Street Gaussians without 3D Object Tracker

Supplementary Material

1. Evaluation on Static Scenes in Waymo-NOTR Dataset

We evaluate our method on the static32 subset of the Waymo-NOTR dataset [5, 13], following the experimental setup of EmerNeRF [13] for novel view synthesis. As shown in Table 1, while our primary focus is on handling dynamic objects, our method also demonstrates robust performance in static scenes.

Method	PSNR↑	SSIM↑	LPIPS↓
3DGS [3]	26.82	0.836	0.134
EmerNeRF [13]	28.89	0.814	0.212
S3Gaussian [2]	27.05	0.825	0.142
MARS [9]	27.63	0.848	0.193
Ours	28.72	0.857	0.092

Table 1. Comparison with state-of-the-art methods on the static32 subset of Waymo-NOTR dataset. StreetGS represents Street Gaussian [12]. The best and the second best results are denoted by pink and blue.

2. Runtime Analysis

As shown in Table 2, we evaluate the inference speed of our method and several state-of-the-art methods at a resolution of 960×640 on the same device.

Method	3DGS [3]	S3G [2]	SG [12]	Ours
Speed (FPS)	200	15	160	100

Table 2. Inference speed at 960×640. S3G and SG represent S3Gaussian [2] and Street Gaussians [12] respectively.

3. Editing Examples

We provide editing demonstrations in Fig. 1. Gaussians corresponding to cars are associated at initialization and consistently maintained throughout the optimization process. This enables object editing by directly applying rigid transformations to the corresponding Gaussians.

4. Choice of 3D tracker

To further illustrate the generalization challenges of 3D trackers, we employ CasTrack [7, 8] as the 3D tracker for Street Gaussians [12], using the same detection and tracking algorithm as in the original paper. Since pretrained nuScenes weights are unavailable, we instead use weights pretrained on KITTI. As shown in Table 3, this change leads to a significant performance drop, indicating that merely changing the detection or tracking algorithm does not resolve generalization issues.

5. Tracking Errors Analysis

We evaluate 3D trajectories from 2D and 3D trackers on Waymo-NOTR, measuring translation (Euclidean) and rotation errors (clipped at $1m$ or $30°$; with missing detections treated as max error). Our 2D tracker-based method is more robust. Error distributions are shown in Fig. 2; a qualitative comparison is shown in Fig. 1 in the main text.

6. Comparison with Rigid-transformation-based Motion Modeling

On Waymo-NOTR, we use 3D trajectories computed from 2D tracking to model vehicle motion via rigid transformations, following Street Gaussians (SG) [12] (see Tab. 4). While improved trajectories help, SG still underperforms—its joint optimization of pose and Gaussians struggles to converge in frames with large pose noise or missing detections, which in turn degrades the Gaussians across all frames. In contrast, we use HexPlane to enforce temporal smoothness by interpolating motion features from a spatio-temporal voxel grid. Since we remove trajectory guidance after the first 40% of training iterations, HexPlane can recover from noisy or missing detections by learning consistent motion from correctly tracked frames.

7. Details of Loss Functions

As described in the main paper, the total loss function is expressed as:

$$\begin{aligned}\mathcal{L} =& \lambda_{\text{rgb}}\mathcal{L}_{\text{rgb}} + \lambda_{\text{ssim}}\mathcal{L}_{\text{ssim}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{tv}}\mathcal{L}_{\text{tv}} \\ &+ \lambda_{\text{color-reg}}\mathcal{L}_{\text{color-reg}} + \lambda_{\text{motion}}\mathcal{L}_{\text{motion}}.\end{aligned} \tag{1}$$

The components of the loss function are detailed below:

1. **Photometric L1 Loss ($\mathcal{L}_{\text{rgb}}$):** This L1 loss measures the photometric difference between the rendered image and the ground truth:

$$\mathcal{L}_{\text{rgb}} = ||\mathbf{I}_{\text{render}} - \mathbf{I}_{\text{gt}}||_1, \tag{2}$$

where $\mathbf{I}_{\text{render}}$ and $\mathbf{I}_{\text{gt}}$ represent the rendered and ground truth images, respectively.

2. **Structural Similarity Loss ($\mathcal{L}_{\text{ssim}}$):** This loss evaluates the structural similarity between $\mathbf{I}_{\text{render}}$ and $\mathbf{I}_{\text{gt}}$:

$$\mathcal{L}_{\text{ssim}} = 1.0 - \text{SSIM}(\mathbf{I}_{\text{render}}, \mathbf{I}_{\text{gt}}). \tag{3}$$

Figure 1. Editing demonstrations on Waymo-NOTR.

	3D tracker	PSNR↑	SSIM↑	LPIPS↓	DPSNR↑	DSSIM↑
Street Gaussians	CasTrack [7, 8]	25.61	0.816	0.163	21.07	0.597
Street Gaussians	VoxelNext [1]	26.98	0.838	0.149	24.62	0.742

Table 3. Ablation study on the choice of 3D tracker for Novel View Synthesis on the Waymo-NOTR dataset.

Method	PSNR↑	SSIM↑	DPSNR↑	DSSIM↑
2D tracker + rigid transformation	27.69	0.850	25.04	0.758
3D tracker + rigid transformation	26.98	0.838	24.62	0.742
Ours (2D tracker + HexPlane)	28.85	0.867	25.58	0.779

Table 4. Comparison of rigid-transformation-based motion modeling [12] with our method on the Waymo-NOTR dataset.

3. **Depth Loss ($\mathcal{L}_{\text{depth}}$):** This L1 loss computes the difference between the rendered depth map $\mathbf{D}_{\text{render}}$ and the ground truth depth map derived from LiDAR data $\mathbf{D}_{\text{gt}}$:

$$\mathcal{L}_{\text{depth}} = \frac{1}{d}||\mathbf{D}_{\text{render}} - \mathbf{D}_{\text{gt}}||_1, \tag{4}$$

where $d = 80$ is the predefined maximum depth used for normalization. Depth loss is calculated only for pixels with ground truth depth values between 0.01 and 80 meters.

4. **Total Variation Loss ($\mathcal{L}_{\text{tv}}$):** A grid-based total variation loss is employed to encourage smooth gradients for HexPlane feature grids, following K-Planes [4]:

$$\mathcal{L}_{\text{tv}} = \text{avg}_{c,i,j}\left(||P_c^{i,j} - P_c^{i-1,j}||_2^2 + ||P_c^{i,j} - P_c^{i,j-1}||_2^2\right), \tag{5}$$

where avg denotes the average operator, c is the plane index, and i, j are indices on the plane resolution.

5. **Color Regularization Loss ($\mathcal{L}_{\text{color-reg}}$):** This L1 regularization loss minimizes the predicted color change $\Delta\mathcal{C}$ for each point to regularize the deformation network:

$$\mathcal{L}_{\text{color-reg}} = \Sigma||\Delta\mathcal{C}||_1. \tag{6}$$

6. **Motion Loss $\mathcal{L}_{\text{motion}}$:** The motion loss is introduced in the main paper as,

$$\mathcal{L}_{\text{motion}} = \text{avg}_{\mathcal{X}\in\mathcal{O}}\,|\Delta\mathcal{X}_t - (\mathbf{T}_t\mathcal{X} - \mathcal{X})|, \tag{7}$$

where $\mathcal{X}$ is the center position of a Gaussian in object $\mathcal{O}$.

The weights assigned to each loss component are: $\lambda_{\text{rgb}} = 1.0$, $\lambda_{\text{ssim}} = 0.1$, $\lambda_{\text{depth}} = 1.0$, $\lambda_{\text{tv}} = 0.1$, $\lambda_{\text{color-reg}} = 0.01$, and $\lambda_{\text{motion}} = 1.0$.

8. Limitation and Failure Cases

- Our approach primarily focuses on modeling moving vehicles while using 4DGS [6] to model humans without explicit motion guidance. This leads to inaccurate human motion in some cases, which can be improved by incorporating human pose estimation as prior information (see Fig. 3). Incorporating human pose estimation techniques [10, 11] in future work could enhance human motion modeling.
- Dynamic objects beyond humans and vehicles, such as animals and traffic lights, are currently treated as static objects. Further refinement is needed to more effectively

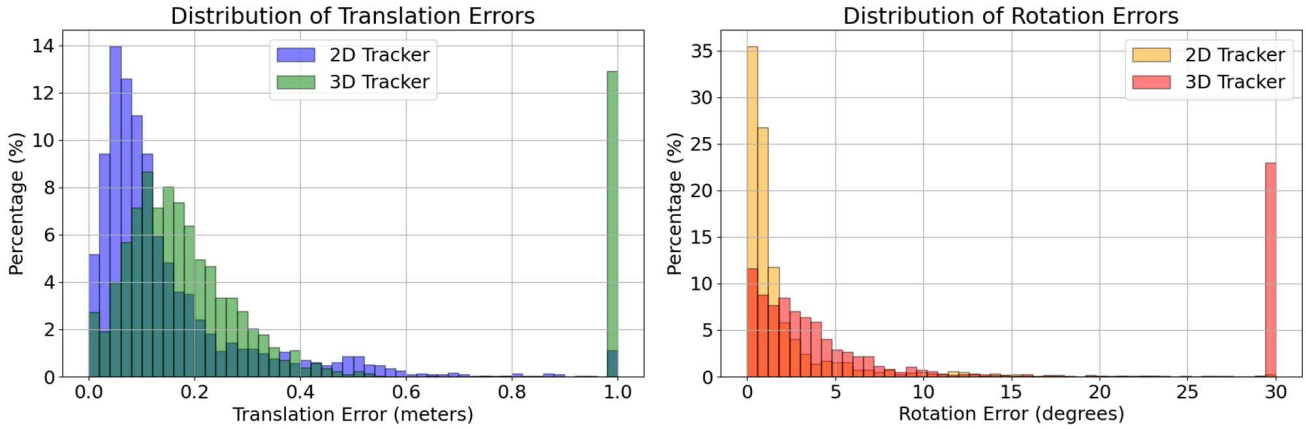

Figure 2. Distributions of tracking errors.

Figure 3. Failure cases

distinguish between static and dynamic elements within a scene.

- The proposed method requires per-scene optimization. A promising direction for future work is the development of a feed-forward approach for predicting generalizable 3D Gaussians.

References

[1] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21674–21683, 2023. 2

[2] Nan Huang, Xiaobao Wei, Wenzhao Zheng, Pengju An, Ming Lu, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. S3gaussian: Self-supervised street gaussians for autonomous driving. *arXiv preprint arXiv:2405.20323*, 2024. 1

[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1

[4] Sara Fridovich-Keil and Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023. 2

[5] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 1

[6] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20310–20320, 2024. 2

[7] Hai Wu, Wenkai Han, Chenglu Wen, Xin Li, and Cheng Wang. 3d multi-object tracking in point clouds based on prediction confidence-guided data association. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):5668–5677, 2021. 1, 2

[8] Hai Wu, Jinhao Deng, Chenglu Wen, Xin Li, Cheng Wang, and Jonathan Li. Casa: A cascade attention network for 3-d object detection from lidar point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2022. 1, 2

[9] 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. In *CAAI International Conference on Artificial Intelligence*, pages 3–15. Springer, 2023. 1

[10] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple vision transformer baselines for human pose estimation. In *Advances in Neural Information Processing Systems*, 2022. 2

[11] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose+: Vision transformer foundation model for generic body pose estimation. *arXiv preprint arXiv:2212.04246*, 2022. 2

[12] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *ECCV*, 2024. 1, 2

[13] 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