Supplementary Material for 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

Overview

In this supplementary material, we provide additional details and experimental results for the main paper, including:

- Further details of MagicArticulate (Section 1) and Articulation-XL (Section 3);
- Additional experimental results on skeleton generation and skinning weight prediction (Section 2);
- A discussion of the limitations of our work and future works (Section 4).

1. More details of MagicArticulate

1.1. Implementation details

Skeleton generation. Our skeleton generation pipeline utilizes a pre-trained shape encoder [21] to process input meshes. For each mesh, we sample 8,192 points which are encoded into 257 shape tokens following MeshAnything [5]. To ensure consistent point cloud sampling across different data sources, we first extract the signed distance function from input mesh using [17], followed by generating a coarse mesh via Marching Cubes [10]. We then sample point clouds and their corresponding normals from this coarse mesh.

For training on Articulation-XL, we use 8 NVIDIA A100 GPUs for approximately two days with a batch size of 64 per GPU, resulting in an effective batch size of 512. When training on ModelsResource, we utilize 4 NVIDIA A100 GPUs for about 9 hours with a batch size of 32 per GPU, which yields an effective batch size of 128. During inference, the model generates skeleton tokens autoregressively from shape tokens until reaching the $<\text{eos}>$ token, followed by detokenization to recover the final skeleton coordinates in $[-0.5, 0.5]$ range.

Skinning weight prediction. Our functional diffusion model employs the Denoising Diffusion Probabilistic Model (DDPM) with 1,000 timesteps and a linear beta schedule. During training, we condition the model on ground truth skeletons and supervise it with corresponding ground truth skinning weights. We add noise to the skinning weight function (the process is illustrated in Figure S2) and then feed the noised skinning weights into our denoising network (Figure S1). Following [19], our network architecture processes the noised set $\{(x, f_t(x)) \mid x \in \mathcal{P}\}$ by splitting it into smaller subsets and handling them through multiple cross-attention stages. The time embedding at timestep t is incorporated into each self-attention layer via adaptive layer normalization. For visual clarity, Figure S1 shows only one processing stage.

We train the model on Articulation-XL using 8 NVIDIA A100 GPUs for approximately one day, with a batch size of 16 per GPU (effective batch size 128). Training on ModelsResource uses the same configuration for about 4 hours. During inference, we perform 25 denoising steps to generate predictions $\mathcal{W} \in \mathbb{R}^{v \times n}$ in the range $[-1, 1]$. These results are then normalized to $[0, 1]$, ensuring that each row of the skinning weight matrix sums to 1. To handle varying joint counts across different models, we employ a valid joint mask during both training and testing, with a maximum joint count of 55 as discussed in the main paper (Sections 4.2 and 5.3).

1.2. Experimental details

For baseline comparisons, we use the implementations of RigNet [18] and Pinocchio [2] from the GitHub repositories[1]. The Geodesic Voxel Binding (GVB) [6] comparison is conducted using the implementation in Autodesk Maya [8]. When training RigNet on our Articulation-XL, we strictly follow the authors' data processing pipeline and six-stage training strategy as specified in their official implementation.

† Corresponding authors.

[1] https://github.com/zhan-xu/RigNet, https://github.com/haoz19/Automatic-Rigging

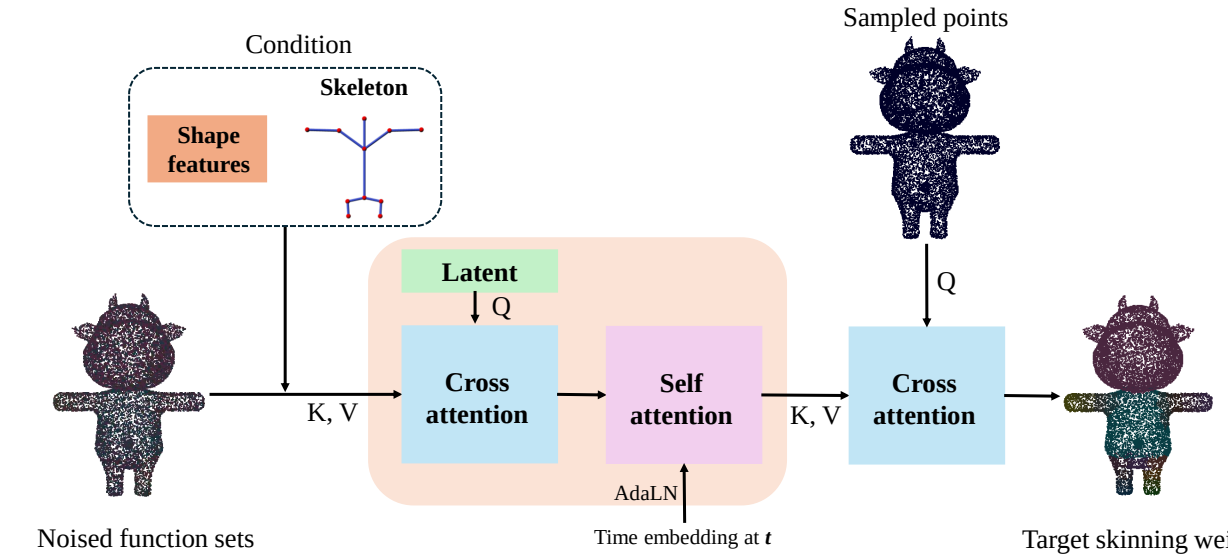

Figure S1. **Overview of the function diffusion architecture for skinning weight prediction.** Given a set of noised skinning weight functions $\{(x, f_t(x)) \mid x \in \mathcal{P}\}$, conditioned on skeleton and shape features from [21], we denoise the skinning weight functions to approximate the target weights.

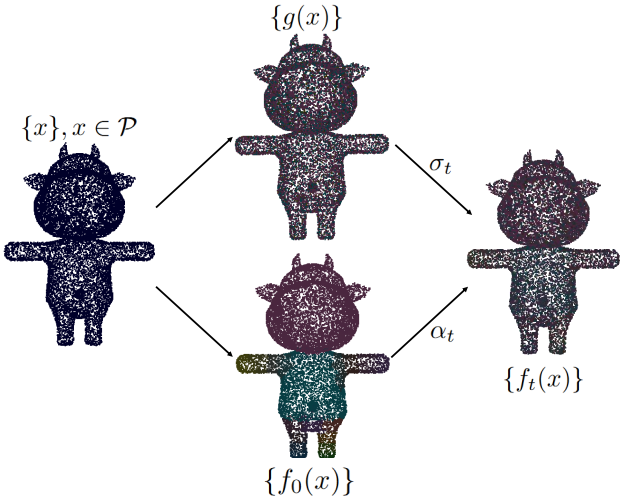

Figure S2. **Process of adding noise to the skinning weight function.** Given $x \in \mathcal{P}$ and the original skinning weight function $f_0(x)$, we add the noise function $g(x)$ to obtain the noised function $f_t(x)$.

1.3. Animation

Many recent works have explored 3D animation, including skeleton-free pose transfer [9, 12–14], skeleton-driven pose transfer [20], and physics-driven animation [7]. In this paper, we propose a method that enables automatic articulation generation for any input 3D model, whether artist-created or AI-generated. The pipeline first generates a skeleton for the input model, then predicts skinning weights conditioned on both the model geometry and the generated skeleton. The resulting articulated model can be exported in standard formats (e.g., FBX, GLB), making it directly compatible with popular animation software such as Blender [3] and Autodesk Maya [8].

2. Additional experimental results

2.1. More results of skeleton generation

We provide additional qualitative comparisons among MagicArticulate, RigNet [18], and Pinocchio [2] for skeleton generation.

More qualitative results on out-of-domain data. We evaluate our method's generalization capability on diverse out-of-domain data sources: AI-generated meshes from Tripo2.0 [1], unregistered 3D scans from FAUST [4], and video-based 3D reconstructions [16]. As shown in Figure S3, while existing methods struggle with generalization (RigNet fails across all cases, and Pinocchio shows misalignments even for human bodies, see skeleton results on the 3D scan), our method maintains robust performance across different data sources and categories. Notably, for human models, our method generates more detailed skeletal structures, including accurate hand skeletons, surpassing Pinocchio's template-based results.

More qualitative results on Articulation-XL and ModelsResource. We provide additional qualitative results on both Articulation-XL and ModelsResource datasets. As illustrated in Figure S4, our method consistently generates

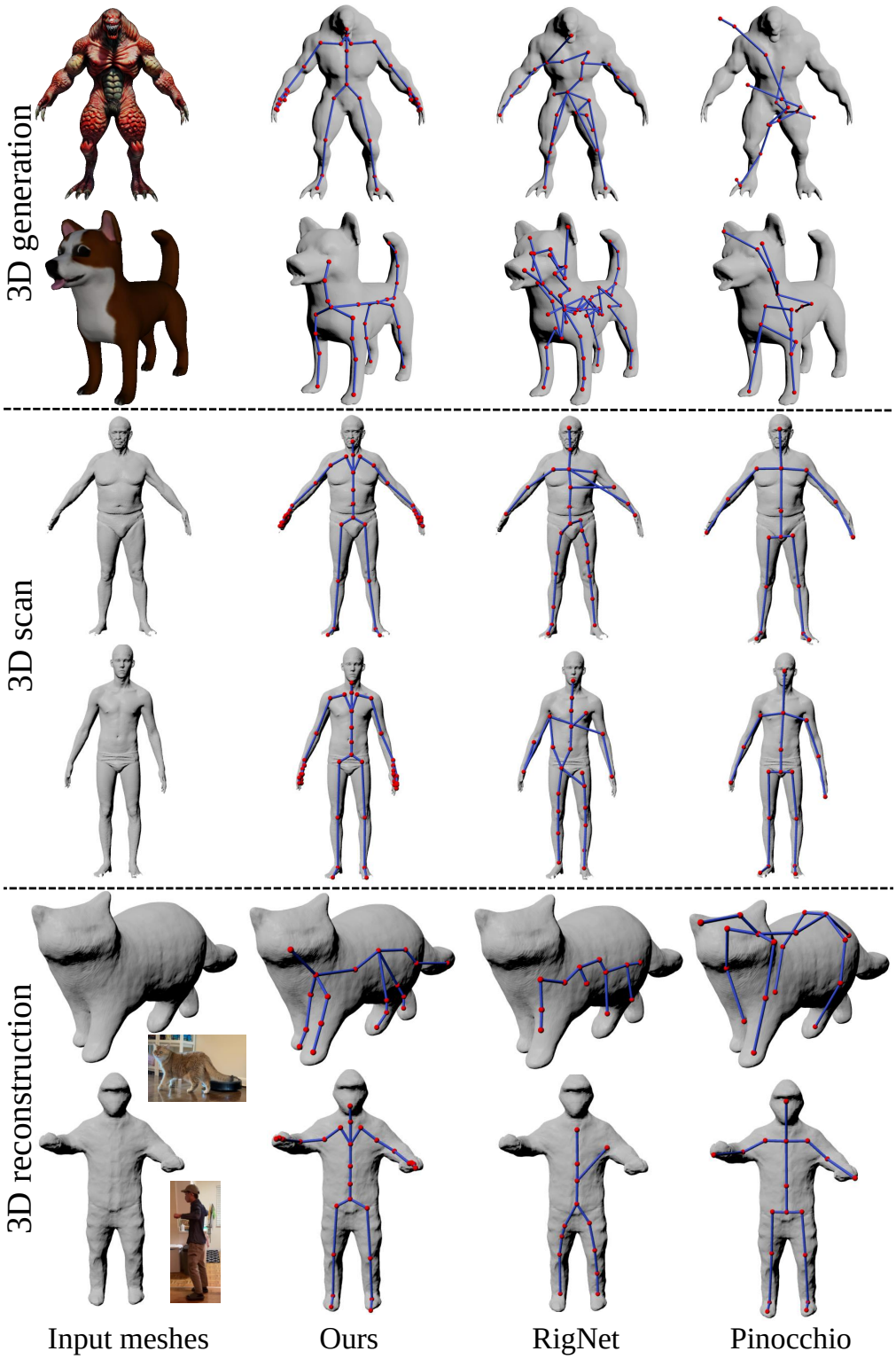

Figure S3. **Comparison of skeleton generation methods on out-of-domain data.** The input meshes are from 3D generation, 3D scan, and 3D reconstruction.

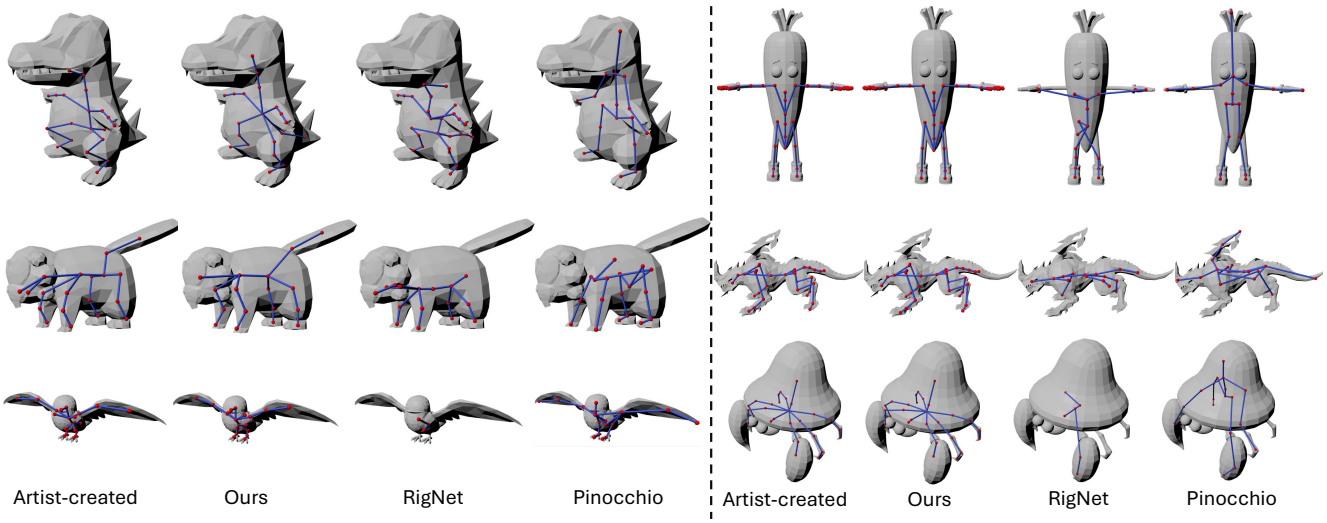

Figure S4. **Comparison of skeleton generation methods on ModelsResource (left) and Articulation-XL (right).** Our results more closely resemble the artist-created references, while RigNet and Pinocchio struggle to handle various object categories.

Table S1. **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 and average distance error, smaller values are preferred.

	Dataset	Precision	Recall	avg L1	avg Dist.
GVB		69.3%	79.2%	0.687	0.0067
RigNet	*ModelsResource*	77.1%	**83.5%**	0.464	0.0054
Ours		**82.1%**	81.6%	**0.398**	**0.0039**
GVB		75.7%	68.3%	0.724	0.0095
RigNet	*Articulation-XL*	72.4%	71.1%	0.698	0.0091
Ours		**80.7%**	**77.2%**	**0.337**	**0.0050**

high-quality skeletons that accurately match artist-created references across diverse object categories.

Robustness to various mesh orientations. To further validate our model's robustness to various orientations, we include mesh rotations at multiple angles in Figure S5. These examples show that our approach remains largely rotation-stable. While minor skeleton variations may occur, all generated results maintain anatomically valid and suitable for rigging purposes.

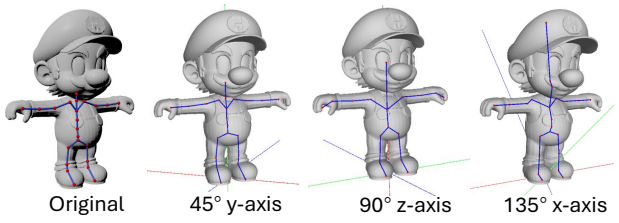

Figure S5. **Skeleton results on 3D models with different orientations.** Although minor differences may appear in the generated skeletons, all results remain anatomically valid and suitable for rigging purposes.

2.2. More results of skinning weight prediction

Quantitative results with deformation error. Beyond the precision, recall, and L1-norm metrics reported in the main paper, we evaluate the practical effectiveness of predicted skinning weights through deformation error analysis. This metric computes the average Euclidean distance between vertices deformed using predicted weights and ground truth weights across 10 random poses. The comprehensive results, shown in Table S1, demonstrate our method's superior performance across most metrics on both datasets. We also include deformation error analysis in our ablation studies (Table S2), further validating the effectiveness of our design choices.

More qualitative results. We present additional qualitative comparisons between MagicArticulate, RigNet [18], and Geodesic Voxel Binding (GVB) [6] for skinning weight prediction. Figure S6 shows both the predicted skinning

Table S2. **Ablation studies on ModelsResource for skinning weight prediction.**

	Precision	Recall	avg L1	avg Dist.
w/o geodesic dist.	81.5%	77.7%	0.444	0.0046
w/o weights norm	82.0%	77.9%	0.436	0.0045
w/o shape features	81.4%	81.3%	0.412	0.0042
Ours	**82.1%**	**81.6%**	**0.398**	**0.0039**

Table S3. **Object counts for each category in the Articulation-XL dataset.**

Category	# Objects	Category	# Objects	Category	# Objects
character	16020	miscellaneous	584	architecture	132
anthropomorphic	13393	scanned data	546	planet	49
animal	4760	plant	382	paper	46
mythical creature	4734	accessories	293	musical instrument	25
toy	1360	vehicle	283	sporting goods	21
weapon	1257	sculpture	276	armor	13
anatomy	1227	household items	274	robot	4
clothing	595	food	206		

weights and their L1 error maps compared to artist-created references, demonstrating our method's superior accuracy across diverse object categories.

3. More details of Articulation-XL

3.1. Data Curation

Our dataset curation process filters out duplicates, objects with extreme joint/bone counts, and multi-component objects. A detailed category-wise object distribution is provided in Table S3.

3.2. Quality assessment

We employ GPT-4o [11] for quality assessment of skeleton annotations. For each model, we generate four-view renders using Pyrender[2] showing both the 3D model and its skeleton (Figure S9). These renders are evaluated using specific quality criteria detailed in Figure S7.

3.3. Category annotation

For the Visual-Language Model (VLM)-based category labeling, we render each 3D model along with its normal maps from four viewpoints using Blender [3] (see example in Figure S10). We then utilize GPT-4o [11] to classify the categories of the 3D models based on specific instructions, as outlined in Figure S8.

4. Limitations and future work

Despite its strong performance, our method has several notable limitations. First, our approach struggles with coarse mesh inputs, often producing inaccurate skeletons as shown in Figure S11. While we employ preprocessing techniques to handle inputs from different sources, the significant domain gap between training data and coarse meshes remains challenging. Potential solutions include incorporating mesh quality augmentation during training to enhance robustness.

A second limitation lies in our dataset composition. Although Articulation-XL is large in scale, it lacks sufficient coverage of common articulated objects like laptops, staplers, and scissors, which affects our model's generalization to these categories.

Future work will address these limitations by: 1) Developing more robust preprocessing and training strategies for handling varying mesh qualities; 2) Expanding dataset coverage to include a broader range of everyday articulated objects; 3) Exploring techniques to better bridge the domain gap between different data sources.

[2] https://github.com/mmatl/pyrender

References

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

[2] 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] Blender Foundation. Blender - a 3d modelling and rendering software, 2024. Version 3.6. 2, 5

[4] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3794–3801, 2014. 2

[5] Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu,

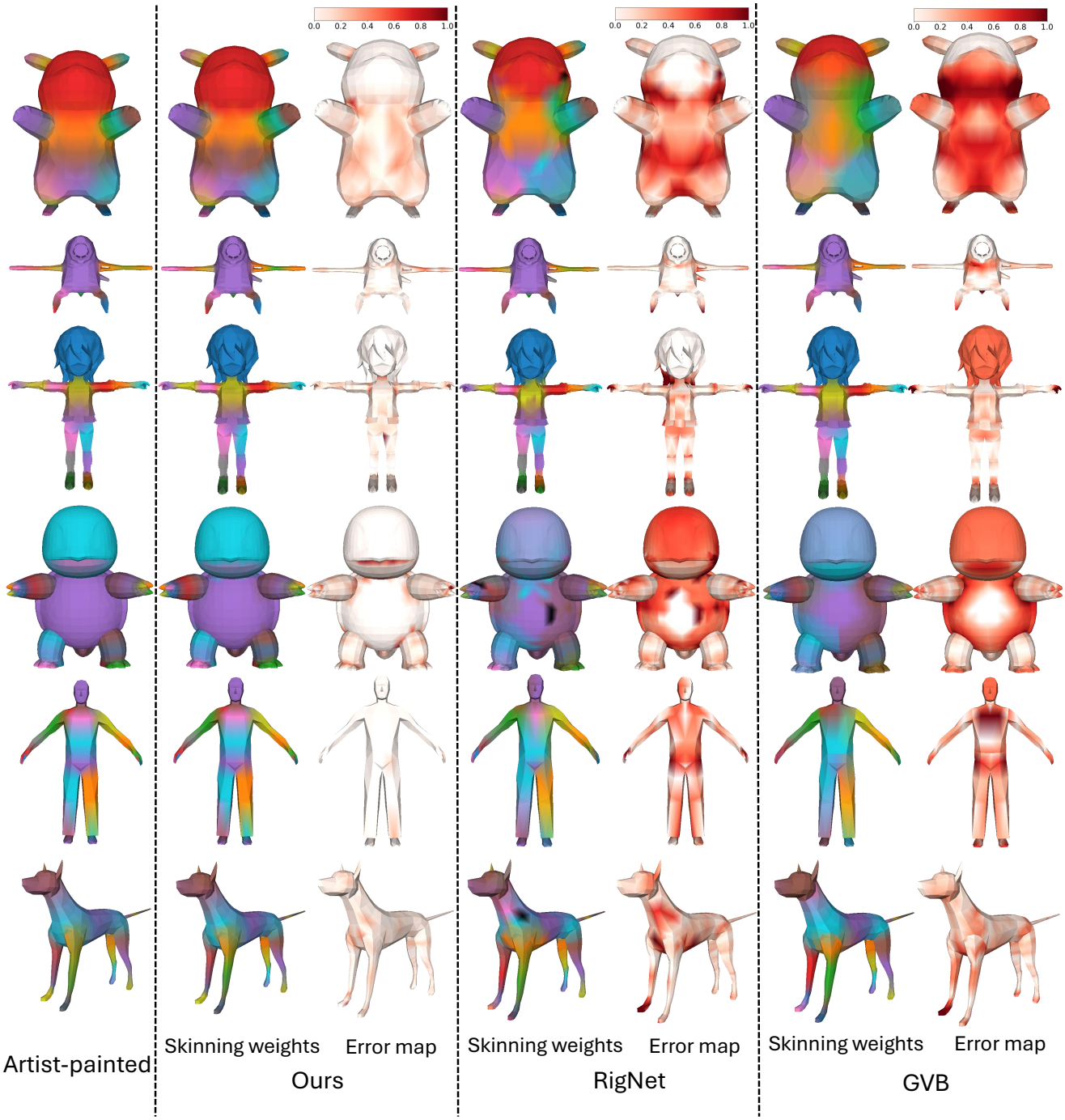

Figure S6. **Comparison of skinning weight prediction methods on ModelsResource (first three rows) and Articulation-XL (last three rows).** We visualize the predicted skinning weights alongside their corresponding L1 error maps.

et al. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024. 1

[6] 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. 1, 4

[7] Zhoujie Fu, Jiacheng Wei, Wenhao Shen, Chaoyue Song, Xiaofeng Yang, Fayao Liu, Xulei Yang, and Guosheng Lin. Sync4d: Video guided controllable dynamics for physics-based 4d generation. *arXiv preprint arXiv:2405.16849*, 2024. 2

Task Description:

Our task is to evaluate the quality of the skeleton within a 3D object using an image that contains four sub-images rendered from both the 3D mesh and its skeleton together. The four sub-images are arranged in a 2x2 grid format. These sub-images are rendered from four distinct views of the 3D object. In the images, bones are represented by blue cylinders, and joints are represented by red spheres.

Instructions:

1. Carefully examine the provided image, which contains four different sub-images of the 3D object and its skeleton.

2. Provide a detailed critique of the skeleton within the 3D object based on the following criteria:

Joint and Bone Position Relative to Mesh:

Identify joints or bones that are excessively protruding out of the mesh. A high-quality skeleton should have both joints and bones confined within the mesh boundary. The more joints or bones that protrude, the lower the quality. Since the joint spheres and bone cylinders have a radius, you should consider their centers without accounting for their shapes.

Anatomical Accuracy:

Correct Joint Placement: Joints should mimic the anatomical placement. For example, shoulder joints should align with the mesh's shoulder region.

Natural Poses: The skeleton should maintain natural and plausible poses. Any significant deviations might indicate errors in skeletal rigging.

3. Based on your analysis, provide a rating for each criterion using the following three options. Ensure that the decision aligns with your detailed critique.

Poor: Poor quality, significant issues or errors that severely impact the skeleton's usability or appearance.
Average: Average quality, some issues or errors that moderately impact the skeleton's usability or appearance.
Good: Good quality, no noticeable issues/errors or minor issues/errors that slightly impact the skeleton's usability or appearance.

4. Provide an overall rating for the skeleton within the 3D object based on the ratings for each criterion, with more weight given to Joint and Bone Position Relative to Mesh. If the skeleton has good performance on this part, it can be rated as good even if it has minor issues in other criteria. Ensure that the final rating is consistent with the individual scores and the overall critique. If the provided reasons for each criterion indicate a poor quality, the rating should reflect that. Ensure that the ratings and reasons are aligned.

5. Make sure to provide your evaluation following the example output format below.

Example Output:

```
```Critique:
**Joint and Bone Position Relative to Mesh:**
- External Joints and Bones: Good. Most joints and bones are well confined within the mesh's boundary, with minimal protrusion observed.
**Anatomical Accuracy:**
- Correct Joint Placement: Good. Joints are well-aligned with the mesh's primary structural points.
- Natural Poses: Good. The skeleton maintains a natural and plausible pose, consistent with the object's intended design.
**Final Rating**: Good.
```

```Critique:
**Joint and Bone Position Relative to Mesh:**
- External Joints and Bones: Poor. Several joints and bones, particularly in the lower section, are protruding significantly out of the mesh boundary.
**Anatomical Accuracy:**
- Correct Joint Placement: Average. While most joints are correctly placed, some joints are noticeably out of alignment.
- Natural Poses: Poor. The skeleton maintains an unnatural pose, with significant deviations observed.
**Final Rating**: Poor.
```

```Critique:
**Joint and Bone Position Relative to Mesh:**
- External Joints and Bones: Average. Most joints and bones are positioned correctly within the mesh, but some joints, especially in the middle section, slightly protrude, affecting the alignment.
**Anatomical Accuracy**:
- Correct Joint Placement: Average. Joints are generally placed close to the intended structural points, but there are minor misalignments that mildly impact the skeleton's accuracy.
- Natural Poses: Average. The skeleton maintains a generally natural pose, but some joints appear slightly off, giving a somewhat unnatural appearance.
**Final Rating:** Average.
```
```

Figure S7. **Input instructions to VLM for data filtering.**

Task Description:

Our task is to evaluate the categories of a 3D object based on an image that contains eight sub-images rendered from this 3D object. The eight sub-images are concatenated in a 2x4 format. The top four images are its RGB images rendered from four different angles, and the bottom four are its normal maps rendered from four different angles.

Instructions:

1. Carefully examine the provided image, which contains eight different sub-images of the 3D object.

2. Classify the 3D object into one or more of the following categories:
   Character: Human or humanoid objects.
   Animal: Any kind of animal, including pets, wild animals, and mythical creatures.
   Furniture: Items like chairs, tables, beds, etc.
   Electronic Device: Specifically defined as 3C devices, such as phones, Apple Watch, TVs, and other consumer electronics.
   Mythical Creature: Strange creatures, monsters, elves, etc., including dragons and other legendary creatures.
   Anatomy: Parts of the human body or used for medical purposes.
   Tool: Instruments used to perform tasks, like hammers, screwdrivers, etc.
   Planet: Celestial bodies such as planets, moons, and other astronomical objects.
   Musical Instrument: Objects designed to produce music, such as guitars, pianos, drums, etc.
   Sculpture: Artistic objects created by carving, modeling, or assembling materials, often for decorative or artistic purposes.
   Jewelry: Decorative items worn for personal adornment, such as rings, necklaces, bracelets, etc.
   Accessory: Includes fashion accessories (can overlap with jewelry) and parts or recognizable parts of a larger object.
   Paper: Appears to be a flat texture with no thickness.
   Anthropomorphic: Objects with human-like features, even if they are essentially other types of items. For example, Donald Duck, although essentially a duck, can be marked as anthropomorphic due to its human-like features. Anything with hands and feet counts as anthropomorphic. A bell pepper with a smiley face but no hands or feet. A soda can with human-like features.
   Toy: Items designed for play, like dolls, action figures, etc.
   Clothing: Wearable items like shirts, pants, shoes, etc.
   Food: Edible items like fruits, vegetables, cooked dishes, etc.
   Scanned Data: Objects created from 3D scans. If the object appears to be a scanned model, classify it as scanned data and rate it accordingly.
   Architecture: Buildings and other structures.
   Vehicle: Cars, bikes, planes, boats, etc.
   Plant: Trees, flowers, shrubs, etc.
   Weapon: Items designed for combat, like swords, guns, archery, etc.
   Household Item: Common items used in daily life, like utensils, appliances, etc.
   Sporting Goods: Items used in sports and recreational activities, like balls, bats, etc.
   Miscellaneous: Objects that do not fit into any of the current categories but have a clear meaning.

3. Make sure to provide your evaluation following the example output format below.

Example Output:

```
Categories: furniture.
```

```
Categories: weapon.
```

```
Categories: scanned data, architecture.
```

```
Categories: Character.
```

Figure S8. **Input instructions to VLM for category labeling.**

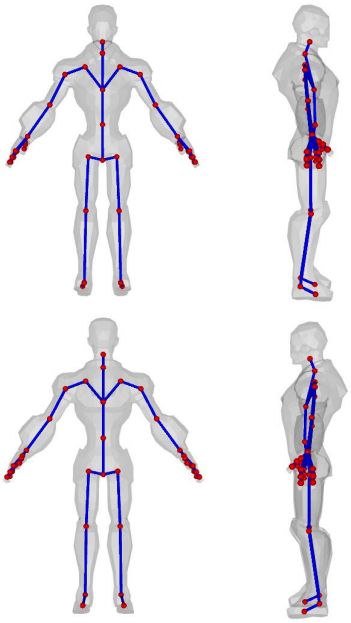

Figure S9. **Input rendered examples to VLM for data filtering.**

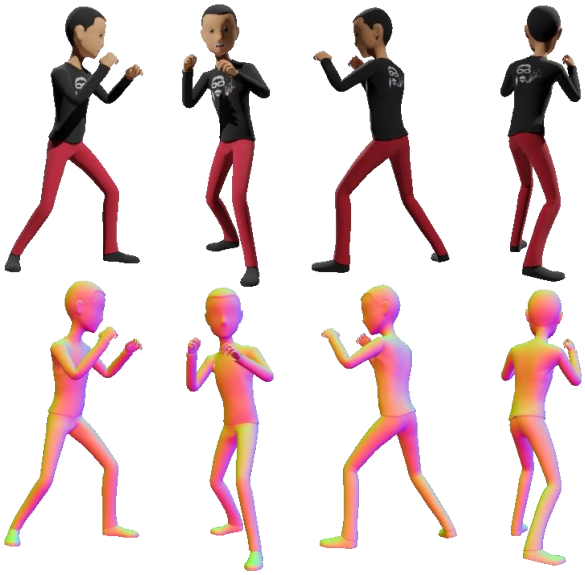

Figure S10. **Input rendered examples to VLM for category labeling.**

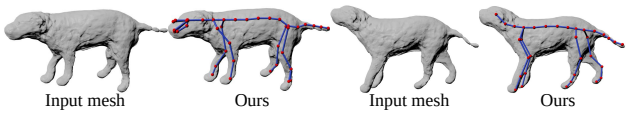

Figure S11. **Failure cases.** When input meshes possess very coarse surfaces (3D reconstruction results from [15]), our generated skeleton may exhibit inaccuracies, such as imperfect connections between the dog's trunk and legs.

[8] Autodesk Inc. Autodesk maya, 2024. Version 2024. 1, 2

[9] 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. 2

[10] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 1

[11] OpenAI. Gpt-4o, 2023. 5

[12] Wenhao Shen, Wanqi Yin, Hao Wang, Chen Wei, Zhongang Cai, Lei Yang, and Guosheng Lin. Hmr-adapter: A lightweight adapter with dual-path cross augmentation for expressive human mesh recovery. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6093–6102, 2024. 2

[13] Chaoyue Song, Jiacheng Wei, Ruibo Li, Fayao Liu, and Guosheng Lin. 3d pose transfer with correspondence learning and mesh refinement. *Advances in Neural Information Processing Systems*, 34:3108–3120, 2021.

[14] Chaoyue Song, Jiacheng Wei, Ruibo Li, Fayao Liu, and Guosheng Lin. Unsupervised 3d pose transfer with cross consistency and dual reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10488–10499, 2023. 2

[15] Chonghyuk Song, Gengshan Yang, Kangle Deng, Jun-Yan Zhu, and Deva Ramanan. Total-recon: Deformable scene reconstruction for embodied view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17671–17682, 2023. 9

[16] 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

[17] Peng-Shuai Wang, Yang Liu, and Xin Tong. Dual octree graph networks for learning adaptive volumetric shape representations. *ACM Transactions on Graphics (TOG)*, 41(4): 1–15, 2022. 1

[18] 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, 4

[19] Biao Zhang and Peter Wonka. Functional diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4723–4732, 2024. 1

[20] 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

[21] 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. 1, 2