GGS: Generalizable Gaussian Splatting for Lane Switching in Autonomous Driving

Anonymous submission
Paper ID: 5255

Appendices

A. Implementation Details

Multi-Lane Diffusion Details

Firstly, we use the Variational AutoEncoder (Kingma and Welling 2013) to encode the input image into latent code. Then, we perform several steps of denoising on this latent code, mainly to prepare for better utilization of the Diffusion Prior. As switching lanes for autonomous driving may cause some floating artifacts, we adapt a method of adding noise first and then removing it to eliminate these noises. Denoising U-Net takes the latent code after adding noise as the initialization parameter, and the input text is fixed as the autonomous driving label. Generated through the CLIP model, denoised through several steps, and then decoded into images using the Variational AutoEncoder.

We construct reasonable multi-lane novel view images, instead of utilizing image of the current lane as input for U-Net denoising. This approach helps ensure that the autonomous driving lane remains visible in the image following a change in viewpoint.

$$y_i = \mathcal{V}(I, \gamma_i), \gamma_i \in (-0.5, 0.5), \tag{1}$$

Where $\mathcal{V}$ represents the lane converter, which calculates pose based on the distance of left and right translation, and renders a new perspective using 3DGS. i represents the $i-th$ rendering view image. γ_i represents the translation distance coefficient, positive numbers indicate movement to the right lane, and negative numbers indicate movement to the left lane. I represents the image rendered by 3DGS in the middle lane. The denoised images are represented as follows:

$$\hat{y}_\pi = \mathcal{V}D(DDIM(\sqrt{\bar{\alpha}_k}\mathcal{V}E(y) + \sqrt{1-\bar{\alpha}_k}\bar{z}_k)), \tag{2}$$

where $\bar{\alpha}_k$ represents the added noise coefficient, and $\bar{z}_k$ represents noise. $DDIM$ is a denoising algorithm derived from (Song, Meng, and Ermon 2020). $\mathcal{V}E$ and $\mathcal{V}D$ respectively represent VAEEncoder and VAEDecoder, derived from (Kingma and Welling 2013).

Datasets Details

KITTI Dataset. The KITTI dataset (Geiger, Lenz, and Urtasun 2012) is an open dataset widely used for autonomous driving and computer vision research, jointly created and maintained by the Karlsruhe Institute of Technology and Toyota Technological Institute. It primarily includes various sensor data collected from vehicles, such as high-resolution images, LiDAR data, and camera calibration information. These data integrate various traffic scenarios in the real world, such as city streets, highways, etc. The dataset mainly includes several categories: City, Residential, Road, Campus, Person and Calibration. We select three of these categories, City, Residential, and Road, for experimentation. Among them, we select 10 scenarios with a total of 6696 frames for training and testing. Finally, we average the PSNR, VGG, LPIPS and SSIM metrics.

Brno Urban Dataset. The Brno Urban dataset (Ligocki, Jelinek, and Zalud 2020) is an open dataset specifically designed for pedestrian detection and pedestrian trajectory analysis, developed and maintained by the research team at Brno University. The Brno Urban dataset includes four views: left view, right view, left front view, and right front view. We select three views, left side view, left front view, and right side view, for a total of 1200 frames in the experiment, using evaluation criteria similar to KITTI.

B. More Results

Evaluation on KITTI and BrnoUrban

In addition, like READ (Li, Li, and Zhu 2023), we perform a more challenging task by discarding 5 frames before and after every 100 frames as test frames. Due to the fast speed of the car, a large amount of scene information would be lost after discarding consecutive frames. As our model is a generalized model, it has some advantages in learning new scenarios that have not been trained before, and it performs best in most scenarios. The experimental table data is shown in Table 1. We add more qualitative analysis experiments, as shown in Figure 1.

Assessing Cross-dataset Generalization

Our method GGS has the advantage of generalization in extending to new scenarios outside the distribution. To evaluate the generalization of our model, We can achieve synthesis effects close to ground truth without any training. Specifically, we train the model on KITTI dataset and test it on Brno Urban dataset (Ligocki, Jelinek, and Zalud 2020). Conversely, we train the model on Brno Urban and test it on

Table 1: Quantitative evaluation of novel view synthesis on KITTI dataset and Brno dataset.

	KITTI Residential				KITTI Road				KITTI City			
	VGG↓	PSNR↑	LPIPS↓	SSIM↑	VGG↓	PSNR↑	LPIPS↓	SSIM↑	VGG↓	PSNR↑	LPIPS↓	SSIM↑
Test on KITTI dataset												
NPBG	924.7	14.98	0.4426	0.4733	791.4	17.63	0.3680	0.5080	994.5	14.97	0.4384	0.4518
ADOP	900.5	14.89	0.3590	0.4734	785.9	17.56	0.3275	0.4701	910.6	15.67	0.3497	0.4774
READ	695.3	17.70	0.2875	0.5963	573.5	20.26	**0.2408**	0.6238	673.2	18.35	0.2529	0.6412
UC-NeRF	694.7	22.19	0.454	0.7806	703.2	20.94	0.506	0.7108	525.2	**24.23**	0.3858	0.7188
3DGaussian	775.2	19.64	0.4878	0.7496	732.2	19.34	0.4654	**0.7631**	549.2	21.04	0.287	0.7455
GaussianPro	677.2	20.89	0.4353	0.7826	733.1	19.6	0.4648	0.7565	545.1	21.19	0.2864	0.7476
DC-Gaussian	618.7	20.62	0.3872	0.8009	737.1	19.33	0.4571	0.7374	591.7	22.27	0.3264	0.735
Ours	**487.3**	**23.64**	**0.1673**	**0.8304**	**565.9**	**21.54**	0.3033	0.6644	**454.2**	23.38	**0.1655**	**0.7721**
Test on Brno Urban dataset												
	Left side view				Left front side view				Right side view			
NPBG	1002.3	13.14	0.5242	0.3978	724.5	17.13	0.4098	0.5596	1024.4	12.22	0.6634	0.4333
ADOP	997.1	14.08	0.4373	0.3915	683.6	18.24	0.3150	0.5618	1091.2	13.21	0.5531	0.3803
READ	842.0	15.28	0.3992	0.4752	523.9	20.51	0.2467	0.6713	928.0	13.88	0.5464	0.4533
UC-NeRF	642.3	23.35	0.5197	**0.8312**	1074.0	14.66	0.6785	0.61	457.7	25.16	0.4443	0.7917
3DGaussian	795.6	19.58	0.5236	0.7774	964.0	14.53	0.6604	0.6852	**288.5**	**26.84**	0.2502	**0.898**
GaussianPro	768.3	20.07	0.5044	0.7835	1012.4	13.89	0.6898	0.6766	288.8	26.73	**0.2418**	0.8965
DC-Gaussian	707.2	19.76	0.5403	0.7805	694.5	20.5	0.4616	0.7738	489.7	25.31	0.3853	0.8424
Ours	**384.1**	**24.6**	**0.1385**	0.8175	**433.7**	**23.8**	**0.2159**	**0.7951**	396.3	25.83	0.2716	0.787

KITTI, as shown in Figure 2.

Assessing Lane Switching

To demonstrate the effectiveness of our model's lane switching, we conduct more lane switching experiments and compare it with other models. Although DC-Gaussian and GaussianPro can render high-quality views of the main lane, the Gaussian sphere looks chaotic after lane switching. However, generalized Gaussian models like MVSplat are difficult to achieve ideal results due to the lack of a multi-lane perspective dataset for training. It can be seen that our model has high quality in lane switching, as shown in Figure 3.

References

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, 3354–3361. IEEE.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Li, Z.; Li, L.; and Zhu, J. 2023. Read: Large-scale neural scene rendering for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 1522–1529.

Ligocki, A.; Jelinek, A.; and Zalud, L. 2020. Brno urban dataset-the new data for self-driving agents and mapping tasks. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 3284–3290. IEEE.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

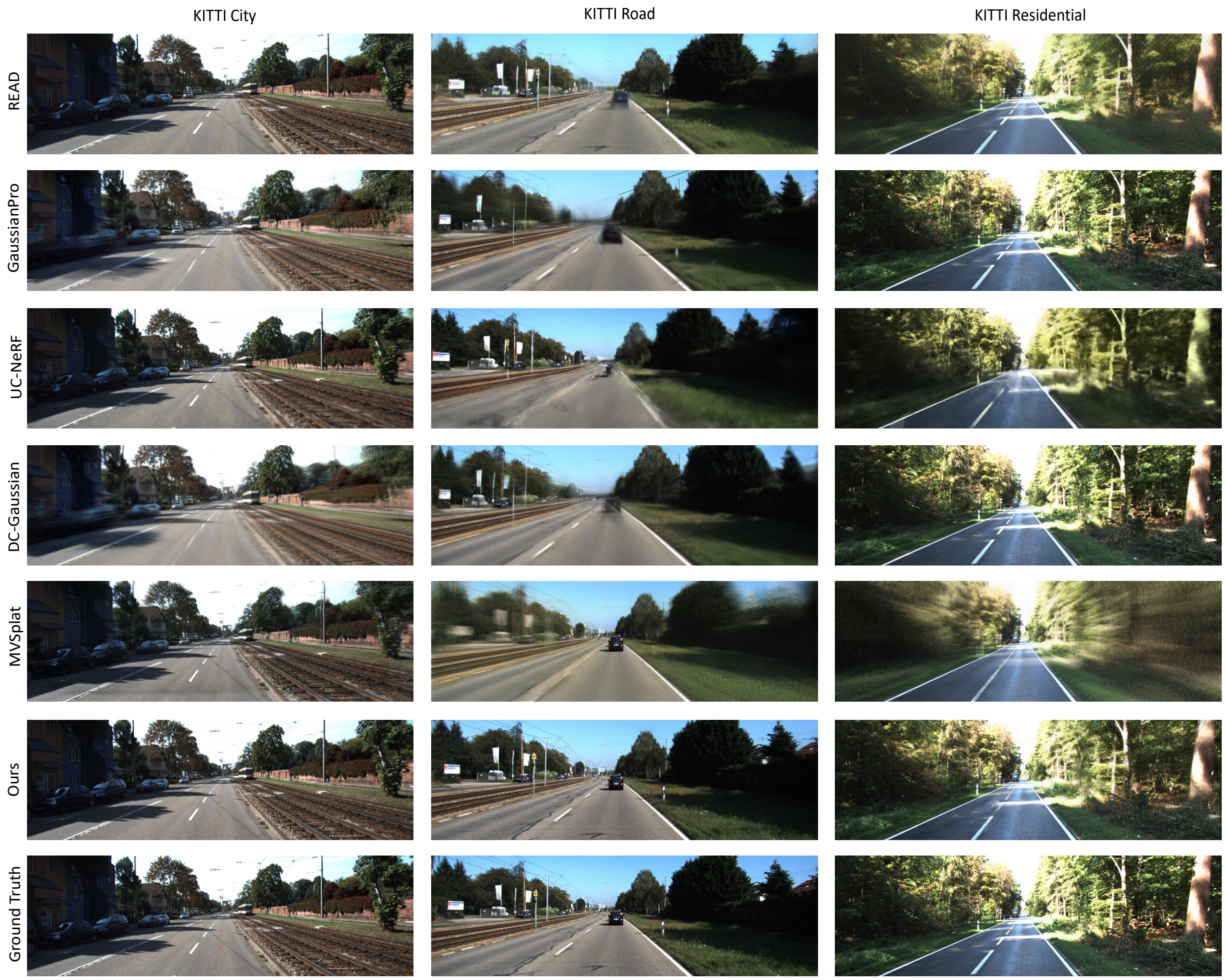

Figure 1: Comparison results of novel view synthesis based on KITTI for residential, road, and urban scenes.

(a) KITTI -> Brno Urban

(b) Brno Urban -> KITTI

Figure 2: Cross-dataset generalization. (a) train the model on the KITTI dataset and test it on the Brno Urban dataset. (b) train the model on the Brno Urban dataset and test it on the KITTI dataset.

Figure 3: Comparison of lane switching between different models on KITTI dataset.