MeshArt: Generating Articulated Meshes with Structure-Guided Transformers

Supplementary Material

In this supplementary document, we provide additional details about MeshArt. In Sec. 7, we give more implementation details of our method and loss functions. We elaborate our data annotation process in Sec. 8. We also include additional quantitative comparisons in Sec. 9. We encourage readers to watch the supplemental video to see more articulated object generations in action.

7. Method Details

We use VQVAEs to model both part articulations and mesh geometries for our hierarchical transformers. Our structure VQVAE encodes extra part-level features (e.g., semantics, geometry feature, and articulation joint), alongside vertex locations, into a compact latent space for articulation-aware structure generation, while our geometry VQVAE predicts additional junction face probabilities for coherent part mesh generation.

7.1. Structure VQ-VAE

The structure VQ-VAE encodes and quantizes features of bounding box triangles to learn a structured embedding space for articulated object structures. We construct a graph for the triangles by treating each triangle face as a node and connecting neighboring faces with undirected edges. The input node features include positionally encoded triangle coordinates, face area, edge angles, and face normal vectors. These features are concatenated with part semantic, geometry, and articulation attributes projected onto the triangle nodes. The combined features are processed through 4 SAGEConv [1] graph convolutional layers, extracting a feature vector of dimension 768 for each triangle.

At the bottleneck, these embeddings are quantized using a codebook of size 8192, enabling a compact representation of the structure. The decoder reconstructs triangle locations by predicting the logits of discretized coordinates, where both triangle and joint locations are mapped to a uniform grid of size 128^3.

Instead of directly predicting a discrete part semantic label, the structure VQ-VAE decoder regresses a continuous semantic feature vector from CLIP. The class label is then determined by computing the cosine similarity between the predicted feature vector and the CLIP features of a predefined set of part labels.

The decoder will output a set of joint information per triangle. To obtain a single set of joint predictions per part, the outputs are averaged across all triangles within the part.

This architecture effectively learns quantized embeddings for articulated object structures. These embeddings serve as the basis for the structure transformer, enabling the autoregressive generation of object structures with articulations.

Loss Functions. As the triangle coordinates and joint locations are discretized, their reconstruction loss can be formulated as a cross-entropy loss:

$$L_{recon} = \sum_{n=1}^{N} \sum_{k=1}^{128} \log \mathbf{P}_k, \tag{5}$$

with n being the face index and $\mathbf{P}_k$ representing the predicted probability distribution over the coordinate bins. For part i, its semantic feature $\mathbf{l}_i$ and geometry feature $\mathbf{g}_i$ are supervised using L_2 regression loss:

$$L_{regression} = ||\mathbf{y}_i - \hat{\mathbf{y}}_i||_2, \tag{6}$$

with $y_i, \hat{y}_i$ being the ground truth and predicted feature vectors.

7.2. Structure Transformer

We use a decoder-only transformer that has a standard GPT-2 architecture, *i.e.*, 12 multi-headed self-attention layers, 12 heads, 768 as feature width, with a context length of 4608. The transformer is trained with cross-entropy loss for next-token index prediction.

7.3. Geometry VQ-VAE

The Geometry VQ-VAE encodes mesh triangle features using an architecture similar to the Structure VQ-VAE. Input triangle features, such as positional encoding, normals, and edge attributes, are processed through 4 SAGEConv [1] layers to extract feature embeddings of dimension 768. These embeddings are quantized at the bottleneck using a vector quantization module with codebook size of 16384, enabling compact and efficient representation.

The 1D-ResNet decoder reconstructs the discretized triangle coordinates by minimizing a cross-entropy loss over a uniform grid. To enforce spatial and structural coherence between parts, the geometry decoder includes an additional channel that predicts the probability of each triangle being a junction triangle, i.e., triangles at the near boundary between adjacent parts. This prediction is supervised with a binary classification loss.

By incorporating junction triangle prediction, the Geometry VQ-VAE not only reconstructs accurate triangle meshes of the target part, but also learns the connectivity information cross parts, supporting smooth articulation and consistent geometry generation.

8. Data Annotation

To effectively learn the distribution of articulated objects, we extend PartNet [4], the largest dataset with object part annotations, by augmenting it with joint information. This augmentation significantly increases the diversity of articulated objects compared to the commonly used PartNet-Mobility [7].

Part Canonicalization. To ensure consistent and meaningful articulation properties, we canonicalize joint annotations. For prismatic joints, all locations are set to the origin of the object's coordinate system. For revolute joints, we address inconsistencies in part orientations, for instance, chair wheels often have arbitrary orientations in the original dataset, resulting in misaligned revolute joints. To canonicalize these, we rotate each wheel around its vertical axis to align their orientations consistently, as shown in Fig. 7.

Joint Location Generation. For storage furniture and tables, revolute joints are typically located at the "hinge" of an articulated part, often corresponding to one of the four bounding box sides of the part. To automate this process, we generate four hypotheses for joint locations based on the bounding box configuration of the articulated part. An interactive viewer is then used to select the most reasonable joint location, as illustrated in Fig. 8.

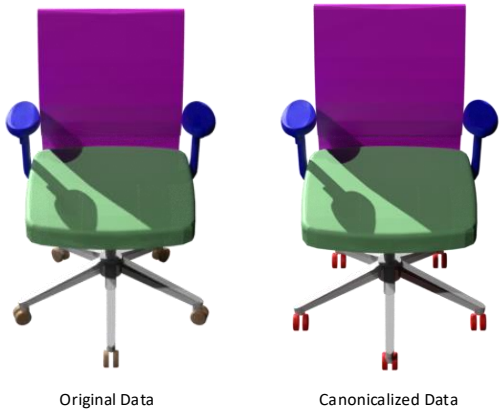

Figure 7. We canonicalize the orientation of different articulated parts for consistent joint annotation.

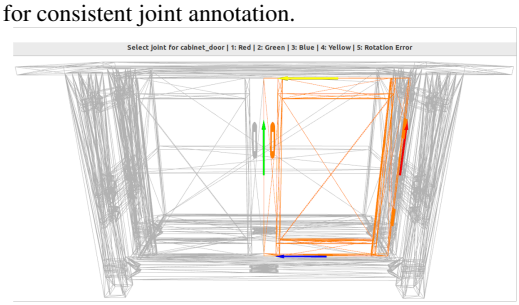

Figure 8. Given a target part, our viewer visualizes the generated joint hypotheses for selection.

Class	Method	COV↑	MMD↓	1-NNA	FID↓	KID↓
Chair	NAP [2]	20.9	5.4	97.3	212.6	0.207
	MeshGPT [6]	**34.5**	4.2	**81.8**	24.0	**0.011**
	MeshArt	27.3	**3.8**	85.8	**23.8**	0.013
Table	NAP [2]	20.0	5.8	96.5	243.8	0.231
	MeshGPT [6]	**43.0**	2.9	**69.9**	**14.5**	**0.005**
	MeshArt	33.4	**2.8**	77.9	15.1	0.007
Storage Furniture	NAP [2]	25.3	2.9	92.4	162.4	0.142
	MeshGPT [6]	38.7	2.3	81.6	9.3	**0.002**
	MeshArt	**40.9**	**2.0**	**78.7**	**8.6**	0.003

Table 6. Quantitative comparison on the task of unconditional mesh generation on a subset of categories from the PartNet [4] dataset. MMD values are multiplied by 10^3. We evaluate the mesh quality at the resting state for all methods. We outperform the baselines in shape quality, visuals, and compactness metrics.

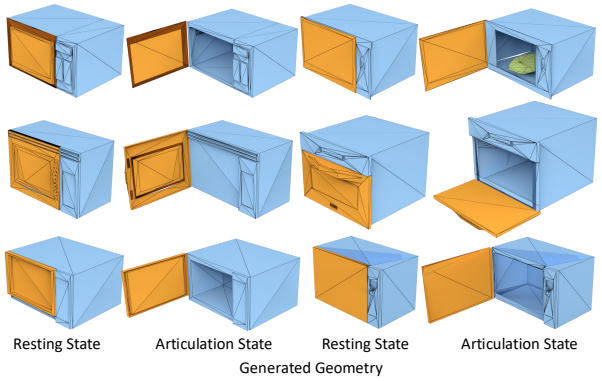

Figure 9. Our method can generate sharp geometry and realistic articulations for microwaves.

Joint Verification. To validate the joint annotations, we render the object at various articulation states and visually inspect the plausibility of the generated motions. This step ensures the accuracy of joint locations and their associated articulation properties, providing high-quality annotations for articulated objects.

9. Additional Results

Quantitative Comparison at Resting State. We compare the mesh generation quality of our method with NAP [2], and the state-of-the-art direct mesh generation approach, MeshGPT [6]. Since MeshGPT does not predict object part and articulation information, the evaluation is performed on generated meshes in their canonical resting state. As shown in Tab. 6, we also achieve comparable COV scores to MeshGPT while outperforming the MMD score, indicating higher fidelity in the generated shapes. Notably, our method shows significant mesh generation improvement over NAP on all metrics.

Additional Categories. We show additional results on microwaves in Fig. 9.

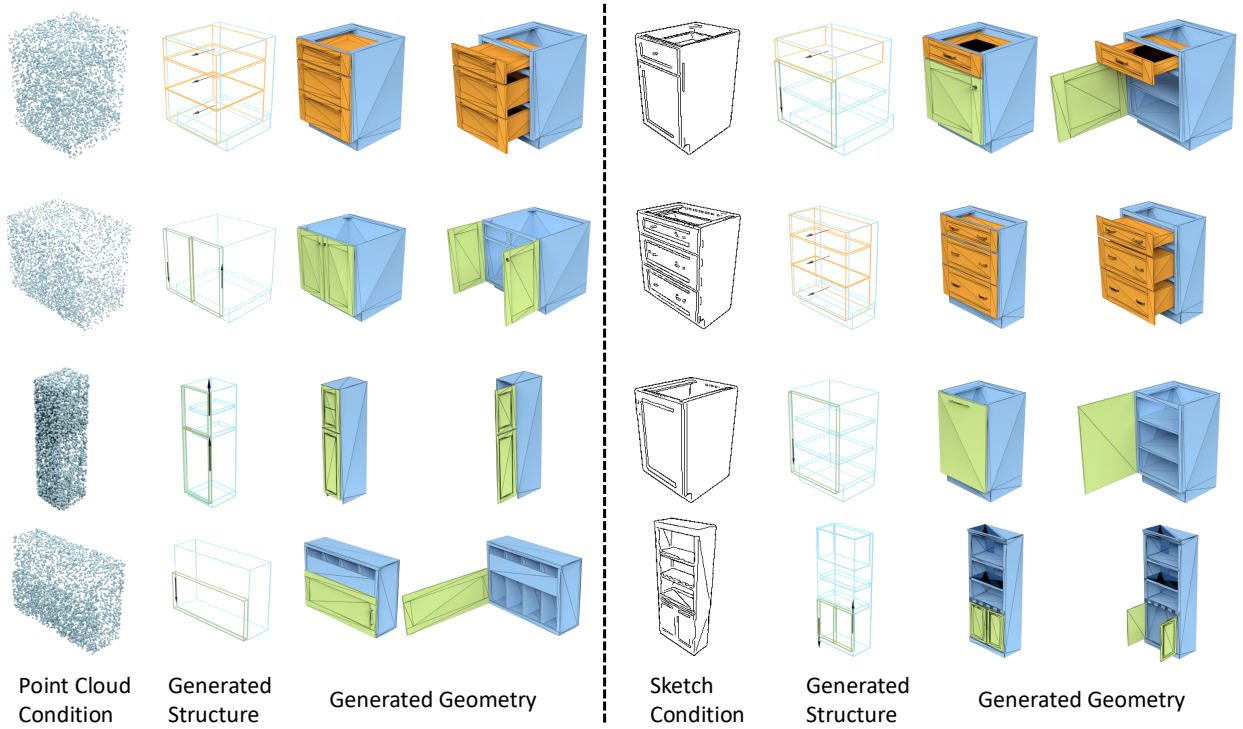

Figure 10. Conditional Generation. Articulated structures and geometry are generated conditioned on point clouds or sketches.

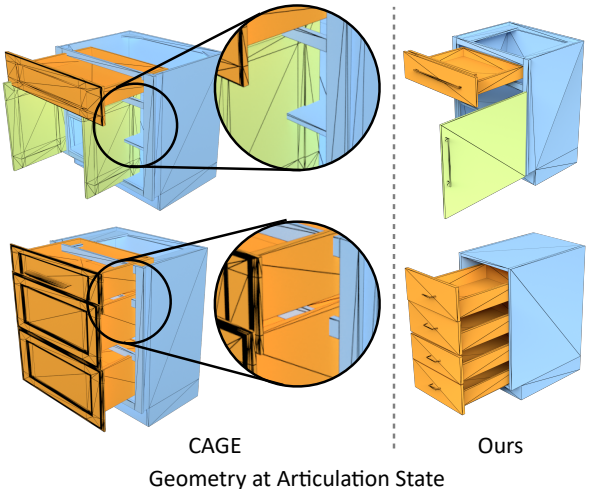

Figure 11. Visual comparison with CAGE. Our method can generate coherent shapes with sharp geometry.

Qualitative Comparison with CAGE. We compare MeshArt with CAGE [3], which retrieves part geometry based on conditionally generated articulated structures. As shown in Fig. 11, our method achieves coherent shape synthesis while achieving sharp geometry details, avoiding undesired part collision and inconsistencies between parts.

Conditional Generation. MeshArt can generate articulated structure and geometry conditioned on point clouds or sketch images. We extract input features using Michelangelo [8] for 3D point clouds and Radio [5] for sketches. A linear layer projects these features to match the structure transformer Φ_s's feature space, appending them to the beginning of the structure sequence. The transformer Φ_s then learns to generate articulation-aware structure bounding boxes. Since the structure codebook $\mathcal{C}_s$ remains fixed, the geometry transformer Φ_g requires no finetuning. As shown in Fig. 10, given the 3D/2D conditions, our method can generate plausible articulated structures, which then guide faithful synthesis of part geometry.

References

[1] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 1

[2] Jiahui Lei, Congyue Deng, William B Shen, Leonidas J Guibas, and Kostas Daniilidis. Nap: Neural 3d articulated object prior. *Advances in Neural Information Processing Systems*, 36:31878–31894, 2023. 2

[3] 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. 3

[4] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019. 2

[5] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation

model reduce all domains into one. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12490–12500, 2024. 3

[6] Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19615–19625, 2024. 2

[7] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11097–11107, 2020. 2

[8] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in Neural Information Processing Systems*, 36, 2024. 3