DrivingGaussian: Composite Gaussian Splatting for Surrounding Dynamic Autonomous Driving Scenes

Supplementary Material

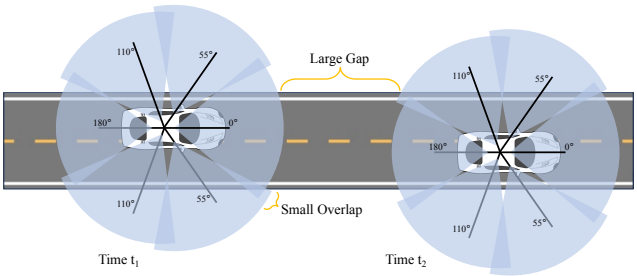

Figure 1. **Visualization of surrounding multi-camera views in nuScenes dataset.** The surrounding views have small overlaps among multi-camera but large gaps across time.

1. Implementation Details

Experimental Details. We report the average results of all camera frames on the selected scenes and assess our models using the average score of PSNR, SSIM, and LPIPS. For the nuScenes [3] dataset, images of full-resolution 1600×900 are rendered with 360-degree horizontal FOV per time-step. We use synchronized images from 6 cameras in surrounding views as inputs. We randomly select every 5th image of different cameras in the sequences as the test set and utilize the remaining images as the training set. For the KITTI-360 [7] dataset, we only use sequential images from a single camera as input, with a resolution of 1408×376. We select every 10th image of cameras in the sequences as the test set.

Details of LiDAR Prior. The LiDAR prior provides more precise and complete initialization oversight for scene modeling, helping to recover the more correct and detailed shape of the scene. Here, we present detailed preprocessing and techniques for using the LiDAR prior.

LiDAR points derived from the dataset are categorized into dynamic foreground and static background. Dynamic foreground can cause misalignment during LiDAR-image registration due to drag, aliasing, etc. So, we first cut out dynamic objects from the LiDAR points based on the segmentation labels, obtaining purely static LiDAR prior to the scenes. We then use multi-frame aggregation to stitch together the LiDAR of the scene according to the currently visible regions of the Incremental Static 3D Gaussians. The coordinates of LiDAR prior are further transformed into the global coordinate system via calibration matrices.

Intuitively, while capturing images with moving platforms, nearby areas will have more pixels to represent finer details. In contrast, distant regions are described using a limited number of coarse points. This principle similarly applies to the 3D Gaussian representation for large-scale driving scenes. In this regard, we utilize an adaptive filtering algorithm to optimize the LiDAR prior. The previously obtained LiDAR point cloud is voxelized into a fixed-size voxel grid. We divide the voxel grid along the rays extending forward from the camera center based on depth. We next apply distance weighting and remove isolated outliers for the points within the voxel grids representing distant views.

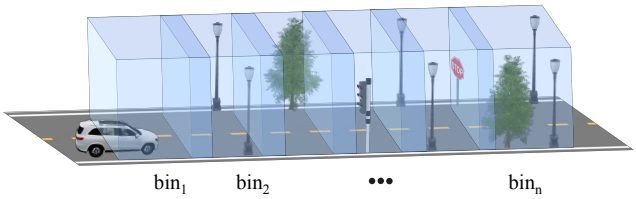

Figure 2. **Visualization of bins arrangement for the Incremental Static 3D Gaussians.** The small overlap between two neighboring bins is used to align the static backgrounds of the two bins.

Details of surrounding multi-camera views. As shown in Figure 1, we show the distribution of the surrounding multi-camera views in driving scenes [3]. We can observe that these surrounding views have only minimal overlap between multi-camera but have large intervals across adjacent frames. Compared with typical NeRF-based captures (e.g., central objects captured by hemisphere views), the surrounding multi-camera views pose a great challenge to modeling the whole scene from such sparse observations.

Details of bins arrangement for static background. We show the arrangement of sequential bins in the Incremental Static 3D Gaussians module. As shown in Figure 2, each bin is distributed according to the scene's depth and contains one or more frames of surrounding images. Neighboring bins have a small overlap region, which is used to align the static backgrounds of two bins. The latter bin is then incrementally fused into the Gaussian field of the previous bins. During the incremental addition of bins, camera poses and LiDAR point positions are utilized to align and merge different bins in a global coordinate, thereby preventing disruption to the previous reconstruction. In addition, we allow the distribution of bins to be specified manually, enabling better adaptation to extreme or depth-unknown scenarios.

Details of sky areas. We treat the sky as an infinitely distant part of the static background in driving scenes. In our work, we follow [16, 18, 21] to explicitly handle the sky using predicted sky masks. The sky masks help in addressing the ill-defined depth of the sky and can be obtained from

Table 1. **Overall perforamnce of DrivingGaussian with existing state-of-the-art approaches on the nuScenes dataset.** Ours-S denotes the DrivingGaussian with SfM points initialization, and Ours-L denotes training the Gaussian model with LiDAR prior. Rendering Time denotes the rendering time for each frame.

Methods	Input	PSNR ↑	SSIM ↑	LPIPS ↓	Rendering Time(s)
Instant-NGP [10]	Images	16.78	0.519	0.570	4.382
NeRF+Time	Images	17.54	0.565	0.532	31.14
Mip-NeRF [1]	Images	18.08	0.572	0.551	24.55
Mip-NeRF360 [2]	Images	22.61	0.688	0.395	11.86
NSG [11]	Images	21.67	0.671	0.424	52.28
Urban-NeRF [12]	Images + LiDAR	20.75	0.627	0.480	41.29
S-NeRF [19]	Images + LiDAR	25.43	0.730	0.302	23.67
SUDS [14]	Images + LiDAR	21.26	0.603	0.466	45.7
EmerNeRF [21]	Images + LiDAR	26.75	0.760	0.311	21.91
3DGS [4]	Images + SfM Points	26.08	0.717	0.298	0.864
4DGS [17]	Images + SfM Points	19.79	0.622	0.473	2.160
Ours-S	Images + SfM Points	28.36	0.851	0.256	0.965
Ours-L	Images + LiDAR	28.74	0.865	0.237	0.963

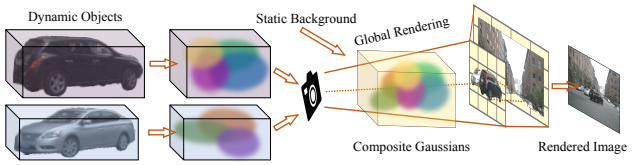

Figure 3. **Visualization of Global Rendering via GS.** DrivingGaussian ensures the reconstruction of multiple dynamic objects along with their accurate positions and occlusion relationships.

semantic segmentation models [5, 13]. We further allocate Gaussian initialization points separately for the segmented sky region and optimize them within the sky mask area.

Moving Objects in Driving Scenes. Dynamic objects are foreground instances that are moving in the current scene, while parked vehicles or static objects are not. We provide two methods of decoupling dynamic objects for our approach, using either a 3D bounding box or a pre-trained dynamic object segmentation model (e.g., SAM [5], Grounded SAM [13], SEEM [22], or OmniMotion [15]).

Using the 3D bounding box, we project the bounding box of each object individually onto 2D images of the surrounding view and mask the objects inside the box. We explicitly align the dynamic objects in each frame with the ID of each object from the label.

Similarly, when using pre-trained dynamic object segmentation models, we separate dynamic objects from static areas by applying pre-trained models and explicitly labeling each object individually with the object ID. Experiments also show that it is unnecessary to precisely segment every pixel while excluding background pixels, as our method is robust to dynamic objects containing background pixels. These imperfect background pixels are eliminated in the optimization of modeling dynamic objects in the scene.

Details of Global Rendering via GS. Global rendering via GS aims to restore the position relationship and occlusion of multiple dynamic objects with static backgrounds in the real driving scene. We transform the Gaussians of each node in the dynamic Gaussian graph to the world coordinate system. We then utilize the fast splatting algorithm introduced by the 3DGS [4] to support the global rendering. As shown in Figure 3, our method enables the re-rendering of multiple objects and static backgrounds in a shared driving scene. Based on the explicit geometry scene structure of Gaussian distribution, the global rendering preserves the original occlusion relationships and exact spatial positions.

2. Additional Results on nuScenes

Quantitative Comparison. We provide more comparisons of results with recent works on large-scale driving scenes and 3D Gaussian-based approaches. For a fair comparison, we also migrate the graph-based method NSG [11] and dynamic Gaussians method 4DGS [17] to the nuScenes dataset. As shown in Table 1, our method boosts the performance of NSG across three metrics. Although NSG similarly uses a graph-based representation for dynamic objects, it only applies to the front-forward monocular views and does not cope well with the dynamic objects under ego vehicle movements. Our method also shows a huge lead compared to the latest work designed for dynamic 3D Gaussians [17]. Since 4DGS [17] employs Gaussians updated over time steps to represent dynamic objects, it only works for slow-moving central objects and fails in complex scenes with multiple high-speed moving foregrounds.

Rendering Speed. Table 1 shows that our method achieves a good balance between rendering quality and rendering speed. Compared to the accelerated NeRF method Instant-NGP [10], our approach achieves higher results with faster rendering speed. Our method achieves the optimal quality with less rendering time compared to those of NeRF-based methods designed for unbounded scenes(e.g., Mip-NeRF [1], Mip-NeRF360 [2], Urban-NeRF [12]). Compared to methods [14, 19, 21], also designed for dynamic driving scenes, our approach obtains better performance and reduction in rendering time. Compared to our baseline

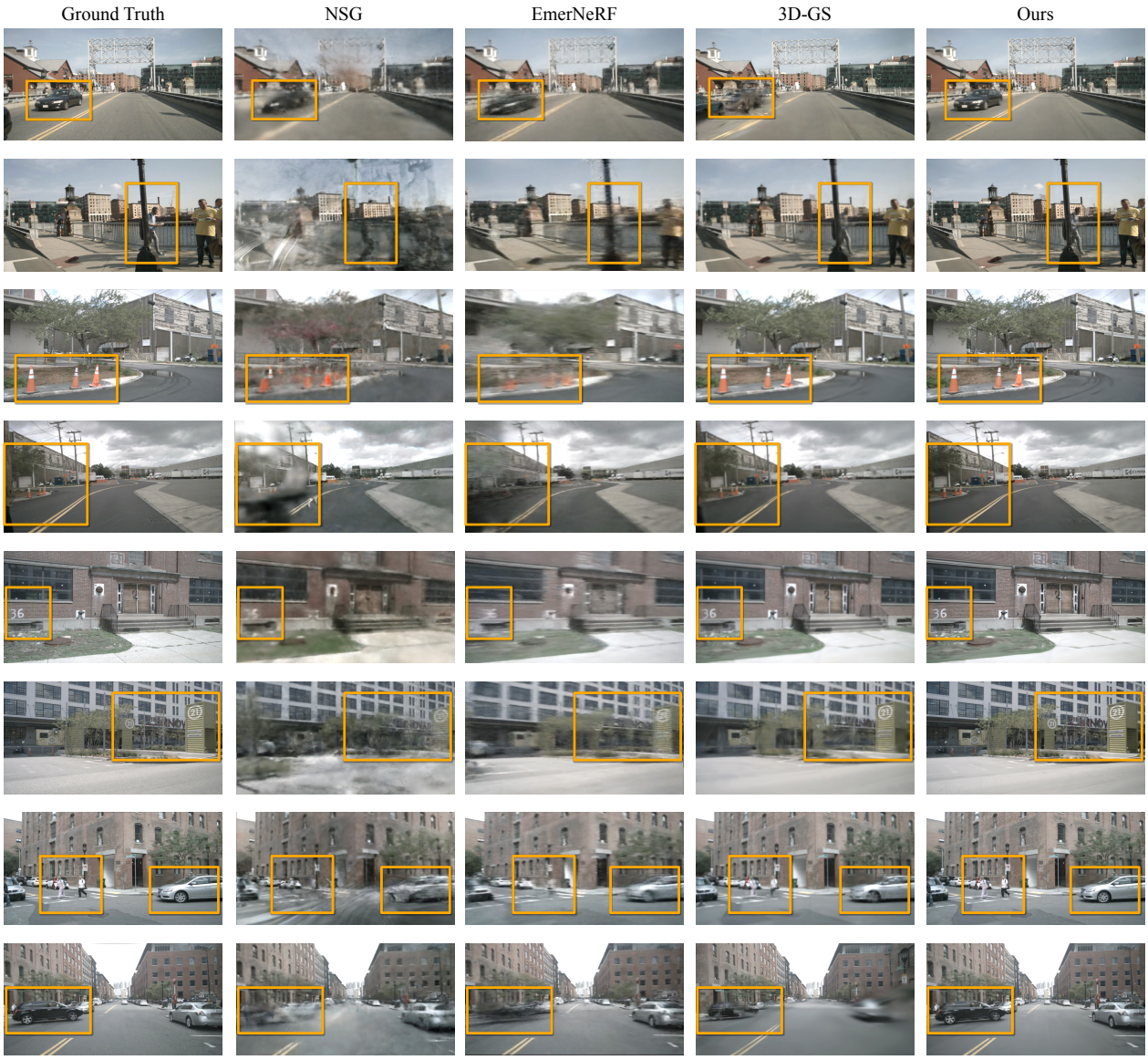

Figure 4. **Qualitative comparison on the nuScenes dataset.** We demonstrate the qualitative comparison results with our main competitors NSG [11], EmerNeRF [21] and 3DGS [4] on driving scenes reconstruction of nuScenes.

method 3DGS [4], our method achieves higher rendering quality with comparable rendering speed.

Qualitative Comparison. We further show more qualitative results compared with the SOTA methods on the nuScenes dataset. As shown in Figure 4, our method surpasses existing works in modeling both static backgrounds and dynamic objects in driving scenes. Please refer to the project page for video results and additional comparisons.

3. Additional Results on KITTI-360

We further evaluate the performance of DrivingGaussian for monocular driving scenes on the KITTI360 dataset. We compare our method with the latest SOTA approaches trained on the KITTI360, including NeRF-based PNF [6] and 3D Gaussian-based 3DGS [4]. As shown in Table 2, our method achieves better performance than all other methods on the leaderboard.

As shown in Figure 5, we show qualitative results compared with our main competitors on the KITTI-360 dataset. DNMP [8] is a NeRF-based method designed for monocular driving scenes with deformable neural mesh and LiDAR prior. Our approach shows more realistic reconstruction results and fine geometry on challenging areas such as traffic signs, vehicles, people, etc. We also find that our baseline method, 3DGS [4], fails in modeling the detail areas, producing unpleasant artifacts, blurring, and unnatural colors.

Figure 5. **Qualitative comparison on the KITTI-360 dataset.** We demonstrate the qualitative comparison results with our main competitors DNMP [8] and 3DGS [4] on driving scenes reconstruction of KITTI-360.

Table 2. **Overall perforamcne of DrivingGaussian with existing state-of-the-art approaches on the KITTI-360 dataset.** We only use sequential images from a single camera as input for modeling driving scenes in the KITTI-360.

Methods	PSNR ↑	SSIM ↑
NeRF [9]	21.94	0.781
Point-NeRF [20]	21.54	0.793
NSG [11]	22.89	0.836
Mip-NeRF360 [2]	23.27	0.836
PNF [6]	23.06	0.839
SUDS [14]	23.30	0.844
DNMP [8]	23.41	0.846
3DGS [4]	22.93	0.847
Ours-S	25.18	0.862
Ours-L	25.62	0.868

In contrast, although our method is not specifically designed for monocular scenarios, it still shows good adaptability and robustness in representing monocular driving scenes with detail areas and outperforms existing SOTA approaches.

4. Additional Ablation Study and Analysis

The quantitative ablation results are presented in the main text. Furthermore, we provide additional qualitative ablation comparisons to demonstrate the effectiveness of each module in our method.

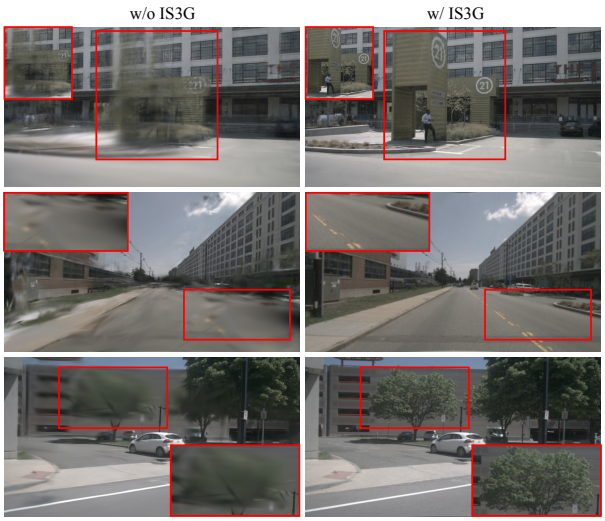

Figure 6. **Rendering with or w/o the Incremental Static 3D Gaussians (IS3G).** IS3G ensures good geometry and topological integrity for static backgrounds in large-scale driving scenes.

Density of Bins. We explore the effect of different densities of bins for reconstructing the driving scenes in Incremental Static 3D Gaussians. Here, we chose a part of the scene close to a straight line (horizontal length of about 400 me-

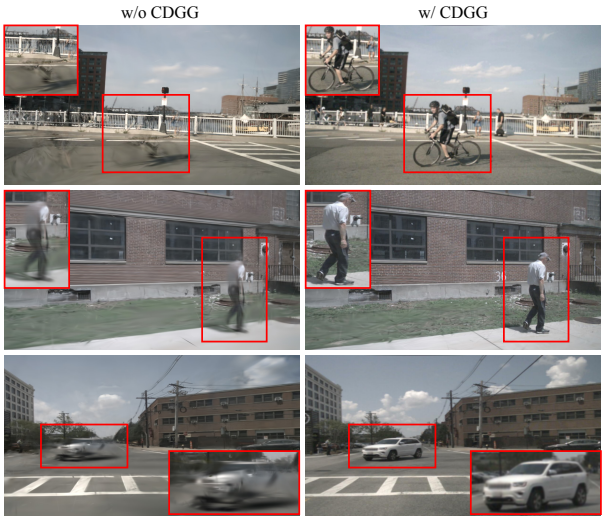

Figure 7. **Rendering with or w/o the Composite Dynamic Gaussian Graph (CDGG).** CDGG enables the reconstruction of dynamic objects at arbitrary speeds in the driving scenes (e.g., vehicles, bicycles, and pedestrians).

Table 3. **Effect of density of bins on the Composite Gaussian model.** N denotes for number of bins in a certain driving scene.

Bins	PSNR ↑	SSIM ↑	LPIPS ↓
N=3	27.94	0.849	0.256
N=4	28.38	0.857	0.249
N=5	28.65	0.861	0.243
N=6	28.72	0.861	0.239
N=7	28.69	0.860	0.242

ters) and cut it according to different densities of bins. The whole scene is divided into 3-7 bins, each containing multiple frames of surrounding views. As shown in Table 3, it is evident that a sparse distribution of bins results in a notable decline in performance, primarily attributable to the absence of overlapping regions among bins. Additionally, this sparse distribution may give rise to an overly extensive scale of scenes within each bin, making it impractical to adequately represent this aspect of the scene with an appropriate number of Gaussians. Alternatively, an overly dense distribution of bins may affect the Gaussian optimization efficiency between adjacent bins, leading to performance fluctuations. An appropriate distribution of bins contributes to the performance of modeling large-scale static backgrounds without wasting excessive Gaussians, thereby avoiding high computational costs.

The effectiveness of the Incremental Static 3D Gaussians. As shown in Figure 6, the Incremental Static 3D Gaussians ensure improved geometric structure and topological integrity for the static background in driving scenes. Undesirable visual effects such as blurring, artifact, and distortion have been eliminated in the incremental reconstruction process. Due to the displacement of the ego vehicle, IS3G also ensures a good consistency of the static background captured during the ego vehicle's movement.

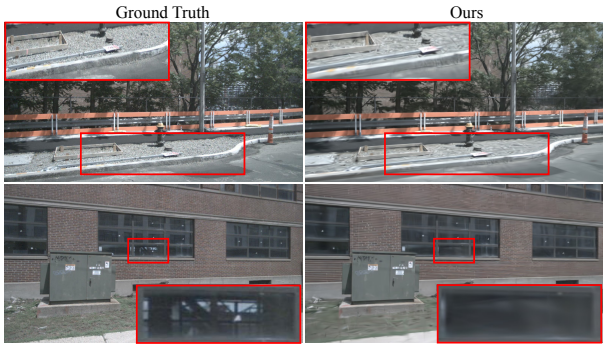

Figure 8. **Failure Cases.** Distortions exist in small objects and reflective materials (e.g., roadside pebbles and glass surfaces).

The effectiveness of the Composite Dynamic Gaussian Graph. As shown in Figure 7, without the proposed Composite Dynamic Gaussian Graph, it would result in "invisible" or distorted dynamic objects, leading to low-quality rendering results. We can also observe that CDGG exhibits good robustness towards dynamic objects, whether they are relatively fast-moving objects (e.g., vehicles and bicycles) or slower pedestrians. CDGG enables the construction of multiple fast-moving dynamic objects in large-scale, long-term driving scenes.

5. Failure Cases

Our primary limitation lies in modeling extremely small and numerous objects (such as roadside stones) and materials with total reflection properties (such as glass mirrors and water surfaces), as shown in Figure 8. We suspect that the distortions are mainly due to 3D Gaussain's shortcomings in representing densely reflected light and errors in calculating the density of fully reflective surfaces. How to reconstruct these challenging and delicate regions quickly and efficiently will be a focus of our future research.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, pages 5855–5864, 2021. 2

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, pages 5470–5479, 2022. 2, 4

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 1

[4] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *TOG*, 42(4):1–14, 2023. 2, 3, 4

[5] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2

[6] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *CVPR*, pages 12871–12881, 2022. 3, 4

[7] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022. 1

[8] Fan Lu, Yan Xu, Guang Chen, Hongsheng Li, Kwan-Yee Lin, and Changjun Jiang. Urban radiance field representation with deformable neural mesh primitives. In *ICCV*, pages 465–476, 2023. 3, 4

[9] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, pages 99–106, 2021. 4

[10] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *TOG*, 41(4):1–15, 2022. 2

[11] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *CVPR*, pages 2856–2865, 2021. 2, 3, 4

[12] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *CVPR*, pages 12932–12942, 2022. 2

[13] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 2

[14] Haithem Turki, Jason Y Zhang, Francesco Ferroni, and Deva Ramanan. Suds: Scalable urban dynamic scenes. In *CVPR*, pages 12375–12385, 2023. 2, 4

[15] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. *arXiv preprint arXiv:2306.05422*, 2023. 2

[16] Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *CVPR*, pages 8370–8380, 2023. 1

[17] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023. 2

[18] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, et al. Mars: An instance-aware, modular and realistic simulator for autonomous driving. *arXiv preprint arXiv:2307.15058*, 2023. 1

[19] Ziyang Xie, Junge Zhang, Wenye Li, Feihu Zhang, and Li Zhang. S-nerf: Neural radiance fields for street views. *arXiv preprint arXiv:2303.00749*, 2023. 2

[20] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *CVPR*, pages 5438–5448, 2022. 4

[21] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023. 1, 2, 3

[22] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *arXiv preprint arXiv:2304.06718*, 2023. 2