CityGS-$\mathcal{X}$: A Scalable Architecture for Efficient and Geometrically Accurate Large-Scale Scene Reconstruction

Supplementary Material

Implementation Details

Given that Mill-19 [8] and UrbanScene3D [6] consist of thousands of high-resolution images, we adhere to the methodology outlined in previous works [8] by downsampling the images by a factor of 4 for both training and validation. For evaluation, we adopt the configuration from Momentum-GS [2], which excludes color correction when computing the metrics. Regarding the depth-prior filter, we set the threshold τ_d to 1. In the training process for the Rubble, Building, Residence, and Sci-Art datasets, we define the total number of training steps as 100,000. The anchor growing process is maintained until the 50,000th step. The Step 2 Depth-Prior loss is introduced at the 10,000th iteration, with its weight progressively decreasing from 1 to 0 as training advances. The Step 3 Batch-Level Geometric Training is initiated at the 30,000th iteration, during which the weight of this loss incrementally rises from 0 to 0.2 throughout the training duration.

Given that MatrixCity [4] comprises over 5000 images, we conducted training for 150,000 iterations. Following the approach of CityGS-V2 [7], we downsampled the images by a factor of 1.2 for training purposes. Owing to the scene's relatively simple geometric structure, the monocular depth estimator method performs with greater accuracy in this context. Consequently, we introduced the Enhanced Depth-Prior Training (Step 2) at the 10,000th iteration and implemented the Batch-Level Geometric Training (Step 3) at the 100,000th iteration.

For mesh reconstruction, we render both visual images and depth maps from multiple viewpoints. These rendered outputs are subsequently fused into a projected truncated signed distance function (TSDF) volume [9], ultimately generating high-quality 3D surface meshes and point clouds. We set the voxel size of 0.01 and SDF truncation of 0.04 in MatrixCity, Residence, and Sci-Art, while 0.001, 0.004 in Rubble and Building datasets.

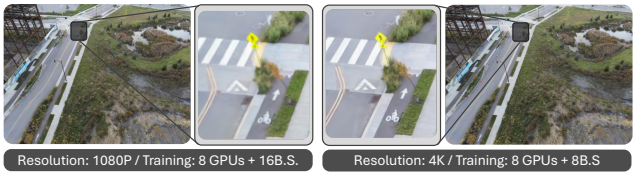

Figure 1. Visual comparison between 1080P and 4K training with our method.

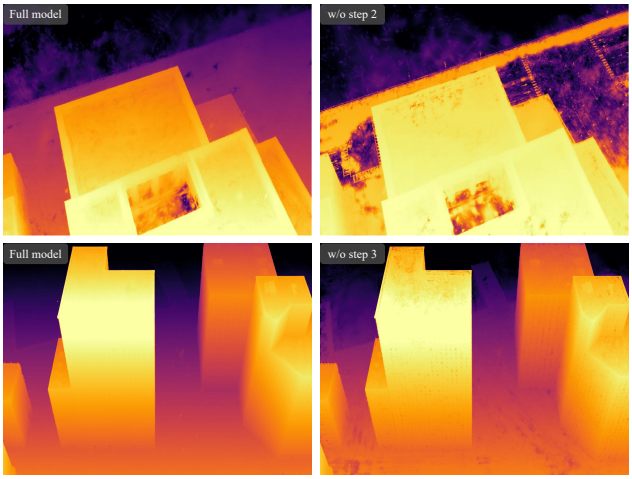

Figure 2. Training ablations for MatrixCity.

4K Training and Rendering.

To demonstrate that our method maintains high-quality reconstruction even at higher resolutions, we compare training results between 1080P and 4K settings, as shown in Fig. 1. The 1080P model is trained with 8 GPUs and a batch size of 16, while the 4K model uses the same number of GPUs but a batch size of 8 due to increased memory consumption. The qualitative comparison shows that our method also achieves consistently high reconstruction quality in 4K settings. The 4K model preserves fine-grained details, such as sharp edges, clear road markings, and accurate object boundaries, without introducing artifacts or degradation. This indicates that our approach effectively scales to higher resolutions, ensuring robustness and reliability in large-scale urban scene reconstruction.

Overall Mesh Visualization

In Fig. 3 and Fig. 4, we visualize textured meshes and meshes of our method and CityGSV2 [7] on Sci-Art, Residence, Rubble, Building. Our mesh exhibits superior geometric structure, demonstrating enhanced accuracy and detail compared to CityGSV2. In the Sci-Art, our mesh has fewer floaters, while at the bottom of the Residence and Rubble we have fewer holes. In the Building datasets, for the central building, we preserve its texture while maintaining accurate structural integrity, a capability that CityGSv2 fails to achieve.

Table 1. **Training Resources Consumption on Mill19 [8] dataset and UrbanScene3D [6] dataset**. We present the allocated memory (GB) during evaluation. For 3DGS-based methods.

Models	Building		Rubble		Residence		Sci-Art	
	Time ↓	Mem ↓	Time ↓	Mem ↓	Time ↓	Mem ↓	Time ↓	Mem ↓
Mega-NeRF [8]	19:49	5.84	30:48	5.88	27:20	5.99	27:39	5.97
Switch-NeRF [10]	24:46	5.84	38:30	5.87	35:11	5.94	34:34	5.92
3DGS [3]	21:37	4.62	18:40	2.18	23:13	3.23	21:33	1.61
VastGS† [5]	03:26	3.07	02:30	2.74	03:12	3.67	02:33	3.54
DOGS [1]	03:51	3.39	02:25	2.54	04:33	6.11	04:23	3.53
CityGS-$\mathcal{X}$	**03:00**	**2.00**	**02:15**	**2.29**	**02:40**	**2.61**	**03:30**	**1.40**

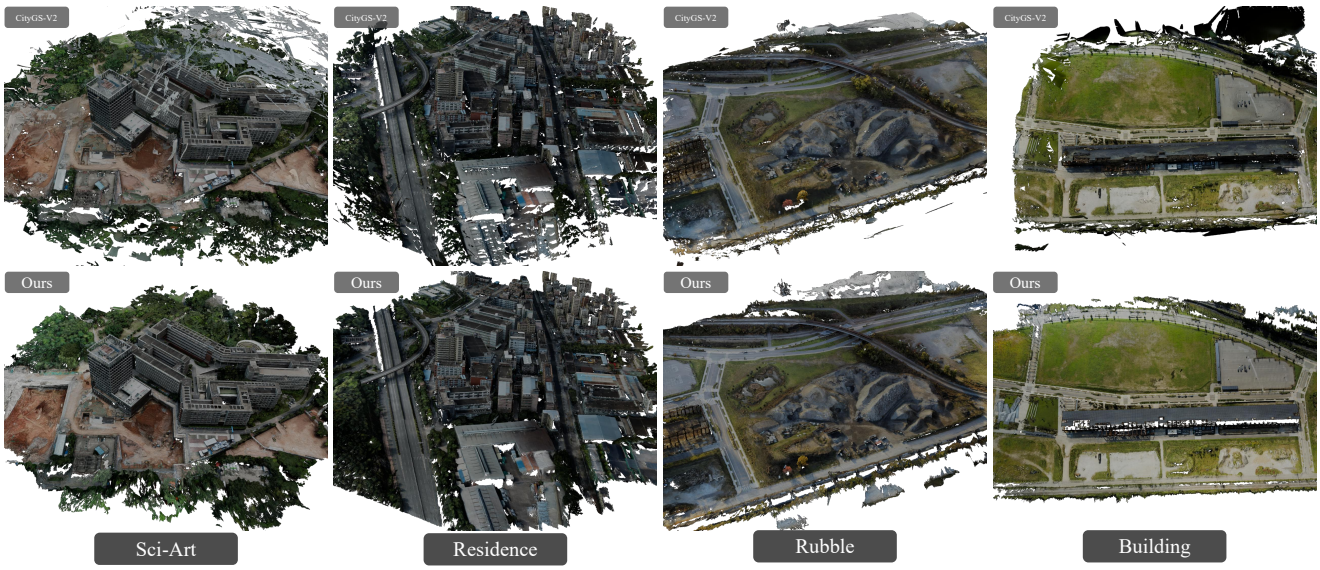

Figure 3. Qualitative comparison between CityGS-V2 [7] and ours in generated textured meshes.

Inference Speed

In Tab. 2, we test the inference performance of our method on the MatrixCity dataset. Our framework achieves 40 FPS on a single GPU at 1920×1080 resolution. When scaling to two and four GPUs, the frame rate slightly decreases to 36 FPS and 29 FPS, respectively, due to inter-GPU communication overhead.

Table 2. **Inference speed on MatrixCity.**

B.S. / GPU	1 / 1	2 / 2	4 / 4	Resolution	Datasets
FPS	40	36	29	1920×1080	MatrixCity

Training Strategy Ablation on MatrixCity

In Fig. 2, in the first row we directly add the Batch-Level Geometric Training (Step 3) without the transition of Step 2, and the large plane of the scene on the ground is hard to converge. In the second row, without Step 3, some geometric details on the ground are not accurate. Therefore, both Step 2 and Step 3 are necessary for optimizing geometry.

Training Resources Consumption

Tab. 1 presents the training resource consumption of various methods on the Mill19 and UrbanScene3D datasets. It reports the training time (Time ↓) and memory usage in GB (Mem ↓) across four different scenes: Building, Rubble, Residence, and Sci-Art.

Among all methods, CityGS-X achieves the fastest training time and lowest memory consumption across all scenes. Specifically, it requires only 3:00 minutes and 2.00 GB for the Building scene, 2:15 minutes and 2.29 GB for Rubble, 2:40 minutes and 2.61 GB for Residence, and 3:30 minutes and 1.40 GB for Sci-Art. In contrast, Mega-NeRF and Switch-NeRF take significantly longer training times (e.g., up to 38:30 minutes for Rubble) and higher memory usage (5.9 GB). 3DGS, VastGS, and DOGS show moderate resource consumption, but CityGS-X consistently outperforms them in both speed and efficiency. These results

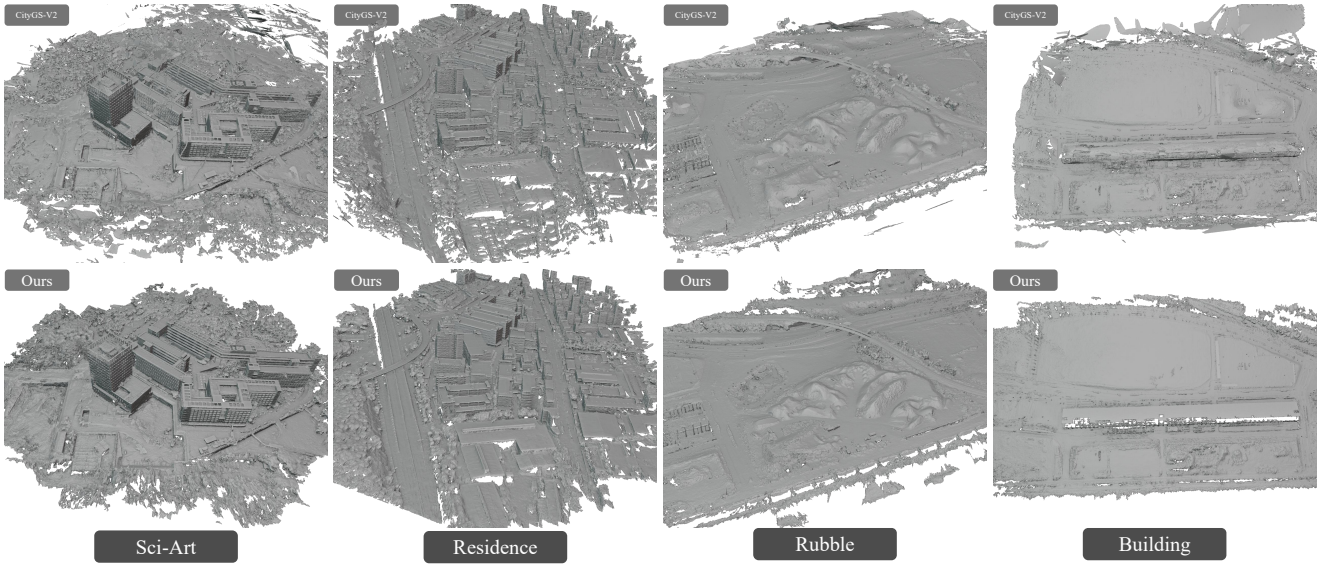

Figure 4. Qualitative comparison between CityGS-V2 [7] and ours in generated meshes.

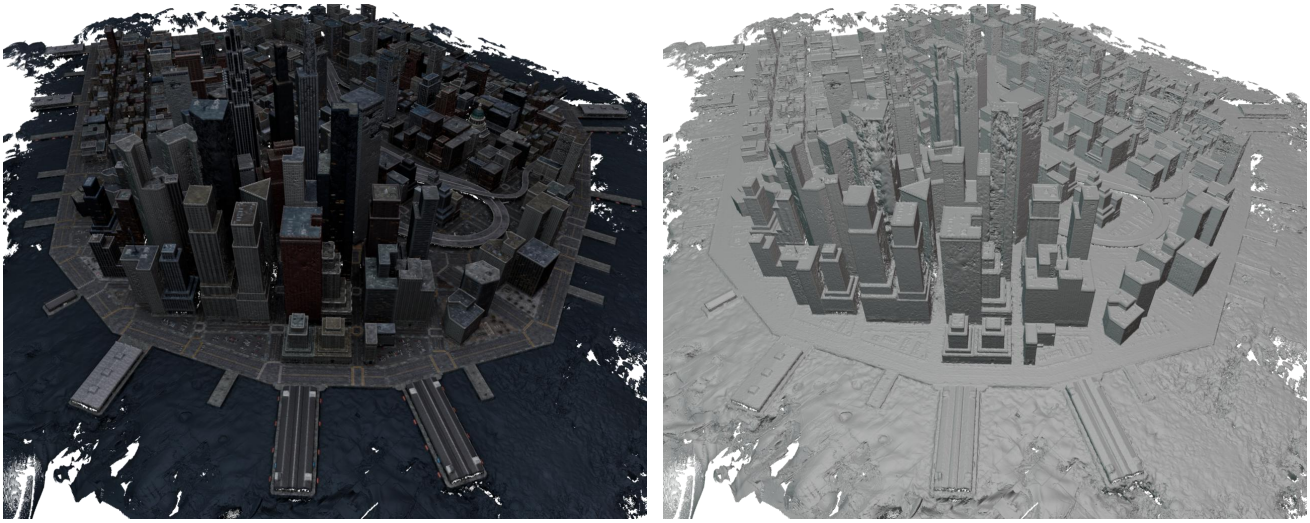

Figure 5. Qualitative results of our generated meshes on MatrixCity [4] dataset.

highlight the superior efficiency of CityGS-X, making it a more practical choice for large-scale scene reconstruction with limited computational resources.

References

[1] Yu Chen and Gim Hee Lee. Dogaussian: Distributed-oriented gaussian splatting for large-scale 3d reconstruction via gaussian consensus. *arXiv preprint arXiv:2405.13943*, 2024.

[2] Jixuan Fan, Wanhua Li, Yifei Han, and Yansong Tang. Momentum-gs: Momentum gaussian self-distillation for high-quality large scene reconstruction, 2024.

[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023.

[4] 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.

[5] Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, et al. Vastgaussian: Vast 3d gaussians for large scene reconstruction. *arXiv preprint arXiv:2402.17427*, 2024.

[6] 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.

[7] 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.

[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.

[9] 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.

[10] MI Zhenxing and Dan Xu. Switch-nerf: Learning scene decomposition with mixture of experts for large-scale neural radiance fields. In *The Eleventh International Conference on Learning Representations*, 2022.