SKDream: Controllable Multi-view and 3D Generation with Arbitrary Skeletons

Supplementary Material

We provide additional discussion and results in the appendix and our project page https://skdream3d.github.io/. Readers can check following contents for the questions they may be curious about:

- A. How does the model trained on synthesized skeletons perform on **human-made skeletons**? [Line-819]
- B. What if we only use **single-view condition**? Why do we need **multi-view condition**? [Line-834]
- C. How does the model perform on **novel categories** beyond the training set? [Line-894]
- D. How are the **mesh skeletonization** results compared with other methods? [Line-902]
- E. Implementation details.
- F. Limitations and future works.

Contents

A. Evaluation on Human-made Skeletons

In the main part, we propose a mesh skeletonization method to generate data, and the evaluation is also based on the synthetic skeletons. Considering the potential gap between synthetic skeletons and human-made skeletons, we also evaluate our method on human-made skeletons. We sample 92 human-made skeletons from ModelResource [70] validation set. The skeletons mainly contain animal and human-shape structures. We obtain the text prompts by using Cap3D [36] on the images rendered from meshes.

Comparison results. We compare the proposed SCM model [skeleton correlation module (SCM) + coordinate color encoding with depth (CCE-D) condition] with SDEdit [38], SDEdit+COSAG and raw model [convolutional module + binary condition]. The qualitative results are shown in Fig. 11. The quantitative comparison results are shown in Tab. 4. The results present consistency with Objaverse-SK evaluation, which suggest that:

- **The proposed mesh skeletonization method can produce skeletons similar to human-made skeletons.**
- **Despite our method is built on synthetic skeletons, it can be applied on human-made skeletons.**

Method	SKA Score	PickScore
SDEdit [38]	67.07	17.52
SDEdit+COSAG	68.94	18.58
Ours-Raw	69.13	30.16
Ours-SCM	**77.38**	**33.73**

Table 4. **Quantitative comparison of SKA Score and PickScore on human-made skeletons from ModelResource** (§A).

B. Single-view Condition and Generation

We elaborate on multi-view conditioned multi-view generation (MV2MV) in our main part. The motivation is that multi-view skeleton images can describe object anatomy and pose better. In the following part, we discuss single-view conditioned single-view generation (SV2SV, Appendix B). In this setting, we train a single-view skeletal conditioned generation model based on StableDiffusion-v2.1-base[2]. We use a single-view skeletal image as the input condition. The model is also trained on Objaverse-SK. We render the training images at the resolution of 512^2 (note the resolution of multi-view generation is 256^2). Similar trade-off on resolution occurs in MVDream [55] and StableDiffusion [53].

Model comparison. We compare the proposed SCM model [skeleton correlation module (SCM) + coordinate color encoding with depth (CCE-D) condition] with the raw model [convolutional module + binary condition]. Two self-attention layers with layer normalization is used to model the skeletal correlation.

Results comparison. The quantitative comparison is in Tab. 5, and the qualitative comparison is in Fig. 12 and Fig. 13. We observe that SV2SV models suffer from more severe condition ambiguity, sometimes produce incorrect results, especially on side or back views. The raw model struggles to tell view pose and object anatomy from the skeletal image due to the severe ambiguity. SCM with CCE-D can alleviate ambiguity and improve anatomy and pose

[2]https://huggingface.co/stabilityai/stable-diffusion-2-1-base/tree/main

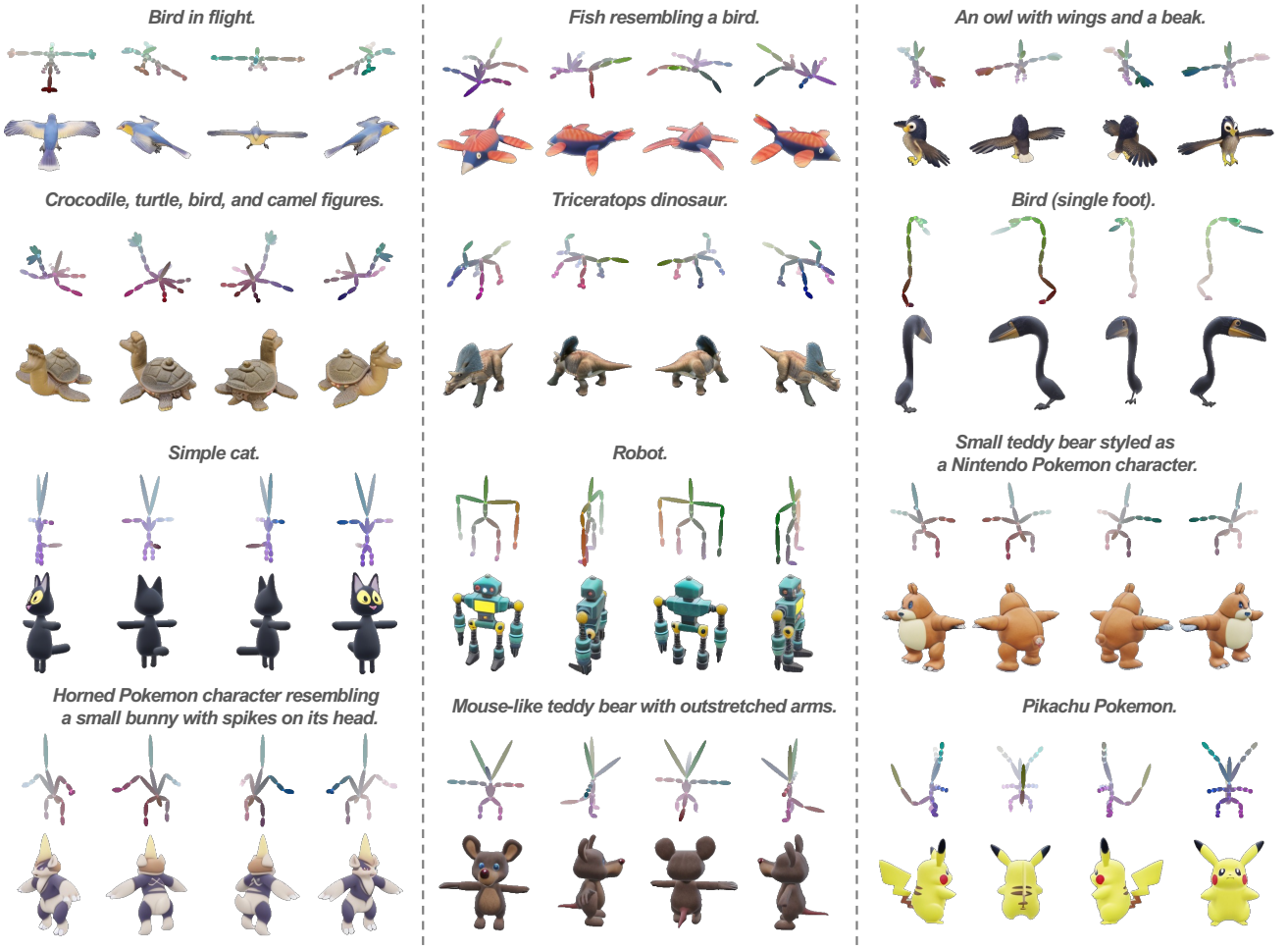

Figure 11. **Qualitative results on human-made skeletons in ModelResource validation set** [70] (§Appendix A). Despite our method is built on synthetic skeletons, the model can also be applied on human-made skeletons.

Method\SKA Score	Mean$^{Inst.}$	Mean^{Class}	Animals	Humans	Plants	Apodes	Bipeds	Quadrupeds	Arthropods	Wings
SV2SV-Raw	64.42	57.66	76.24	63.84	32.89	85.72	83.68	73.41	77.98	68.34
SV2SV-SCM	73.89	68.25	83.55	75.17	46.04	88.85	85.18	82.24	83.21	82.53
MV2MV-Raw	74.69	67.09	88.39	69.90	42.97	**94.74**	82.14	90.18	89.81	84.69
MV2MV-SCM	**81.13**	**74.38**	**91.16**	**78.45**	**53.53**	94.47	**85.20**	**94.19**	**90.68**	**88.40**

Table 5. **Qualitative comparison of Skeleton Alignment Score (SKA)** of single-view conditioned generation (SV2SV) and multi-view conditioned generation (MV2MV) (§Appendix B).

alignment. When using single-view condition, the ambiguity gets more severe, and the improvement is also more significant.

C. Evaluation on Novel Categories

We evaluate our model on ShapeNet [5] to demonstrate the generalization ability. In our training data, animals, human shapes and plants are included while the skeletal conditioned generation can actually generalize to arbitrary categories. We sample 128 instances from three new classes "Airplane", "Chair" and "Guitar" in ShapeNet. Skeletons are extracted and then served as conditions for generation. The qualitative results are in Fig 7. The quantitative comparison results with baseline methods SDEdit [38] and SDEdit+COSAG are in Tab 6. The results suggest that despite the Objaverse-SK mainly covers three categories, the trained model can generalize to other novel categories well, realizing arbitrary skeleton controlled generation.

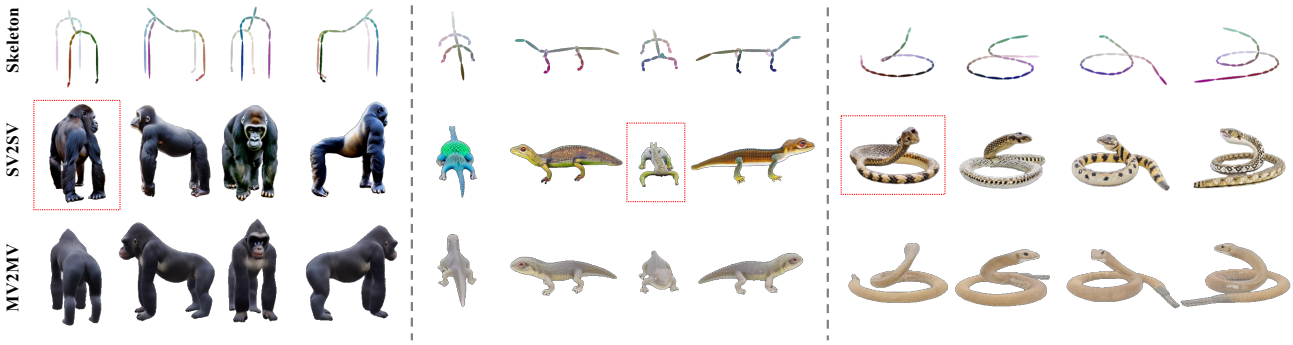

Figure 12. **Qualitative comparison of different generation settings** (§Appendix B). Single-view condition (SV) struggles to control object anatomy and pose precisely, while multi-view (MV) condition performs better.

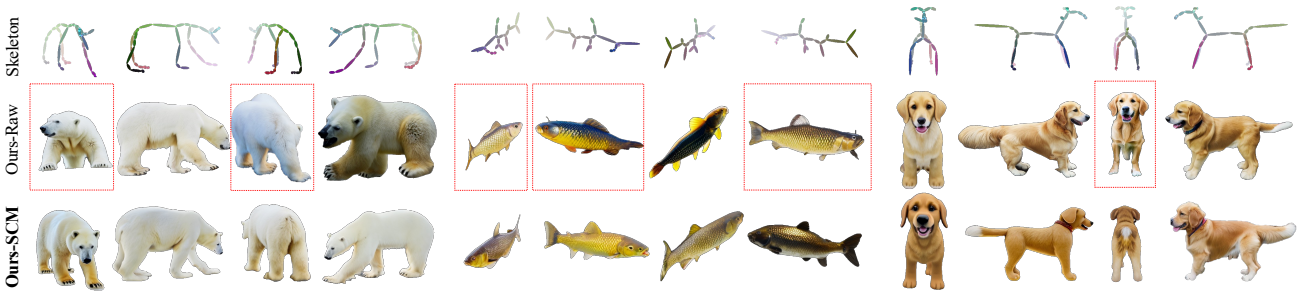

Figure 13. **Qualitative comparison of single-view conditioned single-view generation (SV2SV)** (§Appendix B). SV2SV models suffer from more severe condition ambiguity, while SCM with CCE-D can alleviate ambiguity and improve anatomy and pose alignment.

D. Comparison of Mesh Skeletonization

We compare our method with learning-based method RigNet [71], and the results are shown in Fig. 15. More results of our method are shown in Fig. 14. **The most important reason that RigNet can not be direcly applied is that it assumes that the skeletons are symmetric.** However, the dataset contains many asymmetric objects. Even the object is symmetric, once it is posed, it will also become asymmetric. In addition, since the symmetry constraint is imposed, the object should stay in a determined orientation related to the plane of symmetry. If the orientation is wrong, RigNet will produce wrong results. In addition, RigNet relies on hyperparameters to produce decent results. Using default hyperparameters may produce inaccruate joints and bones. Consequently, the total success rate is around 15% in our test. On the contrary, our method runs without limitation of symmetry and is not sensitive to hyperparameters. It can produce more reliable results with a higher success rate around 80%.

Failure cases. We show the failure cases of our pipeline in Fig. 16. The skeletons may not be properly generated for non-tree like structures, e.g. a ball or a bottle. When the input mesh does not meet the watertight requirement, our pipeline may also fail. For example, the mesh is incomplete/broken, or the mesh consists of multiple parts/contains open surfaces.

E. Implementation Details

E.1. Dataset Construction

Mesh preprocessing. In order to construct the mesh-skeleton pairs with a high success rate, we propose a full pipeline starting from an arbitrary mesh to final skeleton. The mesh preprocessing and rendering are finished in Blender[3]: a) **Normalization**. Given a mesh file, we first normalize it into $(-0.5, 0.5)^3$. Files with a size larger than 200M are filtered to avoid crash. b) **Remeshing**. The remesh modifier is applied, with the voxel size set as 0.005. We need to make sure the mesh is watertight before skeletonization. c) **Decimation**. To accelerate later skeletonization steps, the remeshed result is further decimated with a ratio of 0.2, i.e. the face count is reduced into 1/5.

Mesh skeletonization. We use the implementation of Mean Curvature Flow [58] in CGAL library[4]. After curve graph are generated from the preprocessed mesh, we first find the largest connected component. Only the main object of the mesh is considered. Then the graph is separate into parts by intersection points. The Douglas–Peucker algorithm [12] is

[3]https://www.blender.org/
[4]https://www.cgal.org/

Method	Training	SKA Score				PickScore			
		Mean$_{inst.}$	Airplane	Chair	Guitar	Win Rate	Airplane	Chair	Guitar
SDEdit [38]	❍	68.60	72.92	63.33	69.58	23.92	29.65	26.16	15.77
SDEdit+COSAG	◗	69.40	71.92	65.74	70.57	24.12	31.10	23.84	17.26
Ours-SCM	●	**74.30**	**81.54**	**69.93**	**71.36**	**51.95**	**39.24**	**50.00**	**66.96**

Table 6. **Comparison of Skeleton Alignment Score (SKA) and PickScore of novel categories from ShapeNet** [5] (§Appendix C).

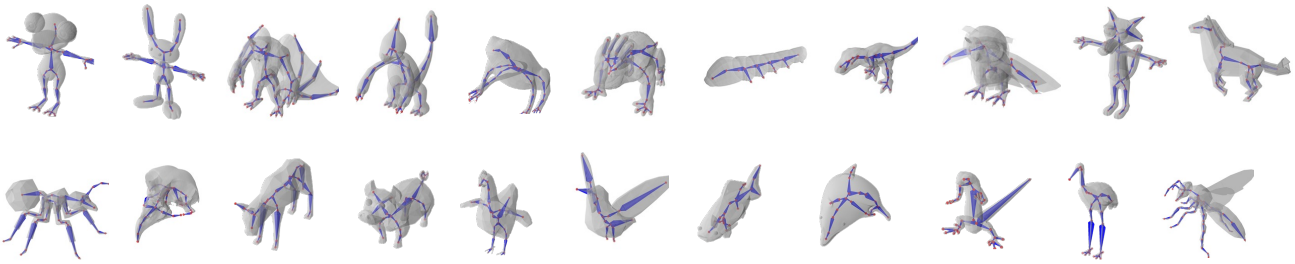

Figure 14. **Demonstration of generated skeletons in our Objaverse-SK dataset** (§Appendix D).

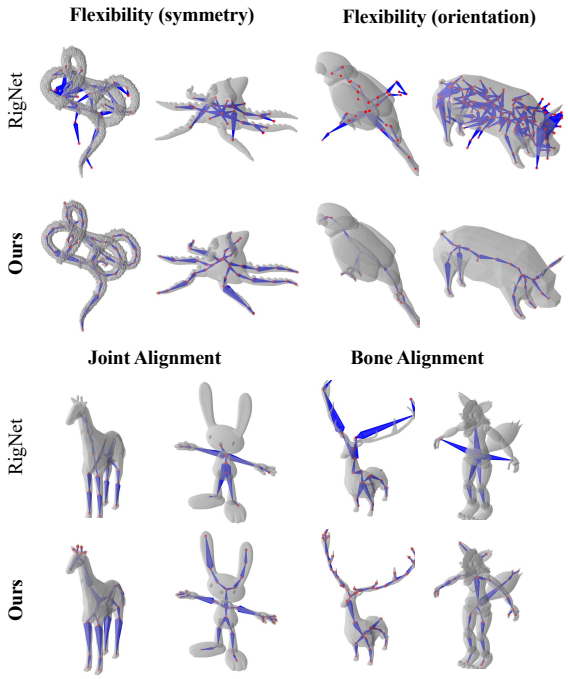

Figure 15. **Comparison of skeletons generated by RigNet [71] and our method** (§Appendix D).

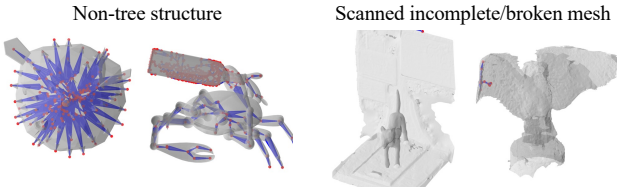

Figure 16. **Failure cases of our mesh skeletonization pipeline** (§Appendix D).

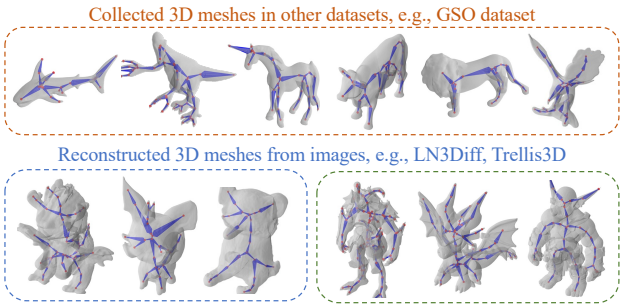

Figure 17. **Examples of more mesh sources for data extension.** (§Appendix F).

used to simplify each part, with the distance threshold set as 0.01. In addition, points with a distance less than 0.01 are also merged. Later, the sparse graph is converted into a spanning tree to remove cycles. Finally, the root of the skeleton is determined by finding the minimum height tree.

Mesh and skeleton rendering. For each mesh file, we randomly select 4 elevation angles in $[-10°, 45°]$ degrees. For each elevation angle, 32 azimuth angles are selected uniformly in $360°$. The FOV of the camera is set as $45°$. The distance between the camera and the object is randomly set between $[2.5, 3.5]$. Finally, 128 RGB images with a size 256×256 are rendered for each object. We use the EEVEE engine in Blender for fast rendering. For each RGB image, the corresponding skeleton is rendered with the same camera parameters. The joints are projected by the perspective transformation and colored by the proposed coordinate color encoding method. Bones are then drawn between joints, and bone colors are determined by the center points. During projection, the depth values are calculated and are inversed and normalized to $[0.2, 1]$ as the alpha channel.

E.2. Model Training

The models are trained on our proposed Objaverse-SK dataset with a learning rate of 1×10^{-5}. Multi-view models are trained with 4k steps, and the batch size is 240*4 (four views). For models without skeletal correlation module, we train 8k steps for convergence. Single-view models are trained with 10k steps, and the batch size is 240. Since the image resolution for multi-view training is 256^2 while that for single-view training is 512^2, the total GPU memory consumption is similar. Diffusers [5] and Accelerate [6] libraries are used for mix-precision training. The implementation of the models is based on MVDream [55] and MVControl [30].

F. Limitation and Future Work

Shape representation. Noticing the limited capacity of text for shape description, we resort to skeletons. However, there are still some objects which can not be well described by skeletons (Fig. 16). A possible future work is to design more general and expressive shape representations as conditions. Some works propose new skeletal shape representations [11], but the utility and simplicity for editing and articulation may be compromised.

Skeleton ambiguity. Although we propose to use multi-view generation to avoid skeleton ambiguity, there are still some cases that the skeleton is not correctly recognized. The key problem is that parts in the skeleton are not bind with specific semantics. A meaningful future work is to inject semantic information into the skeletal conditions. For example, the word "head" is bind with the head joints in the skeleton and can be recognized by the model. This will not only help the model to understand the skeleton and generate correct content but also enable more flexible controlling.

Generation paradigm. We build our generation framework as multi-view to 3D generation. The generation quality is limited by the low resolution of multi-view images. Recently native 3D generation methods [7, 21, 25, 67, 76] have achieved impressive results. Injecting skeletal conditions into native 3D generation frameworks for more accurate spatial control and high-quality generation is also a meaningful topic to study.

Data extension. We construct our dataset upon Objaverse and mainly contain three classes (animals, human shapes, and plants). As for more general skeletal control, extending the dataset into a broader domain is important. Multiple data sources can be considered (Fig. 17). First, collecting meshes from more datasets like GSO and Objaverse-XL. The diversity of skeletons can be further enhanced. Second, given the promising results of LRMs, meshes can be reconstructed from 2D images. As a result, image-mesh-skeleton triplets can be obtained by using LRMs and our skeletonization method. The data scale of 2D image datasets is much larger than 3D mesh datasets and appearance realism is also better.

[5] https://huggingface.co/docs/diffusers/en/index
[6] https://huggingface.co/docs/accelerate/en/index