Supplementary Materials for LGM: Large Multi-View Gaussian Model for High-Resolution 3D Content Creation

Jiaxiang Tang[1⋆], Zhaoxi Chen[2], Xiaokang Chen[1], Tengfei Wang[3], Gang Zeng[1], and Ziwei Liu[2]

[1] National Key Lab of General AI, Peking University
[2] S-Lab, Nanyang Technological University
[3] Shanghai AI Lab

A More Implementation Details

Datasets. The full list of words for filtering the Objaverse dataset is: *'flying, mountain, trash, featuring, a set of, a small, numerous, square, collection, broken, group, ceiling, wall, various, elements, splatter, resembling, landscape, stair, silhouette, garbage, debris, room, preview, floor, grass, house, beam, white, background, building, cube, box, frame, roof, structure'*. The 100 camera views we use form a spiral path on the sphere surface. The camera radius is fixed to 1.5, and the field-of-view angle is fixed to 49.1 degree.

Mesh Extraction. The mesh extraction process contains three stages. 1) Gaussians to NeRF: we train an efficient NeRF similar to Instant-NGP [4] for 512 iterations using the rendered Gaussians as ground truth images. We supervise both RGB and alpha channels at the rendering resolution of 128×128 using MSE loss. The learning rate is set to 0.01 for the grids and 0.001 for MLPs. `nerfacc` [2] is adopted for efficient training. The training camera views are randomly sampled with azimuth from $[-180, 180]$ degree, elevation from $[-45, 45]$ degree, and radius from $[1.5, 3.0]$. 2) NeRF to Mesh: we first extract the mesh using Marching Cubes [3] with a grid resolution of 256 and density threshold of 10. We then train the vertex deformation and the appearance grid for 2048 iterations at the rendering resolution of 512×512. The learning rate for deformation is set to 10^{-4}. Following NeRF2Mesh [9], we apply normal consistency loss and perform remeshing every 512 iterations to make the final mesh smooth. 3) Texture optimization: finally, we unwrap the UV coordinates using the mesh and bake the appearance grid to a 2D texture image of resolution 1024×1024. This texture image is further optimized using a learning rate of 0.001 for another 512 iterations at the rendering resolution of 512×512.

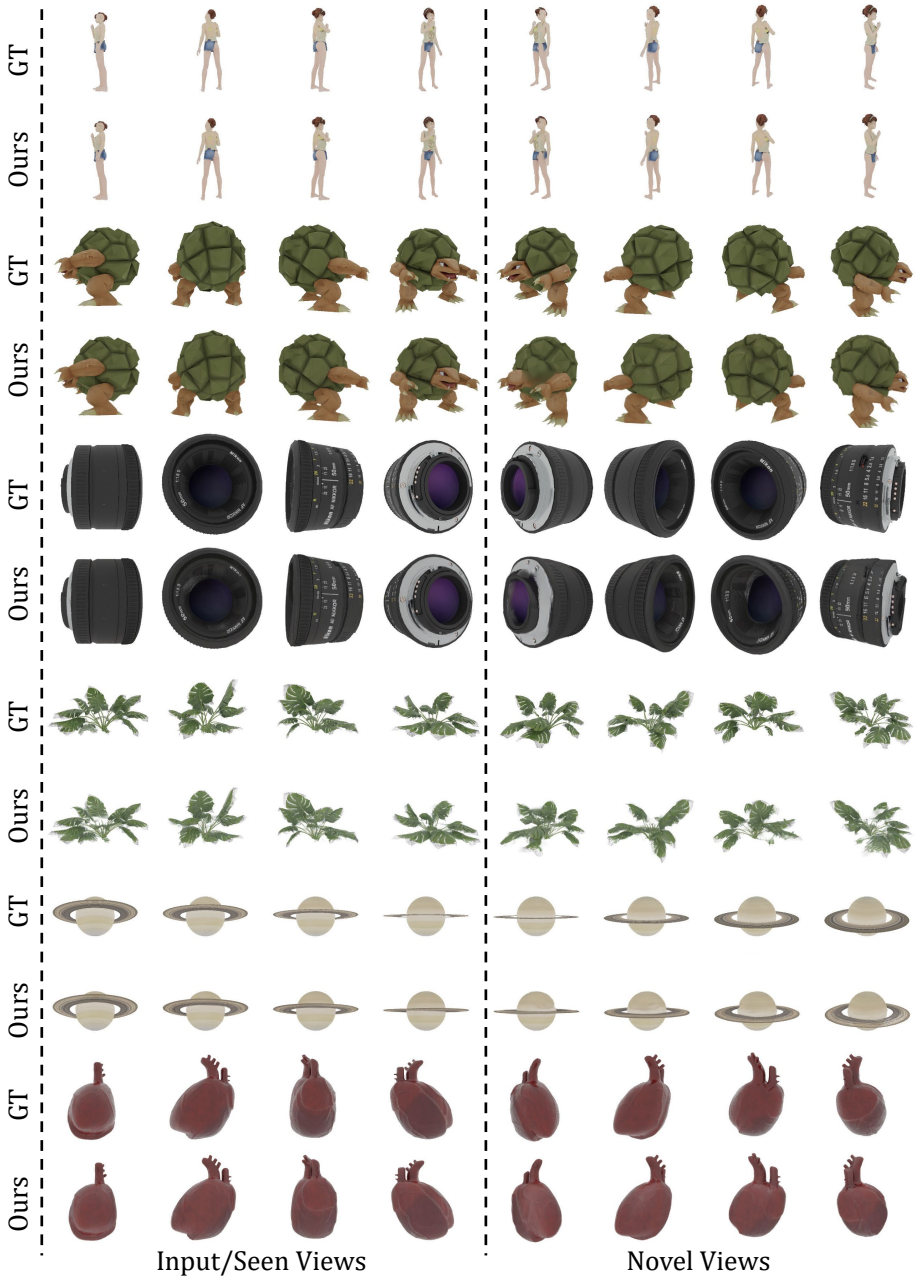

Fig. 1: Visualization of reconstruction results. We show our model's reconstruction results on the test dataset. The left four columns are also used as the input, and the right four columns are novel views.

Fig. 2: Comparisons between different meshing method from Gaussians. We compare our meshing method with DreamGaussian [8].

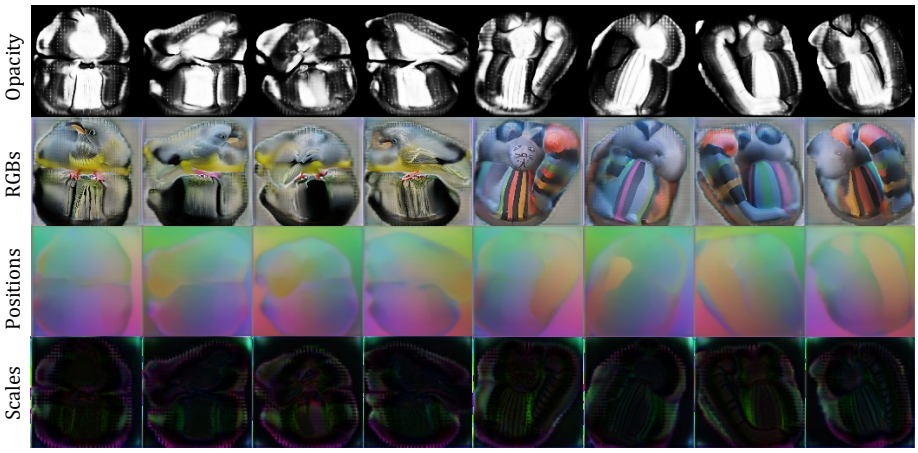

Fig. 3: Visualization of Gaussian feature maps. We visualize the opacity, RGB color, 3D position, and scales of each pixel-aligned Gaussian in our four 128×128 output images.

B More Results

Reconstruction Quality. In Figure 1, we visualize the reconstruction results of our method. The left four views are used as the input to our method, and the right four views are predicted by the model. Our model can reconstruct accurate geometry and faithful details from the input views.

* Work done while visiting S-Lab, Nanyang Technological University.

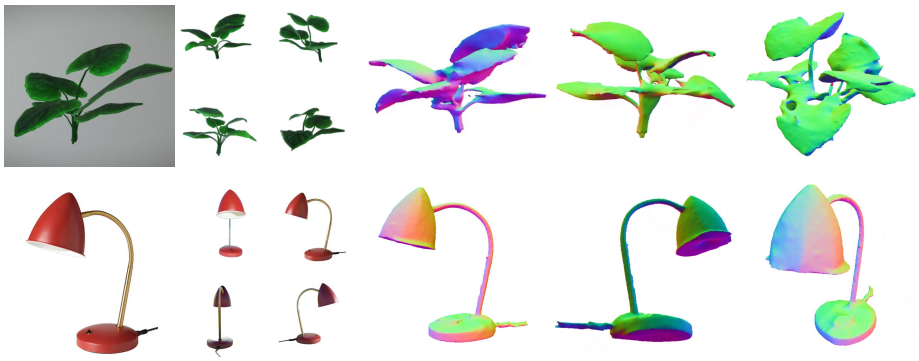

Fig. 4: Mesh Extraction for hard examples.

Fig. 5: Visualization of our limitations. We show three major reasons for failure cases of our method.

CLIP Similarity. To quantitatively evaluate the generation quality of our method, we calculate the CLIP similarity [5] between the generated 3D models and the input images. Specifically, for each method, we render the generated 3D models from zero elevation and 60 evenly distributed azimuths, and average the cosine similarities between the CLIP features of each rendered image and the input image. We use three CLIP backbones [1, 6] with different sizes for a thorough evaluation. Our model performs best consistently compared to other recent methods on 3D Gaussians generation.

	DreamGaussian [8]	TriplaneGaussian [11]	LGM (Ours)
CLIP-ViT-base	81.75	84.65	**88.47**
CLIP-ViT-large	70.08	76.55	**83.21**
CLIP-ViT-bigG	65.59	73.03	**80.16**

Table 1: Comparisons on CLIP-Similarity for Image-to-3D. We calculate the CLIP-similarity between the input image and generated 3D model with different CLIP backbones.

Different Meshing Method. Figure 2 presents a comparison between our meshing algorithm and the technique introduced in DreamGaussian [8]. Our algorithm generates a smoother surface, which is advantageous for subsequent tasks such as relighting. Moreover, our method operates independently of the underlying 3D Gaussians, as it relies solely on the rendered images.

Feature Map Visualization. In Figure 3, we visualize the Gaussian features of the four output images from our U-Net model. It can be observed that each image contains some extra occupied positions similar to [7], which is important to complete the unseen or occluded part of the 3D model (*e.g.*, the top and bottom views).

Further analysis on mesh extraction. Mesh extraction from 3D Gaussians presents a challenging problem, and our method employs NeRF as an intermediate representation to tackle this issue. Additional mesh extraction results are provided in Figure 4. Even for challenging examples such as plant leaves and thin structures, our method is capable of generating plausible meshes.

C Limitations

We visualize failure cases of our method in Figure 5 to gain a deeper understanding of its weaknesses. As previously mentioned in the main paper, the primary causes of these failures stem from the flawed multi-view images produced in the initial step. The resolution of these multi-view images is limited to 256×256, which can diminish the quality of the input image. Despite implementing data augmentation during training to emulate 3D inconsistencies and attempting to bridge the domain gap, this approach still results in inaccuracies for slender structures, such as chairs. Additionally, ImageDream [10] struggles with images that have significant elevation angle, occasionally producing images with a dark appearance.

References

1. Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., Schmidt, L.:

Openclip (Jul 2021). https://doi.org/10.5281/zenodo.5143773, https://doi.org/10.5281/zenodo.5143773, if you use this software, please cite it as below.
2. Li, R., Gao, H., Tancik, M., Kanazawa, A.: Nerfacc: Efficient sampling accelerates nerfs. arXiv preprint arXiv:2305.04966 (2023)
3. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: Seminal graphics: pioneering efforts that shaped the field, pp. 347–353 (1998)
4. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM TOG (2022)
5. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022)
6. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021)
7. Szymanowicz, S., Rupprecht, C., Vedaldi, A.: Splatter image: Ultra-fast single-view 3d reconstruction. In: arXiv (2023)
8. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. arXiv preprint arXiv:2309.16653 (2023)
9. Tang, J., Zhou, H., Chen, X., Hu, T., Ding, E., Wang, J., Zeng, G.: Delicate textured mesh recovery from nerf via adaptive surface refinement. arXiv preprint arXiv:2303.02091 (2022)
10. Wang, P., Shi, Y.: Imagedream: Image-prompt multi-view diffusion for 3d generation. arXiv preprint arXiv:2312.02201 (2023)
11. Zou, Z.X., Yu, Z., Guo, Y.C., Li, Y., Liang, D., Cao, Y.P., Zhang, S.H.: Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. arXiv preprint arXiv:2312.09147 (2023)