# ArtFormer: Controllable Generation of Diverse 3D Articulated Objects

Jiayi Su[1,*,†] Youhe Feng[2,*] Zheng Li[4] Jinhua Song[1] Yangfan He[5,6] Botao Ren[3] Botian Xu[3,‡]
[1]Xiamen University Malaysia [2]Renmin University of China [3]Tsinghua University
[4]Southern University of Science and Technology [5]University of Minnesota-Twin Cities
[6]Henan RunTai Digital Technology Group Co., Ltd.
[*] Equal Contribution. [†]CST2209162@xmu.edu.my [‡]btx0424@outlook.com

github.com/ShuYuMo2003/ArtFormer

For the convenience of readers, the changes made during the rebuttal stage are highlighted in different colors.

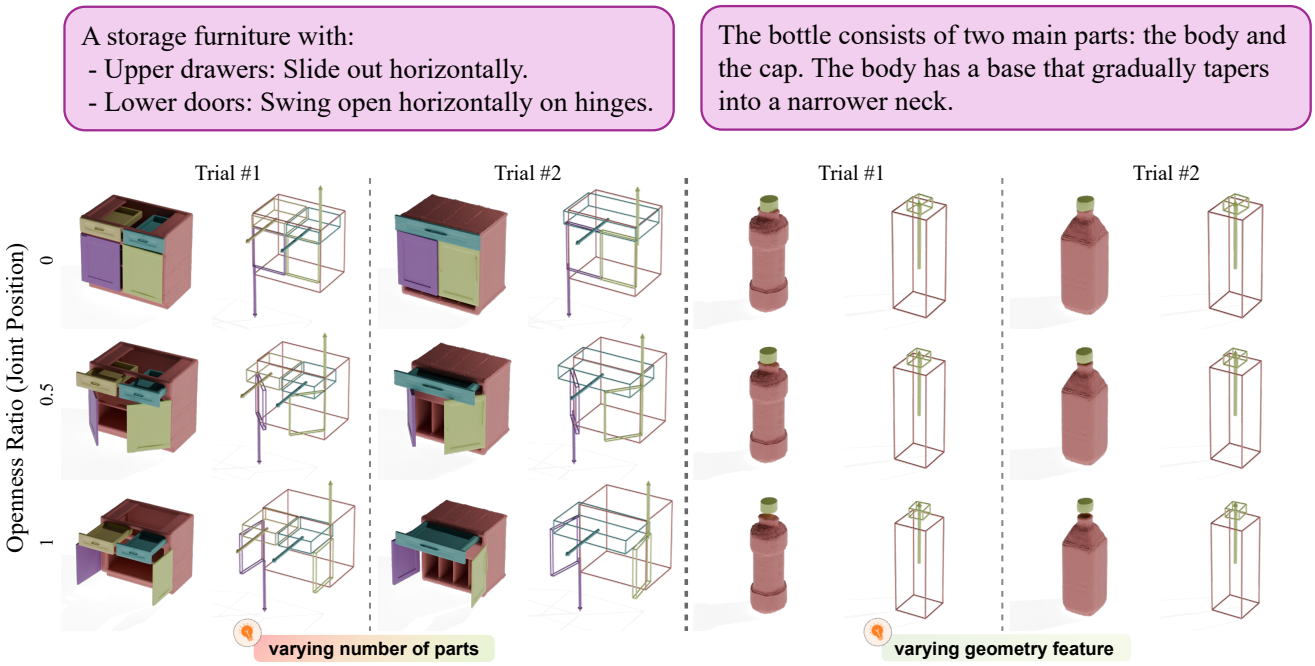


Figure 1. We present the ***Art***iculation Trans***Former***, for high-quality generation articulated objects. This figure illustrates controlled generation across random trials based on text descriptions. Notably, it can generate a diverse range of objects with varying numbers of sub-parts and different geometry features.

## Abstract

*This paper presents a novel framework for modeling and conditional generation of 3D articulated objects. Troubled by flexibility-quality tradeoffs, existing methods are often limited to using predefined structures or retrieving shapes from static datasets. To address these challenges, we parameterize an articulated object as a tree of tokens and employ a transformer to generate both the object's high-level geometry code and its kinematic relations. Subsequently, each sub-part's geometry is further decoded using a signed-distance-function (SDF) shape prior, facilitating the synthesis of high-quality 3D shapes. Our approach enables the generation of diverse objects with high-quality geometry and varying number of parts. Comprehensive experiments on conditional generation from text descriptions demonstrate the effectiveness and flexibility of our method.*

## 1. Introduction

Articulated objects are defined as entities composed of multiple rigid sub-parts connected by various joints which allow the sub-parts to undergo constrained relative motion [33]. Among others, man-made articulated objects con-

stitute most everyday objects around us.

The perception [21, 29, 38, 66] and reconstruction [15, 42, 62] of articulated objects have been extensively studied. However, research on generating articulated objects remains limited. On the one hand, to generate a multi-part articulated object, the model must simultaneously produce both the geometry of each sub-part and the kinematic relationships between them. Existing methods find it challenging to generate both modalities with high quality simultaneously. On the other hand, the complexity of articulated objects makes annotating them very costly, resulting in limited datasets for articulated objects.

Most relevant to this work are NAP [28], CAGE [34], and SINGAPO [32]. They all support the conditional generation of 3D articulated objects but are limited to pre-defined graph structures. However, NAP has limited capability to adhere to the given condition while producing high-quality geometry. Meanwhile, aimed at controllability and quality, CAGE and SINGAPO do not actually generate the geometry but rather perform retrieval from datasets, restricting their ability to produce novel and diverse objects.

To achieve both diversity and usability, this paper proposes a novel framework, ***Art***iculation Trans***former***, to generate high-quality and diverse articulated objects from text descriptions. We parameterize each articulated object with a tree structure. Each node corresponds to a sub-part, encompassing both its geometry and the kinematic relation (joint transform) relative to its parent node. Treating each node as a token, we utilize a transformer architecture to generate the sub-parts of the articulated object. Additionally, we introduce a tree position embedding in place of the ordinary position embeddings to better encode the tree structure from a sequence of tokens. Conditions (such as text descriptions and images) can be flexibly incorporated using cross-attention modules in the transformer layers.

However, simultaneously generating high-quality geometry and accurate joint parameters poses drastic challenges to both the model capacity and training pipeline. Instead of generating the geometries directly, we let the transformer output a compact latent code, which is then decoded by a Signed Distance Function (SDF) shape prior. The shape prior is trained on datasets with its latent space modeled by a diffusion model. This approach allows controllable sampling of sub-parts with varying geometry details.

In this paper, we primarily conduct experiments on text-guided generation of articulated objects, with a pre-trained text-encoder providing conditions to the transformer. Compared to prior works, our results demonstrate that we can generate a more diverse array of articulated objects that exhibit more precise kinematic features and high-quality geometry as well. Moreover, we also validate the flexibility of our framework with image-guided generation.

In summary, our main contributions are:

1. We present a novel framework for modeling and conditional generation of 3D articulated objects.
2. A novel sampling and decoding recipe is designed to facilitate generation of shapes with diverse yet high-quality geometry.
3. Through experiments on text- and image-conditioned generation, we validate the effectiveness and flexibility of our framework.

We believe our method could enable a range of future research and applications such as building Digital Cousins [8] for scaling up robot learning.

## 2. Related Work

### 2.1. Modeling 3D Articulated Objects.

Articulated objects are a specialized type of 3D object distinguished by their segmented, jointed structure, allowing for flexible movement and positioning of individual sub-parts. Modeling 3D articulated objects, an extension of 3D object modeling, involves the prediction [14, 41, 58], reconstruction, and generation of flexible, jointed structures.

Implicit neural representations have become a popular option recently [39, 46, 63, 64] due to their GPU memory efficiency and the ability to generate high-quality geometry. A convenient characteristic of implicit representations is that spatial transforms to the shape can be cast as rigid transforms to the input query points, making them a good choice for dealing with the kinematic relations in articulations.

A-SDF [42] is among the earliest explorers of using SDF to model articulated objects, but did not utilize the aforementioned property. More recently, NAP [28] introduces the first 3D deep generative model for synthesizing articulated objects through a novel articulation tree/graph parameterization and the use of a DDPM [16], enabling masked generation applications. Similarly, CAGE [34] also employs a graph diffusion denoising model but with a primary aim of controllability. SINGAPO [32] further extends controllable generation to single-image conditioning. MeshArt [12] utilizes domain-specific tokenizers to convert articulated objects into sequences and employs a transformer architecture for generation.

However, CAGE and SINGAPO only generate abstractions of sub-parts, which are then used to retrieve similar assets from a dataset. Therefore, they can not produce objects with geometry features that are unseen in the dataset. This limitation is also common to methods that do not use SDF, such as URDFormer [4], which predicts predefined URDF [50] primitives and meshes. A potential reason for such limitation is the difficulty of simultaneously modeling kinematic relations and geometry. Hyper-SDF [10] and Diffusion-SDF [6] propose methods to learn high-quality yet controllable priors of rigid SDFs. This work adopts a

shape prior similar to Diffusion-SDF to ensure geometry quality.

### 2.2. Generating Tree-structured Objects

Generating tree-structured objects differs from conventional sequential generation, as each node can have multiple successors. Traditional approaches model graph distributions using Variational Autoencoders (VAEs) [27, 57, 67], Generative Adversarial Networks (GANs) [9, 37, 61] and Denoising Diffusion Networks [18, 19, 23, 68, 69]. DiGress [60] and FreeGress [43], for example, achieve state-of-the-art generation performance and can handle large, diverse molecular datasets. However, these methods are not tailored for tree-structured graphs and lack autoregressive generation, resulting in unreliable outputs for realistic acyclic, single-edge tree structures. To address this limitation, SceneHGN [13] introduces a recursive auto-encoder-based method that enables the hierarchical tree-structured generation of 3D indoor scenes. Similarly, GRAINS [31] achieves hierarchical indoor scenes generation using a specific recursive VAE. Shiv et al. [55] extend transformers to tree-structured data by proposing a novel tree-to-sequence mapping method. Peng et al. [48] advance this approach and enable Transformers to learn from both pairwise node paths and leaf-to-root paths by integrating tree path encoding into the attention module.

### 2.3. Learning Representations with Conditional Inputs

[1] Reed et al. [53] present a neural language model trained from scratch for zero-shot visual recognition, enabling accurate image retrieval using text-based representations. Hubert et al. [20] use autoencoders to extract visual-semantic joint embeddings. State-of-the-art methods [1, 2, 24, 47, 54] use a domain-specific embedding layer to learn implicit representations for multi-modal inputs, employing functions like cross-attention to model the joint distribution. Compared to image and text data, 3D-formatted data is less common, leading to extensive research on reconstructing 3D objects from image or text inputs[2, 3, 11, 17, 62]. Chen et al. [3] utilize learning by association and metric learning techniques to learn representations conditioned on text. Liu et al. [32] reconstruct 3D articulated objects from a single image, leveraging DINOv2 [44] and GPT-4o to extract articulation information.

## 3. Method

### 3.1. Articulation Parameterization

Our parameterization process encodes an articulated object into a tree structure highly similar to the format used in URDF [50] and MJCF [59] files. We consider each node (part) as a token that stores the geometry and kinematic relations of the corresponding sub-part of the articulated object. The attributes stored at each node are similar to those stored in the data parameterization of CAGE [34]. Regarding the geometry information, for the $i$-th node, we identify the following 2 attributes:

- **Bounding box,** $b_i \in \mathbb{R}^6$: For an articulated object, each sub-part is assigned an initial position, with its bounding box defining the maximum and minimum coordinates that the sub-part occupies along each axis in this initial state.
- **Geometry latent code,** $z_i \in \mathbb{R}^{768}$: We collect the point cloud of the sub-part. The point cloud is then processed through a series of encoders, converting it into a corresponding latent vector with dimension 768 to represent the geometry of the $i$-th sub-part.

The kinematic parameters between $i$-th node and its parent node are represented by 2 attributes:

- **Joint axis,** $j_i \in \mathbb{R}^6$: The joint axis includes an origin point and a unit direction vector. The $i$-th sub-part is capable of rotating around this axis or translating along it relative to its parent sub-part. The direction vector determines the positive direction for both rotational and translational movements.
- **Limit,** $l_i \in \mathbb{R}^4$: The attribute defines the permissible ranges for translational and rotational movements, setting the minimum and maximum extents of both translation and rotation relative to the initial position. If a sub-part is restricted from moving relative to its parent part, both the upper and lower bounds of these ranges are 0.

For the $i$-th node in the tree, we store the aforementioned 4 attributes, as well as the index of its parent node. Consequently, each node is represented by a token of dimension $D = 6+768+6+4+1 = 785$. For all coordinates in each node, we utilize coordinates from the global coordinate system.

### 3.2. Diverse and Controllable Shape Prior

As previously mentioned, simultaneous modeling and generating high-quality geometry and accurate kinematic relationships is challenging. Therefore, we first learn a *shape prior* $p(z)$ of the geometry latent code using a method similar to Diffusion-SDF [6].

**Shape Prior.** An articulated object consists of multiple sub-parts. Given a sub-part sampled from the dataset, we encode its point cloud with a VAE encoder: $q(z|f)$ where $f = \Gamma(\text{point cloud}) \in \mathbb{R}^{3\times256\times64\times64}$ is an intermediate tri-plane feature obtained from a PointNet encoder $\Gamma$. A generalizable SDF network $\Omega(f, x)$ then predicts the part's SDF at query points $x \in R^3$ from decoded features $p(f|z)$. The training objective is:

$$L(q,p,\Gamma,\Omega) = ||\Omega(\hat{f},x) - \text{SDF}(x)||_1 + \beta D_{\text{KL}}\left(q(z|f)||\mathcal{N}(\mathbf{0},\mathbf{I})\right). \quad (1)$$

[1] This subsection is removed in the final version.

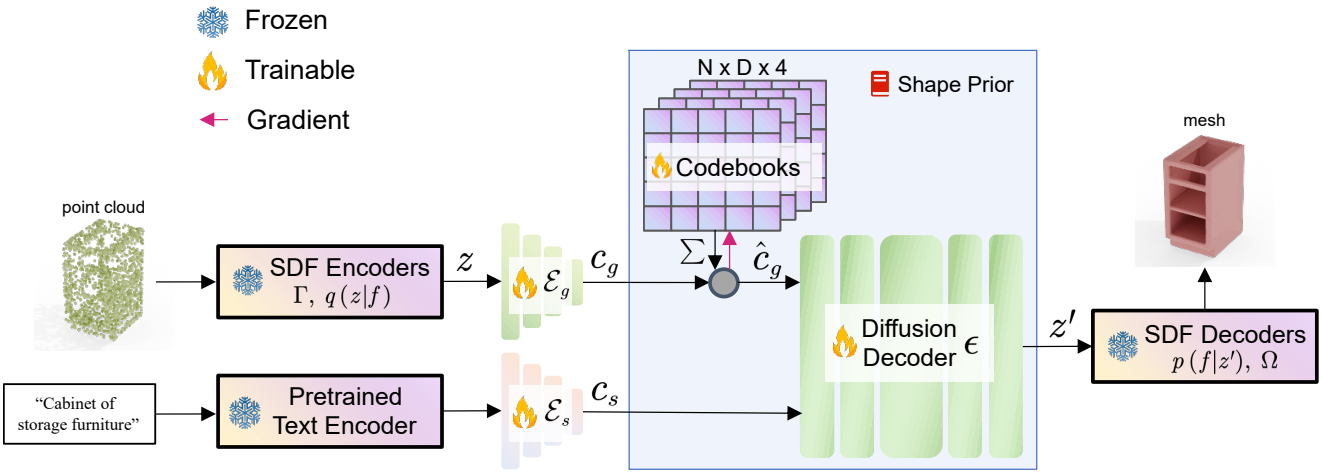


Figure 2. **Training Pipeline of Shape Prior** Mini encoder $\mathcal{E}_g$ compresses the geometry latent code $z$ into $c_g$, which is then processed by the embedding vectors of codebooks to form $\hat{c}_g$. $\hat{c}_g$ is the condition for diffusion decoder $\epsilon$. Each sub-part has a semantic label, such as 'the lid of cup' or 'handle of box'. These labels, encoded by the pre-trained text encoder, pass through mini encoder $\mathcal{E}_s$. The resultant vector $c_s$ is then passed into the diffusion shape prior directly.

where $\hat{f} \sim p(f|q(z|f))$, SDF$(x)$ is the ground-truth signed-distance at point $x$ and $\beta$ balances the degree of regularization to a Gaussian prior.

However, to subsequently enable guided or conditional generation of object sub-parts geometry, we train a conditional diffusion model $\epsilon(z_t, t, \hat{c}_g, c_s)$ on $p(z)$, where the geometry and semantic conditions are given by two corresponding encoders: $c_g = \mathcal{E}_g(z)$ and $c_s = \mathcal{E}_s(name)$. Before geometry condition $c_g$ is input into the diffusion model, it is processed into $\hat{c}_g$ using codebooks. The diffusion model is trained to denoise random latent $z_T \sim \mathcal{N}(0, \mathbf{I})$ into a meaningful $z_0 \sim p(z)$, following the objective used in [52]:

$$L(\epsilon) = ||\epsilon(z_t, t, \hat{c}_g, c_s) - z_0||_2. \tag{2}$$

The aforementioned pipeline and structure can be referenced in Fig. 2.

To process semantic information in text format (e.g., part name), we prepend a pre-trained text encoder [51] to $\mathcal{E}_s$. Note that after shape prior training, the two encoders $\mathcal{E}$ are discarded, leaving only the codebooks and diffusion decoder as our final shape prior.

**Sampling Diverse Shapes.** A particularly desirable capability is to generate parts with diverse geometry features given the semantic information. For example, we would like USB caps of different shapes and styles. To enable our model for such capability, we discretize the space of geometry code $c_g = \mathcal{E}_g(z)$ to allow for sampling. A geometry condition $c_g$ is chunked into 4 segments $(c_g^0, c_g^1, c_g^2, c_g^3)$, which is designed to enlarge the capacity of the latent space from $4N$ to $N^4$. Then, these 4 segments are used to retrieve $(\hat{c}_g^0, \hat{c}_g^1, \hat{c}_g^2, \hat{c}_g^3)$ from 4 different codebooks $M_t \in \mathbb{R}^{N \times D}$ ($t$ is the index of codebook) using Gumbel-Softmax sampling:

$$\mathcal{D}_{t,l} = -||m_l^t - c_g^t||_2 \tag{3}$$

$$\hat{c}_g^t = \sum_{j=1}^{N} m_j^t \cdot \mathrm{GS}\left(\{\mathcal{D}_{t,l}\}_{l=1}^{N}\right)_j \tag{4}$$

where $\mathcal{D}_{t,l}$ is the distance matrix and $m_l^t \in \mathbb{R}^D$ denotes the $l$-th out of $N$ embedding vector in the codebook $M_t$. The Gumbel-Softmax operation is defined as:

$$\mathrm{GS}\left(\{x_l\}_{l=1}^{N}\right)_j = \frac{\exp\left((x_j + g_j)/\tau\right)}{\sum_{k=1}^{N} \exp\left((x_k + g_k)/\tau\right)} \tag{5}$$

where $g_1, \cdots, g_k$ are samples from $Gumbel(0, 1)$ [22]. The softmax temperature $\tau$ controls the diversity of shape prior, which we do not specifically tune in this work. Since GS sampling is differentiable, the model can still be trained end-to-end.

The diffusion model helps sample from the full continuous space of $z$, given $\hat{c}_g$ which is either quantised from $c_g$ (produced by $\mathcal{E}_g$, during shape prior training) or sample by logits $P_i \in \mathbb{R}^{4 \times N}$ from codebook $M$ directly (after shape prior training). By leveraging the shape prior, the sampled $z$ is guaranteed to align well with the target distribution. Meanwhile, the stochasticity introduced by discrete sampling improves the generation diversity.

### 3.3. Articulation Transformer

After the articulated object is parameterized into a tree structure, each node is treated as a token in the classical transformer architecture. The $i$-th token is composed of

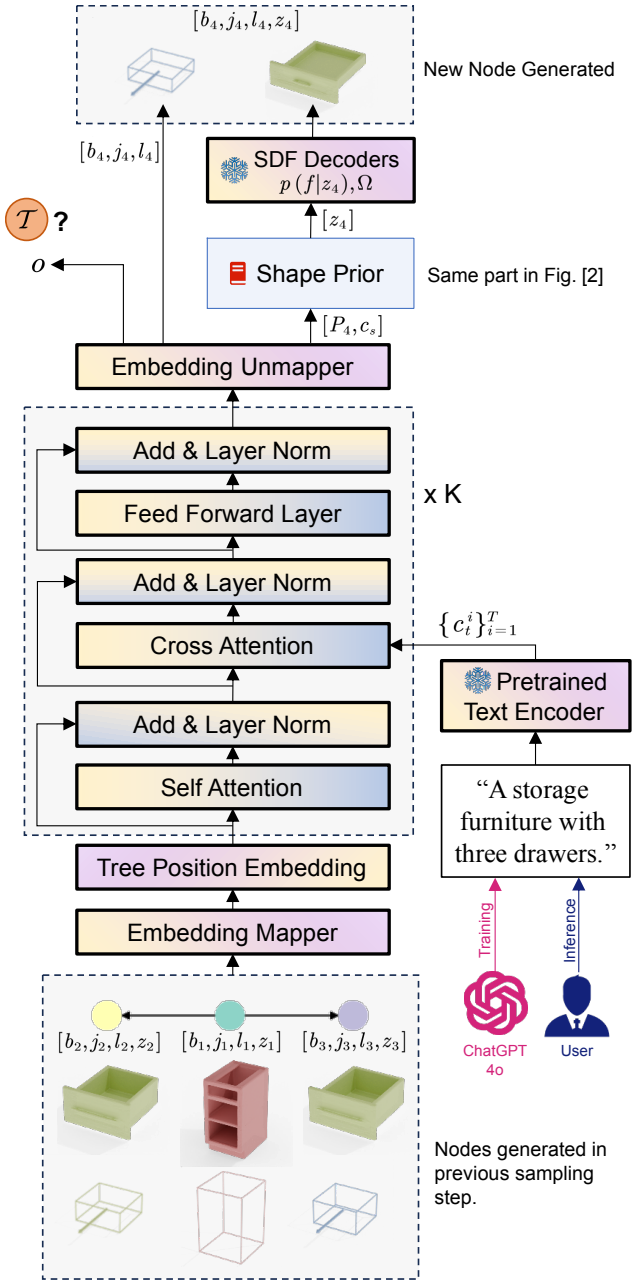


Figure 3. **Articulation Transformer**: In the tree structure, $i$-th node carries 4 attributes: $b_i$, $j_i$, $l_i$ and $z_i$, which respectively represent the bounding box, joint axis, limit, and geometry latent code. $\hat{o}$ represents the logits indicating whether the current output token is a terminal token $\mathcal{T}$ (a special token).

$[fa_i, b_i, z_i, j_i, l_i]$, where $fa_i$ is the parent index of the $i$-th node. In addition to the index of the parent node $fa_i$, the remaining attributes are concatenated and subsequently processed through an MLP Embedding Mapper. An overview of the articulation transformer is illustrated in Fig. 3.

**Tree Position Embedding.** In order for the transformer to recognize the specific position of each token, we proposed a novel position encoding scheme specifically designed for tree structures building upon the works of [25, 49, 56]. We first calculate the absolute position encoding $a_i \in \mathbb{R}^{64}$ for each $i$-th node:

$$a_i = \mathrm{GRU}\left(\{\mathcal{A}_{p_k}\}_{k=1}^{K}\right) \tag{6}$$

where $\mathcal{A}_i = [b_i, z_i, j_i, l_i]$, $K$ is the depth of $i$-th node and $\{p_k\}_{k=1}^{K}$ is the sequence of indices along the path on the tree from the root to $i$-th node. It pushes the tokens on the path from the root to a bi-directional GRU [25, 49] to compress the information. We define the position embedding of the $i$-th node $p_i \in \mathbb{R}^{1024}$ to represent the relative position:

$$p_i = \mathrm{CAT}\left(\{a_{p_k}\}_{k=K}^{1}\right), \tag{7}$$

where CAT denotes concatenation. We employ truncation or padding with zeros to ensure that $p_i$ has a uniform length across all nodes.

**Conditioning.** We primarily demonstrate conditional generation based on text descriptions. A text input is processed by a pre-trained text encoder [51], producing a sequence of conditioning tokens $\{c_t^i\}_{i=1}^T$, which are incorporated into the transformer through cross-attention layers. For training, we generate paired data using the following recipe: (1) sample an object from the dataset, (2) render images from different views using Blender [7], and then (3) query ChatGPT-4o for text descriptions. For image conditioning, we can simply replace the text encoder with an image encoder and adopt a similar procedure. More details are provided in Supplementary Materials.

**Iterative Decoding.** Instead of predicting all parts at once (which assumes they are conditionally independent), we adopt an iterative decoding procedure to capture the interdependence between parts.

In each iteration, we input all previously generated nodes and predict a child for all of the input nodes, starting from a special start token $\mathcal{S}$, which conditions the generation of the root node. If no child nodes can be added to a current node, a special terminal token $\mathcal{T}$ is outputted. The self-attention layer ensures that the same child token is not repeatedly generated for any node across different prediction iterations. This process can be better understood through the illustration in Fig. 4.

The decoding procedure terminates when all output tokens are $\mathcal{T}$. We then project the generated tokens back to the format described in Sec. 3.1, i.e., the kinematic characteristics and relations $(b_i, j_i, l_i)$, and conditions $(P, c_s)_i$ for SDF decoding. To obtain the final object, cast joint transforms to rigid transforms to the input of SDF $\Omega$ and extract mesh using the Marching Cubes [35] algorithm.

**Training Objective.** The output of the articulation transformer are some tuples $([o, b, j, l, c_s, P]_i$ for the $i$-th node).

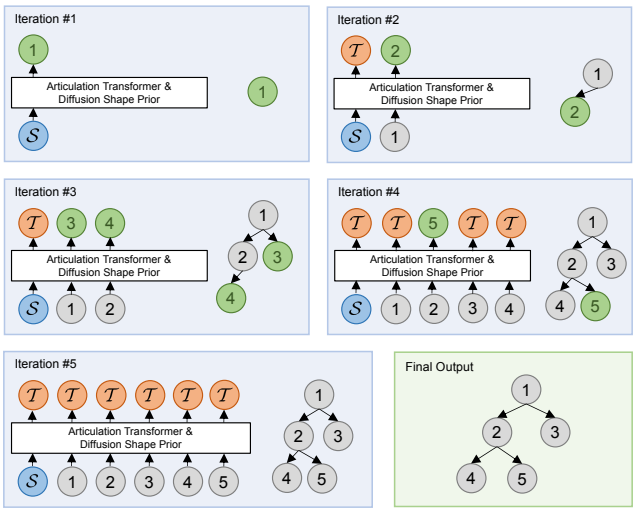


Figure 4. Each blue card represents a round in the predicting process. On each blue card, the left side shows the input given to the model and the expected output. The right side displays the tree structure of the articulated object formed after this prediction round, with green nodes indicating the nodes generated in this round. Orange nodes are terminal nodes.

Binary cross-entropy loss $L_o$ is employed to supervise $o$, which is the logits indicating whether this token is a terminal token. MSE loss $L_a$ is used to supervise the attributes $b, j, l, c_s$.

The training of the shape prior is conducted first. Upon completion, for each sub-part's geometry latent code $z$ in dataset, we compute the matrix $\mathcal{D}$ defined in Eq. (3).

$L_P$ denotes the loss function to supervise $P$:

$$L_P = \frac{1}{4}\sum_{i=0}^{3} D_{KL}\left(\mathcal{H}(P_{i,*})||\mathcal{H}(\mathcal{D}_{i,*})\right), \tag{8}$$

where $\mathcal{H}(P_{i,*})$ and $\mathcal{H}(\mathrm{D}_{i,*})$ are categorical distributions defined as:

$$\mathcal{H}(P_{i,*})(X=j) = \mathrm{softmax}(\{P_{i,l}\}_{l=1}^{N})_j, \tag{9}$$

$$\mathcal{H}(\mathcal{D}_{i,*})(X=j) = \mathrm{softmax}(\{\mathcal{D}_{i,l}\}_{l=1}^{N})_j. \tag{10}$$

The total loss function for Articulation Transformer is defined as:

$$L_{\text{trans}} = \beta_o L_o + \beta_P L_P + L_a, \tag{11}$$

where $\beta_o$ and $\beta_P$ are coefficients to balance the losses.

# 4. Experiments

## 4.1. Experimental Setup

We train the shape prior on PartNet [40] and PartNet-Mobility [65]. Although PartNet does not provide kinematic information, it still contributes to learning the geometry. The ArtFormer and other baseline models are trained exclusively on 6 categories (Storage Furniture, Safe, Oven, USB, Bottle and Washer) in PartNet-Mobility. For each articulated object in the dataset, we use Blender to create high-resolution thumbnails and employ ChatGPT-4o to generate corresponding descriptions, which are used for training the baselines. Detailed implementation steps can be found in Supplementary Material.

## 4.2. Baselines

Previous works, such as NAP [28] and CAGE [34], differ from ours in several key aspects. NAP uses a simple shape prior with hidden dimensions that are not consistent with ours, while CAGE retrieves shapes from a dataset rather than generating them. To enable a fair comparison, we made modifications to these original models. A cross-attention layer, with the same structure as ours, is added to enable them to process text instructions. The compared models are:

1. **NAP-128**: The original NAP model, modified to use our shape prior, generating a 128-dimensional shape code consistent with the original work.
2. **NAP-768**: Building on NAP-128, we increase the size of the shape code to 768 dimensions to align with our model.
3. **CAGE**: The original CAGE model, modified to retrieve outputs based on our shape prior.
4. **Ours**: Our proposed model and articulation parameterization, the geometry is generated through shape prior, bypassing the part retrieval.
5. **Ours-PR**: Building on our original model. We perform part retrieval after iterative decoding, as CAGE does, for a fair comparison.
6. **Ours-NAPSP**: Building on our original model, where the geometry is generated by the shape prior of NAP.

## 4.3. Metrics

We adopt the Instantiation Distance (ID) from NAP to evaluate the kinematic relations and geometry. A smaller ID value between two articulated objects indicates greater similarity, and vice versa. The following metrics are defined: (1) Minimum Matching Distance (**MMD**) describes the minimum distance between corresponding matches of generated objects and ground truths. (2) Coverage (**COV**) represents the ratio of ground truth instances that have a corresponding match in the generated objects, reflecting the similarity between the two distributions. (3) Nearest Neighbor Accuracy (**1-NNA**) measures the mismatch rate between generated and ground truth objects after performing 1-NN clustering.

To examine whether the generated objects are physically plausible, we also propose Part Overlapping Ratio (**POR**)

Table 1. Comparison of Generation Quality

| | Part Retrieval | POR ↓ $\times 10^{-2}$ | ID | | | HS | |
|---|---|---|---|---|---|---|---|
| | | | MMD ↓ | COV ↑ | 1-NNA ↓ | AL ↑ | DS ↑ |
| NAP-128 | ✗ | 0.805 | 0.0710 | 0.3085 | 0.7021 | 0.105 | 0.13 |
| NAP-768 | ✗ | 1.620 | 0.0632 | 0.3723 | 0.6543 | 0.093 | 0.12 |
| Ours-NAPSP | ✗ | 2.761 | 0.0375 | 0.4831 | 0.8315 | - | - |
| Ours | ✗ | **0.709** | **0.0292** | **0.5213** | **0.5266** | **0.459** | **0.67** |
| CAGE | ✓ | **0.251** | **0.0193** | 0.6064 | 0.5319 | 0.343 | 0.07 |
| Ours-PR | ✓ | 0.556 | 0.0214 | **0.6400** | **0.3950** | - | - |

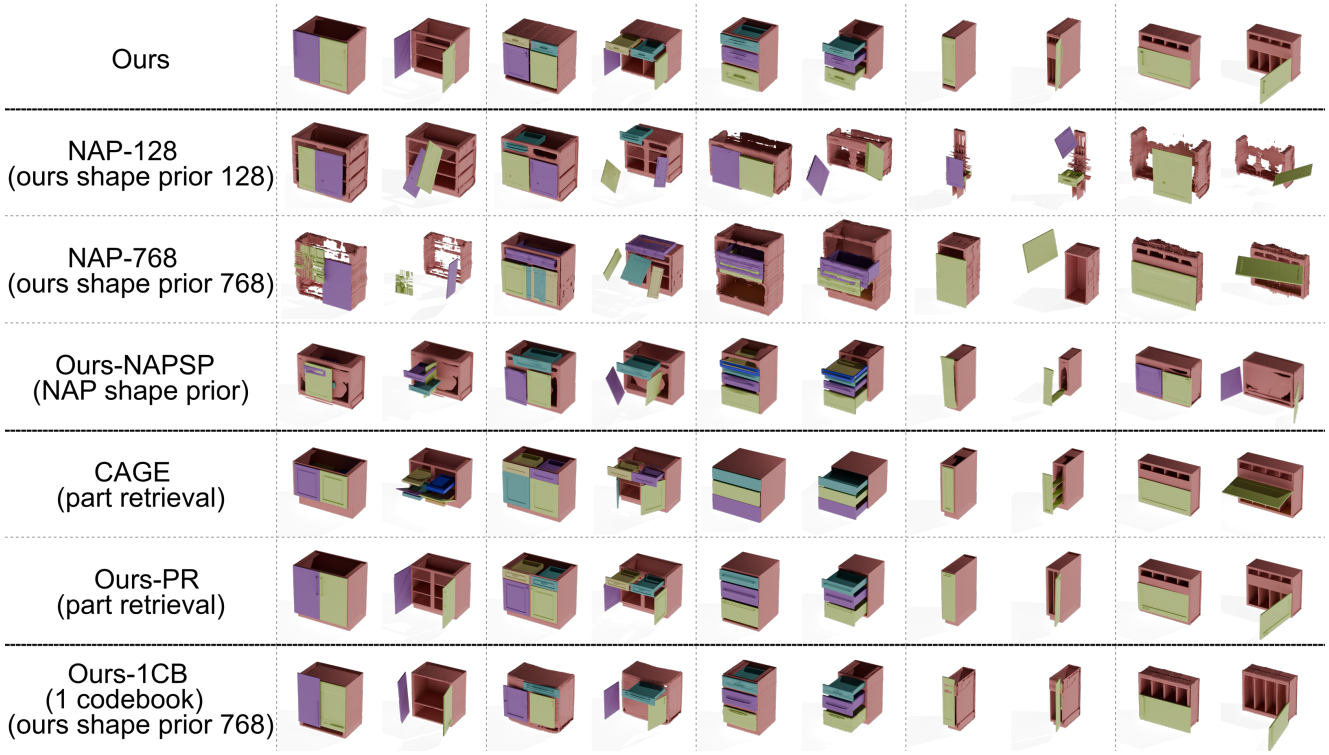


Figure 5. Qualitative comparison between *ArtFormer* and baselines (`Ours-1CB` will be discussed in Sec. 4.5). Our method is capable of generating high-quality geometry and accurate joint relations.

which assesses the degree of interpenetration between sub-parts. Let $E$ represent the articulated object. We define the interpenetration metric between any two sub-parts $P_1, P_2 \in E$ as the vIoU (volume Intersection over Union) of their corresponding 3D geometries:

$$I(P_1, P_2) = \frac{|\mathcal{G}_1 \cap \mathcal{G}_2|}{|\mathcal{G}_1 \cup \mathcal{G}_2|}, \tag{12}$$

where $\mathcal{G}_1$ and $\mathcal{G}_2$ represent the geometries of the sub-parts, respectively. Given a set of joint states $\mathcal{J}$, we can calculate the mean interpenetration between every pair of parts, denoted as $\mathrm{MI}(E, \mathcal{J})$. We uniformly sample $N_j = 10$ joint states $\mathcal{J}_1, \mathcal{J}_2, ..., \mathcal{J}_{N_j}$ within the limits and define the Part Overlapping Ratio as:

$$\mathrm{POR}(E) = \frac{1}{N_j} \sum_{i=1}^{N_j} \mathrm{MI}(E, \mathcal{J}_i). \tag{13}$$

A human study methodology (HS) is used to assess the alignment between generated objects and their text descriptions, as well as the diversity of the generated objects. $P = 20$ participants are presented with objects generated from $M = 20$ distinct descriptions, each by 4 baseline models. Participants select the object that best matches the description. In a separate task, they choose the most diverse group from 4 options, each containing objects gener-

ated from the same instruction by different baselines. This task is repeated $T = 5$ times. The alignment score (**AL**) and diversity score (**DS**) are defined as the mean win rate. Further details are provided in Supplementary Material.

### 4.4. Results

**Generation Quality and Diversity.** We evaluate the generation quality of baselines, as shown in Tab. 1. Since CAGE cannot directly generate geometry, comparisons are divided into two groups. The first includes NAP-128, NAP-768, and our model, which generate geometry features directly. The second approach consists of CAGE and ours-PR, which retrieve the suitable shape from the dataset to generate objects. Our model outperforms NAP on all metrics in the first group, producing more realistic articulated objects with less part interpenetration. In the second group, while CAGE achieves better MMD and POR, indicating superior object-level reconstruction, our model excels in COV and 1-NNA, capturing the overall distribution and generating more diverse objects. HS results in Tab. 1 suggest that our model produces greater diversity and aligns better with text instructions from an ordinary user's perspective.

**Image Guided Generation.** In our study, we replaced the original pre-trained text encoder [51] with a pre-trained image encoder [30] to validate the flexibility of our method to support various conditioning modalities. We utilized Blender [7] to render each object in the dataset as input images. The results of our experiment are shown in Fig. 6. Our model is capable of generating high-quality articulated objects from a single image. This outcome further demonstrates its potential to scale to more complex and multi-modality settings.

### 4.5. Ablations

In our ablation study (Tab. 2), we evaluated the impact of removing the tree position embedding (TPE) and shape prior (SP). Without the shape prior, the transformer must directly generate sub-part geometry, complicating feature generation, eliminating randomness, and significantly degrading all performance metrics. Removing the tree position embedding causes the model to lose positional information, increasing POR due to sub-part overlap during motion and reducing COV by impairing its ability to capture structural nuances and dataset distribution. We conduct the experiment with a single codebook (1 Codebook), and the performance drop is shown in Fig. 5 and Tab. 2.

## 5. Conclusion

We propose a novel method for modeling and generating 3D articulated objects, addressing limitations in diversity and usability. Representing articulated objects as a tree structure with nodes for rigid parts simplifies articulation parameterization, enabling part-level definition and generation. To ensure well-aligned yet diverse outputs, we develop a controllable shape prior and the Articulation Transformer, which captures articulation features effectively. A tree position embedding layer enhances part relationship modeling, supporting autoregressive generation. Our method achieves state-of-the-art performance, generating high-quality, diverse objects from text or image conditions.

**Limitations and Future Work.** (1) The limited dataset restricts the application to a small range of object types with few sub-part, preventing the full potential of the approach from being realized. Our model is expected to show more advantage when the number of parts is large ($> 10$) and varies (which are rare in existing dataset), due to transformer's capability to handle long sequences of varying lengths. Future work may explore this capability on a large scale. (2) Multi-modal instructions beyond text and im-

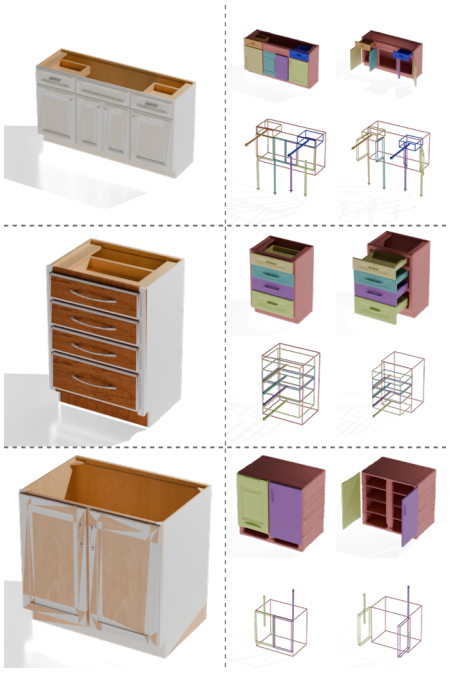

Figure 6. The figure displays 3 pairs of image condition inputs alongside the articulated object outputs produced by the model. Each pair contains a large image on the left as the input and generated articulated object on the right as the outputs.

Table 2. Ablation Studies with Reconstruction Quality

| | POR ↓ | ID | | |
|---|---|---|---|---|
| | $\times 10^{-2}$ | MMD↓ | COV↑ | 1-NNA↓ |
| Full | 0.709 | 0.0292 | **0.5213** | 0.5266 |
| No TPE | 1.170 | 0.0257 | 0.5000 | **0.5053** |
| No SP | 2.502 | 0.0339 | 0.4574 | 0.7606 |
| 1 Codebook | **0.687** | **0.0291** | 0.4948 | 0.5928 |

ages have not yet been explored, such as point cloud or joint structure of expected articulated object. Investigating diverse instruction formats could greatly enhance flexibility and usability of our method in practical application. (3) Capturing joint quantitative details in the text condition, such as rotation angles, is more challenging than joint type and geometry condition. Further research is needed to improve representation and learning of this data. (4) It is observed that increasing the number of object categories during training leads to a decline in geometry reconstruction quality, likely due to the limited generalization of SDF models. Methods from GenSDF [5] may enhance generalization, and we also recommend using modern 3D representations like 3D Gaussian Splatting [26] in future work.

**Acknowledgements.** This paper is a *pure student work*, as all authors are either undergraduate or PhD students. The computational resources were provided by Ningxia Westcloud Computing Technology Co., Ltd. (Damodel), which offered us certain discounts and waivers. Partial financial support was provided by Xiamen University Malaysia.

## References


[1] Sherwin Bahmani, Ivan Skorokhodov, Victor Rong, Gordon Wetzstein, Leonidas Guibas, Peter Wonka, Sergey Tulyakov, Jeong Joon Park, Andrea Tagliasacchi, and David B. Lindell. 4D-fy: Text-to-4d generation using hybrid score distillation sampling. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7996–8006, 2024. 3

[2] Dave Zhenyu Chen, Haoxuan Li, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. SceneTex: High-quality texture synthesis for indoor scenes via diffusion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21081–21091, 2024. 3

[3] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2Shape: Generating shapes from natural language by learning joint embeddings. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 100–116. Springer, 2019. 3

[4] 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*, abs/2405.11656, 2024. 2

[5] Gene Chou, Ilya Chugunov, and Felix Heide. Gensdf: Two-stage learning of generalizable signed distance functions. In *Proc. of Neural Information Processing Systems (NeurIPS)*, 2022. 9

[6] Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-SDF: Conditional generative modeling of signed distance functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2262–2272, 2023. 2, 3, 1

[7] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 5, 8

[8] Tianyuan Dai, Josiah Wong, Yunfan Jiang, Chen Wang, Cem Gokmen, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Automated creation of digital cousins for robust policy learning, 2024. 2

[9] Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018. 3

[10] Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. HyperDiffusion: Generating implicit neural fields with weight-space diffusion. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14254–14264, 2023. 2

[11] Rao Fu, Xiao Zhan, Yiwen Chen, Daniel Ritchie, and Srinath Sridhar. ShapeCrafter: A recursive text-conditioned 3d shape generation model. *Advances in Neural Information Processing Systems*, 35:8882–8895, 2022. 3

[12] Daoyi Gao, Yawar Siddiqui, Lei Li, and Angela Dai. Meshart: Generating articulated meshes with structure-guided transformers, 2024. 2

[13] Lin Gao, Jia-Mu Sun, Kaichun Mo, Yu-Kun Lai, Leonidas J. Guibas, and Jie Yang. SceneHGN: Hierarchical graph networks for 3d indoor scene generation with fine-grained geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8902–8919, 2023. 3

[14] Søren Hauberg and Kim Steenstrup Pedersen. Predicting articulated human motion from spatial processes. *International Journal of Computer Vision*, 94:317–334, 2011. 2

[15] Nick Heppert, Muhammad Zubair Irshad, Sergey Zakharov, Katherine Liu, Rares Andrei Ambrus, Jeannette Bohg, Abhinav Valada, and Thomas Kollar. CARTO: Category and joint agnostic reconstruction of articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21201–21210, 2023. 2

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[17] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 3

[18] Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pages 8867–8887. PMLR, 2022. 3

[19] Han Huang, Leilei Sun, Bowen Du, Yanjie Fu, and Weifeng Lv. GraphGDP: Generative diffusion processes for permutation invariant graph generation. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 201–210. IEEE, 2022. 3

[20] Yao-Hung Hubert Tsai, Liang-Kang Huang, and Ruslan Salakhutdinov. Learning robust visual-semantic embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3571–3580, 2017. 3

[21] Ajinkya Jain, Rudolf Lioutikov, Caleb Chuck, and Scott Niekum. ScrewNet: Category-independent articulation model estimation from depth images using screw theory. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 13670–13677. IEEE, 2021. 2

[22] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. 4

[23] Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International Conference on Machine Learning*, pages 10362–10383. PMLR, 2022. 3

[24] Heewoo Jun and Alex Nichol. Shap-E: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 3

[25] Guolin Ke, Di He, and Tie-Yan Liu. Rethinking positional encoding in language pre-training. In *International Conference on Learning Representations*, 2020. 5

[26] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 9

[27] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016. 3

[28] Jiahui Lei, Congyue Deng, Bokui Shen, Leonidas Guibas, and Kostas Daniilidis. NAP: Neural 3D articulated object prior. In *Advances in Neural Information Processing Systems*, 2024. 2, 6

[29] Hao Li, Guowei Wan, Honghua Li, Andrei Sharf, Kai Xu, and Baoquan Chen. Mobility fitting using 4d ransac. *Computer Graphics Forum*, 35(5):79–88, 2016. 2

[30] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pages 19730–19742. PMLR, 2023. 8, 3

[31] Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. Grains: Generative recursive autoencoders for indoor scenes, 2019. 3

[32] Jiayi Liu, Denys Iliash, Angel X. Chang, Manolis Savva, and Ali Mahdavi-Amiri. SINGAPO: Single image controlled generation of articulated parts in object. *arXiv preprint arXiv:2410.16499*, 2024. 2, 3

[33] Jiayi Liu, Manolis Savva, and Ali Mahdavi-Amiri. Survey on modeling of articulated objects. *arXiv preprint arXiv:2403.14937*, 2024. 1

[34] 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. 2, 3, 6

[35] William E. Lorensen and Harvey E. Cline. Marching Cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, page 163–169. Association for Computing Machinery, 1987. 5

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. 1

[37] Łukasz Maziarka, Agnieszka Pocha, Jan Kaczmarczyk, Krzysztof Rataj, Tomasz Danel, and Michał Warchoł. MolCycleGAN: a generative model for molecular optimization. *Journal of Cheminformatics*, 12(1):2, 2020. 3

[38] Niloy J. Mitra, Yong-Liang Yang, Dong-Ming Yan, Wilmot Li, and Maneesh Agrawala. Illustrating how mechanical assemblies work. *ACM Trans. Graph.*, 29(4), 2010. 2

[39] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. AutoSDF: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 306–315, 2022. 2

[40] 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*, 2019. 6, 1

[41] Carlota Parés Morlans, Claire Chen, Yijia Weng, Michelle Yi, Yuying Huang, Nick Heppert, Linqi Zhou, Leonidas Guibas, and Jeannette Bohg. AO-Grasp: Articulated object grasp generation. *arXiv*, 2023. 2

[42] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan L. Yuille, Nuno Vasconcelos, and Xiaolong Wang. A-SDF: learning disentangled signed distance functions for articulated shape representation. *ICCV*, pages 12981–12991, 2021. 2

[43] Matteo Ninniri, Marco Podda, and Davide Bacciu. Classifier-free graph diffusion for molecular property targeting. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 318–335. Springer, 2024. 3

[44] 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. 3

[45] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to-image synthesis. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 2

[46] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 2

[47] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4195–4205, 2023. 3

[48] Han Peng, Ge Li, Wenhan Wang, YunFei Zhao, and Zhi Jin. Integrating tree path in transformer for code representation. In *Advances in Neural Information Processing Systems*, pages 9343–9354, 2021. 3

[49] Han Peng, Ge Li, Wenhan Wang, YunFei Zhao, and Zhi Jin. Integrating tree path in transformer for code representation. In *Advances in Neural Information Processing Systems*, pages 9343–9354. Curran Associates, Inc., 2021. 5

[50] 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. 2, 3

[51] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 4, 5, 8, 1, 3

[52] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 4

[53] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016. 3

[54] 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 International Conference on Computer Vision*, pages 10684–10695, 2022. 3

[55] Vighnesh Shiv and Chris Quirk. Novel positional encodings to enable tree-based transformers. In *Advances in Neural Information Processing Systems*, pages 12058–12068, 2019. 3

[56] Vighnesh Shiv and Chris Quirk. Novel positional encodings to enable tree-based transformers. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 5

[57] Martin Simonovsky and Nikos Komodakis. GraphVAE: Towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*, pages 412–422. Springer, 2018. 3

[58] Jürgen Sturm, Cyrill Stachniss, and Wolfram Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41:477–526, 2011. 2

[59] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. 3

[60] Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete denoising diffusion for graph generation. In *International Conference on Learning Representations*, 2023. 3

[61] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. GraphGAN: Graph representation learning with generative adversarial nets. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2508–2515, 2018. 3

[62] Fangyin Wei, Rohan Chabra, Lingni Ma, Christoph Lassner, Michael Zollhoefer, Szymon Rusinkiewicz, Chris Sweeney, Richard Newcombe, and Mira Slavcheva. Self-supervised neural articulated shape and appearance models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3

[63] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. MarrNet: 3d shape reconstruction via 2.5d sketches. In *Advances in Neural Information Processing Systems*, pages 540–550, 2017. 2

[64] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 2

[65] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11097–11107, 2020. 6, 1, 3

[66] Weiwei Xu, Jun Wang, KangKang Yin, Kun Zhou, Michiel van de Panne, Falai Chen, and Baining Guo. Joint-aware manipulation of deformable models. *ACM Trans. Graph.*, 28 (3), 2009. 2

[67] Muhan Zhang, Shali Jiang, Zhicheng Cui, Roman Garnett, and Yixin Chen. D-VAE: A variational autoencoder for directed acyclic graphs. In *Advances in Neural Information Processing Systems*, pages 1586–1598, 2019. 3

[68] Cai Zhou, Xiyuan Wang, and Muhan Zhang. Unifying generation and prediction on graphs with latent graph diffusion. In *Advances in Neural Information Processing Systems*, 2024. 3

[69] Huaisheng Zhu, Teng Xiao, and Vasant G Honavar. 3M-Diffusion: Latent multi-modal diffusion for language-guided molecular structure generation. In *First Conference on Language Modeling*, 2024. 3

# ArtFormer: Controllable Generation of Diverse 3D Articulated Objects

## Supplementary Material

For the convenience of readers, the changes made during the rebuttal stage are highlighted in different colors.

## 6. Implementation Details

### 6.1. SDF Model

As we describe in Sec. 3.2, we firstly stack the PointNet $\Gamma$, $q(z|f)$, $p(f|z)$ and SDF Decoder $\Omega$, which is shown in Fig. 7. This stacked network is used to generate the latent code $z$ from the point cloud and decode the mesh from $z$. To strengthen the generalization of this network, we adopt a similar training method as [6]. During one training step of each sub-part, we randomly sample the point cloud which contains 4096 points, 16,000 query points $Q$, and compute the SDF value of $Q$. The training objective is $L(q, p, \Gamma, \Omega)$ as mentioned in Sec. 3.2.

### 6.2. Training Details

The training process employs the AdamW optimizer [36] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for all of the models.

**SDF Model.** We utilize articulated objects from two datasets, PartNet [40] and PartNet-Mobility [65], to train the SDF Model as displayed in Fig. 7. The training on a single NVIDIA 4090 GPU with a batch size of 24 takes approximately 8 hours for 1.5k epochs.

**Diffusion Shape Prior.** We use pretrained SDF model to generate the geometry latent code $z$ for each sub-part, which is used to train diffusion shape prior. The mini-encoders are implemented as 4-layer multilayer perceptron (MLP) networks. The dimensions of $c_g$ and $c_s$ are set to 64 and 32, respectively. Furthermore, the diffusion denoiser comprises 4 blocks of normal transformers with self-attention layers. The training on a single NVIDIA 4090 GPU with a batch size of 64 takes approximately 11 hours for 4k epochs.

**Articulation Transformer.** The PartNet-Mobility dataset is exclusively used for training the Articulation Transformer. This network is composed of 8 transformer blocks, each with 8 attention heads and a token dimension of 1024. For the pre-trained text encoder, the encoder component of the T5 model [51] is employed. The full training process takes 16 hours on a single NVIDIA 4090 GPU with a batch size of 128 for 17k epochs.

### 6.3. Text Condition Generation Using GPT-4o

The training and testing of our model rely on text descriptions as conditions that highlight both kinematic and geometry features of articulated objects. Using prompt engineering, GPT-4o (`gpt-4o-2024-08-06`) excels in image-to-text generation, producing detailed and precise descriptions for each object. The prompt we provide consists of two parts, $\mathcal{P}_{\text{base}} : \mathcal{P}_{\text{len},i}$, where : denote concatenation. The content of $\mathcal{P}_{\text{base}}$ is shown in Fig. 8, and $\mathcal{P}_{\text{len},i}$, used to control the expected output length, is shown in Fig. 9. For each object in the dataset, we supply GPT-4o with its corresponding snapshot and a series of text prompts $\{(\mathcal{P}_{\text{base}} : \mathcal{P}_{\text{len},i})\}_{i=0}^{3}$ sequentially, generating 4 text conditions of varying lengths for the same object.

In some cases, GPT-4o may fail to produce a valid description (e.g., returning *"I'm sorry, I can't assist with that."*), with a failure rate of 26.10%. In the final dataset for text-guided generation, such failed descriptions are excluded.

### 6.4. Human Study

We randomly selected 20 participants without prior knowledge of articulated object generation for the human study. Each participant completed the same questionnaire, divided into two sections for the alignment and diversity experiments, containing 20 and 5 questions, respectively.

In the first section, participants evaluated four sets of images generated by different models from the same text instruction, with each set including three snapshots corresponding to openness ratios (linear interpolation between the predicted joint limits) of 0, 0.5, and 1. The text instruction used for generation was provided. The question is: *select the set that best matched the described articulation characteristics and aligned with reality.*

In the second section, participants reviewed four snapshots with an openness ratio of 0, generated from the same instruction, repeated four times. The question is: *select the set that shows the richest diversity while remaining consistent with reality.*

## 7. Additional Experiments and Results

### 7.1. Novel Shape Generation

We conducted an experiment inspired by Diffusion-SDF to demonstrate that our shape prior, guided by an articulation transformer, can generate new geometry shapes that never appear in the dataset. We used our model to produce various objects and dissected them into sub-parts. Then, we calculated the Chamfer Distance between each sub-part and those in the training set and ranked them from nearest to farthest. The results, shown in Fig. 10, indicate that the sub-parts generated by our model are distinct from those in the training set, confirming the model's ability to create novel geometry shapes.

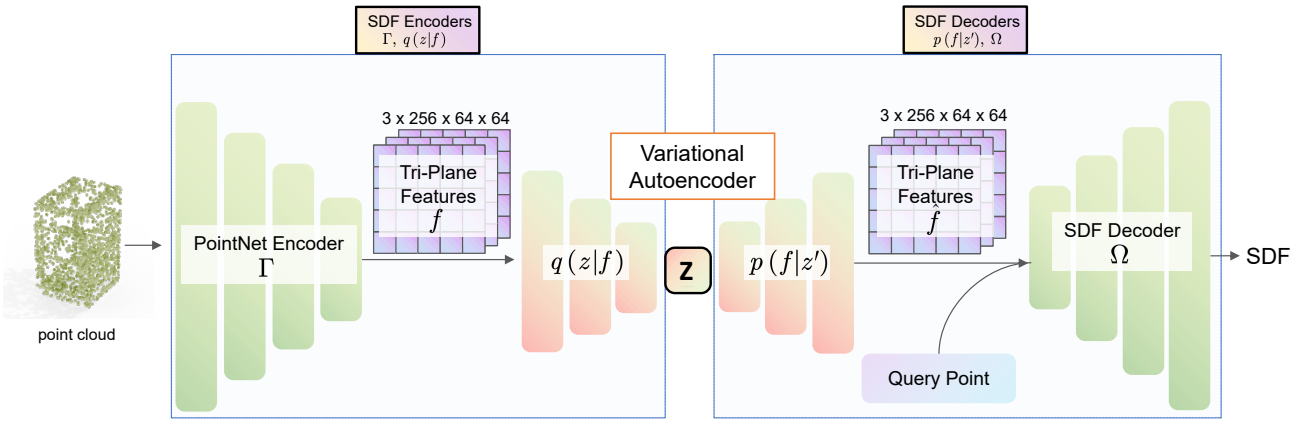


Figure 7. Training pipline for PointNet $\Gamma$, $q(z|f)$, $p(f|z)$ and SDF decoder $\Omega$. For brevity, we refer to $\Gamma$ and $q(z|f)$ collectively as SDF encoders. And, similarly, we refer to $p(f|z)$ and SDF decoder $\Omega$ as SDF decoders.

```
This is a type of [Storage Furniture/Bottle/Toilet...].
    Please focus on the shape of each part and its articulation
characteristics, and describe the possible motion characteristics
and shape of each part.
    In the given image, there are different colored parts that can
move relative to each other.
    In your description, you should ignore the color, texture, and
other non-structural features.
```

Figure 8. Prompt for GPT-4o to generate text description for objects.

```
[
    "You can describe it in detail with more sentences.",
    "You can describe it with fews sentences.",
    "You can describe it with only one sentence.",
    "You can describe it with with only fews words.",
]
```

Figure 9. Prompt Used to Restrict the Length of Output.

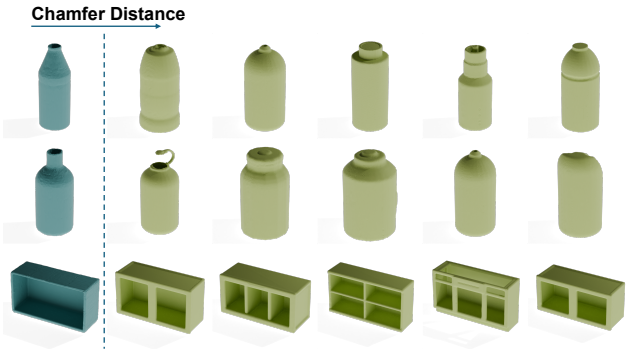


Figure 10. The first column (blue) shows shapes generated by our framework. The subsequent columns (green) are sub-parts retrieved from the training set, ordered according to increasing chamfer distance from the generated sub-parts.

### 7.2. Text Condition Attention

To verify the weights of text in the transformer's cross-attention mechanism, we provided a brief description and calculated the average text token weight across each cross-attention layer. As shown in Fig. 11, the words with higher attention weights describe the main sub-parts ('drawers' and 'doors') of the object and the kinematic feature ('slide' and 'swing') of these sub-parts. This confirms that our cross-attention mechanism effectively establishes relationships, allowing text conditioning to guide the generation of articulated objects.

### 7.3. Editing of Articulated Objects

To demonstrate the flexibility of autoregressive generation achieved through iterative decoding, we employed ArtFormer to edit existing articulated objects. In iterative decoding, each iteration generates a child node for each input node. This allows us to remove specific sub-parts from articulated objects and input desired text conditions, enabling the system to regenerate the missing sub-parts based on the provided text. In our experiment, we removed sub-parts from several objects in the training dataset and used ArtFormer to regenerate these incomplete parts based on alternative text instructions. The results are shown in Fig. 12.

### 7.4. Text Guided Generation

Additional visualization results are provided in Fig. 13 to illustrate the text guided generation for articulated objects using ArtFormer.

A quantitative evaluation is conducted to assess text-guided generation alignment, complementing previous human study results. Inspired by Park et al. [45], we adopt CLIP-R precision to measure the alignment between generated objects and instruction text. Snapshots are created for each generated object at an openness ratio of 0. The CLIP-R precision is then computed using these images and text instructions, leveraging the openai/clip-vit-large-patch14 model. The results for R=10 are presented in Tab. 3.

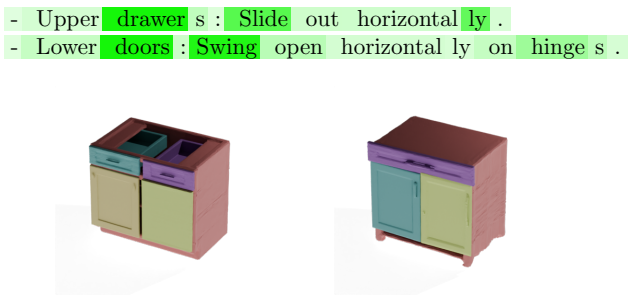


Figure 11. The text in the upper part of the figure represents the input text given to the model, where the intensity of the colors indicates the strength of attention; darker colors correspond to higher attention weights, while lighter colors indicate lower attention weights. The lower part of the figure displays the articulated objects generated by the model based on the input text.

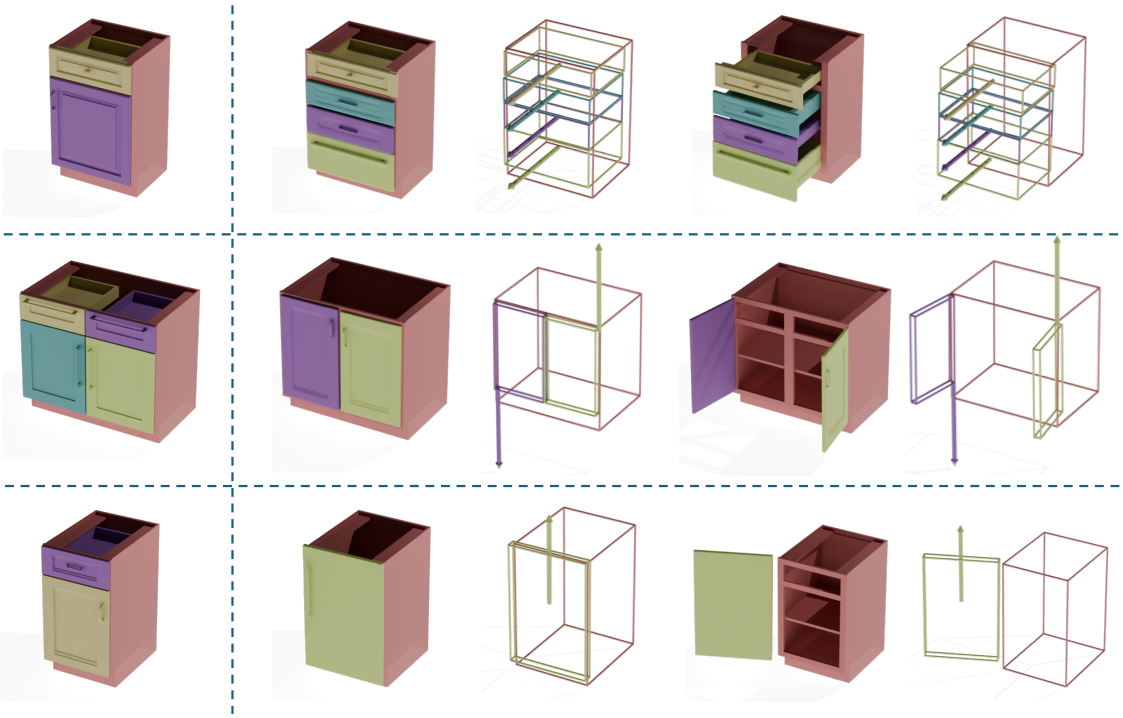

Figure 12. The figure illustrates the process of editing existing articulated objects using ArtFormer. (1) For the first object, the cabinet door (purple) is removed, and the following text condition is provided: *This storage furniture consists of a rectangular frame with multiple horizontally aligned drawers that slide in and out on tracks*. The edited object is displayed on the right. (2) For the second object, the drawers (purple and yellow) and cabinet doors (green and blue) are removed. The condition is: *This storage furniture consists of a rectangular base with two front panels that pivot on vertical hinges to open outward*. (3) For the third object, the drawer (purple) and cabinet door (yellow) are removed. The condition is: *Rectangular frame: stationary base. Front panel: hinged door, pivots outward*.

### 7.5. Image Guided Generation

Preliminary experiments of ArtFormer's capability to generate articulated objects based on a single image, achieved by substituting the pretrained text encoder [51] with a pretrained image encoder [30], as discussed Sec. 4.4. Rendered images of articulated objects from PartNet-Mobility [65] using Blender, used as image conditions for generating articulated objects in the images with ArtFormer. The results are displayed in Fig. 14. In addition, we employ real-world photographs as the image condition to generate articulated objects. The results are illustrated in Fig. 15.

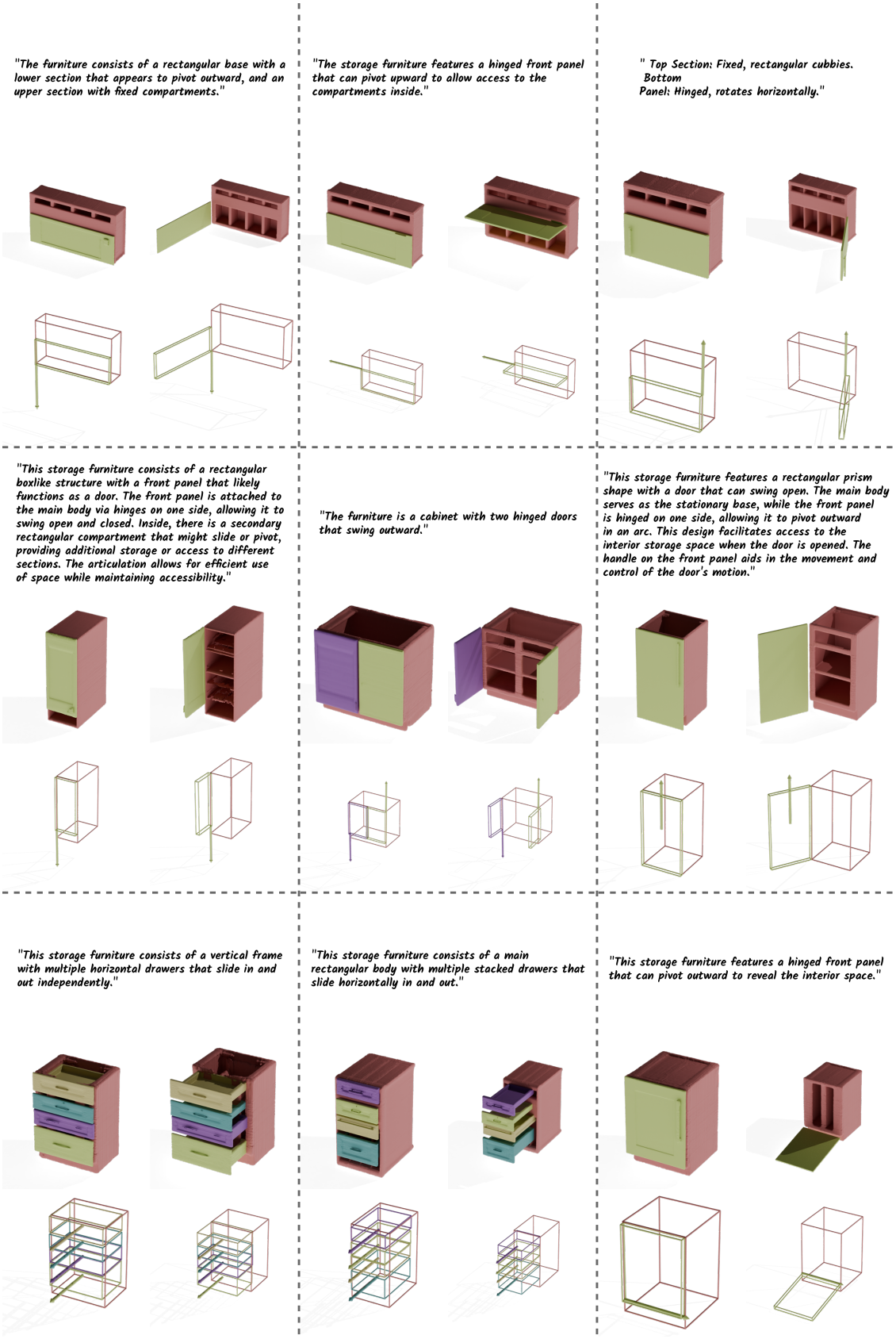


Figure 13. *Continued on next page*

*Continued from previous page*

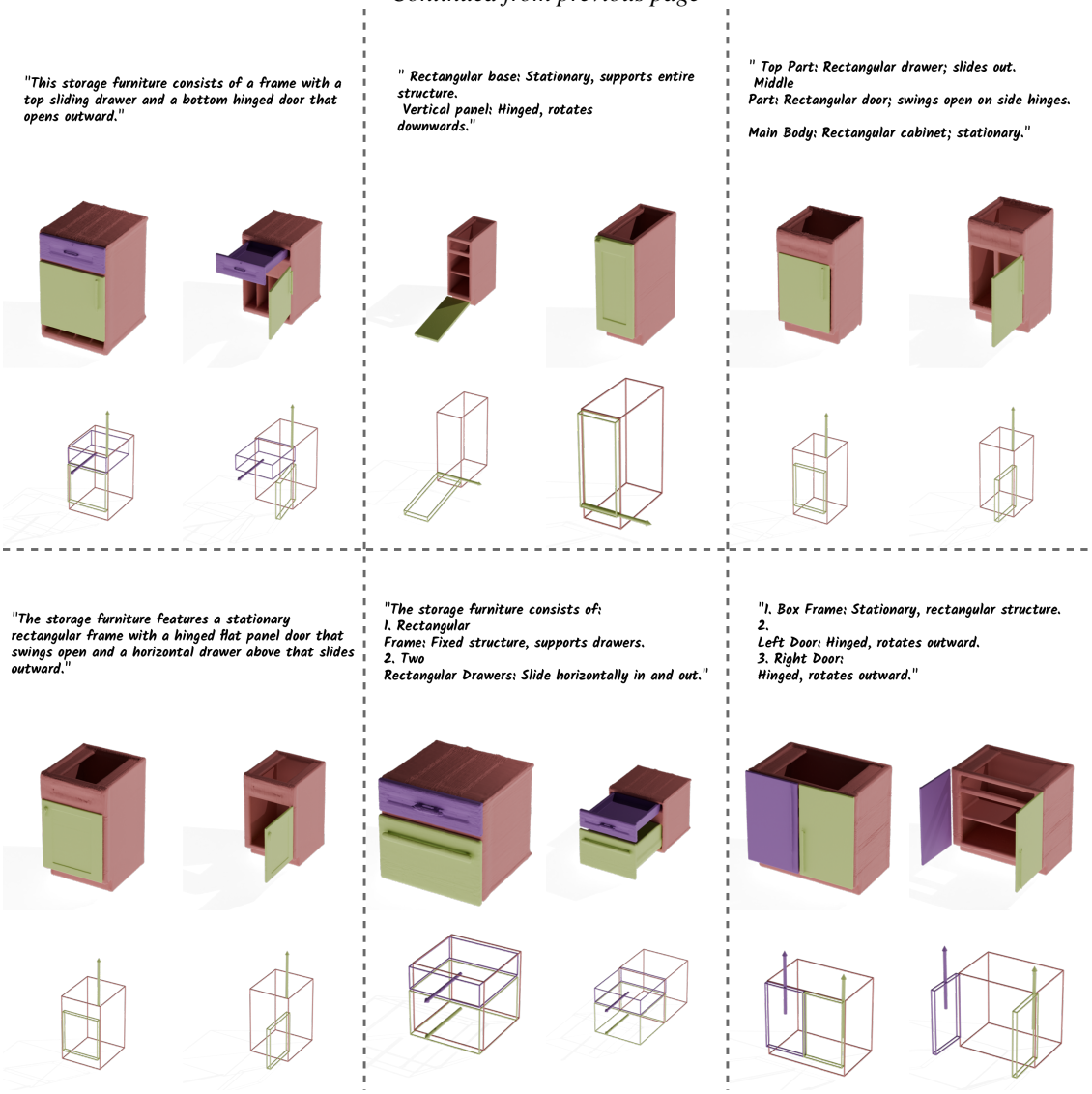


Figure 13. The figure presents 15 pairs of input text conditions and articulated objects generated by ArtFormer. For each pair, the text on the top serves as the input text condition, while the bottom side illustrates the articulated object output, showcasing the predicted motion relationship with the joint in both the fully closed and fully open states.

Table 3. Text-Guided Object Alignment Results

| | CAGE | NAP-128 | NAP-768 | Ours |
|---|---|---|---|---|
| CLIP-R@10↑ | 0.1429 | 0.1648 | 0.1319 | **0.2198** |

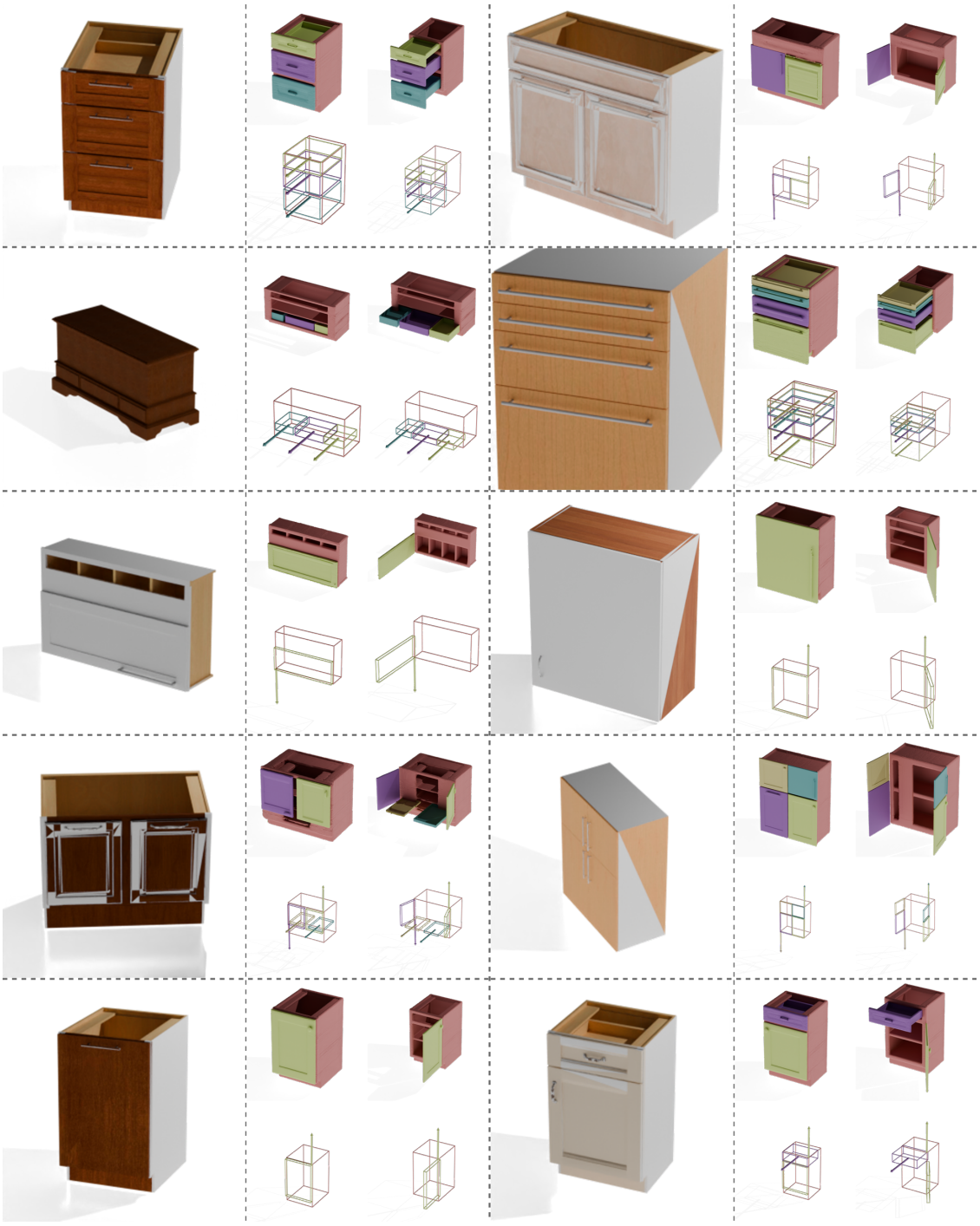

Figure 14. The figure presents 10 pairs of input images and articulated objects generated by ArtFormer. For each pair, the larger image on the left serves as the input image condition, while the right side illustrates the articulated object output, showcasing the predicted motion relationship with the joint in both the fully closed and fully open states.

Figure 15. We present 4 pair of photographs for articulated objects from the real world (shown on the left side of each pair). Using our model, we generate these articulated object and make the visualization of them (shown on the right side of each pair).