MagicArticulate: Make Your 3D Models Articulation-Ready

Chaoyue Song[1,2], Jianfeng Zhang[†2], Xiu Li[2], Fan Yang[1], Yiwen Chen[1], Zhongcong Xu[2],
Jun Hao Liew[2], Xiaoyang Guo[2], Fayao Liu[3], Jiashi Feng[2], Guosheng Lin[†1]
[1]Nanyang Technological University [2]ByteDance Seed
[3]Institute for Inforcomm Research, A*STAR

Abstract

With the explosive growth of 3D content creation, there is an increasing demand for automatically converting static 3D models into articulation-ready versions that support realistic animation. Traditional approaches rely heavily on manual annotation, which is both time-consuming and labor-intensive. Moreover, the lack of large-scale benchmarks has hindered the development of learning-based solutions. In this work, we present MagicArticulate, an effective framework that automatically transforms static 3D models into articulation-ready assets. Our key contributions are threefold. First, we introduce Articulation-XL, a large-scale benchmark containing over 33k 3D models with high-quality articulation annotations, carefully curated from Objaverse-XL. Second, we propose a novel skeleton generation method that formulates the task as a sequence modeling problem, leveraging an auto-regressive transformer to naturally handle varying numbers of bones or joints within skeletons and their inherent dependencies across different 3D models. Third, we predict skinning weights using a functional diffusion process that incorporates volumetric geodesic distance priors between vertices and joints. Extensive experiments demonstrate that MagicArticulate significantly outperforms existing methods across diverse object categories, achieving high-quality articulation that enables realistic animation. Project page: https://chaoyuesong.github.io/MagicArticulate.

1. Introduction

The rapid advancement of 3D content creation has led to an increasing demand for articulation-ready 3D models, especially in gaming, VR/AR, and robotics simulation. Converting static 3D models into articulation-ready versions traditionally requires professional artists to manually place skeletons, define joint hierarchies and specify skinning weights, which is both time-consuming and demands significant expertise, making it a major bottleneck in modern content creation pipelines.

To address these issues, various automatic approaches for skeleton extraction have been proposed, which can be categorized into template-based [3, 19] and template-free methods [2, 14, 34, 35]. Template-based methods, like Pinocchio [3], fit predefined skeletal templates to input shapes. While they achieve satisfactory results for specific categories like human characters, they struggle to generalize to objects with varying structural patterns. Moreover, these methods mostly rely on distance metrics between joints and vertices for skinning weight prediction, which often fail on shapes with complex topology. Many template-free methods [2, 4, 14, 21, 29] extract curve skeletons from meshes or point clouds using shape medial axis or the centerline of shapes, but often produce densely packed joints that are unsuitable for animation. Recent deep learning methods like RigNet [35] have shown promise in predicting skeletons and skinning weights directly from input shapes. However, they rely heavily on carefully crafted features and make strong assumptions about shape orientation, limiting their ability to handle diverse object categories. These limitations stem from two fundamental challenges: the lack of a large-scale, diverse dataset for training generalizable models, and the inherent difficulty in designing an effective framework capable of handling complex mesh topologies, accommodating varying skeleton structures, and ensuring the coherent generation of both accurate skeletons and skinning weights.

To overcome these challenges, we first introduce Articulation-XL, a large-scale dataset containing over 33k 3D models with high-quality articulation annotations carefully curated from Objaverse-XL [9, 10]. Built upon this benchmark, we propose MagicArticulate, a novel framework that addresses both skeleton generation and skinning weight prediction. Specifically, we reformulate skeleton generation as an auto-regressive sequence modeling task, enabling our model to naturally handle varying numbers of bones or joints within skeletons across different 3D models. For skinning weight prediction, we develop a functional diffusion framework that learns to generate smoothly

† Corresponding authors.

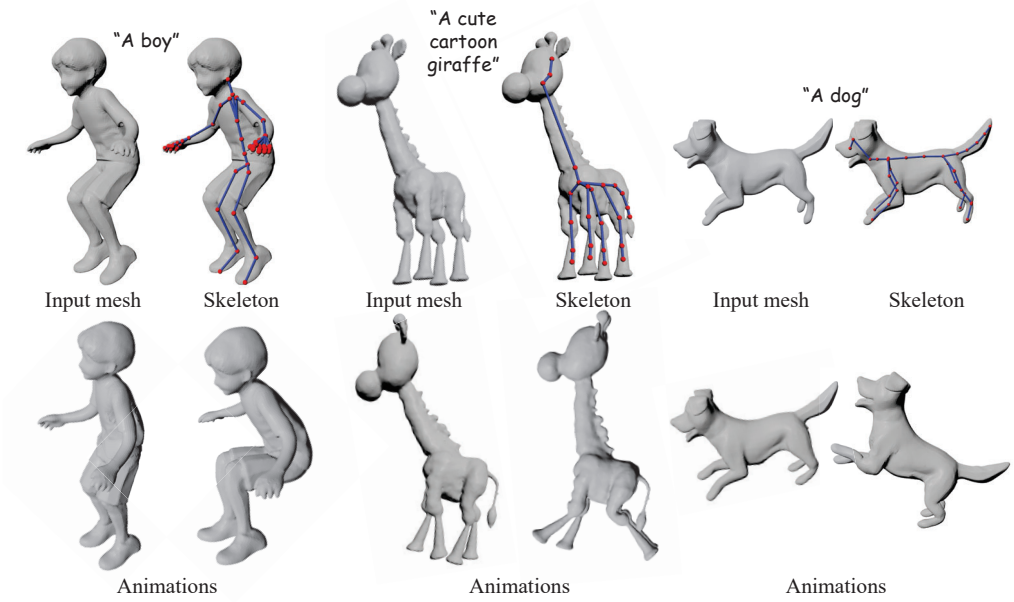


Figure 1. **Given a 3D model, MagicArticulate can automatically generate the skeleton and skinning weights, making the model articulation-ready without further manual refinement.** The input meshes are generated by Rodin Gen-1 [42] and Tripo 2.0 [1].

transitioning skinning weights over mesh surfaces by incorporating volumetric geodesic distance priors between vertices and joints, effectively handling complex mesh topologies that challenge traditional geometric-based methods. These designs demonstrate superior scalability on large-scale datasets and generalize well across diverse object categories, without requiring assumptions about shape orientation or topology.

Extensive experiments on our Articulation-XL and ModelsResource [31] collected by Xu et al. [34, 35] demonstrate the effectiveness of MagicArticulate in both skeleton generation and skinning weight prediction. The proposed methods also generalize well to 3D models from various sources, including artist-created assets, and models generated by AI techniques. With the generated skeleton and skinning weights, our method automatically creates ready-to-animate assets that support natural pose manipulation without manual refinement (Figure 1), particularly beneficial for large-scale animation content creation.

Our key contributions include: (1) The first large-scale articulation benchmark containing over 33k models with high-quality articulation annotations; (2) A novel two-stage framework that effectively handles both skeleton generation and skinning weight prediction; (3) State-of-the-art performance and demonstrated practicality in real-world animation pipelines.

2. Related works

2.1. Skeleton generation

There are two categories of methods for creating skeletons in 3D models. The first category relies on predefined templates [3, 19] or additional annotations [8, 15, 18, 36]. Pinocchio [3] is a pioneering method for automatically extracting an animation skeleton from an input 3D model. It fits a predefined skeleton template to the 3D model, evaluating the fitting cost for different templates and selecting the most suitable one for a given model. Li et al. [19] proposed a deep learning-based method to estimate joint positions for a given human skeletal template. However, these template-based methods are limited to rigging characters whose articulation structures are compatible with the predefined templates, making it difficult to generalize to objects with distinct structures.

There are also methods that rely on additional inputs or annotations to generate skeletons for 3D models, including point cloud sequences [36], mesh sequences [8, 18], and manual annotations [15]. Additionally, recent works [27, 28, 37, 39–41] have focused on learning the joints and bones of articulated objects directly from videos to reconstruct object motion. In contrast, our approach aims to generate skeletons using only 3D models as input.

The second category consists of template-free methods that operate without relying on predefined templates or additional annotations. Many approaches [2, 4, 14, 21, 29] are designed to extract curve skeletons from meshes or point clouds by utilizing the medial axis or the centerline of shapes. These methods often result in densely packed joints that are unsuitable for effective articulation and animation. Recent deep-learning approaches have also been developed to learn skeletons directly from input shapes without relying on predefined templates. These methods are generally trained on datasets containing thousands of rigged characters, allowing them to generate skeletons that align with articulated components. For instance, Xu et al. [34] intro-

duced a volumetric network designed to generate skeletons for input 3D models. RigNet [35] leverages graph convolutions to learn mesh representations, thereby enhancing the accuracy of skeleton extraction. However, it relies on the strong assumption that the input training and test shapes maintain a consistent upright and front-facing orientation.

In this work, we formulate skeleton generation as an auto-regressive problem to accommodate the varying number of bones in different 3D models. By generating bones auto-regressively, our method dynamically adapts to each model's specific requirements, ensuring flexibility and accuracy in skeleton creation.

2.2. Skinning weight prediction

To make 3D models ready for articulation, we also predict skinning weights conditioned on the 3D shape and corresponding skeleton, which define the influence of each joint on each vertex of the mesh.

Several geometric-based techniques have been introduced for skinning [3, 11, 12, 17]. These methods assign skinning weights based on the distance between joints and vertices. However, this distance-based assumption often fails when the 3D shape has a complex topology. Deep learning-based methods [20, 22, 23, 35], such as NeuroSkinning [22], take a skeleton template as input and predict skinning weights using a learned graph neural network. RigNet [35] utilizes intrinsic shape representations that capture geodesic distances between vertices and bones, often struggles with highly intricate mesh topologies and may require extensive feature engineering to maintain performance across varied object categories. SkinningNet [23] employs a two-stream graph neural network to compute skinning weights directly from input meshes and the corresponding skeletons. However, the performance of these GNN-based methods can degrade when applied to datasets with highly varying orientations, leading to reduced accuracy and robustness in complex and varied scenarios.

In this work, we predict skinning weights in a functional diffusion process by incorporating volumetric geodesic distance priors between vertices and joints. This approach effectively handles complex mesh topologies and diverse skeletal structures without constraints of shape orientations.

2.3. Auto-regressive 3D generation

Recently, auto-regressive models have been widely used in 3D mesh generation [5–7, 24, 26, 30, 33]. MeshGPT [26] models meshes as sequences of triangles and tokenizes them using a VQ-VAE [32]. It then employs an auto-regressive transformer to generate the token sequences. This approach enables the creation of meshes with varying face counts. However, most subsequent methods [5, 6, 33] are limited to generating meshes up to 800 faces, due to the computational cost of mesh tokenization. MeshAnythingV2 [7] introduces Adjacent Mesh Tokenization (AMT), doubling the maximum face count to 1,600. EdgeRunner [30] further increases this limit to 4,000 faces by enhancing mesh tokenization techniques. In this work, we explore the potential of auto-regressive models for shape-conditioned skeleton generation. To achieve this, we formulate skeletons as sequences of bones. Unlike mesh generation, which focuses on creating detailed and realistic shapes by utilizing a high number of faces, skeleton generation prioritizes accuracy over complexity. Accurate skeletons are crucial for realistic articulation and animation, and typically consist of fewer than 100 bones, as indicated by the statistics in Articulation-XL.

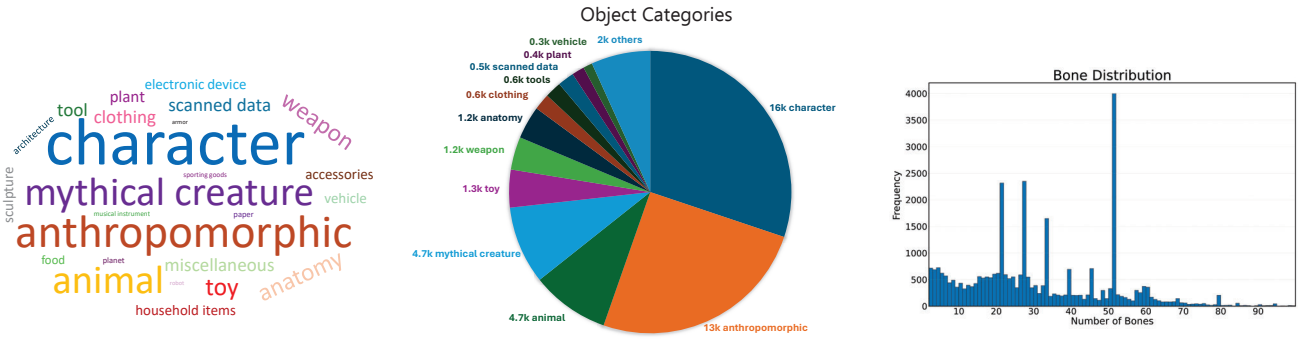
3. Articulation-XL

To facilitate large-scale learning of 3D model articulation, we present Articulation-XL, a comprehensive dataset curated from Objaverse-XL [9, 10]. Our dataset construction pipeline consists of three main stages: initial filtering, VLM-based filtering, and category annotation.

Initial data collection. We begin by identifying 3D models from Objaverse-XL that contain both skeleton and skinning weight annotations. To ensure data quality and practical utility, we apply the following filtering criteria: 1) we remove duplicate data based on both skeleton and mesh similarity; 2) we exclude models with only a single joint/bone structure; 3) we filter out data with more than 100 bones, which constitute a negligible portion of the dataset. This initial filtering yields 38.8k candidate models with articulation annotations.

VLM-based filtering. However, we observe that many initial candidates contain poorly defined skeletons that may impair learning (see Figure 3). To ensure dataset quality, we further implement a Vision-Language Model (VLM)-based filtering pipeline: 1) we render each object with its skeleton from four viewpoints; 2) and then utilize GPT-4o [25] to assess skeleton quality based on specific criteria (detailed in supplementary). This process results in a final collection of over 33k 3D models with high-quality articulation annotations, forming the curated dataset Articulation-XL. The dataset exhibits diverse structural complexity: the number of bones per model ranges from 2 to 100, and the number of joints ranges from 3 to 101. The distribution of bone numbers is illustrated in Figure 2c.

Category label annotation. Additionally, we also leverage a Vision-Language Model (VLM) to automatically assign category labels to each model using specific instructions. The distribution of these categories is illustrated via a word cloud and a pie chart, as shown in Figure 2a and Figure 2b, respectively. We observe a rich diversity of object categories, with human-related models forming the largest subset. Detailed statistics and distribution analyses are provided in the supplementary material.



(a) **Word cloud of Articulation-XL categories.**

(b) **Breakdown of Articulation-XL categories.**

(c) **Bone number distributions of Articulation-XL.**

Figure 2. **Articulation-XL statistics.**

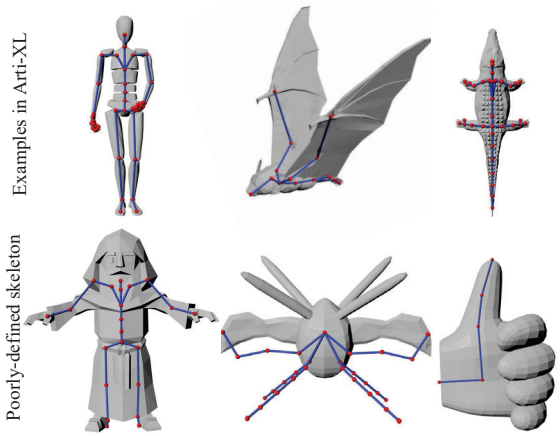


Figure 3. **Some examples from Articulation-XL alongside examples of poorly defined skeletons that were curated out.**

4. Methods

We propose a two-stage pipeline to make 3D models articulation-ready. Given an input 3D mesh, our method first employs an auto-regressive transformer to generate a structurally coherent skeleton (Section 4.1). Subsequently, we predict skinning weights in a functional diffusion process, conditioning on both the input shape and its corresponding skeleton (Section 4.2).

4.1. Auto-regressive skeleton generation

In the initial stage of MagicArticulate, we generate skeletons for 3D models. Unlike previous approaches that rely on fixed templates, our method can handle the inherent structural diversity of 3D objects through an auto-regressive generation framework, as presented in Figure 4.

4.1.1. Problem formulation

Given an input 3D mesh $\mathcal{M}$, our goal is to generate a structurally valid skeleton $\mathcal{S}$ that captures the articulation structure of the object. A skeleton consists of two key components: a set of joints $\mathbf{J} \in \mathbb{R}^{j\times 3}$ defining spatial locations, and bone connections $\mathbf{B} \in \mathbb{N}^{b\times 2}$ specifying the topological structure through joint indices. Formally, we aim to learn the conditional distribution:

$$p(\mathcal{S}|\mathcal{M}) = p(\mathbf{J}, \mathbf{B}|\mathcal{M}), \tag{1}$$

where $\mathcal{M}$ can be sourced from various inputs, including direct 3D models, text-to-3D generation, or image-based reconstruction.

A key challenge in skeleton generation lies in the variable complexity of articulation structures across different objects. Traditional approaches [3, 19] often adopt predefined skeleton templates, which work well for specific categories like human bodies but fail to generalize to objects with diverse structural patterns. This limitation becomes particularly apparent when dealing with our large-scale dataset that contains a wide range of object categories.

To address this challenge, we draw inspiration from recent advances in auto-regressive mesh generation [7, 26] and reformulate skeleton generation as a sequence modeling task. This formulation allows us to: 1) handle varying numbers of bones or joints within skeletons across different 3D models; 2) capture the inherent dependencies between bones; 3) scale effectively to diverse object categories.

4.1.2. Sequence-based generation framework

Our framework transforms the skeleton generation task into a sequence modeling problem through three key components: skeleton tokenization, shape conditioning, and auto-regressive generation. Following the sequence ordering strategies from recent mesh generation methods [24, 26], we order bones by their lower joint index, followed by the higher one. Joints are sorted in ascending z-y-x order (with z representing the vertical axis). For each bone, joint indices are cyclically permuted so that the lower index appears first.

Skeleton tokenization. We represent each skeleton $\mathcal{S}$ as a sequence of bones, where each bone is defined by its two connecting joints (6 coordinates in total). To ensure consistent and discrete representation, we employ a carefully designed tokenization process. We first scale and translate the input mesh and corresponding skeleton to a unit

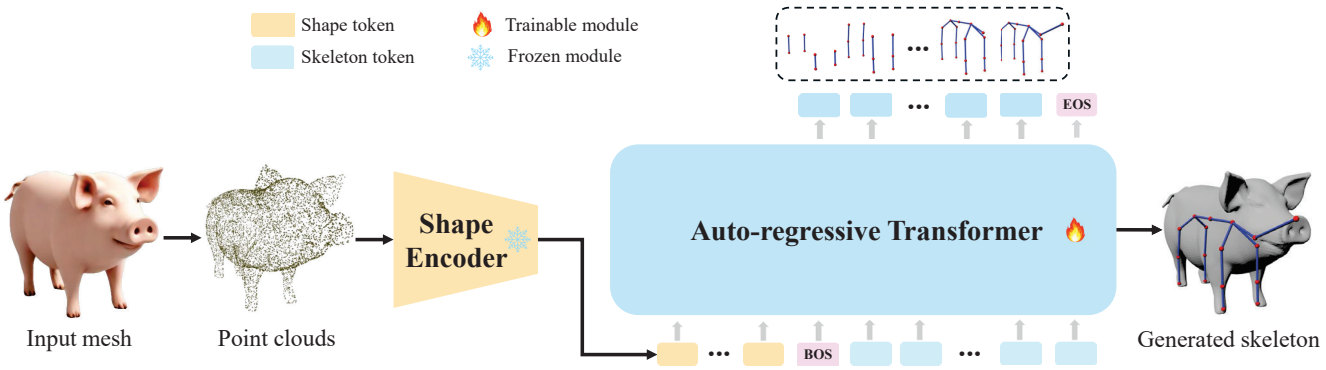


Figure 4. **Overview of our method for auto-regressive skeleton generation.** Given an input mesh, we begin by sampling point clouds from its surface. These sampled points are then encoded into fixed-length shape tokens, which are appended to the start of skeleton tokens to achieve auto-regressive skeleton generation conditioned on input shapes. The input mesh is generated by Rodin Gen-1 [42].

cube $[-0.5, 0.5]^3$, ensuring their spatial alignment. Subsequently, we map the normalized joint coordinates to a discrete 128^3 space, leading to a sequence length of $6b$ for b bones. As such, the discretized coordinates are converted into tokens, which serve as input to the auto-regressive transformer. Unlike MeshGPT [26], we omit the VQ-VAE compression step based on our dataset analysis. Specifically, in Articulation-XL, most of the models have fewer than 100 bones (i.e., 600 tokens). Given these relatively short sequence lengths, using VQ-VAE compression would potentially introduce artifacts without significant benefits in computational efficiency.

Shape-conditioned generation. Following the conventions in [6, 7], we utilize point clouds as the shape condition by sampling 8,192 points from the input mesh $\mathcal{M}$. We then process this point cloud through a pre-trained shape encoder [44], which transforms the raw 3D geometry into a fixed-length feature sequence suitable for transformer processing. This encoded sequence is then appended to the start of the transformer's input skeleton sequence for auto-regressive generation. Additionally, for each sequence, we insert a <bos> token after the shape latent tokens to signify the beginning of the skeleton tokens. Similarly, a <eos> token is added following the skeleton tokens to denote the end of the skeleton sequence.

Auto-regressive learning. For skeleton generation, we employ a decoder-only transformer architecture, specifically the OPT-350M model [43], which has demonstrated strong capabilities in sequence modeling tasks. During training, we provide the ground truth sequences and utilize cross-entropy loss for next-token prediction to supervise the model:

$$\mathcal{L}_{pred} = \mathrm{CE}(\mathbf{T}, \hat{\mathbf{T}}), \tag{2}$$

where $\mathbf{T}$ represents the one-hot encoded ground truth token sequence, and $\hat{\mathbf{T}}$ denotes the predicted sequence.

At inference time, the generation process begins with only the shape tokens as input, and the model sequentially generates each skeleton token, ending when the <eos> token is produced. The resulting token sequence is then detokenized to recover the final skeleton coordinates and connectivity structure.

4.2. Skinning weight prediction

The second stage focuses on predicting skinning weights, which controls how the mesh deforms with skeleton movements. In this work, we represent skinning weights as an n-dimensional function defined on mesh surfaces, which are continuous, high-dimensional, and exhibit significant variation across different skeletal structures. To address these complexities, we employ a functional diffusion framework for accurate skinning weight prediction.

4.2.1. Preliminary: Functional diffusion

Functional diffusion [38] extends classical diffusion models to operate directly on functions, making it particularly suitable for our task. Consider a function f_0 mapping from domain $\mathcal{X}$ to range $\mathcal{Y}$:

$$f_0 : \mathcal{X} \to \mathcal{Y}. \tag{3}$$

The diffusion process gradually adds functional noise g (mapping the same domain to range) to the original function:

$$f_t(x) = \alpha_t \cdot f_0(x) + \sigma_t \cdot g(x), \quad t \in [0, 1] \tag{4}$$

where α_t and σ_t control the noise schedule. The goal is to train a denoiser D that recovers the original function:

$$D_\theta[f_t, t](x) \approx f_0(x). \tag{5}$$

This formulation naturally aligns with our task requirements. By treating skinning weights as continuous functions over the mesh surface, we can capture smoothly transitioning weights between vertices. Additionally, the framework's flexibility allows it to adapt to diverse mesh topologies and skeletal structures.

4.2.2. Skinning weight prediction

Building upon the functional diffusion framework, we formulate skinning weight prediction as learning a mapping $f : \mathbb{R}^3 \rightarrow \mathbb{R}^n$ from 3D points to their corresponding weights. Specifically, the input to our model consists of 3D points $\mathcal{P} \in \mathbb{R}^{v\times 3}$ sampled from the surface of the mesh. The output is an n-dimensional skinning weight matrix $\mathcal{W} \in \mathbb{R}^{v\times n}$. Here, the ground truth skinning weights of sampled points for training are copied from their nearest vertices and will also be copied back when inference. n denotes the maximum number of joints in the dataset.

To enhance prediction accuracy, we introduce two key components. First, we condition the generation on both joint coordinates and global shape features extracted by a pre-trained encoder [44]. Second, we leverage volumetric geodesic priors calculated from [11]. Specifically, we compute the volumetric geodesic priors from each mesh vertex to each joint. We then assign these priors to sampled points based on their nearest vertices and normalize them to match the range of skinning weights, forming a volumetric geodesic matrix $\mathcal{G} \in \mathbb{R}^{v\times n}$. Our model learns to predict the residual between the actual skinning weights and this geometric prior, i.e., $f : \mathcal{P} \rightarrow (\mathcal{W} - \mathcal{G})$, enabling more stable predictions.

Following [38], we optimize our model using x_0-prediction with the objective:

$$\mathcal{L}_{denoise} = \left\| D_\theta \left(\{x, f_t(x)\}, t \right) - f_0(x) \right\|_2^2, \quad x \in \mathcal{P}. \tag{6}$$

We employ the Denoising Diffusion Probabilistic Model (DDPM) [13] as our scheduler. In practice, we normalize the skinning weights and volumetric geodesic priors to the range $[-1, 1]$ before adding noise. We will conduct ablation studies on this design in Section 5.4.2.

5. Experiments

5.1. Implementation details

Datasets. We evaluate our method on two datasets: our proposed Articulation-XL and ModelsResource [31, 35]. Articulation-XL contains 33k samples, with 31.4k for training and 1.6k for testing. ModelsResource is a smaller dataset, containing 2,163 training and 270 testing samples. The number of joints for each object varies from 3 to 48, with an average of 25.0 joints. While the data in ModelsResource maintains a consistent upright and front-facing orientation, the 3D models in Articulation-XL exhibit varying orientations. We have verified that there are no duplications between Articulation-XL and ModelsResource.

Training details. Our training process consists of two stages. For skeleton generation, we train the auto-regressive transformer on 8 NVIDIA A100 GPUs for approximately two days. For skinning weight prediction, models are trained on the same hardware configuration for about one day. To enhance model robustness, we apply data augmentation including scaling, shifting, and rotation transformations. For more details, please refer to the appendix.

5.2. Skeleton generation results

Metrics. We adopt three standard metrics following [35] to evaluate skeleton quality: CD-J2J, CD-J2B, and CD-B2B. These Chamfer Distance-based metrics measure the spatial alignment between generated and ground truth skeletons by computing distances between joints-to-joints, joints-to-bones, and bones-to-bones respectively. Lower values indicate better skeleton quality.

Baselines. We compare our method against two representative approaches: Pinocchio [3], a traditional template-fitting method, and RigNet [35], a learning-based method using graph convolutions. All methods are evaluated on the Articulation-XL and ModelsResource datasets.

Comparison results. Qualitative comparisons are presented in Figure 5, where we compare different methods across various object categories. Pinocchio struggles with objects that differ from its predefined templates, especially obvious in non-humanoid objects (as shown in the 2nd row and the 3rd row on the right). RigNet demonstrates improved performance when tested on ModelsResource, where the data maintains a consistent upright and front-facing orientation. However, it still struggles with complex topologies (as illustrated in the 1st and 2nd rows on the left). Furthermore, RigNet performs worse on Articulation-XL, where the data exhibit varying orientations. In contrast, our method generates high-quality skeletons that closely match artist-created references across diverse object categories.

The quantitative results are shown in Table 1. Our method consistently outperforms baselines across all metrics on both datasets.

Generalization analysis. To evaluate the generalization capability, we perform cross-dataset evaluation by training RigNet and our MagicArticulate on Articulation-XL and testing on ModelsResource. As shown in Table 1 (marked with *), our method maintains competitive performance compared to RigNet trained directly on ModelsResource, while RigNet's performance degrades significantly when tested on unseen data distributions, performing even worse than the template-based method Pinocchio.

To further assess real-world applicability, we evaluate all methods on AI-generated 3D meshes from Tripo 2.0 [1] (Figure 6). Our method successfully generates plausible skeletons for diverse object categories, while RigNet fails to produce valid results despite being trained on our large-scale dataset. Notably, even Pinocchio's template-based approach struggles to generate accurate skeletons for basic

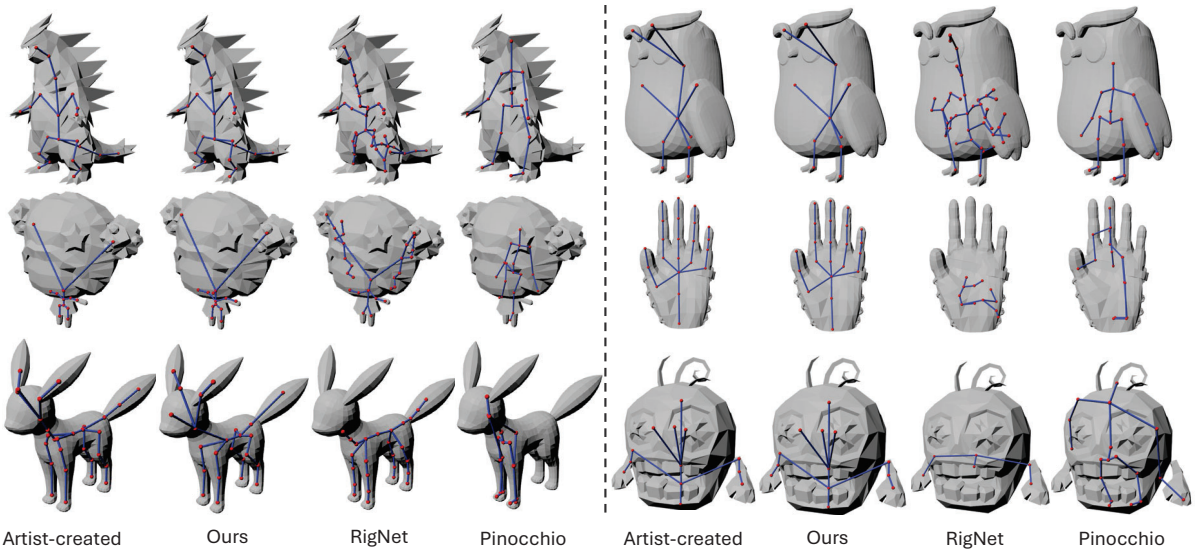


Figure 5. **Comparison of skeleton creation results on ModelsResource (left) and Articulation-XL (right).** Our generated skeletons more closely resemble the artist-created references, while RigNet and Pinocchio struggle to handle various object categories.

Table 1. **Quantitative comparison on skeleton generation.** We compare different methods using CD-J2J, CD-J2B, and CD-B2B as evaluation metrics on both Articulation-XL (Arti-XL) and ModelsResource (Modelres.). Lower values indicate better performance. The metrics are in units of 10^{-2}. Here, * denotes models trained on Articulation-XL and tested on ModelsResource.

	Dataset	CD-J2J	CD-J2B	CD-B2B
RigNet*		7.132	5.486	4.640
Pinocchio		6.852	4.824	4.089
RigNet	*ModelsRes.*	4.143	2.961	2.675
Ours*		4.103	3.101	2.672
Ours		**3.343**	**2.455**	**2.140**
Pinocchio		8.360	6.677	5.689
RigNet	*Arti-XL*	7.478	5.892	4.932
Ours		**2.586**	**1.959**	**1.661**

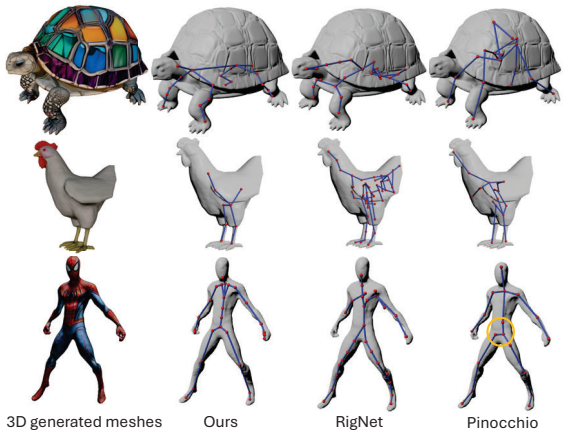


Figure 6. **Skeleton creation results on 3D generated meshes.** Our method has a better generalization performance than both RigNet [35] and Pinocchio [3] across difference object categories. The 3D models are generated by Tripo 2.0 [1].

categories like humans and quadrupeds, highlighting the advantage of our method in handling novel object structures.

5.3. Skinning weight prediction results

Metrics. We evaluate skinning weight quality using three metrics: precision, recall, and L1-norm error. Precision and recall measure the accuracy of identifying significant joint influences (defined as weights larger than $1e-4$ following [35], while the L1-norm error computes the average difference between predicted and ground truth skinning weights across all vertices. We will also report the deformation error in appendix.

Baselines. We compare our method against Geodesic Voxel Binding (GVB) [11], a geometric-based method available in Autodesk Maya [16] and RigNet [35]. When trained on Articulation-XL, we filter out a subset containing 28k training and 1.2k testing samples, excluding data with more than 55 joints (which constitute a small fraction of both real-world cases and Articulation-XL).

Comparison results. Qualitative comparisons in Figure 7 visualize the predicted skinning weights and their L1 error maps against artist-created references. Our method predicts more accurate skinning weights with significantly lower errors across diverse object categories. In contrast, both GVB and RigNet show larger deviations, particularly in regions around joint boundaries.

The quantitative results are shown in Table 2, which support qualitative observations, demonstrating that our method consistently outperforms baselines across most metrics on both datasets.

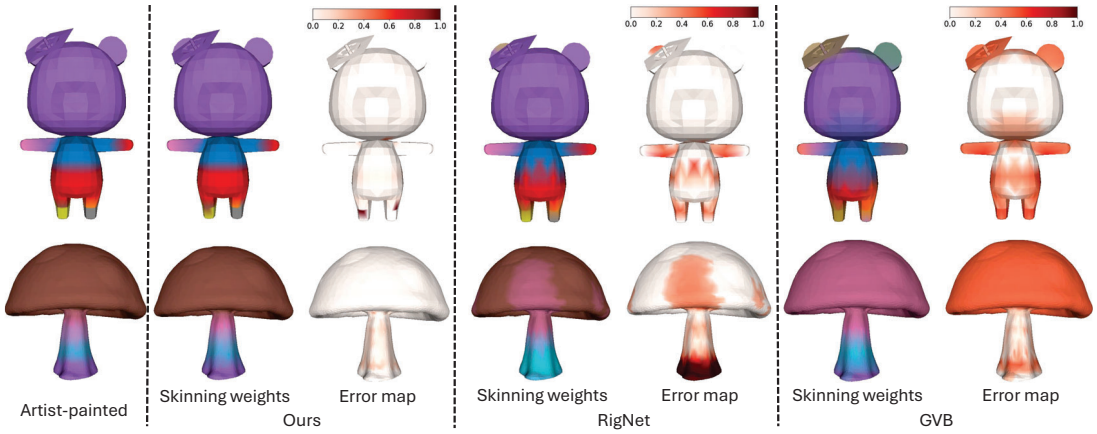


Figure 7. **Comparisons with previous methods for skinning weight prediction on ModelsResource (top) and Articulation-XL (bottom).** We visualize skinning weights and L1 error maps. For more results, please refer to the supplementary materials.

Table 2. **Quantitative comparison on skinning weight prediction.** We compare our method with GVB and RigNet. For Precision and Recall, larger values indicate better performance. For average L1-norm error, smaller values are preferred.

	Dataset	Precision	Recall	avg L1
GVB		69.3%	79.2%	0.687
RigNet	*ModelsResource*	77.1%	**83.5%**	0.464
Ours		**82.1%**	81.6%	**0.398**
GVB		75.7%	68.3%	0.724
RigNet	*Articulation-XL*	72.4%	71.1%	0.698
Ours		**80.7%**	**77.2%**	**0.337**

Table 3. **Ablation studies for skeleton generation.**

	CD-J2J	CD-J2B	CD-B2B
w/o data filtering	2.982	2.327	2.015
w/ data balance	2.691	2.033	1.731
Ours	**2.586**	**1.959**	**1.661**

5.4. Ablation studies

5.4.1. Ablation studies on skeleton generation

We conduct ablation studies to evaluate the effects of VLM-based data filtering and category balance strategies on skeleton generation. All experiments are performed on Articulation-XL with the same number of iterations to ensure a fair comparison. The results, presented in Table 3, show notable performance degradation without data filtering, highlighting the importance of high-quality training data. We investigate the impact of category imbalance (see Figure 2b) by replicating data from non-human-like categories and applying augmentations such as scaling, shifting, and rotation. This balanced training strategy shows no improvement over the original results, which could be attributed to the dominance of humanoid data in the test set.

Table 4. **Ablation studies on skinning weight prediction.**

	Precision	Recall	avg L1
w/o geodesic dist.	81.5%	77.7%	0.444
w/o weights norm	82.0%	77.9%	0.436
w/o shape features	81.4%	81.3%	0.412
Ours	**82.1%**	**81.6%**	**0.398**

5.4.2. Ablation studies on skinning weight prediction

We conduct ablation studies on three critical components of our skinning weight prediction framework. The quantitative results on ModelsResource are shown in Table 4. First, removing the volumetric geodesic distance initialization reduces precision by 0.6% and recall by 3.9%, demonstrating its crucial role in guiding accurate weight distribution. Second, eliminating our normalization strategy, which scales both skinning weights and geodesic distances to $[-1, 1]$ before noise addition, leads to an 8.7% increase in L1 error. Finally, excluding global shape features from the pretrained encoder [44] results in less accurate predictions. All these results validate our design choices and show that each component contributes notably to the final performance.

6. Conclusion

In this work, we present MagicArticulate to convert static 3D models into articulation-ready assets that support realistic animation. We first introduce a large-scale dataset Articulation-XL with high-quality articulation annotations, which is carefully curated from Objaverse-XL. Built upon this dataset, we develop a novel two-stage pipeline that first generates skeletons through auto-regressive sequence modeling, naturally handling varying numbers of bones or joints within skeletons across different 3D models. Then we predict skinning weights in a functional diffusion process that incorporates volumetric geodesic distance priors between vertices and joints. Extensive experiments demonstrate our method's superior performance and generalization ability across diverse object categories.

Acknowledgements

This research is supported by the MoE AcRF Tier 2 grant (MOE-T2EP20223-0001).

References

[1] TriPo AI. Tripo 3d, 2023. 2, 6, 7

[2] Oscar Kin-Chung Au, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, and Tong-Yee Lee. Skeleton extraction by mesh contraction. *ACM transactions on graphics (TOG)*, 27(3):1–10, 2008. 1, 2

[3] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Transactions on graphics (TOG)*, 26(3):72–es, 2007. 1, 2, 3, 4, 6, 7

[4] Junjie Cao, Andrea Tagliasacchi, Matt Olson, Hao Zhang, and Zhinxun Su. Point cloud skeletons via laplacian based contraction. In *2010 Shape Modeling International Conference*, pages 187–197. IEEE, 2010. 1, 2

[5] Sijin Chen, Xin Chen, Anqi Pang, Xianfang Zeng, Wei Cheng, Yijun Fu, Fukun Yin, Yanru Wang, Zhibin Wang, Chi Zhang, et al. Meshxl: Neural coordinate field for generative 3d foundation models. *arXiv preprint arXiv:2405.20853*, 2024. 3

[6] Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024. 3, 5

[7] Yiwen Chen, Yikai Wang, Yihao Luo, Zhengyi Wang, Zilong Chen, Jun Zhu, Chi Zhang, and Guosheng Lin. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization. *arXiv preprint arXiv:2408.02555*, 2024. 3, 4, 5

[8] Edilson De Aguiar, Christian Theobalt, Sebastian Thrun, and Hans-Peter Seidel. Automatic conversion of mesh animations into skeleton-based animations. In *Computer Graphics Forum*, pages 389–397. Wiley Online Library, 2008. 2

[9] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 1, 3

[10] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3

[11] Olivier Dionne and Martin de Lasa. Geodesic voxel binding for production character meshes. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 173–180, 2013. 3, 6, 7

[12] Ana Dodik, Vincent Sitzmann, Justin Solomon, and Oded Stein. Robust biharmonic skinning using geometric fields. *arXiv preprint arXiv:2406.00238*, 2024. 3

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 6

[14] Hui Huang, Shihao Wu, Daniel Cohen-Or, Minglun Gong, Hao Zhang, Guiqing Li, and Baoquan Chen. L1-medial skeleton of point cloud. *ACM Trans. Graph.*, 32(4):65–1, 2013. 1, 2

[15] Adobe Inc. Mixamo. 2

[16] Autodesk Inc. Autodesk maya, 2024. Version 2024. 7

[17] Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 30(4):78, 2011. 3

[18] Doug L James and Christopher D Twigg. Skinning mesh animations. *ACM Transactions on Graphics (TOG)*, 24(3): 399–407, 2005. 2

[19] Peizhuo Li, Kfir Aberman, Rana Hanocka, Libin Liu, Olga Sorkine-Hornung, and Baoquan Chen. Learning skeletal articulations with neural blend shapes. *ACM Transactions on Graphics (TOG)*, 40(4):1–15, 2021. 1, 2, 4

[20] Zhouyingcheng Liao, Jimei Yang, Jun Saito, Gerard Pons-Moll, and Yang Zhou. Skeleton-free pose transfer for stylized 3d characters. In *European Conference on Computer Vision*, pages 640–656. Springer, 2022. 3

[21] Cheng Lin, Changjian Li, Yuan Liu, Nenglun Chen, Yi-King Choi, and Wenping Wang. Point2skeleton: Learning skeletal representations from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4277–4286, 2021. 1, 2

[22] Lijuan Liu, Youyi Zheng, Di Tang, Yi Yuan, Changjie Fan, and Kun Zhou. Neuroskinning: Automatic skin binding for production characters with deep graph networks. *ACM Transactions on Graphics (ToG)*, 38(4):1–12, 2019. 3

[23] Albert Mosella-Montoro and Javier Ruiz-Hidalgo. Skinningnet: Two-stream graph convolutional neural network for skinning prediction of synthetic characters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18593–18602, 2022. 3

[24] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pages 7220–7229. PMLR, 2020. 3, 4

[25] OpenAI. Gpt-4o, 2023. 3

[26] 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. 3, 4, 5

[27] Chaoyue Song, Jiacheng Wei, Tianyi Chen, Yiwen Chen, Chuan-Sheng Foo, Fayao Liu, and Guosheng Lin. Moda: Modeling deformable 3d objects from casual videos. *International Journal of Computer Vision*, pages 1–20, 2024. 2

[28] Chaoyue Song, Jiacheng Wei, Chuan Sheng Foo, Guosheng Lin, and Fayao Liu. Reacto: Reconstructing articulated objects from a single video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5384–5395, 2024. 2

[29] Andrea Tagliasacchi, Ibraheem Alhashim, Matt Olson, and Hao Zhang. Mean curvature skeletons. In *Computer Graphics Forum*, pages 1735–1744. Wiley Online Library, 2012. 1, 2

[30] Jiaxiang Tang, Zhaoshuo Li, Zekun Hao, Xian Liu, Gang Zeng, Ming-Yu Liu, and Qinsheng Zhang. Edgerunner: Auto-regressive auto-encoder for artistic mesh generation. *arXiv preprint arXiv:2409.18114*, 2024. 3

[31] The Models-Resource. The models-resource, 2019. 2, 6

[32] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 3

[33] Haohan Weng, Yikai Wang, Tong Zhang, CL Chen, and Jun Zhu. Pivotmesh: Generic 3d mesh generation via pivot vertices guidance. *arXiv preprint arXiv:2405.16890*, 2024. 3

[34] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, and Karan Singh. Predicting animation skeletons for 3d articulated models via volumetric nets. In *2019 international conference on 3D vision (3DV)*, pages 298–307. IEEE, 2019. 1, 2

[35] Zhan Xu, Yang Zhou, Evangelos Kalogerakis, Chris Landreth, and Karan Singh. Rignet: Neural rigging for articulated characters. *arXiv preprint arXiv:2005.00559*, 2020. 1, 2, 3, 6, 7

[36] Zhan Xu, Yang Zhou, Li Yi, and Evangelos Kalogerakis. Morig: Motion-aware rigging of character meshes from point clouds. In *SIGGRAPH Asia 2022 conference papers*, pages 1–9, 2022. 2

[37] Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2863–2873, 2022. 2

[38] Biao Zhang and Peter Wonka. Functional diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4723–4732, 2024. 5, 6

[39] Hao Zhang, Di Chang, Fang Li, Mohammad Soleymani, and Narendra Ahuja. Magicpose4d: Crafting articulated models with appearance and motion control. *arXiv preprint arXiv:2405.14017*, 2024. 2

[40] Hao Zhang, Fang Li, Samyak Rawlekar, and Narendra Ahuja. Learning implicit representation for reconstructing articulated objects. *arXiv preprint arXiv:2401.08809*, 2024.

[41] Hao Zhang, Fang Li, Samyak Rawlekar, and Narendra Ahuja. S3o: A dual-phase approach for reconstructing dynamic shape and skeleton of articulated objects from single monocular video. *arXiv preprint arXiv:2405.12607*, 2024. 2

[42] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 43(4):1–20, 2024. 2, 5

[43] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 5

[44] 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. 5, 6, 8