GRM: Large Gaussian Reconstruction Model for Efficient 3D Reconstruction and Generation Supplementary Material

Yinghao Xu[1⋆], Zifan Shi[1,2⋆], Yifan Wang[1], Hansheng Chen[1]
Ceyuan Yang[3], Sida Peng[4], Yujun Shen[5], and Gordon Wetzstein[1]

[1] Stanford University
[2] The Hong Kong University of Science and Technology
[3] Shanghai AI Laboratory
[4] Zhejiang University
[5] Ant Group

This supplementary material is organized as follows. We first introduce implementation details of our GRM (Appendix A) . Then, we evaluate the geometry quality of our GRM against the baselines (Appendix B). We also present the details of mesh extraction from 3D Gaussians in Appendix C. Finally, we show additional results on 3D reconstruction and generation to evaluate the flexibility and effectiveness of our approach (Appendix D).

A Implementation Details

Network Architecture and Training Details. We illustrate the details of network architecture and training in Tab. 1.

Training Data. We obtain multi-view images from Objaverse [2] as training inputs. Objaverse contains more than $800k$ 3D objects with varied quality. Following [5], we filter $100k$ high-quality objects, and render 32 images at random viewpoints with a fixed 50° field of view under ambient lighting.

Test Data. We use Google Scanned Objects (GSO) [3], and render a total of 64 test views with equidistant azimuth at $\{10, 20, 30, 40\}$ degree elevations. In sparse-view reconstruction, the evaluation uses full renderings from 100 objects to assess all models. For single-view reconstruction, we restrict the analysis to renderings generated at an elevation angle of 20 from 250 objects.

Deferred Backpropagation. Our model generates $4 \times 512 \times 512$ Gaussians, consuming a significant amount of GPU memory. We would only be able to train our model with a batch size of 2 on 80GB A100 GPUs. Deferred backpropagation [13] is an important technique for saving GPU memory in large batch-size training. With it, we are able to scale up the batch size to 8, consuming only 38GB per GPU. We provide a pseudo code (Algorithm 1) to demonstrate how we implement it in our model training.

⋆ Equal Contribution

Table 1: Implementation details.

Encoder	Convolution layer	1, kernel size 16, stride 16
	Att layers	24, channel width 768, # heads 12
Upsampler block	Pixelshuffle per block	1, scale factor 2
	Att layers per block	2, # heads 12
	Channel width	starting from 768, decay ratio of 2 per block
	# Blocks	4
Gaussian splatting	Color activation	sigmoid
	Rotation activation	normalize
	Opacity activation	sigmoid
	Scale activation	sigmoid
	Position activation	None
Training details	Learning rate	3e-4
	Learning rate scheduler	Cosine
	Optimizer	AdamW
	(Beta1, Beta2)	(0.9, 0.95)
	Weight decay	0.05
	Warm-up	3000
	Batch size	8 per GPU
	# GPU	32

Perceptual loss. We have experimented with an alternative loss to the conventional perceptual loss mentioned in the paper, known as the Learned Perceptual Image Patch Similarity (LPIPS) loss [14]. However, we observe severe unstable training and the model cannot converge well.

B Geometry Evaluation

Here, we demonstrate the geometry evaluation results on sparse-view reconstruction and single-image-to-3D generation. We report Chamfer Distance (CD) and F-score as the evaluation metrics. Specifically, we use different thresholds for F-score to reduce the evaluation uncertainty. We use ICP alignment to register all the 3D shapes into the same canonical space. All metrics are evaluated on the original scale in the GSO dataset.

Sparse-view Reconstruction. We compare with SparseNeuS [9] which trained in One-2-3-45 [7], and LGM [10] in Tab. 2. The SparseNeuS exhibits a very high CD score with 16 views for reconstruction (32 views in the original paper) because of the far-away floaters. GRM achieves better geometry scores across all metrics, particularly on the F-score with small thresholds.

Single-Image-to-3D Generation. We compare `GRM` against baselines on geometry quality in Tab. 3. The original implementation of One-2-3-45++ [6] suffers from a limitation where it can only generate a single component in multi-object scenes, resulting in geometry metrics that are not as good as other baseline methods. `GRM` outperforms all baseline methods across all metrics. Moreover, when compared to optimization-based methods, such as DreamGaussian [11],

Algorithm 1 Pseudocode of Deferred Backpropagation on Gaussian Rendering in PyTorch-like style.

`Render`: Rendering process;

```
class DBGaussianRender(torch.autograd.Function):
    def forward(ctx, gaussians, cameras):
        # save for backpropagation
        ctx.save_for_backward(gaussians, cameras)

        with torch.no_grad():
            images = Render(gaussians, cameras)

        return images

    def backward(ctx, grad_images):
        # restore input tensor
        gaussians, cameras = ctx.saved_tensors

        with torch.enable_grad():
            images = Render(gaussians, cameras)
            images.backward(grad_images)

        return gaussians.grad
```

Table 2: Geometry evaluation on Sparse-view Reconstruction. SparseNeuS [7, 9] exhibits an exceptionally high CD due to the far-away floaters.

Method	#Views	CD↓	F-Score(0.01)↑	F-Score(0.005) ↑
SparseNeuS [7, 9]	16	0.02300	0.3674	0.5822
LGM [10]	4	0.00393	0.9402	0.7694
`GRM` (Ours)	4	**0.00358**	**0.9560**	**0.8239**

our approach demonstrates notable runtime advantages along with superior geometry quality.

Table 3: Geometry evaluation on Single-image-to-3D generation. Note that the original implementation of One-2-3-45++ [6] suffers from a limitation where it can only generate a single component in multi-object scenes.

Method	CD↓	F-Score(0.01)↑	F-Score(0.025) ↑
One-2-3-45++ [6]	0.0145	0.6419	0.8362
Wonder3D [8]	0.0131	0.6384	0.8576
One-2-3-45 [7]	0.0134	0.6689	0.8682
Shap-E [4]	0.0118	0.6990	0.8820
LGM [10]	0.0123	0.6853	0.8591
DreamGaussian [11]	0.0077	0.7616	0.9506
`GRM` (Ours)	**0.0058**	**0.8758**	**0.9775**

Fig. 1: Blender scene constructed with our textured mesh.

C Mesh Extraction from 3D Gaussians

We utilize the Fibonacci sampling method to sample 200 uniformly distributed cameras on sphere for rendering images and depth maps based on the 3D Gaussians of the scene. Subsequently, we fuse the RGB-D data using the TSDFVolume [1] method to generate a mesh. We must take into account that due to the Gaussian distribution, some points may scatter outside the surface of the object. Therefore, we employ clustering to remove very small floaters outside the object's surface in order to smooth the generated mesh.

D Additional Visual Results

We assemble the extracted texture mesh in Blender to construct a 3D scene. We attach the scene image in Fig. 1. We include more qualitative results on sparse-view reconstruction, text-to-3D generation and image-to-3D generation in Fig. 2, Fig. 3 and Fig. 4, respectively.

E Limitations

The output quality of our sparse-view reconstructor suffers when the input views are inconsistent. The reconstructor is deterministic in nature and future work could embed it in a probabilistic framework, akin to DMV3D [12]. Our current framework is limited to object-centric scenes due to the lack of large-scale 3D scene datasets. Future work could explore the generation of larger and more complicated scenes.

Despite the high-quality reconstruction, image-to-3D and text-to-3D generation results we achieved, our model relies on the input information for reconstruction and lacks the capability for hallucination. For example, if a region is not observed in any of the input images, the model may produce blurry textures for it.

Ethics. Generative models pose a societal threat—we do not condone using our work to generate deep fakes intending to spread misinformation.

References

1. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (1996) 4
2. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 13142–13153 (2023) 1
3. Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., McHugh, T.B., Vanhoucke, V.: Google scanned objects: A high-quality dataset of 3d scanned household items. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 2553–2560. IEEE (2022) 1
4. Jun, H., Nichol, A.: Shap-e: Generating conditional 3d implicit functions. arXiv preprint arXiv:2305.02463 (2023) 3
5. Li, J., Tan, H., Zhang, K., Xu, Z., Luan, F., Xu, Y., Hong, Y., Sunkavalli, K., Shakhnarovich, G., Bi, S.: Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. https://arxiv.org/abs/2311.06214 (2023) 1
6. Liu, M., Shi, R., Chen, L., Zhang, Z., Xu, C., Wei, X., Chen, H., Zeng, C., Gu, J., Su, H.: One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. arXiv preprint arXiv:2311.07885 (2023) 2, 3
7. Liu, M., Xu, C., Jin, H., Chen, L., T, M.V., Xu, Z., Su, H.: One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization (2023) 2, 3
8. Long, X., Guo, Y.C., Lin, C., Liu, Y., Dou, Z., Liu, L., Ma, Y., Zhang, S.H., Habermann, M., Theobalt, C., et al.: Wonder3d: Single image to 3d using cross-domain diffusion. arXiv preprint arXiv:2310.15008 (2023) 3
9. Long, X., Lin, C., Wang, P., Komura, T., Wang, W.: Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In: Eur. Conf. Comput. Vis. (2022) 2, 3
10. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: Lgm: Large multi-view gaussian model for high-resolution 3d content creation. arXiv preprint arXiv:2402.05054 (2024) 2, 3

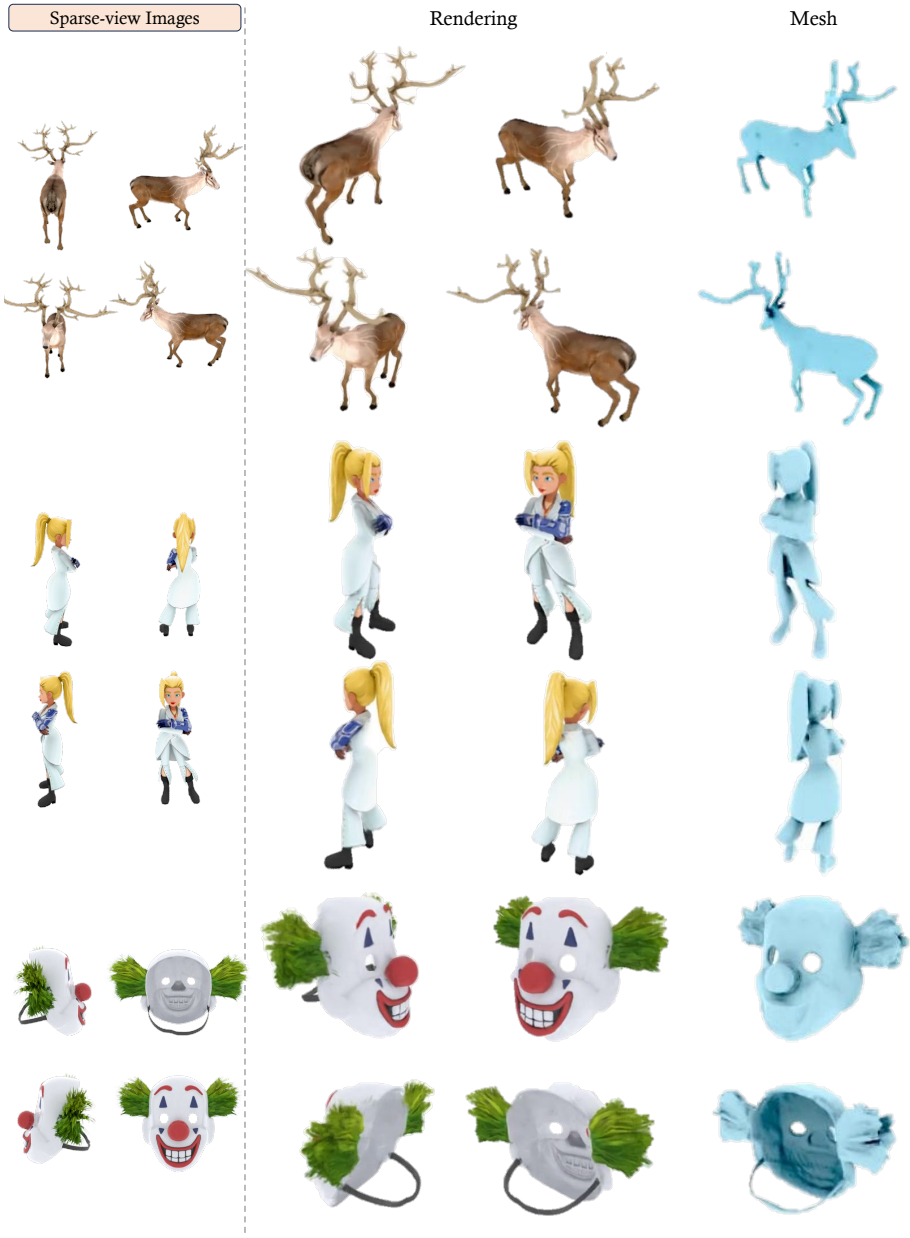

Fig. 2: Sparse-view Reconstruction.

11. 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) 2, 3

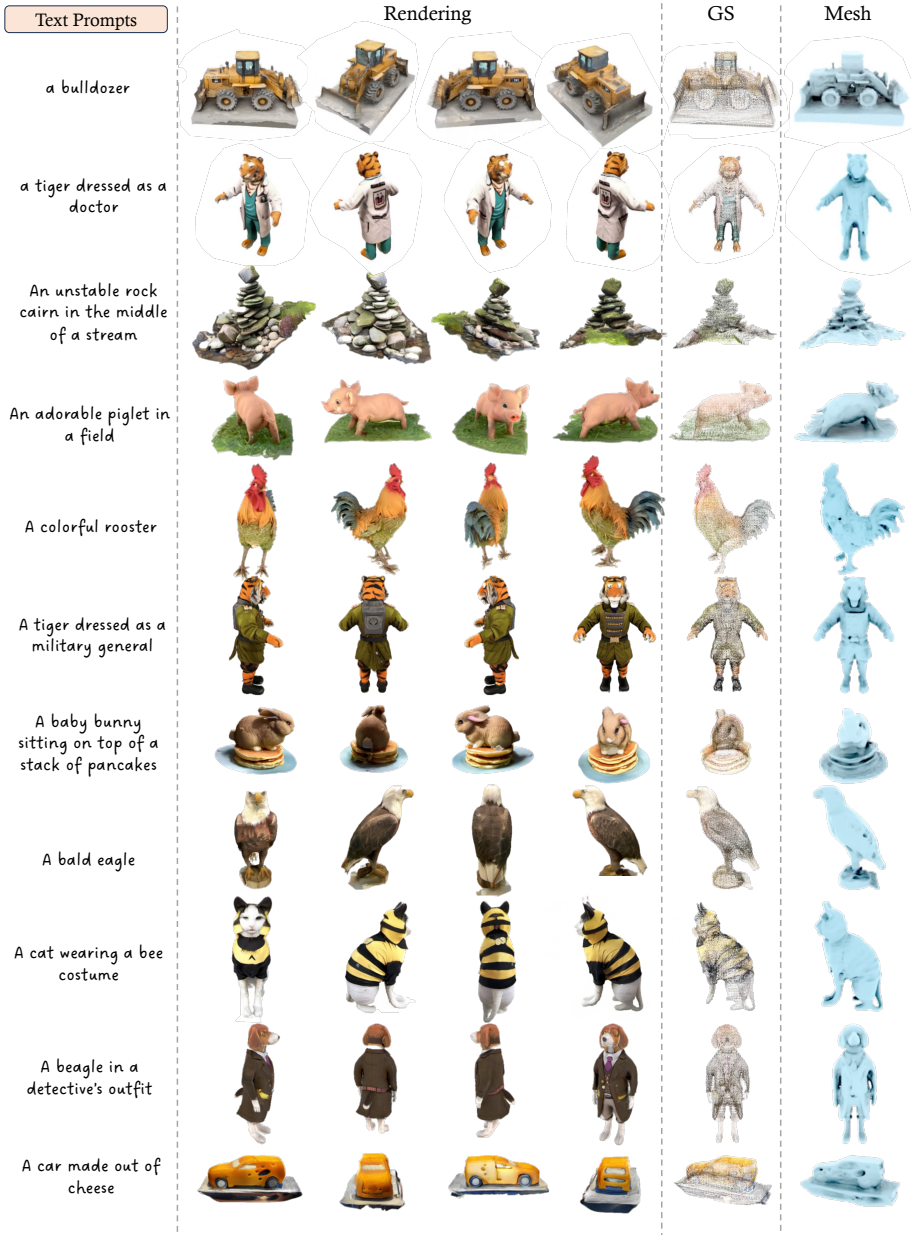

Fig. 3: Text-to-3D Generation.

12. Xu, Y., Tan, H., Luan, F., Bi, S., Wang, P., Li, J., Shi, Z., Sunkavalli, K., Wetzstein, G., Xu, Z., et al.: Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. arXiv preprint arXiv:2311.09217 (2023) 5

Fig. 4: Single-image-to-3D Generation.

13. Zhang, K., Kolkin, N., Bi, S., Luan, F., Xu, Z., Shechtman, E., Snavely, N.: Arf: Artistic radiance fields (2022) 1

14. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: IEEE Conf. Comput. Vis. Pattern Recog. (2018) 2