripting toolkit and produce a python dict that contains the information of parts layout in grid-level.

Visual Filter:

You are an expert in **3D object structure verification**.
You will be shown a pair of rendered images of a 3D articulated object: one in the **closed state**, and one in the **open state**. These images are generated based on a predicted structure and joint configuration.

Your task is to determine whether the observed articulation behavior is **physically plausible and logically consistent**. That is, check if the object's opening and closing behavior matches how real-world articulated objects work.

You must analyze whether:

- The joints behave correctly (e.g., drawers slide outward, doors rotate from hinges).
- Each handle or knob is correctly positioned and attached to a moving part.
- There are no unreasonable collisions, floating parts, or detached motion.
- The motion (from closed to open) is consistent with the structure and joint types.

Final Output: After your analysis, respond with exactly one of the following:

- **Yes** — if the object's motion and structure are physically and functionally plausible.
- **No** — if there are any structural, physical, or semantic inconsistencies.

Figure 9: The system prompt of Visual Filter.

B Details of Graph Reasoner

The proposed Graph Reasoner can infer articulated connectivity from a dual-state image pair based on chain-of-thought [41, 17] prompt, which is illustrated as followed:

Graph Reasoner:

You are an expert in the **recognition**, **structural parsing**, and **physical-feasibility validation** of articulated objects from image inputs.
You will be provided with two rendered images of the same object:

1. A **closed-state image** (all movable parts in their fully closed positions)
2. An **open-state image** (all movable parts in their fully opened positions)

Your task is to analyze the object's articulated structure and generate a **connectivity graph** describing the part relationships.

Workflow:

1. **Part Detection**
   - Detect candidate parts in the **closed-state image**, optionally using the open-state image to resolve ambiguity or occlusion.
   - Allowed part types: `['base', 'door', 'drawer', 'handle', 'knob', 'tray']`
   - Ignore small decorative elements attached directly to the base.
   - There must be exactly one `"base"`; `"tray"` is only allowed for microwaves (but not required).
2. **Step-by-Step Reasoning**
   (a) **Part Listing**: List all detected parts and their counts (no attachment inference yet).
   (b) **Validation**: Enforce structural rules:
      - Exactly one base
      - Each door or drawer may have at most two handles or knobs
      - Every handle/knob must be attached to a door or drawer
      - Trays may only appear in microwaves

   (c) **Attachment Inference**: For each non-base part, infer its parent (e.g., `"drawer_1 (attached to base)"`). Use the open-state image if necessary.
   (d) **Connectivity Graph Construction**: Output a JSON tree where `"base"` is the root and all other parts are children with proper hierarchy.

   Example Output:

```
{
  "base": [
    { "door": [ { "handle": [] } ] },
    { "drawer": [ { "handle": [] } ] }
  ]
}
```

Final Output: You **MUST** output a single JSON tree representing the part connectivity of the object. Use the open-state image to enhance accuracy and completeness, but base your interpretation primarily on the closed-state image.

Figure 10: The system prompt of Graph Reasoner.

Moreover, we use GPT-4o [3, 1] to generate dual-state image pairs, which are example visual prompts to make the Graph Reasoner learn how to generate articulated graph in a few-shot manner. The generated image pairs are shown in Figure 11.

Figure 11: Each pair shows a closed-state image (left) and an open-state image (right) of an articulated object generated by GPT-4o.

C More Visual Examples of PM-X

The proposed PM-X dataset provides a large amount of diverse and structurally complex articulated objects. Figure 12 illustrates more visual examples of PM-X dataset. As we can see, each object has a reasonable structure and a rich set of operable parts. In addition, we annotate the description generated by the first stage of LEGO-Art. Objects precisely match the natural language descriptions, enabling LEGO-Art to further serve as a pipeline for text-to-articulated-object generation.

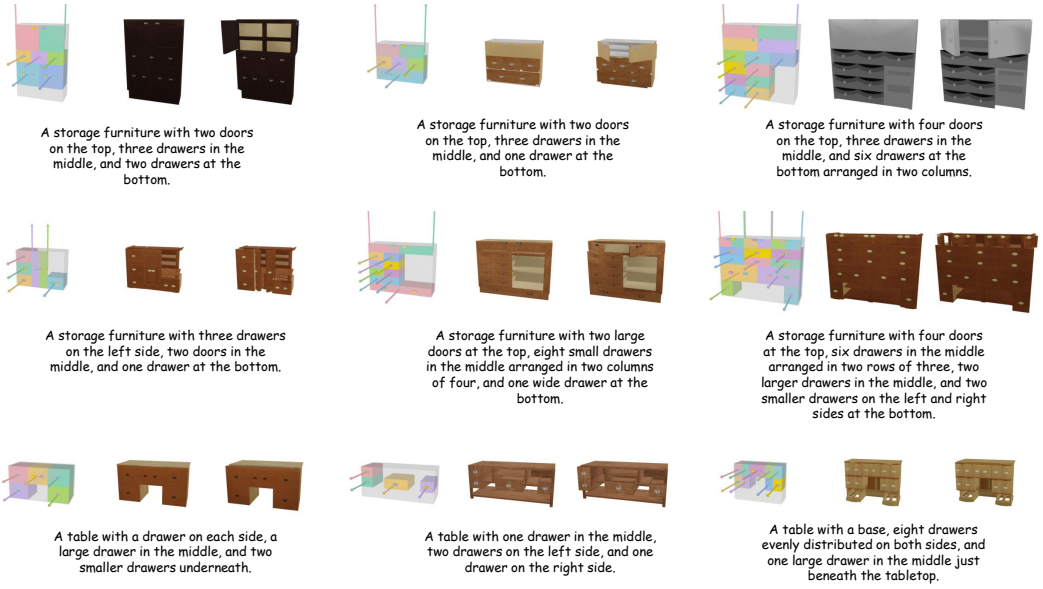


Figure 12: More visual examples of PM-X dataset. Each example includes: (1) the part layout and joints in resting state, (2) a rendered image pair in dual-state, (3) nature language description generated in the fisrt stage of LEGO-Art.

D Data Augmentation

We employ several data augmentation during the training state to enhance the robustness and controllability. Data augmentation can be categorized into two types. One focuses on part-level augmentation:

1. Randomly replace small parts like handles and knobs with those of other objects, and perturb their positions.
2. Randomly rescale the whole object.
3. Rotate the whole object upside-down.
4. Stacking several objects together to build more complex objects.

The other one focus on joint-level augmentation:

1. Change the revolute joint into prismatic joint.

2. Randomly modify the direction of revolute joint.
3. Randomly fix the joint.

E Limitations & Future Work

We follow the experimental settings of SINGAPO [23] for a fair comparison. However, the benchmark used in SINGAPO only contains several categories, which especially focuses on cabinet-like objects. This limited object diversity may constrain the generalization ability of our model to other articulated categories, such as appliances, tools, or deformable structures. In future work, we plan to build a benchmark that cover a broader range of articulated object types, including both everyday household items and more complex mechanical systems. Our research primarily focuses on predicting more accurate part layouts and joint configurations. We adopt a retrieval-based approach to construct the final 3D objects. Incorporating 3D generation techniques to synthesize more precise and diverse part geometries represents a meaningful direction for future work.

F Broader Impact

Our work facilitates controllable generation of articulated 3D objects from dual-state images, enabling structured reasoning over part layout and connectivity. This contributes to downstream applications in embodied AI, virtual environment simulation, and robotics manipulation. By releasing a large-scale synthetic dataset and a modular pipeline, we aim to lower the barrier for research on articulated perception and generation. However, as with any generative framework, care must be taken to avoid misuse such as creating physically implausible or unsafe designs. Moreover, biases in the data distribution or articulation patterns may influence downstream decision-making, highlighting the need for interpretability and robustness in practical deployments.