

A APPENDIX

A.1 MESH SKELETONIZATION

Result comparison. We compare our method with learning-based method RigNet (Xu et al., 2020), and the results are shown in Fig. 13. More results of our method are shown in Fig. 14. **The most important reason that RigNet can not be directly 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 inaccurate 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. 12. The skeletons may not be properly generated for non-tree like structures, e.g. a ball or a bottle. When the input mesh is incomplete or broken (e.g. mesh scanned from real-world), our pipeline may also fail, since it requires the input mesh to be watertight.

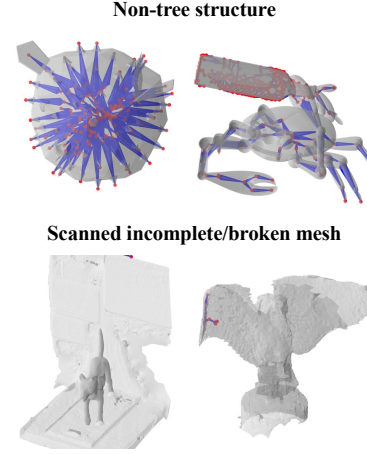


Figure 12: Failure cases of our mesh skeletonization pipeline.

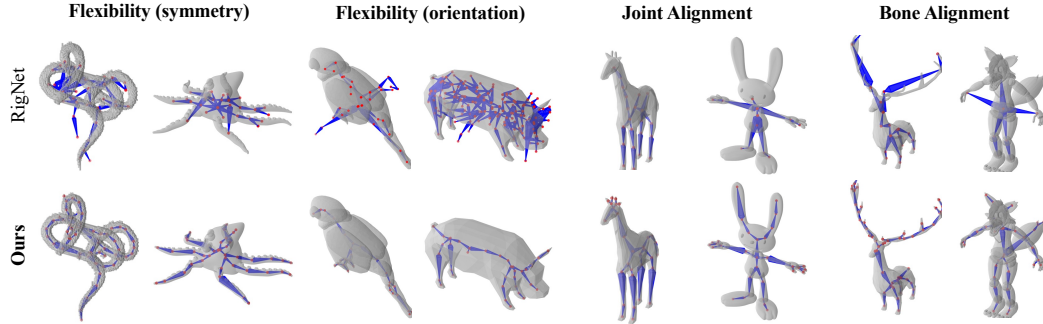


Figure 13: Comparison of skeletons generated by RigNet (Xu et al., 2020) and our method.

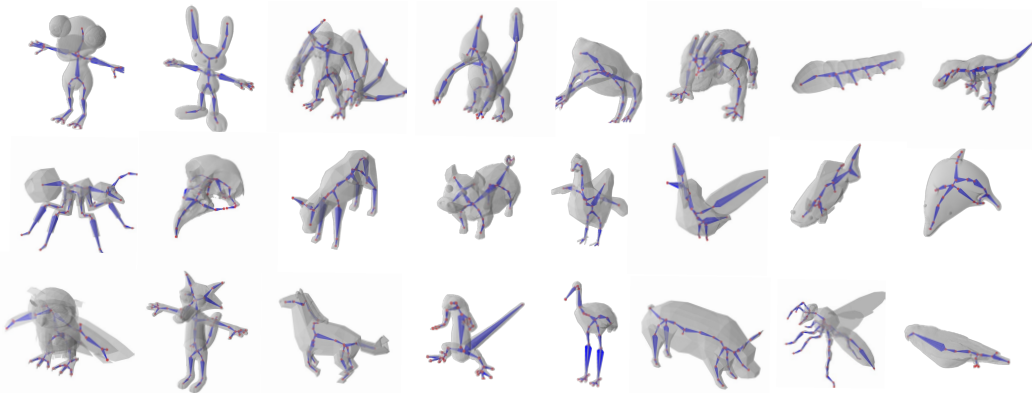


Figure 14: Demonstration of generated skeletons in our Objaverse-SK dataset.

A.2 SKELETAL CONDITIONED SINGLE-VIEW GENERATION

We train a single-view (SV) skeletal conditioned generation model. The base diffusion model is StableDiffusion 2.1 base¹. We use a single-view skeletal image with CCE-D as the input condition. Similar to ControlNet (Zhang et al., 2023), the UNet encoder is copied and trained for conditional generation. The model is also trained on Objaverse-SK. We render the training images at the resolution of 512² (note the resolution of multi-view training is 256²). The same resolution trade-off occurs in MVDream (Shi et al., 2023) and StableDiffusion (Rombach et al., 2022).

A.2.1 SINGLE-VIEW SKELETON CORRELATION MODELING

We also apply our skeleton correlation module (SCM) in single-view generation, in which the cross-view attention is replaced with a self-attention layer. The alignment scores are evaluated and results are shown in Tab. 3. Similar to the multi-view scenario, SCM also facilitates conditioned learning and achieves higher skeleton alignment score for single-view conditions.

Method/SKA Score	Mean ^{Inst.}	Mean ^{Class}	Animals	Humans	Plants	Apodes	Bipeds	Quadrupeds	Arthropods	Wings
SV w/o SCM	67.58	57.06	83.22	60.44	27.53	89.24	83.06	82.94	82.97	75.44
SV with SCM	74.39	65.86	86.78	70.15	40.66	88.93	83.93	89.60	82.50	86.45

Table 3: Comparison of Skeleton Alignment Score (SKA) between models with and without SCM.

A.2.2 SINGLE-VIEW VS. MULTI-VIEW GENERATION

We compare the result of single-view (SV) and multi-view (MV) generation. The SV model generates images with a higher resolution of 512², but suffer from skeleton ambiguity. Although the MV model produces lower resolution images, the anatomy and pose can be determined by multiple views. As a result, when considering the possible poses and orientations of the skeletal conditions, MV model tends to perform better.

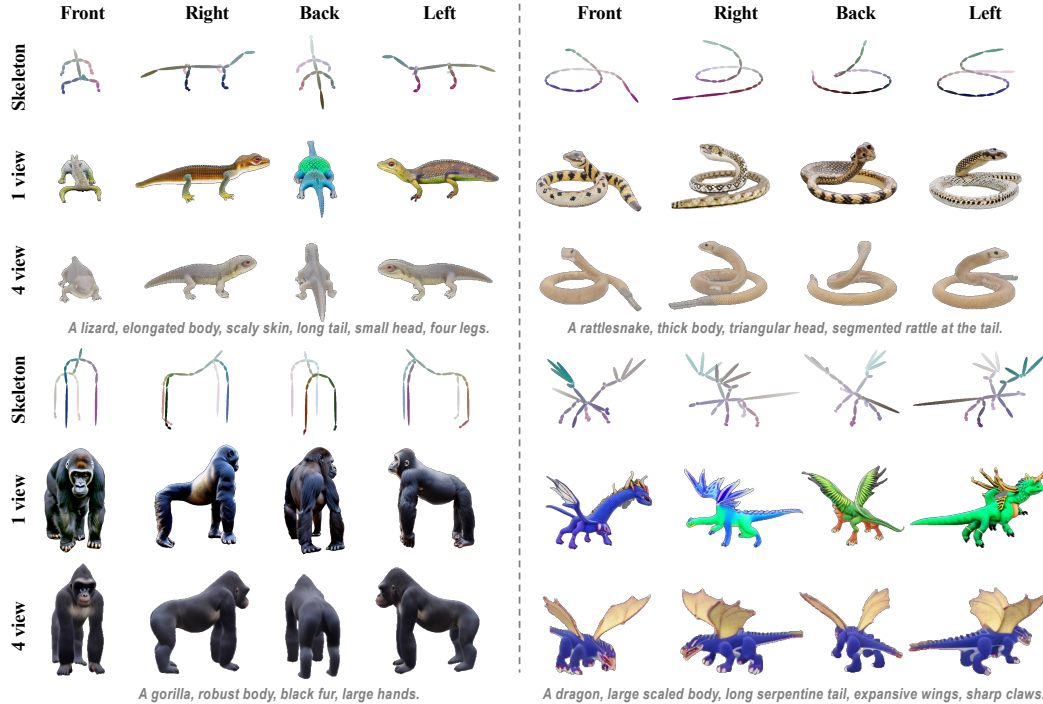


Figure 15: Comparison of single-view and multi-view generation. Since the single view condition may be ambiguous, the correctness of generated content could be affected.

¹<https://huggingface.co/stabilityai/stable-diffusion-2-1-base/tree/main>

A.3 ADDITIONAL EVALUATION RESULTS

We evaluate our model on ShapeNet (Chang et al., 2015) 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 150 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 16. The quantitative evaluation results are in Tab 4.

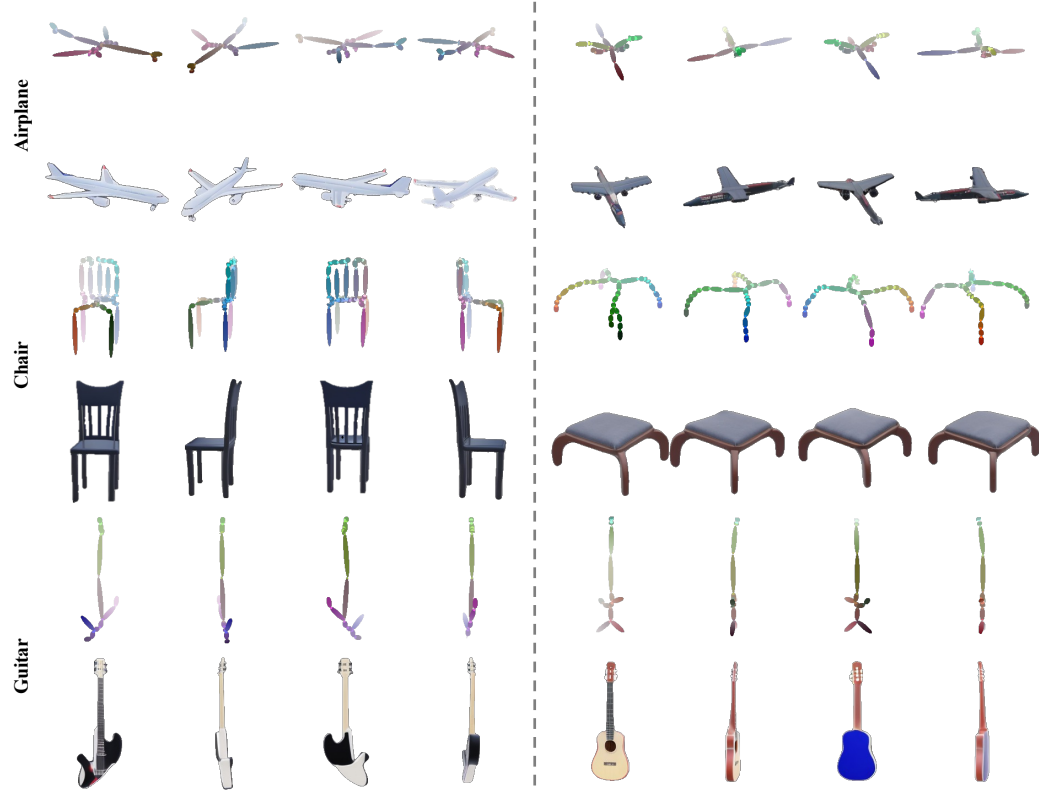


Figure 16: Skeletal conditioned generation on categories not covered by the training set.

Method	Training	PickScore				SKA Score			
		Win Rate	Airplane	Chair	Guitar	Mean _{inst.}	Airplane	Chair	Guitar
SDEdit (Meng et al., 2021)	○	24.57	33.43	21.22	19.05	70.43	76.61	65.34	69.38
SDEdit+COSAG	◐	24.29	32.56	19.19	21.13	69.84	75.43	64.54	69.54
Ours	●	51.14	34.01	59.59	59.82	74.55	81.74	70.00	71.91

Table 4: Comparison of Skeleton Alignment Score (SKA) and PickScore on ShapeNet (Chang et al., 2015).

A.4 IMPLEMENTATION DETAILS

A.4.1 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 Blender²: 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

²<https://www.blender.org/>

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 (Tagliasacchi et al., 2012) in CGAL library³. 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 (Douglas & Peucker, 1973) is 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 $[-10^\circ, 45^\circ]$ 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.

A.4.2 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 6k 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. Diffusers⁴ and Accelerate⁵ libraries are used for mix-precision training. The implementation of the models is based on MVDream (Shi et al., 2023) and MVControl (Li et al., 2023d).

A.5 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. 12). A possible future work is to design more general and expressive shape representations as conditions. Some works propose new skeletal shape representations (Dou et al., 2022; Guo et al., 2023), 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.

³<https://www.cgal.org/>

⁴<https://huggingface.co/docs/diffusers/en/index>

⁵<https://huggingface.co/docs/accelerate/en/index>