MetroGS: Efficient and Stable Reconstruction of Geometrically Accurate High-Fidelity Large-Scale Scenes

Supplementary Material

A. Implementation Details

For the GauU-Scene [9] dataset, we conducted parallel training with a batch size of 4, targeting a total of 60,000 iterations. Subsequently, we train both the single-view and multi-view geometric refinement stages of $\mathcal{L}_{geo}$ for 30,000 iterations. During this process, λ_d decreases from 0.5 to 0.005 as training progresses, while λ_n is set to 0.0125, λ_s to 0.1, and λ_{mv} to 2.5. For $\mathcal{L}_{app}$, the weight λ is set to 0.8. Densification terminates after the 15,000th iteration, with sparsity compensation parameters set to $S_{\text{th}} = 20$ and $V_{\text{th}} = 10$. The voxel size is set to 0.1 or 0.01 depending on the scale of the scene. For evaluation, only the view embeddings from the training set are available. Since the image filenames encode temporal information, we first use it to identify the two training views that are temporally closest to each test view. We select the candidate with the most similar camera pose to the test view. This nearest-neighbor assignment provides the interpolated view embedding for the test view.

For the MatrixCity [3] dataset, the Aerial and Street scenes were trained for 150,000 and 180,000 iterations, respectively. For $\mathcal{L}_{geo}$, single-view optimization is performed until the 50,000th iteration, followed by the switch to multi-view refinement. Densification is also terminated at the 50,000th iteration. All other training configurations follow those used for the GauU-Scene dataset. For evaluation, test image filenames lack temporal information, so each test view selects its most relevant training view solely based on camera-pose similarity. The corresponding view embedding is then used for image rendering.

For geometric quality evaluation, we follow the parameter settings used in CityGSV2 [5]. Specifically, we render RGB images and depth maps from the training viewpoints and fuse them into a projected truncated signed distance functio (TSDF) volume [10] to extract surface meshes and point clouds. GauU-Scene uses a voxel size of 0.01, an SDF truncation of 0.04, and a depth truncation of 2.0. In MatrixCity, the Aerial split uses 0.01 / 0.04 / 5.0 for voxel size, SDF truncation, and depth truncation, respectively, whereas the Street split adopts 1 / 4 / 500.

B. Hyperparameters of Other Methods

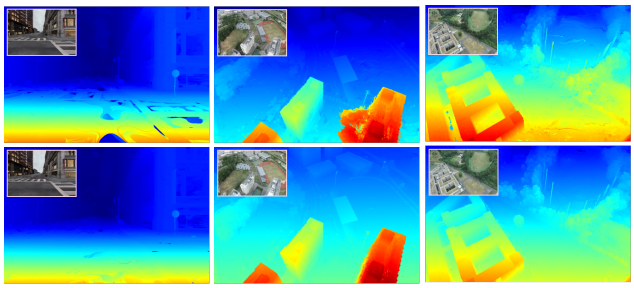
For the visualization results of 2DGS, CityGS, and CityGSV2, we train the models using the default parameter settings provided in the CityGSV2 codebase, and for CityGSV2, we use the provided checkpoints. For the comparison with CityGS-$\mathcal{X}$, we utilized its provided Mill19 configuration to train the GauU-Scene dataset. Crucially, we disabled the progressive LOD (Level of Detail) training within this configuration to ensure better preservation

Table 1. **Efficiency performance comparison on the GauU-Scene [9] dataset.** Entries marked with an asterisk (*) represent the intermediate results obtained after 30,000 training iterations.

Scene	Method	PSNR ↑	F1 ↑	#G(M)	T(min)
Russian	V2-coarse*	23.46	0.509	**7.98**	110
	Ours*	**24.60**	**0.559**	8.20	**50**
	CityGSV2	24.12	0.542	**7.77**	363
	Ours	**24.94**	**0.585**	8.20	**106**
Residence	V2-coarse*	22.09	0.437	**9.29**	103
	Ours*	**23.96**	**0.470**	11.33	**78**
	CityGSV2	23.55	0.466	**8.08**	311
	Ours	**24.51**	**0.494**	11.33	**156**
Morden	V2-coarse*	25.08	0.479	**7.61**	98
	Ours*	**26.68**	**0.508**	9.27	**70**
	CityGSV2	25.79	0.492	**7.89**	332
	Ours	**27.07**	**0.524**	9.27	**149**

Table 2. **Efficiency performance comparison on MatrixCity-Aerial [3].** In CityGS-$\mathcal{X}$, which uses an anchor-based Gaussian representation, "×10" denotes the Gaussians derived per anchor.

Scene	Method	PSNR ↑	F1 ↑	#G(M)	T(min)
MC-Aerial	CityGS-$\mathcal{X}$	**27.53**	0.582	2.48×10	716
	Ours	27.52	**0.677**	17.09	**415**



(a) w/ Ini. (b) w/ Geo. (c) w/ App.

Figure 1. **Supplementary Visualization of ablation study results.** The top row shows results without the modules, and the bottom row shows results with them. Our components yield a significant improvement in depth quality, effectively addressing challenges across diverse and complex scenes.

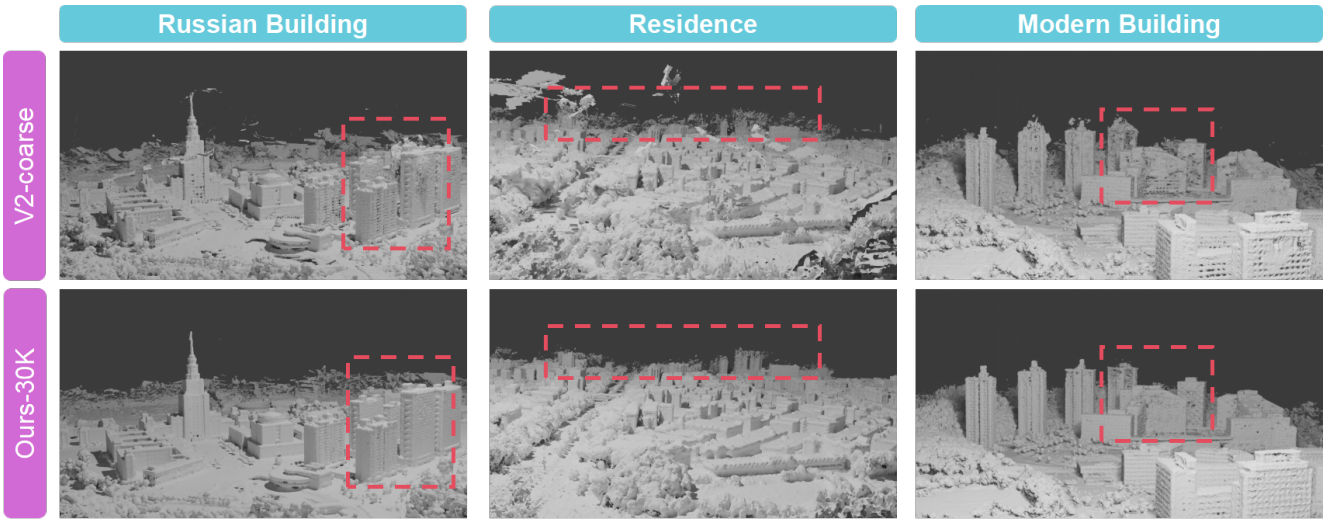


Figure 2. **Qualitative comparison of meshes on the GauU-Scene [9] dataset.** Our method achieves higher-quality results.

Figure 3. **Mesh visualization comparison on MatrixCity-Aerial [9].** Our method provides better results than the baselines.

of scene details. For the MatrixCity dataset, we directly applied the corresponding official configuration provided by CityGS-$\mathcal{X}$ for training.

C. Additional Results

C.1. Training Efficiency Analysis

Using a system with four RTX 3090 GPUs, we conducted a training efficiency comparison between CityGSV2 and CityGS-$\mathcal{X}$ on the GauU-Scene and MatrixCity-Aerial datasets, respectively. As shown in Tab. 1, our method consistently outperforms CityGSV2 in both rendering quality and geometric fidelity, while also demonstrating a significant improvement in training efficiency. Notably, even the intermediate results of our model at 30k iterations already surpass the final performance of CityGSV2, while requiring less than 25% of its training time. Across the GauU-Scene dataset, our final model achieves an average 2.55× training speedup relative to CityGSV2. Tab. 2 presents a comparison of training efficiency between CityGS-$\mathcal{X}$ and our method on the MatrixCity-Aerial dataset. Our approach achieves superior geometric fidelity (F1: 0.677 vs. 0.582) with a 1.7× reduction in training time, while maintaining comparable PSNR performance. Overall, these results highlight the remarkable speed and efficiency of our method. It is worth noting that CityGSV2 and CityGS-$\mathcal{X}$ adopt model-size reduction strategies such as trimming [1] and anchor-based Gaussian compression [7]. Enhancing model-size compactness therefore remains a promising direction for further improving the efficiency of our method.

C.2. Additional Qualitative Comparison

Fig. 1 presents further visualization results for the ablation study. Our adopted pointmap assisted initialization effectively supplements sparse point cloud regions, thereby laying a solid geometric foundation for subsequent reconstruction. Progressive hybrid geometric refinement and depth-guided appearance modeling then collaboratively ensure the final geometric quality exhibits high accuracy and completeness.

In addition, we include more comprehensive qualitative comparisons with the baseline methods. Fig. 2 presents the mesh reconstruction visualization comparison on the GauU-Scene dataset. Given the relatively small size of the im-

Table 3. **Quantitative results on the Mill19 [8] dataset and UrbanScene3D [4] dataset.** The **best** and second best results are highlighted. All missing results are denoted by a "–".

Methods	Building			Rubble			Residence			Sci-Art		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeuS	18.01	0.463	0.611	20.46	0.480	0.618	17.85	0.503	0.533	18.62	0.633	0.472
Neuralangelo	17.89	0.582	0.322	20.18	0.625	0.314	18.03	0.644	0.263	19.10	0.769	0.231
SuGaR	17.76	0.507	0.455	20.69	0.577	0.453	18.74	0.603	0.406	18.60	0.698	0.349
PGSR	16.12	0.480	0.573	23.09	0.728	0.334	20.57	0.746	0.289	19.72	0.799	0.275
PGSR+VastGS	21.63	0.720	0.300	25.32	0.768	0.274	–	–	–	–	–	–
CityGS	21.55	0.778	0.246	25.77	0.813	0.228	22.00	0.813	0.211	21.39	0.837	0.230
CityGS-$\mathcal{X}$	22.76	**0.817**	0.191	26.15	0.823	0.210	22.44	0.819	0.194	22.77	0.867	0.179
CityGSV2	19.07	0.650	0.397	23.75	0.720	0.322	21.15	0.769	0.234	20.66	0.810	0.266
Ours	**23.06**	0.787	**0.173**	**27.48**	**0.826**	**0.147**	**23.38**	**0.824**	**0.166**	**25.96**	**0.872**	**0.152**

age data, we conducted an equivalent comparison in terms of training time: we trained our method for 30,000 iterations and compared its results with those of CityGSV2-coarse. The reconstructed meshes from our method are much cleaner, containing minimal spurious artifacts or floating mesh fragments. Fig 3 further presents a comparison of our method's results against CityGSV2 and CityGS-$\mathcal{X}$ on the MatrixCity-Aerial dataset. The results indicate that our approach achieves a better balance between geometric accuracy and completeness.

C.3. Additional Dataset Evaluation

We have also conducted supplementary evaluations on the Mill19 [8] and UrbanScene3D [4] datasets, which are widely used for assessing rendering quality in the field of large-scale scene reconstruction. Four scenes were selected: Building, Rubble, Residence, and Sci-Art. The configuration uses 100,000 training iterations, with 50,000 iterations allocated to each of the two geometric optimization stages. The densification process is terminated at the 30,000th iteration. The weight λ_s set to 0.001. The remaining settings follow those used for GauU-Scene, as detailed in Sec. A.

Quantitative results are presented in Tab. 3, where we compare against other state-of-the-art surface reconstruction methods. Our method achieves state-of-the-art performance among surface reconstruction approaches in terms of PSNR and LPIPS, and ranks first in SSIM for most scenes. In addition, Fig. 4 provides a qualitative comparison among our methodd and CityGS (Public Checkpoints), showing that our approach performs better under challenging illumination conditions and renders fine-grained details more faithfully. Overall, our method achieves superior visual quality and robustness.

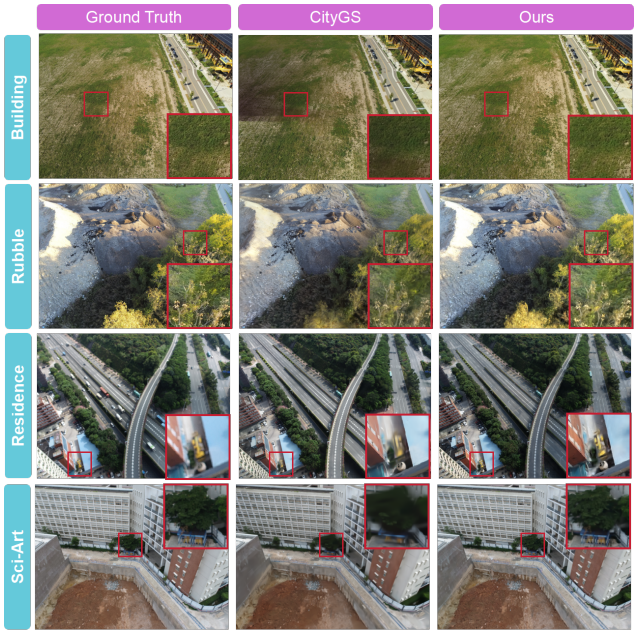


Figure 4. **Qualitative results on Mill-19 [8] and UrbanScene3D [4] datasets**. We compare against CityGS.

D. Discussion

While our method successfully delivers efficient training, accurate geometry, and high rendering quality for large-scale scene reconstruction, it still presents the following limitations: Firstly, due to hardware constraints, memory consumption remains the primary bottleneck limiting the training scale, which to some extent weakens the model's potential performance. Therefore, it is necessary to introduce techniques such as advanced pruning [6] and cache management [11] to mitigate memory challenges. Additionally, our method is based on 2DGS. Although it achieves excellent geometric reconstruction, its upper bound for rendering quality may still lag behind 3DGS. To address this, future work could consider introducing a new geometry representation similar to [2] for complete decoupling of geometry and appearance to further realize improved geometric accuracy and rendering performance.

References

[1] Lue Fan, Yuxue Yang, Minxing Li, Hongsheng Li, and Zhaoxiang Zhang. Trim 3d gaussian splatting for accurate geometry representation. *arXiv preprint arXiv:2406.07499*, 2024. 2

[2] Changjian Jiang, Kerui Ren, Linning Xu, Jiong Chen, Jiangmiao Pang, Yu Zhang, Bo Dai, and Mulin Yu. Halogs: Loose coupling of compact geometry and gaussian splats for 3d scenes. *arXiv preprint arXiv:2505.20267*, 2025. 3

[3] Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3205–3215, 2023. 1

[4] Liqiang Lin, Yilin Liu, Yue Hu, Xingguang Yan, Ke Xie, and Hui Huang. Capturing, reconstructing, and simulating: the urbanscene3d dataset. In *European Conference on Computer Vision*, pages 93–109. Springer, 2022. 3

[5] Yang Liu, Chuanchen Luo, Zhongkai Mao, Junran Peng, and Zhaoxiang Zhang. Citygaussianv2: Efficient and geometrically accurate reconstruction for large-scale scenes. *arXiv preprint arXiv:2411.00771*, 2024. 1

[6] Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 3

[7] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *arXiv preprint arXiv:2403.17898*, 2024. 2

[8] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12922–12931, 2022. 3

[9] Butian Xiong, Nanjun Zheng, Junhua Liu, and Zhen Li. Gauu-scene v2: Assessing the reliability of image-based metrics with expansive lidar image dataset using 3dgs and nerf. *arXiv preprint arXiv:2404.04880*, 2024. 1, 2

[10] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1802–1811, 2017. 1

[11] Hexu Zhao, Xiwen Min, Xiaoteng Liu, Moonjun Gong, Yiming Li, Ang Li, Saining Xie, Jinyang Li, and Aurojit Panda. Clm: Removing the gpu memory barrier for 3d gaussian splatting. *arXiv preprint arXiv:2511.04951*, 2025. 3