Supplementary Materials for "GaussianGrow: Generative Gaussian Growing from 3D Point Clouds"

Weiqi Zhang[1*], Junsheng Zhou[1*†], Haotian Geng[1], Kanle Shi[2], Shenkun Xu[2], Yi Fang[3], Yu-Shen Liu[1†]
School of Software, Tsinghua University, Beijing, China[1]
Kuaishou Technology, Beijing, China[2]
CAIR and CIDSAI, NYU Abu Dhabi, UAE[3]
{zwq23, zhou-js24, genght24}@mails.tsinghua.edu.cn
{shikanle, xushenkun}@kuaishou.com yfang@nyu.edu liuyushen@tsinghua.edu.cn

A. Video

We include a supplementary video that showcases our results across a variety of different tasks and applications of our method.

B. Source Codes

The demo code is included as part of our supplementary materials. We will make the source code, data, and implementation instructions publicly available upon acceptance of our paper.

C. More Implementation Details

C.1. Novel View Synthesis

We provide additional details on multi-view image generation process. For each object, we generate information for six cardinal views and four additional views focused on key overlapping regions. At each camera position, our UDF-based pipeline renders normal maps N_i and position maps C_i. For the primary view, we also generate a depth map D_i using ray marching through our UDF field. This depth map, combined with the text prompt c, is processed using Depth-Aware-ControlNet [5] and Stable Diffusion [2] to create a high-quality reference view I_r that serves as an anchor for maintaining consistent appearance across multiple viewpoints.

Hunyuan3D-Paint [3] implements a sophisticated multi-view generation framework through a double-stream image conditioning architecture. To ensure high-fidelity image generation and multi-view consistency, the framework consists of two key components: a dual-stream feature processing pipeline and a multi-task attention mechanism. In the dual-stream pipeline, the reference stream processes input features $Z_r = E(I_r)$ at timestep $t = 0$, where E denotes the VAE encoder, while the generation stream processes noise-conditioned features Z_t at timestep t. These two streams are integrated through a multi-task attention mechanism that combines three parallel modules:

$$\begin{aligned} Z_{MVA} = Z_{SA} + \lambda_{ref} \cdot \text{Softmax}\left(\frac{Q_{ref}K_{ref}^T}{\sqrt{d}}\right)V_{ref} \\ + \lambda_{mv} \cdot \text{Softmax}\left(\frac{Q_{mv}K_{mv}^T}{\sqrt{d}}\right)V_{mv}, \end{aligned} \tag{1}$$

where Z_{SA} represents self-attention features with frozen SD2.1 [2] weights, and the subsequent terms capture reference image guidance and multi-view consistency respectively. To maintain geometric accuracy, the framework incorporates geometric information through encoded normal and position maps $G = E([N_i, C_i])$. These geometric features are combined with view-specific embeddings e_v to form the complete conditioning signal $[G; e_v]$, enabling the generation of geometrically accurate and view-consistent outputs $I = I_{i}{}_{i=1}^{K}$ across all K camera positions.

C.2. Spatial Inpainting

For each unoptimized Gaussian g_i, the process locates its L nearest valid neighbors g_j and computes weighted contributions based on spatial proximity and normal alignment. For each valid neighbor g_j of an unoptimized Gaussian g_i, we define its weight w_j as:

$$w_j = \frac{1/dis_{\{i,j\}}}{\sum_{k=1}^{L} 1/dis_{\{i,k\}}} \cdot (\mathbf{n}_i \cdot \mathbf{n}_j) \cdot \mathbb{I}_{\{(\mathbf{n}_i \cdot \mathbf{n}_j) > 0.5\}}, \tag{2}$$

where $dis_{\{i,j\}}$ is the distance between Gaussians. The distance term prevents far Gaussians with inconsistent appearances from significantly affecting the attributes, while the normal consistency term preserves geometric features by prioritizing propagation between Gaussians with similar

Figure 1. Additional visual comparisons on the Objaverse dataset. GaussianGrow produces more detailed and geometrically consistent results.

surface orientations. The final color c_i for the unoptimized Gaussian is then calculated as:

$$c_i = \frac{\sum_{j=1}^{L}(c_j \cdot w_j)}{\sum_{j=1}^{L} w_j}. \tag{3}$$

The inpainting also considers local spatial density when determining appropriate scale values and adjusts opacity parameters to prevent over-accumulation in densely populated regions. Through this comprehensive spatial inpainting process, GaussianGrow produces complete Gaussian representations with consistent appearance, even in initially challenging-to-observe areas.

D. More Ablation Studies

In Section 4.4 of the main paper, we conducted ablation studies to demonstrate the effectiveness of various components in our framework. To further investigate the robustness of GaussianGrow, we conduct additional ablation experiments on a key hyperparameter: the number of input points N. These results are summarized in Table 1, clearly indicating how different settings of these hyperparameters affect the performance of GaussianGrow.

E. More Results

To complement our quantitative evaluation, we provide additional visual comparisons on the Objaverse dataset [1] in Fig. 1. These examples further illustrate GaussianGrow's capability to generate high-quality appearances across diverse object categories. While Section 4 established our method's performance through metrics, these visualizations highlight the qualitative differences between GaussianGrow and existing approaches. The visual comparison reveals that previous methods often struggle with complex geometries, particularly in regions with intricate details where UV mapping can introduce distortions. GaussianGrow, by directly growing Gaussians from point clouds, preserves fine geometric structures without introducing the artifacts commonly seen in traditional approaches.

Table 1. More Ablation Studies.

Hyper parameters	FID↓	KID↓	CLIP↑
$N = 10^5$	36.60	3.29	26.47
$N = 2.5 \times 10^5$	**36.07**	**3.04**	**27.30**
$N = 3.5 \times 10^5$	36.32	3.17	26.49

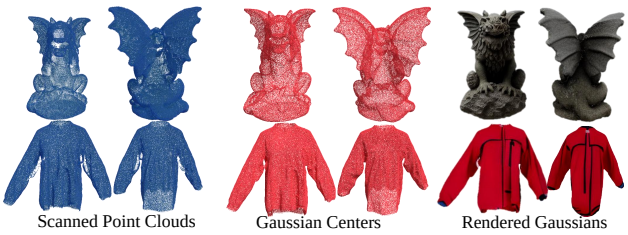

Figure 2. Experiments on the Corrupted Data.

F. Corrupted Data Input

GaussianGrow demonstrates robust capability in generating high-fidelity 3D Gaussians from degraded real-world point clouds through the incorporation of image-level supervision for Gaussian densification. We conduct comprehensive evaluations on two challenging real-world datasets: the Scan-to-Reality Benchmark (SRB) [4] and DeepFashion3D [6], both of which contain point clouds acquired from real-world scanning processes and exhibit substantial noise artifacts and geometric occlusions. As illustrated in Fig. 2, our approach achieves perceptually plausible reconstructions across both experimental settings, effectively mitigating the adverse effects of input data corruption.

References

[1] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A Universe of Annotated 3D Objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023. 2

[2] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1

[3] Tencent Hunyuan3D Team. Hunyuan3D 2.0: Scaling Diffusion Models for High Resolution Textured 3D Assets Generation, 2025. 1

[4] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10130–10139, 2019. 3

[5] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models . In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 1

[6] Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. Deep Fashion3D: A Dataset and Benchmark for 3D Garment Reconstruction from Single Images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 512–530. Springer, 2020. 3