CAGE: Controllable Articulation GEneration

Jiayi Liu Hou In Ivan Tam Ali Mahdavi-Amiri Manolis Savva
Simon Fraser University
3dlg-hcvc.github.io/cage

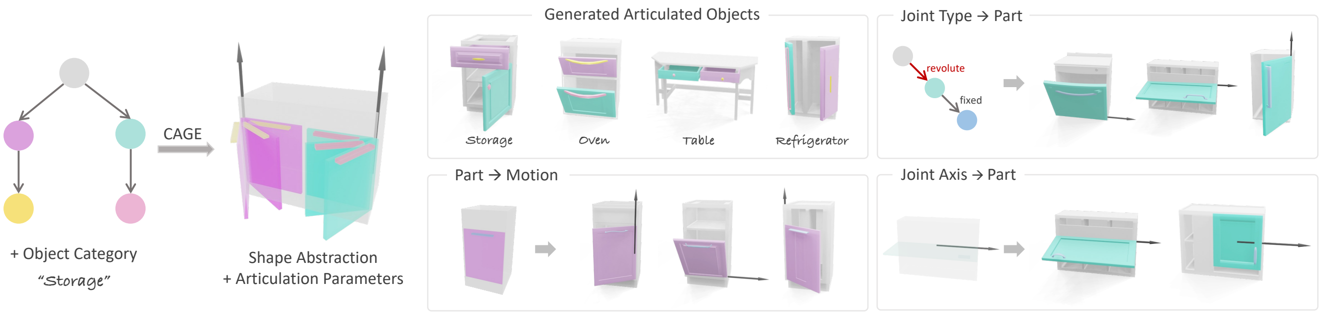

Figure 1. We present CAGE: a user-controllable generative model for 3D articulated objects. **Left**: given an object category label and a directed graph describing the interconnections among constituent parts, our model generates an abstraction of the articulated object specifying both geometry and motion parameters for each part. **Right**: the generated shape abstraction combined with appropriately constrained part retrieval allows for generating high-quality articulated objects under various user-specified constraints. Users can specify a desired object category, part shape, articulation type, or articulation axis and obtain generated objects that respect the provided constraints.

Abstract

We address the challenge of generating 3D articulated objects in a controllable fashion. Currently, modeling articulated 3D objects is either achieved through laborious manual authoring, or using methods from prior work that are hard to scale and control directly. We leverage the interplay between part shape, connectivity, and motion using a denoising diffusion-based method with attention modules designed to extract correlations between part attributes. Our method takes an object category label and a part connectivity graph as input and generates an object's geometry and motion parameters. The generated objects conform to user-specified constraints on the object category, part shape, and part articulation. Our experiments show that our method outperforms the state-of-the-art in articulated object generation, producing more realistic objects while conforming better to user constraints.

1. Introduction

Articulated objects are ubiquitous in real-world scenes. Kitchen cabinetry, refrigerators, storage drawers, and wardrobes are a few examples. Thus, digital 3D models of these objects are useful for a variety of tasks in robotics [4, 51], 3D vision [10, 22, 42], and embodied AI systems [38, 47]. Unfortunately, despite this clear value in many research areas, 3D models of articulated objects are predominantly authored manually and existing datasets of such models are few and relatively small [19, 26].

Unsurprisingly, there is much recent work on reconstructing articulated objects from real-world observations [9, 11, 18, 40] and on predicting how object parts can articulate for existing 3D object models [16, 44, 49]. However, these methods either rely on laborious acquisition of real-world data, or assume the existence of large datasets of 3D objects with sufficiently complete part geometry to enable articulation. These assumptions limit the scalability and practical utility of these approaches.

An alternative strategy recently proposed in NAP [13] is to learn a generative model for articulated 3D objects that can directly generate a complete articulated object. Although the NAP generative model conceptually supports conditional generation of articulated objects for partially specified inputs, as we will see it often fails to respect input constraints, and exhibits limited controllability. This limitation stands in stark contrast with the natural desire for user-driven control over the output of generative models, especially for highly structured and compositional output as is the case with articulated objects.

In this paper, we tackle the challenge of controllable articulated object generation. To address the limitations of prior work and enable fine-grained control for 3D articulated object generation, we take a directed graph and category label describing a desired object as input conditions. Since articulated objects are composed of hierarchies of rigidly moving parts organized in a kinematic chain it is natural and straightforward for designers to use an abstraction such as a graph when creating a new articulated 3D object. We then develop a denoising diffusion probabilistic model (DDPM) [8] based generative model for 3D articulated objects that: 1) disentangles graph structure and part attributes by representing parts as a set of bounding primitives with motion parameters associated with the input graph; 2) models a joint distribution of articulation and shape abstraction among parts; and 3) controls the generation using the input graph structure and object category, and additional conditions on desired part attributes.

One of our key insights is that lifting part geometry to a high-level shape abstraction is effective in succinctly capturing important shape-motion correlations. Our experiments show that this abstraction coupled with a series of appropriately designed attention modules improve joint modeling of parts and motion compared to prior work which attempts to capture detailed part geometry [13]. Another insight is that the position of key “actionable” parts (e.g., a door handle that is grasped to open the door) in relation to other parts provides a strong signal about likely motion patterns. For example, should a handle be positioned on the top-left corner of a door, the door is likely to rotate around an axis either on the right or bottom sides. We leverage this insight by incorporating actionable parts into the *articulation* graph to form an augmented *action* graph, with each actionable part connected to and influencing a parent part.

Our quantitative and qualitative evaluations compare our proposed method against ablations and baselines from prior work, and show that our method generates more realistic and more complex articulated 3D objects, exhibiting physically plausible variations. Our method also demonstrates better compatibility with various conditional input scenarios enabling better user-controlled generation. In summary:

- We present a generative model for articulated objects that learns a joint distribution over part shape and motion, under the constraints of graph structure and object category.
- We design a denoising network that enables strong conditioning through a series of attribute-attribute level attentions to inject user constraints effectively.
- Our evaluation with several proposed metrics shows that our method generates samples of higher quality and achieves better controllability compared to prior work.

2. Related Work

Generation of structured objects. There is prior work on structure-aware generative models for 3D objects that models part geometry. Wang et al. [43] use a GAN to generate semantically labeled voxel object parts and refine the shape with an auto-encoder. Similarly, Wu et al. [46] use a VAE to encode part geometry and pairwise part relations into a latent code and decode it to generate objects. These methods do not explicitly consider the part structure hierarchy. Li et al. [14] represent the part hierarchies as binary trees of symmetry hierarchy [45] and train a generative recursive autoencoder to encode parts and their geometry. Mo et al. [25] improve upon this work by removing the binary tree constraint and represent objects in n-ary graphs for robust generation of more complex objects. SDM-NET [5] synthesizes deformable meshes enabling more detailed object part generation. DSG-Net [50] disentangles part geometry and structure for fine-grained controllable generation.

Our work also addresses the challenge of modeling geometric relations between structured components in a controllable generative model. However, we also generate articulation parameters and thus tackle the additional challenge of generating physically plausible part motions.

Conditional diffusion for structured geometry. Diffusion models lack effective control over outputs from noise alone. Therefore, various techniques condition diffusion models on inputs such as text, sketches, low-resolution images, etc. SDEdit [23] injects a guide image into the noise, while Voynov et al. [41] use a sketch as a condition. ControlNet [52] takes inputs such as edge or depth maps and adjusts Stable Diffusion [33] weights without significant perturbation. Text-conditioned image and 3D shape generation is currently the most popular. Various methods [27, 32–34] leverage classifier-free guidance [7], text-image embeddings such as CLIP [31], or BERT [2]. DreamFusion [30] uses text-to-image diffusion and image-based guidance to generate 3D shape neural fields [24]. Magic3D [17] extends this approach to 3D meshes using neural marching tetrahedra [37]. Voxels [15, 35] and point clouds [28] are also employed for 3D object generation based on text prompts.

Unlike images and unstructured shapes, generation of structured 2D or 3D geometry such as articulated objects requires *part-to-part* relations to be captured in a representation such as a graph. Such relations are not modeled in image or shape representations such as NeRF, voxel grids, point clouds, or polygonal meshes. A common approach to overcome this challenge uses transformer-based diffusion and incorporates conditioning through attention mechanisms [29, 36, 39]. Our approach is inspired by this line of work, and focuses on the design of a multi-stage attention mechanism enabling fine-grained attention over part-to-part attributes and conditioning through user-provided constraints on the parts and their attributes.

Articulated 3D object modeling. Jiang et al. [11] introduced Ditto to build digital twins of articulated objects from point clouds. Heppert et al. [6] and Liu et al. [20] reconstruct articulated objects from a single stereo

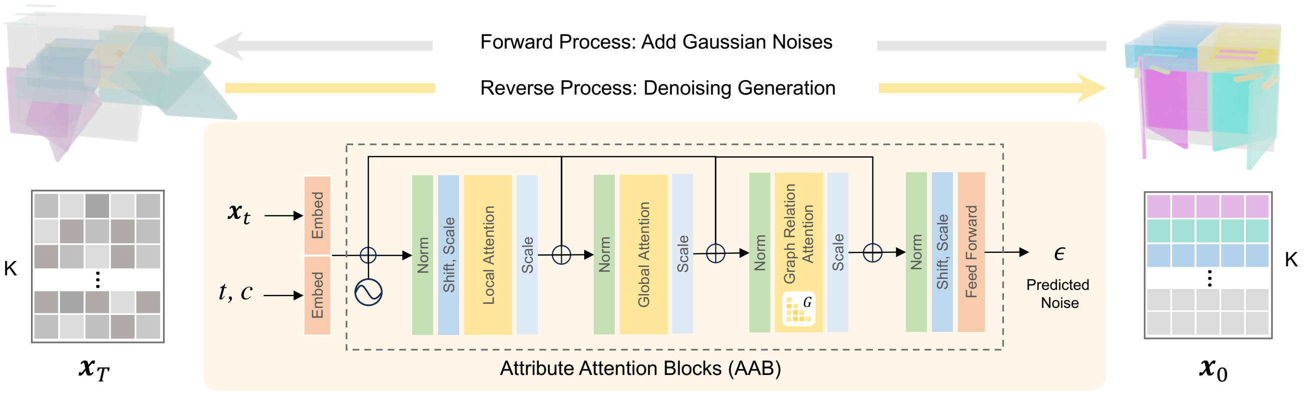

Figure 2. Method overview. Our generative model is based on DDPM [8]. In the forward pass, Gaussian noise is iteratively added to corrupt the data from $\mathbf{x}_0$ to random noise $\mathbf{x}_T$. During the reverse process, our denoiser (in yellow highlight) predicts the residual noise to be subtracted from the input data $\mathbf{x}_t$ at timestep t conditioned on the category label c and a graph adjacency G as an attention mask injected in the Graph Relation Attention module. All the timesteps share the same denoiser that is built on layers of our Attribute Attention Blocks.

RGB observation and point cloud videos, respectively. PARIS [18] simultaneously reconstructs part geometry and articulation parameters from multi-view images. However, these reconstruction-based methods require real-life observations, limiting their scalability and practicality.

Recently, Lei et al. [13] proposed NAP to tackle the task of 3D articulated object generation using a part relation tree formalism. This is similar to our approach, however NAP requires postprocessing to obtain a valid articulation tree and performs quite poorly in conditional generation, often disregarding input constraints. Moreover, NAP's fully connected graph has difficulty handling objects with a large number of parts, which limits its adaptability. Our work addresses these shortcomings and focuses specifically on a variety of conditional generation scenarios, which are particularly important in practical use.

3. Method

Given an object category label c and a graph structure G represented as an adjacency matrix, our objective is to generate an object within category c that is comprised of N parts that adhere to the graph. We learn the joint distribution of shape abstraction and articulation with a diffusion model using a series of attention modules to capture the interrelation among shape-motion attributes effectively. Figure 2 provides an overview of our generation pipeline and denoising network architecture.

3.1. Preliminaries

Diffusion models work by corrupting training data via successive addition of Gaussian noise in a forward process, and then learn to recover the original data via iterative denoising in a reverse process. Our work follows the original DDPM [8] formulation. In the forward pass, given a real articulated object $\mathbf{x}_0$ from an underlying distribution $q(\mathbf{x}_0)$, a series of noisy samples $\mathbf{x}_t$ is obtained by gradually adding Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}), \qquad (1)$$

where $t = 1, \ldots, N$ indicates the denoising step and α_t is determined by a noise variance scheduler. Practically, the noisy sample $\mathbf{x}_t$ is obtained by $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t$, where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. In the reverse process, the denoising model aims to predict the noise added in each step $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$.

Training loss. The training objective of a diffusion model is to minimize the negative log-likelihood of the data by maximizing the variational lower bound. Following DDPM [8], a simplified training objective is used as our training loss:

$$\mathcal{L} = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}_t}\left[\left\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t)\right\|^2\right] \qquad (2)$$

Building on this, we train our model to be conditioned on the graph structure G and object category label c.

3.2. Data Parameterization

The design of the data representation is crucial when working with diffusion models. Given a graph structure as a constraint for generation, our denoising target $\mathbf{x}_T$ is represented as a set of part parameters associated with the input graph (see Figure 3 top right). Each part is represented as a node with 5 node attributes. The attributes are aligned to be an M-dimensional vector by repeating if necessary. The node attributes include:

- **Part bounding box**: we canonicalize all objects to a "resting" state (i.e. doors and drawers closed). The canonicalization simplifies the problem such that each part bounding box can be assumed to be axis-aligned. We represent each axis-aligned bounding box with the 3D positions of the box maximum and minimum corners.

- **Joint type**: we categorize joints into 5 types as *fixed*, *revolute*, *prismatic*, *continuous*, or *screw* joint. The *fixed* joint is for non-rigid parts, *continuous* is a rotation-only joint without limits, while *revolute* is bounded at two ends. *Screw* joints exhibit both unbounded rotational motion and translational motion. We represent the joint type as a scalar which we expand to an M dimensional vector.
- **Joint axis**: each articulation is constrained by an axis. We represent each axis direction with a 3D unit vector and the position of the axis origin with a 3D vector.
- **Joint range**: we represent joint range with a 2D vector (left and right bounds), associated with the joint type. For continuous and revolute joints this indicates the rotation angle limits and for prismatic and screw joints this indicates the translation distance limits.
- **Semantic label**: Each part is associated with one of 8 semantic category labels (i.e., base, drawer, door, tray, shelf, knob, wheel, and handle), represented by a scalar which we expand to an M dimensional vector.

Formally, let $P = \{p_1, p_2, ..., p_N\}$ denote an articulated object with N parts to be generated. Each part p_i is represented as $p_i = \{a_{i,j} | a_{i,j} \in \mathbb{R}^M, 1 \leq j \leq 5\}$, where $a_{i,j}$ denotes the j^{th} attribute for part i. To generate objects with parts of variable length and leverage the diffusion model, we pad the nodes to a maximum number K. In summary, the denoising target is a vector of node attributes $\mathbf{x} = \{a_{i,j}\} \in \mathbb{R}^{5\times K\times M}$. We define K as 32 and M as 6.

3.3. Denoising Network

Our denoising network uses a series of Attribute Attention Blocks (AAB). The components in each AAB are shown in Figure 2. We design three attention modules interleaved with adaptive layer normalization to inject object category and graph adjacency as constraints in the conditional generation process. We discuss each component below.

Feature embedding. Each AAB takes a vector of node attributes $\mathbf{x}_t$ as input, with the timestep t, object category label c, and graph adjacency G as conditions. Each node attribute serves as a token in the attention modules after embedding. We embed the j^{th} attributes for part i ($a_{i,j}$) to a feature vector $\hat{a}_{i,j} \in \mathbb{R}^{128}$ along with two kinds of positional encoding: 1) indicates the attribute type, ranging from 1 to 5; 2) indicates the node it belongs to, ranging from 1 to K. The timestep t and category label c are embedded as a $128D$ feature vector through a linear layer.

Norm layer. Adaptive layer norm (adaLN) [48] has been adopted in GANs [1, 12] and diffusion models with a U-Net denoiser [3]. Recent conditional image generation work [29] showed that the adaLN-Zero variant improves condition injection by initializing each residual block as the identity function. Specifically, in addition to regressing scale and shift parameters, dimension-wise scaling parameters are regressed and applied immediately prior to residual connections. We follow this design in our attribute attention blocks. Once the timestep and category label are embedded in the feature vector as $\hat{t}$ and $\hat{c}$, we pass them into an additional embedding layer to learn all the scaling and shift parameters used between attention layers.

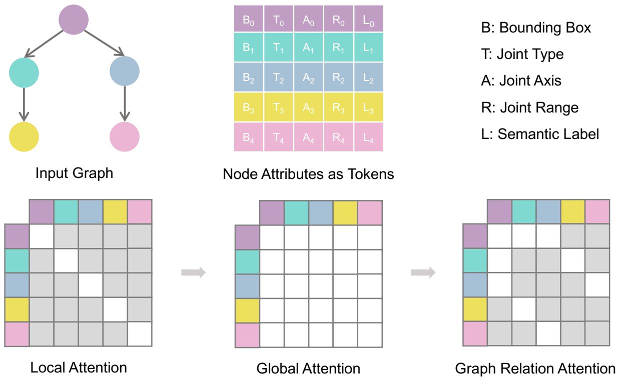

Figure 3. Design of the attention modules within our attribute attention blocks (AAB). Each node attribute is projected to a separate token and sequentially passed to three attention modules with varied masking strategies. White cells signify activated attention positions, whereas grey cells indicate attention that has been masked out. In graph relation attention, the activated cells represent the parent and child parts associated with each node.

Attention modules. Prior to being input into the attention layers, the attribute features $\hat{a}_{i,j}$, timestep features $\hat{t}$, and category features $\hat{c}$ are fused in the adaLN-Zero layer. This is when the condition of timestep and object category gets injected. The normalized and embedded tokens $\{\hat{f}^{t,c}_{i,j}\}$ are ready to go through three attention layers sequentially with structured masking (see Figure 3). The intuition behind the design of each attention module is as follows.

- *Local Attention (LA)* captures the relationship between attributes within the part itself. As the dependency between shape and articulation is both within and among parts, we intend to make node attributes carry the relationship among themselves to exchange information between nodes later. The masking in LA only activates attention positions among attributes within the same node.
- *Global Attention (GA)* allows for attention between every pair of attributes across all valid nodes. The valid nodes are indicated by a key padding mask referring to the number of parts in each graph. This module is designed to capture the relationship among nodes regardless of the distance in the graph. Carrying the information from distant nodes to the subsequent module, which only concentrates on a local neighbourhood, can enhance overall hierarchical understanding. For masking, we only apply a key padding mask to ensure attention is applied only between non-padded attributes.
- *Graph Relation Attention (GRA)* focuses on the relation-

	MMD		COV		Realism
	ID↓	AID↓	ID↑	AID↑	AOR↓
NAP [13]	0.118	0.751	0.752	0.794	0.026
NAP-light	0.060	0.741	**0.773**	0.866	0.062
Ours-light	**0.043**	**0.636**	0.753	**0.867**	**0.007**

Table 1. We evaluate the distribution modeling of a variation of our approach and compare against NAP and its variation when training on *articulation* graph with $K = 8$. Our approach largely outperforms the baselines in terms of similarity to ground truth test set samples (measured by the MMD metrics), coverage of the ground truth test set distribution (measured by the COV metrics), and realism as measured by the AOR metric.

ship between attributes only from parent and children nodes. This is where we leverage the graph structure as a condition by explicitly masking the attention using the graph adjacency matrix. The root node (purple in Figure 3) is the sole self-connected node in the matrix. The model latches onto this signal through attention, and deduces edge direction by tracing down the tree from it. After propagating to the zero-ring and N-ring in the graph in the preceding attention layers, here we focus on one-ring relations, which are the strongest factor for articulations.

Our experiments show the effectiveness of these attention modules in Section 5.3.

4. Experiments

4.1. Experimental Setup

We use the PartNet-Mobility dataset [26] with an 80/20 train-test split ratio per object category. We use eight object categories (i.e., Storage, Table, Refrigerator, Dishwasher, Safe, Oven, Washer, and Microwave) with a maximum of 32 nodes in the *action* graph. The architecture of the denoising network stacks 12 layers of attribute attention blocks with 32 heads in each attention module. We train with the AdamW [21] optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\lambda = 0.01$) for 5000 epochs, taking 13 hours training on a single NVIDIA A40 GPU with batch size as 64.

4.2. Baselines

We compare our approach with NAP [13] and several variants. NAP's data representation differs from ours as: 1) an additional $128D$ latent code is used to represent the surface for each part; 2) the joint type is not explicitly modelled. The specific baseline variations are:

- **NAP**: the original NAP model and data representation, trained on *articulation* graphs ($K = 8$).
- **NAP-light**: NAP with our data parameterization, trained on *articulation* graphs ($K = 8$).
- **Ours-light**: variant of our method trained on *articulation* graphs with $K = 8$ to be comparable with NAP-light.

	MMD		COV		Realism	
	ID↓	AID↓	ID↑	AID↑	AOR ↓	HS % ↑
NAP-large	0.067	0.860	0.716	0.716	0.034	17.39
Ours	**0.049**	**0.816**	**0.753**	**0.852**	**0.008**	**82.61**

Table 2. We evaluate the distribution modeling of our model and compare against NAP's variation when training on *action* graph with $K = 32$. In this more challenging setting requiring handling of complex input graphs, our approach significantly outperforms the NAP baseline along all metric axes, including in perceived generation realism as measured by human judgment, reported in HS column as a preference rate.

- **NAP-large**: uses our data parameterization and is trained on *action* graphs with $K = 32$.

4.3. Metrics

Our evaluation metrics rely on a notion of distance between articulated objects. We use the Instantiation Distance (**ID**) from NAP [13] which considers both part geometry and motion. This is the minimum pairwise Chamfer-L1 distance per articulation state using 2048 point samples per part per object.

We refine this metric to consider the pairwise distance for temporally synchronized states, rather than enumerating across all state pairs as in Lei et al. [13] This refinement enhances computation efficiency and eliminates spurious point pair distance computations caused by erroneous part matching. We also introduce the Abstract Instantiation Distance (**AID**), which minimizes the influence of fine-grained part geometry by using volumetric IoU (vIoU) on part bounding boxes instead of Chamfer distance.

Armed with these two distances we define the following evaluation metrics: 1) Minimum Matching Distance (**MMD**) reports the average minimum matching distance between the ground truth set and the generated set. It measures the similarity between the approximated and ground truth distribution. 2) Coverage (**COV**) is the percentage of ground truth objects with at least one matched generated sample. A large value indicates better diversity of the generation and better distribution coverage. 3) 1-Nearest Neighbor Accuracy (**1-NNA** [13]) reported using AID. This metric measures the distance between the generated and ground truth distribution using 1-NN classification accuracy.

We also report the Average Overlapping Ratio (**AOR**), computed as the average ratio of overlapping volume between any two sibling parts in the object. We design this metric by assuming that the sibling nodes in the graph should never overlap in any articulation state. Overlapping volumes are determined from part oriented bounding boxes and computed using vIoU. This metric measures the physical plausibility of the generated part structure. Lastly, we report human judgments of generated object quality and

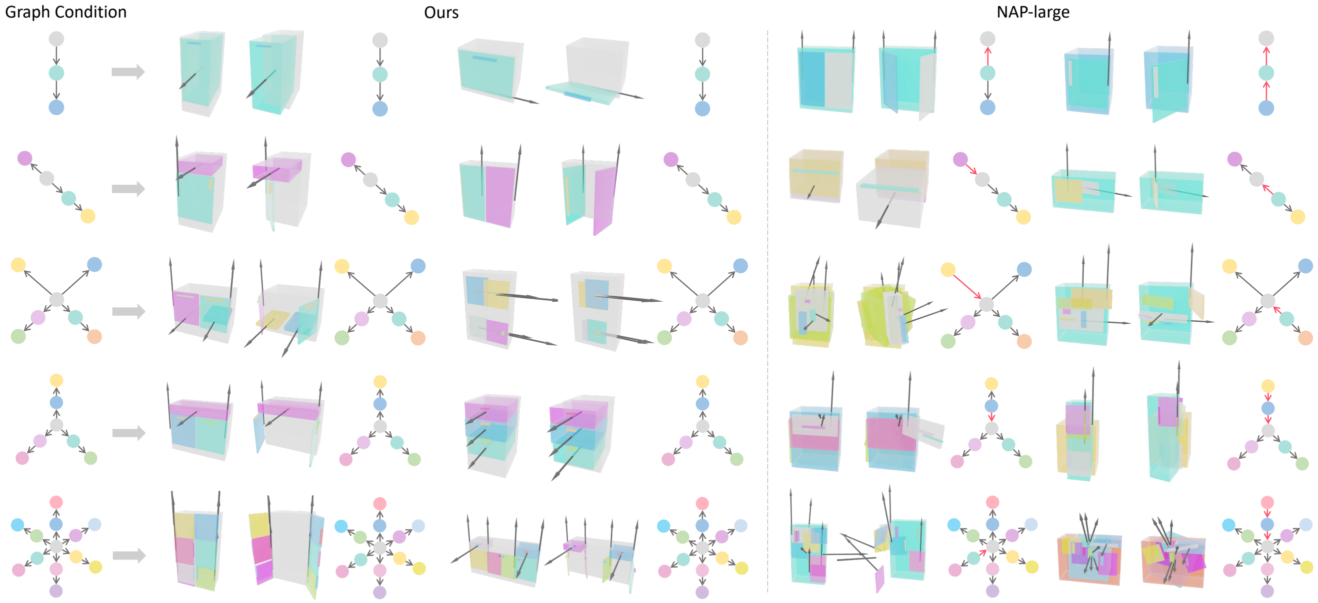

Figure 4. Qualitative results conditioned on graph structures (on the left) at different levels of complexity. We compare our method with a comparable version of NAP. Our generated objects are faithfully compatible with the graph input. In contrast, NAP fails to conform to the input constraint with flipped or disordered node connections. We denote inconsistent graph connections using red arrows.

plausibility through a human study (**HS**). We conducted this study using a two-alternative forced choice setup (A/B choice). Participants were shown 20 pairs of randomly generated results from ours and NAP-large, and asked to choose the object with highest quality in terms of part articulations, part arrangement (i.e. no overlaps), and plausibility compared to real-world objects. We collected responses from 44 participants not involved with this work and report the preference rate in Table 2.

4.4. Part Retrieval

Once we have a generated articulated object abstraction specifying all the parts and their attributes (and optionally an object category constraint specified in the input), we use a part retrieval strategy to extract part surfaces from the training data and build the final 3D object mesh. The training data contains a total 527 objects composed of 2690 parts which can be composited into a generated object through part-level retrieval. We use a two-step approach: 1) we compute the AID metric of the generated object against candidate objects in the train set. We identify the candidate with the best AID metric and pick the base part from it. This procedure ensures that the base part is compatible with the part motions as much as possible. 2) For the remaining parts, we pick as many parts as possible from a single candidate object to maintain style consistency. We start with the candidate selected in Step 1 and consider other candidates while there are still parts left unretrieved. Please refer to the supplement for more details.

5. Results

5.1. Overall Generation Quality

We evaluate how well our method models the real data distribution by randomly generating samples conditioned on the category labels and graphs in the testing set and computing the metrics described in Section 4.3 against the test objects. We generate five times as many objects as the test set and report the results in Tables 1 and 2. Table 1 compares baseline models trained on articulated parts with $K = 8$. Table 2 shows a more challenging setting that includes actionable parts and more complex graph topology, with $K = 32$. The lower MMD and higher COV values demonstrate our method better captures the training data distribution than NAP and other baselines. Our method outperforms both NAP-large (0.567 vs. 0.728) and NAP-light (0.495 vs. 0.521) on 1-NNA as well. In addition, our lower AOR indicates that generated objects suffer less from overlapping volumes among sibling parts during articulation, which suggests more physically realistic objects. The HS score further supports the finding that our generated objects are of higher fidelity and realism, as perceived by people.

5.2. Conditional Generation

A key focus of our work is controllability. Thus, we evaluate performance in five conditional generation tasks illustrating different forms of control over the output. For graph-conditioned generation, we explicitly inject the graph structure into the denoiser as the condition. For generation

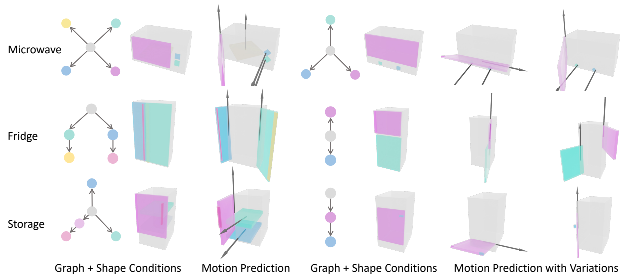

Figure 5. Part→Motion: generated results conditioned on graphs specifying part bounding boxes.

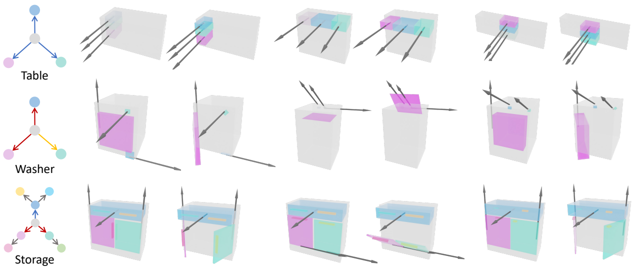

Figure 6. Joint Type → Part: qualitative results conditioned on specific articulation joint type. Edge colors in the graph indicate the input joint type: blue for prismatic, red for revolute, yellow for continuous, and grey for fixed. The outputs conform to these constraints while exhibiting variety.

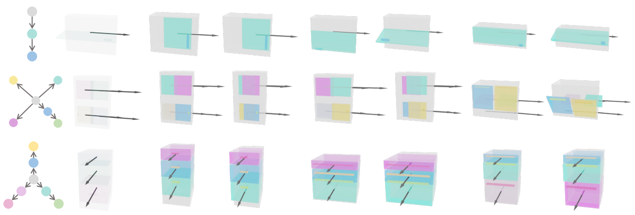

Figure 7. Joint Axis → Part: generated objects for input graphs specifying joint axis constraints (shown by arrows). Output objects have parts with motions corresponding to the given axes but varying part type and overall arrangement.

conditioned on other attributes associated with the graph, we mask the conditioning attributes with ground truth values and fill in other node attributes via denoising (akin to "inpainting"). For NAP, all the conditional generation are implemented using "inpainting" generation as above. Note that conditioning on particular node attributes presupposes the inclusion of the graph as a condition.

Graph-conditioned generation. Figure 4 shows qualitative examples at different levels of complexity for the input conditioning graph topology. Our results are consistently plausible and of high quality, regardless of graph complexity. In contrast, NAP often fails to respect the input graph topology and generates objects and graphs with flipped and

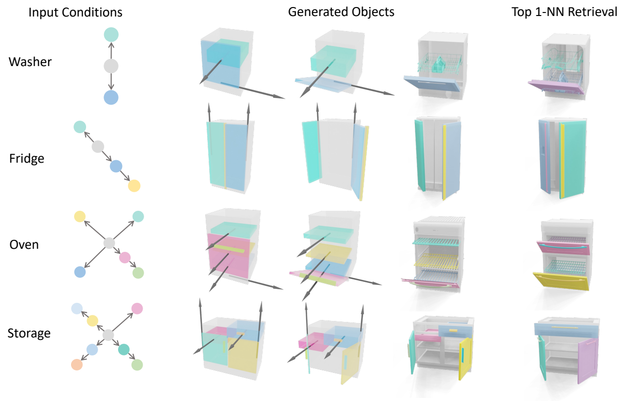

Figure 8. Generated objects for out of distribution input graphs (structure unseen in train set) and the corresponding nearest neighbor retrieval from the train set. The generated objects for various categories exhibit realistic arrangement even though the exact number of parts and their arrangement were not in the train set.

disordered node connections. NAP has a notably high rate of generating inconsistent graphs with the input condition, and this issue becomes even more pronounced when the graph constraints are more complex.

Part → Motion. Given the position and size of the bounding box for each part, this experiment aims to generate compatible motion parameters (joint type, joint axis, and joint range). Figure 5 shows the qualitative results under different categories. On the left, we present three examples that are expected to generate more deterministic motions, given the specified arrangement of parts. On the right, we show generated motion with variations, each reasonably compatible with the specified shape conditions.

Joint Type → Part. Given the joint type for each part, this experiment aims to generate compatible bounding boxes and joint parameters (joint axis and joint range). We show qualitative results in Figure 6. Given the specified joint type (denoted in varied colours on the graph edges) for each part, we show variations of generated shapes adhering to these conditions.

Joint Axis → Part. In this scenario, the position and direction of the joint axis for each part is given and we generate compatible bounding boxes, joint types, and joint ranges. Figure 7 shows some results, demonstrating variation while conforming to the axis constraint. Note that in cases where a part is connected via a prismatic joint, the part considers only the direction of the axis as its constraint.

OOD graph-conditioned generation. This experiment shows how well our model generalizes the graph constraint by conditioning the generation on graph topologies that are out of the distribution of the training samples. Figure 8 shows the qualitative results for various object categories. These results illustrate that our method can generate reasonable objects even for unseen graphs as conditions.

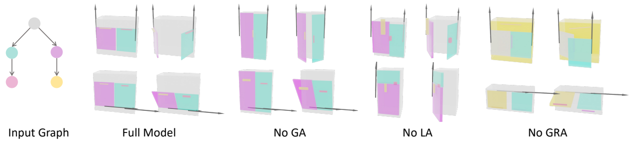

Figure 9. Ablation of each attention module in our denoising architecture. The significance of each module increases from left to right. Removing these modules leads to lower quality objects, with inconsistent connections between articulating parts, misassigned object base part (i.e. non-moving root for the object), and unrealistically floating part motions.

	full	no LA	no GA	no GRA
MMD-AID↓	**0.816**	0.840	0.831	0.876
MMD-ID↓	**0.049**	0.053	0.052	0.157
AOR↓	**0.008**	0.016	0.011	0.013

Table 3. Quantitative results for ablations of our attention modules. MMD increases as individual attention modules are removed, indicating their positive impact on the design of our architecture. There is also a corresponding drop in sample realism, as indicated by the increasing AOR metric.

5.3. Ablations

To show the effectiveness of our architecture we ablate the local attention (LA), global attention (GA), and graph relation attention (GRA) modules in each attribute attention block (AAB) (see Figure 9 and Tab. 3). Figure 9 shows qualitative examples conditioned on the same graph across ablations removing one module at a time. The significance of each module increases progressively from left to right. GA is designed to learn the relation between nodes beyond the 1-ring neighborhood (pink and yellow nodes in this case). Removing GA makes handles less symmetric compared to the full model. LA is designed to learn the relation between attributes within each part (e.g., handle should be at far end from joint axis). By removing LA, this correlation is weakened. GRA is the main module for learning part arrangements that conform to the input graph topology. By removing GRA, the generated parts do not respect the specified part hierarchy. In Table 3, the MMD score increases as modules are removed, indicating lower quality objects. There is also a corresponding drop in sample realism, as indicated by the AOR score. Generally, the removal of explicit attention to part relations leads to generated objects that tend to not conform to input constraints, making coverage-based metrics less meaningful.

5.4. Failure Cases and Limitations

Figure 10 shows typical failure patterns in terms of generation quality and controllability. 1) Overlapping parts, or collisions during motion. 2) Reliance on part retrieval can lead to unrealistic part combinations (e.g., inconsistent drawers under desk). 3) Mismatched parts and joint axes in the *Part* → *Motion* conditional scenario, leading to physically unrealistic motion. 4) Floating parts in the more challenging *Joint Axis* → *Part* setting due to generated part not perfectly aligning with joint axis.

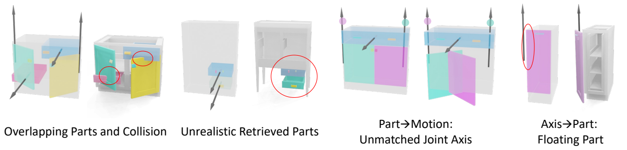

Figure 10. We show several failure cases in terms of both generation quality and controllability.

6. Conclusion

We address conditional generation of articulated 3D objects, allowing for fine-grained user-specified constraints on object parts and articulation. We develop a denoising diffusion architecture with a set of part attribute attention blocks that guide generation based on the input conditions. We thus leverage relations between part attributes and generate higher-quality objects that better conform to the user constraints compared to prior work. Our qualitative and quantitative evaluations show that we significantly outperform the state-of-the-art, generating more realistic and more complex articulated objects, and exhibiting greater diversity.

Some limitations of our work suggest future work directions. Due to significant data imbalance, our method has better performance on higher frequency objects. Data augmentation schemes for obtaining a more uniform performance would be an interesting direction to explore. Similarly to NAP [13], we face the challenge that motion attributes exhibit relatively weaker controllability than geometric attributes. While we enhance the dependency through attention among attributes, further investigation into reinforcing this connection is warranted. Additionally, our part synthesis currently relies on a retrieval strategy which is limited by the diversity of available objects and parts. Combining our work with part geometry generation is another interesting avenue for future work.

We believe that the fine-grained controllable generation of 3D articulated objects that our approach provides will enable scalable generation of interactive 3D assets in support of tasks in computer vision, robotics, and embodied AI.

Acknowledgements. This work was funded in part by a Canada Research Chair, NSERC Discovery, and enabled by support from WestGrid and the Digital Research Alliance of Canada. We thank Xiang Xu, Yizhi Wang, and Hanhung Lee for helpful discussions, as well as Angel X. Chang, Sonia Raychaudhuri, Xiaohao Sun, Qirui Wu, Yiming Zhang, and Ning Wang for proofreading.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *ICLR*, 2019. 4

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[3] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *NeurIPS*, pages 8780–8794, 2021. 4

[4] Haoyuan Fu, Jieyi Zhang, Chen Bao, Qiang Zhang, Yongxi Huang, Wenqiang Xu, Animesh Garg, Cewu Lu, et al. RoboTube: Learning Household Manipulation from Human Videos with Simulated Twin Environments. In *6th Annual Conference on Robot Learning*, 2022. 1

[5] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao(Richard) Zhang. SDM-NET: Deep Generative Network for Structured Deformable Mesh. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2019)*, 38(6):243:1–243:15, 2019. 2

[6] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 21201–21210, 2023. 2

[7] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 2, 3

[9] Cheng-Chun Hsu, Zhenyu Jiang, and Yuke Zhu. Ditto in the House: Building Articulation Models of Indoor Scenes through Interactive Perception. *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2023. 1

[10] Hanxiao Jiang, Yongsen Mao, Manolis Savva, and Angel X Chang. OPD: Single-view 3D Openable Part Detection. In *ECCV*, pages 410–426. Springer, 2022. 1

[11] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building Digital Twins of Articulated Objects from Interaction. In *CVPR*, pages 5606–5616, 2022. 1, 2

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 4

[13] Jiahui Lei, Congyue Deng, Bokui Shen, Leonidas Guibas, and Kostas Daniilidis. NAP: Neural 3D Articulation Prior. *NeurIPS*, 2023. 1, 2, 3, 5, 8

[14] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. GRASS: Generative Recursive Autoencoders for Shape Structures. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*, 36(4):to appear, 2017. 2

[15] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-SDF: Text-to-Shape via Voxelized Diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12642–12651, 2023. 2

[16] Xiaolong Li, He Wang, Li Yi, Leonidas Guibas, A Lynn Abbott, and Shuran Song. Category-Level Articulated Object Pose Estimation. *CVPR*, pages 3703–3712, 2020. 1

[17] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-Resolution Text-to-3D Content Creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 2

[18] Jiayi Liu, Ali Mahdavi-Amiri, and Manolis Savva. PARIS: Part-level Reconstruction and Motion Analysis for Articulated Objects. In *ICCV*, pages 352–363, 2023. 1, 3

[19] 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 *CVPR*, pages 14809–14818, 2022. 1

[20] Shaowei Liu, Saurabh Gupta, and Shenlong Wang. Building rearticulable models for arbitrary 3d objects from 4d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21138–21147, 2023. 2

[21] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2018. 5

[22] Yongsen Mao, Yiming Zhang, Hanxiao Jiang, Angel Chang, and Manolis Savva. MultiScan: Scalable RGBD scanning for 3D environments with articulated objects. *NeurIPS*, 35: 9058–9071, 2022. 1

[23] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *International Conference on Learning Representations*, 2022. 2

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[25] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas Guibas. StructureNet: Hierarchical Graph Networks for 3D Shape Generation. *ACM Transactions on Graphics (TOG), Siggraph Asia 2019*, 38 (6):Article 242, 2019. 2

[26] 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 *CVPR*, pages 909–918, 2019. 1, 5

[27] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2

[28] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-E: A System for Generat-

ing 3D Point Clouds from Complex Prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2

[29] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023. 2, 4

[30] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2

[32] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents, 2022. 2

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution Image Synthesis with Latent Diffusion Models. In *CVPR*, pages 10684–10695, 2022. 2

[34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *NeurIPS*, 35:36479–36494, 2022. 2

[35] Etai Sella, Gal Fiebelman, Peter Hedman, and Hadar Averbuch-Elor. Vox-E: Text-guided Voxel Editing of 3D Objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 430–440, 2023. 2

[36] Mohammad Amin Shabani, Sepidehsadat Hosseini, and Yasutaka Furukawa. HouseDiffusion: Vector Floorplan Generation via a Diffusion Model With Discrete and Continuous Denoising. In *CVPR*, pages 5466–5475, 2023. 2

[37] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep Marching Tetrahedra: a Hybrid Representation for High-Resolution 3D Shape Synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021. 2

[38] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *NeurIPS*, 34:251–266, 2021. 1

[39] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. DiffuScene: Scene Graph Denoising Diffusion Probabilistic Model for Generative Indoor Scene Synthesis. *arXiv preprint arXiv:2303.14207*, 2023. 2

[40] Wei-Cheng Tseng, Hung-Ju Liao, Lin Yen-Chen, and Min Sun. CLA-NeRF: Category-Level Articulated Neural Radiance Field. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 8454–8460, 2022. 1

[41] Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. Sketch-Guided Text-to-Image Diffusion Models. In *ACM SIGGRAPH 2023 Conference Proceedings*, New York, NY, USA, 2023. Association for Computing Machinery. 2

[42] Johanna Wald, Torsten Sattler, Stuart Golodetz, Tommaso Cavallari, and Federico Tombari. Beyond Controlled Environments: 3D Camera Re-Localization in Changing Indoor Scenes. In *ECCV*, pages 467–487. Springer, 2020. 1

[43] Hao Wang, Nadav Schor, Ruizhen Hu, Haibin Huang, Daniel Cohen-Or, and Hui Huang. Global-to-Local Generative Model for 3D Shapes. *ACM TOG*, 37(6):214:1–214:10, 2018. 2

[44] 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 *CVPR*, pages 8876–8884, 2019. 1

[45] Yanzhen Wang, Kai Xu, Jun Li, Hao Zhang, Ariel Shamir, Ligang Liu, Zhiquan Cheng, and Yueshan Xiong. Symmetry Hierarchy of Man-Made Objects. *Computer Graphics Forum (Eurographics 2011)*, 30(2):287–296, 2011. 2

[46] Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. SAGNet: Structure-aware Generative Network for 3D-Shape Modeling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019)*, 38(4):91:1–91:14, 2019. 2

[47] 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 *CVPR*, pages 11097–11107, 2020. 1

[48] Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. *NeurIPS*, 32, 2019. 4

[49] 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. *ACM TOG*, 38(6):240:1–240:15, 2019. 1

[50] Jie Yang, Kaichun Mo, Yu-Kun Lai, Leonidas J Guibas, and Lin Gao. Dsm-net: Disentangled structured mesh net for controllable generation of fine geometry. *arXiv preprint arXiv:2008.05440*, 2020. 2

[51] Qiaojun Yu, Junbo Wang, Wenhai Liu, Ce Hao, Liu Liu, Lin Shao, Weiming Wang, and Cewu Lu. GAMMA: Generalizable Articulation Modeling and Manipulation for Articulated Objects. *arXiv preprint arXiv:2309.16264*, 2023. 1

[52] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models, 2023. 2