GGS: Generalizable Gaussian Splatting for Lane Switching in Autonomous Driving

Huasong Han[1*], Kaixuan Zhou[2*], Xiaoxiao Long[3], Yusen Wang[1], Chunxia Xiao[1†]

[1]School of Computer Science, Wuhan University, Wuhan, China,
[2]Huawei Technologies Riemann Lab, Wuhan, Hubei, China,
[3]The University of HongKong, Hong Kong, China
hanhuasong@whu.edu.cn, zhoukaixuan2@huawei.com, xxlong@connect.hku.hk, wangyusen@whu.edu.cn, cxxiao@whu.edu.cn

Abstract

We propose GGS, a Generalizable Gaussian Splatting method for Autonomous Driving that can achieve realistic rendering under large viewpoint changes. Previous generalizable 3D gaussian splatting methods are limited to rendering novel views that are very close to the original pair of images, which cannot handle large difference in viewpoint. Especially in autonomous driving scenarios, images are typically collected from a single lane. The limited training perspective makes rendering images of a different lane very challenging. To further improve the rendering capability of GGS under large viewpoint changes, we introduce a novel virtual lane generation module into GSS method to enable high-quality lane switching even without a multi-lane dataset. Besides, we design a diffusion loss to supervise the generation of virtual lane images to further address the problem of data lacking in the virtual lanes. Finally, we also propose a depth refinement module to optimize depth estimation in the GSS model. Extensive validation of our method, compared to existing approaches, demonstrates state-of-the-art performance.

Introduction

Novel view synthesis is an essential task in the field of computer vision, with significant potential applications in autonomous driving (Yang et al. 2020; Wu et al. 2023; Liu et al. 2023; Yang et al. 2023b, 2024a; Yu et al. 2024), object detection, scene reconstruction (Wang et al. 2024b; Qin et al. 2024) and digital human representation (Li, Luo, and Xiao 2024; Luo et al. 2024). To enhance the robustness of autonomous driving systems, it is imperative to establish a simulation environment for testing these systems effectively. However, the majority of existing datasets are limited to single-lane scenarios. This limitation presents significant challenges in inferring adjacent lane scenarios from the current viewpoint. If lane switching is not supported, the test samples provided to the autonomous driving simulation system will be incomplete, making it impossible to conduct better simulation testing and requiring a significant amount of data collection costs.

*Both authors contributed equally to this research.
†Corresponding author.

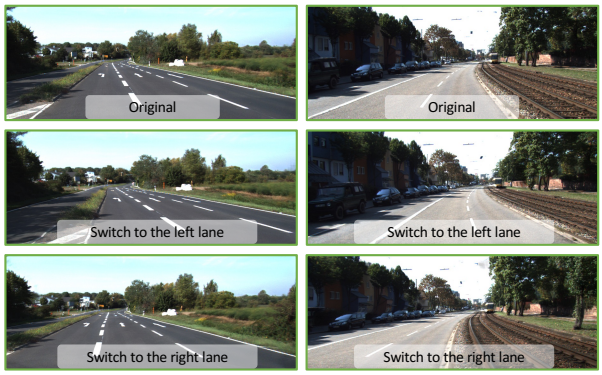

Figure 1: Our GGS method can achieve high-quality lane switching in autonomous driving scenarios.

The methods based on NeRF (Mildenhall et al. 2021; Yang et al. 2023a) often rely on LiDAR to better generate novel views in autonomous driving scenarios. READ (Li, Li, and Zhu 2023) introduces a new rendering method that adopts a neural rendering approach different from NeRF. It learns neural descriptor of the original point cloud with explicit geometry to render image, instead of learning implicit geometry in NeRF methods. However, the efficiency of training and rendering using these methods remains quite low.

The efficiency in training and rendering speed, coupled with the high reconstruction quality of 3D Gaussian Splatting (Kerbl et al. 2023), contributes to its widespread application in novel view synthesis in autonomous driving. GaussianPro (Cheng et al. 2024) introduces multi-view stereo to improve the geometry of generated gaussian splats. DC-Gaussian (Cheng et al. 2023) introduces an adaptive image decomposition module to mitigate the impact of glass reflections on the quality of novel view synthesis. However, these methods still cannot perform effective novel view synthesis in lane switching, as they do not address the main problem that only single lane of data are collected.

To address the problem of the sparse view synthesis, many methods have sought to optimize this process using generative models (Yu et al. 2021; Chen et al. 2024; Liu et al. 2024; Wu et al. 2024; Tang et al. 2024). Generative models

are trained across large amount of scenes to enhance performance in sparse view scenarios. However, the generative model still lacks data from multi-lanes to learn how to synthesize novel views for other lanes from single lane data.

Therefore, we introduce a virtual lane module into generative Gaussian splatting to address the synthesis of new views involving lane change, despite the lack of multi-lane training datasets for supervision. In the module, we first use 3D gaussians generated from images in the single lanes using a generative model to predict images from virtual lanes, then use 3D gaussians generated from the virtual images to predict back the image collected in the single lanes. In this way, we can let generative model learn how to generate the best images in the other lanes even with only single lane of data. In addition, we introduce a diffusion loss from a latent diffusion model (Sohl-Dickstein et al. 2015; Song, Meng, and Ermon 2020; Nichol and Dhariwal 2021; Rombach et al. 2022) to virtual generated images to further improve the lane switching of our GGS. Finally, as improving geometry of generated 3D gaussians also improves novel view synthesis in sparse view collections, we employ points from traditional multiview stereo reconstruction to refine the depth estimated in GGS.

The main contributions of this paper can be summarized as follows:

- We propose a novel virtual lane module into the generative gaussian splatting to improve the quality of novel lane switching view with only single lane of data.
- We introduce a diffusion loss to directly supervise the image from virtual lanes predicted by GGS to further improve the novel view synthesis from limited collected views.
- We propose to fuse MVS geometry into the generative 3D gaussian splating to improve geometry estimation.
- We conduct extensive experiments on a wide range of scenarios to validate the effectiveness of our algorithm, and achieve state-of-the-art street novel view synthesis even without LiDAR.

Related Work

3D Gaussian Splatting

3D Gaussian Splatting (Kerbl et al. 2023) employs a point-cloud-based 3D reconstruction method, which combines the position information of each point with Gaussian distribution to convert point cloud data into a 3D surface. However, the quality of street novel view synthesis is still problematic due to limited view collection in the street.

GaussianPro (Cheng et al. 2024) has improved 3D Gaussian Splatting by introducing a novel progressive propagation strategy to guide Gaussian densification based on the scene's surface structure. Although improving geometries helps to mitigate the issues in novel synthesis in sparse views, the quality of novel synthesis in other lanes is still low. Deformable 3D Gaussians (Yang et al. 2024b) employ a framework for extending 3D Gaussian Splatting in dynamic scenes using a deformation field, enabling the learning of 3D Gaussians in a normalized space. There are also other methods based on 3D Gaussian Splatting such as (Zhou et al. 2024; Paliwal et al. 2024; Niedermayr, Stumpfegger, and Westermann 2024). Although street view synthesis has been improved on the collected lanes, they have not solved the problem of sparse views, leaving lower level of novel view synthesis when changing lanes.

Generalized Model

To solve the problem of novel view synthesis in sparse views, some methods propose a generalized model-based approach. PixelNeRF (Yu et al. 2021) employs a generalized model for novel view synthesis based on volume rendering method, which can be trained directly from images without explicit 3D supervision. However, the generation quality is not high and the training efficiency is low. MVSplat (Chen et al. 2024) introduces an efficient feedforward 3D Gaussian splash model learned from sparse multi-view images, and constructs a cost volume to represent the cross view feature similarity of different candidate depths, providing valuable clues for depth estimation (Li, Luo, and Xiao 2023; Li et al. 2023). MVSGaussian (Liu et al. 2024) employs a mixture Gaussian rendering method that integrates efficient volume rendering design for novel view synthesis. Compared with the original 3D Gaussian Splatting, MVSGaussian achieves better view synthesis results while reducing training computational costs. However, it cannot perform well for scenes with obstacles.

Diffusion Model

For occluded scenes, using generalized models cannot generate better results, so some algorithms introduce diffusion model (Sohl-Dickstein et al. 2015; Rombach et al. 2022; Nichol and Dhariwal 2021; Song, Meng, and Ermon 2020) to imagine unknown regions. ReconFusion (Wu et al. 2024) further utilizes the generative capacity of large models to infer unknown areas, and integrates diffusion prior into NeRF's 3D reconstruction process. DrivingDiffusion (Li, Zhang, and Ye 2023) introduces a spatiotemporally consistent diffusion framework, incorporating multi-view attention to generate realistic multi-view videos controlled by 3D layouts. These diffusion-model-based methods only consider a single lane and do not utilize multi-lane features for better completion.

Methodology

Although generalized models can assist in synthesizing novel views in sparse views, insufficient view information leads to inaccurate depth estimation. Our method further optimizes the generalization model. The overall framework diagram of our GGS method is shown in Figure 2. We input four different frame images and introduce neighboring features in the **Multi-View Depth Refinement Module** to better address scenes with occlusions. And we introduce more global information to optimize the predicted depth map by using MVS. In the **Virtual Lane Generation Module**, we introduce the concept of virtual lanes and solve the problem of not having a multi-lane dataset by switching lanes and

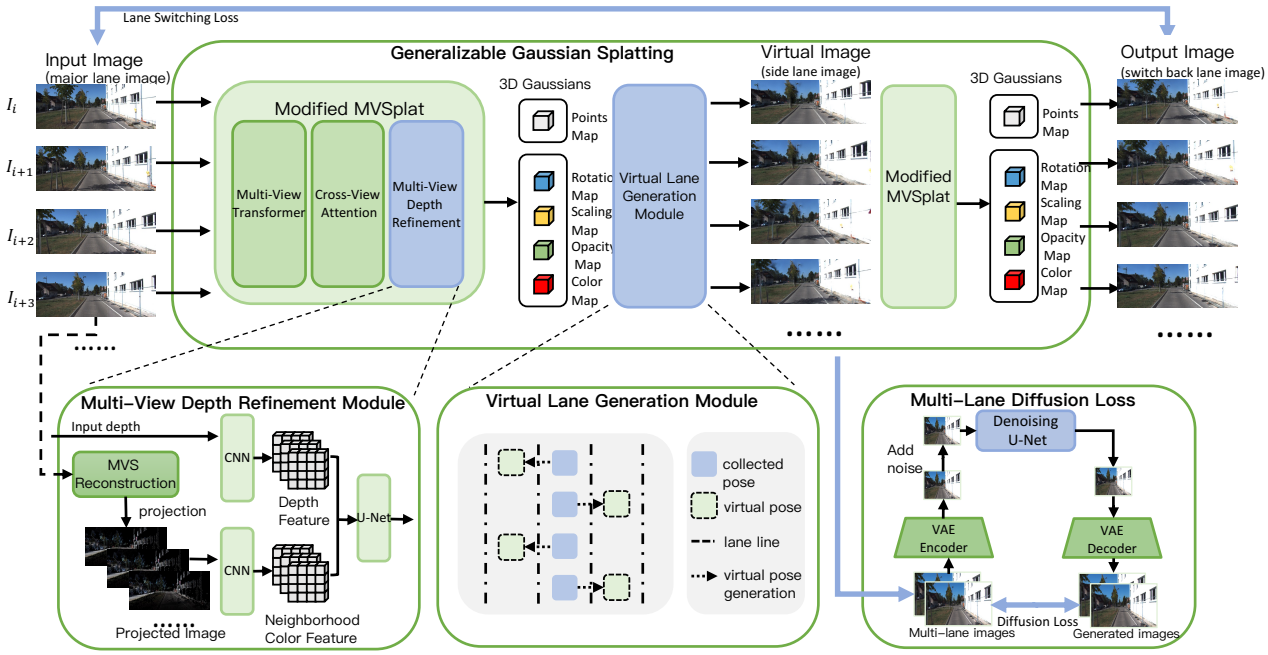

Figure 2: The overall framework of the GGS. We input multiple frames and estimate depth maps through MVS and multi-view depth refinement modules, combined with 3DGS to synthesize novel views. Through the virtual lane generation module, we switch lanes with high quality. In addition, multi-lane diffusion loss is introduced to supervise the novel view synthesis in the presence of obstacles.

then switching back, allowing the model to flexibly switch lanes. In addition, we introduce the **Multi-Lane Diffusion Loss** to supervise the novel view synthesis.

Background

MVSplat (Chen et al. 2024) is a generalizable 3D Gaussian Splatting method, which can synthesize novel views from sparse inputs. MVSplat takes a transformer-based structure and adopts cross view attention strategy to build a cost volume for each input view, then following a U-Net to predict the depth and the parameters of Gaussian primitives for each pixel. The 3D Gaussian parameters consist of the Gaussian center position x, scale s, rotation angle q, opacity α, and color c. Given the predicted depth map D and the projection matrix P with camera parameter K, pixels located at x are back-projected from the image plane to 3D space as follows:

$$x_{p_x} = \Pi_P^{-1}(p_x, D), \tag{1}$$

where Π represents the back-projection operation, and p_x and D represent pixel coordinate and estimated depth, respectively. The opacity α is represented by the matching confidence directly. The remaining Gaussian parameters, scale s, rotation angle q, and color c are decoded from the encoded features as follows:

$$s_{p_x} = Softplus(h_s(\Gamma(p_x))), \tag{2}$$

$$q_{p_x} = Norm(h_q(\Gamma(p_x))), \tag{3}$$

$$c_{p_x} = Sigmoid(h_c(\Gamma(p_x))), \tag{4}$$

where Γ represents the high-dimensional feature vector, p_x represents pixel coordinate, and h_s, h_q, and h_c represent the scaling head, rotation head, and color head, respectively.

Multi-View Depth Refinement Module

We enhance MVSplat by our Multi-View Depth Refinement Module, i.e. Modified MVSplat. It can produce more accurate 3D gaussian primitives and improve the quality of novel view synthesis. To better infer unknown regions, we incorporate the neighboring color feature information of this view. We use a back-projected point cloud map reconstructed through Agisoft Metashape as an additional input color feature for U-Net. The neighboring feature is represented as:

$$F_{neighbor_i} = \{F_m | m \in [i-k, i+k]\}, \tag{5}$$

where i represents the i-th frame in the video, and F_i represents the color feature of the i-th frame, k represents neighboring distance.

Neighboring color features are merged into depth features through concatenation, high-dimensional Gaussian parameter features are output through U-Net, decoded using a Gaussian parameter decoder, and finally generate Gaussian parameter representation:

$$dep_{ref} = \mathcal{U}(F_{neighbor_i}, dep_i), \tag{6}$$

where $\mathcal{U}$ represents the U-Net. By introducing color information from multiple neighboring perspectives in this way,

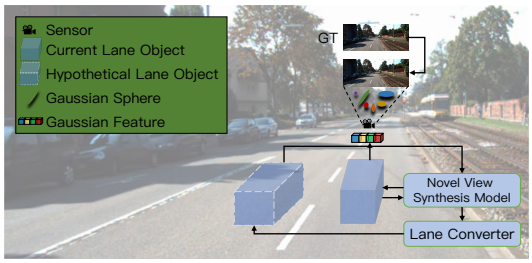

Figure 3: Using a lane converter to create a virtual lane and then switching back to the real lane enables the model to improve the quality of lane switching.

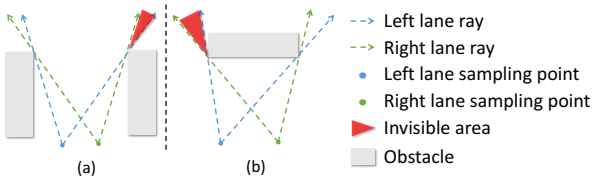

Figure 4: If we switch from the right lane to the left lane, the red area represents the blind spot of the right lane. When rendering the left lane, in order to avoid voids, the content of that area needs to be imagined in some way.

the synthesis ability of the generalized model under obstacle occlusion is enhanced.

In addition, to refine the depth, we introduce a confidence based method. The lower the transparency of the predicted 3D Gaussian, the lower the confidence level of the predicted depth. When the confidence level is high, the predicted depth remains unchanged. When the confidence level is low, we correct the predicted depth map by reconstructing the back-projected depth map through Agisoft Metashape (met 2019). The optimized depth value is:

$$dep_i = \begin{cases} \beta \hat{dep}_i + (1-\beta) D_i, if \alpha_i < \alpha \\ \hat{dep}_i, if \alpha_i \geq \alpha \end{cases}, \quad (7)$$

where D_i represents the depth of projected depth map. $\hat{dep}_i$ represents the predict depth. α and β represent the transparency threshold and depth threshold, respectively.

Virtual Lane Generation Module

Previous generalizable 3D gaussian splatting methods are limited to rendering novel views that are very close to the original pair of images, which cannot handle large difference in viewpoint. Especially in autonomous driving scenarios, images are typically collected from a single lane. The limited training perspective makes rendering images of a different lane very challenging. We have obtained a 3D Gaussian using our modified depth refinement module. To further improve the rendering capability of GGS under large viewpoint changes, we introduce the virtual lane approach that enables high-quality lane switching even without a multi-lane dataset, inspired by (Huang et al. 2023).

The virtual lane converter is used to select the appropriate virtual lane, so that after lane switching, no information can be seen from the virtual perspective due to excessive switching amplitude. Then we generate a pose for the virtual lane by performing a vertical translation along the lane. Finally, we generate a virtual perspective based on the pose of the virtual lane. After introducing the virtual lane module, our GGS module process mainly includes two stages.

In the first stage, we input a set of N images:

$$ISet_1 = \{I_1, I_2, ...I_N\}, \quad (8)$$

then we output the target image through the model:

$$\hat{I^1} = \mathcal{G}(ISet_1), \quad (9)$$

where $\mathcal{G}$ represents GGS module, and $ISet_1$ is a rendered image without shifting the view, and the rendered view is consistent with the ground truth. The current lane generates a collection of virtual lane rendering images through lane converter. The rendered image of the virtual lane is represented as:

$$ISet_2 = \{\mathcal{V}(\hat{I_k^1}, \gamma sin\theta) | k_f \leq k \leq k_l, \theta = \omega k\}, \quad (10)$$

where $\mathcal{V}$ represents the virtual lane converter, γ represents the translation coefficient, k_f and k_l represent the index of the first and last frames of the input, respectively. ω represents the switching period angle, and the switching angle of each frame changes periodically in order.

In the second stage, we use the virtual lane generated in the first stage as input. Using our model, we switch back from the virtual lane to the real lane and output a rendered image of the real lane:

$$\hat{I^2} = \mathcal{G}(ISet_2), \quad (11)$$

where $\mathcal{G}$ represents GGS module. This forms a closed-loop process of switching to a new lane and then switching back. The advantage of doing so is that even without the ground truth of the left and right lanes, we can still enhance the quality of the model's rendering of the left and right lanes by establishing virtual lanes, allowing the model to improve the quality of lane switching, as shown in Figure 3.

Multi-Lane Diffusion Loss

There is no ground truth available for training when switching lanes. When the lane switching amplitude is large, obstacles can obstruct the view during lane changes, making it impossible to collect information about the new lane from the current lane, as shown in Figure 4. Therefore, to better address this issue, we use diffusion prior knowledge to imagine color information from a novel lane view.

The traditional diffusion model denoising method directly completes the generated image, but due to the diversity of the generated models, it can lead to inconsistent results between frames. Our method calculates the loss of the denoised image and the image before denoising, and generates a new perspective supervised by diffusion. Additionally, we construct multi-lane novel view images, instead of utilizing image of the current lane as input for U-Net denoising. This

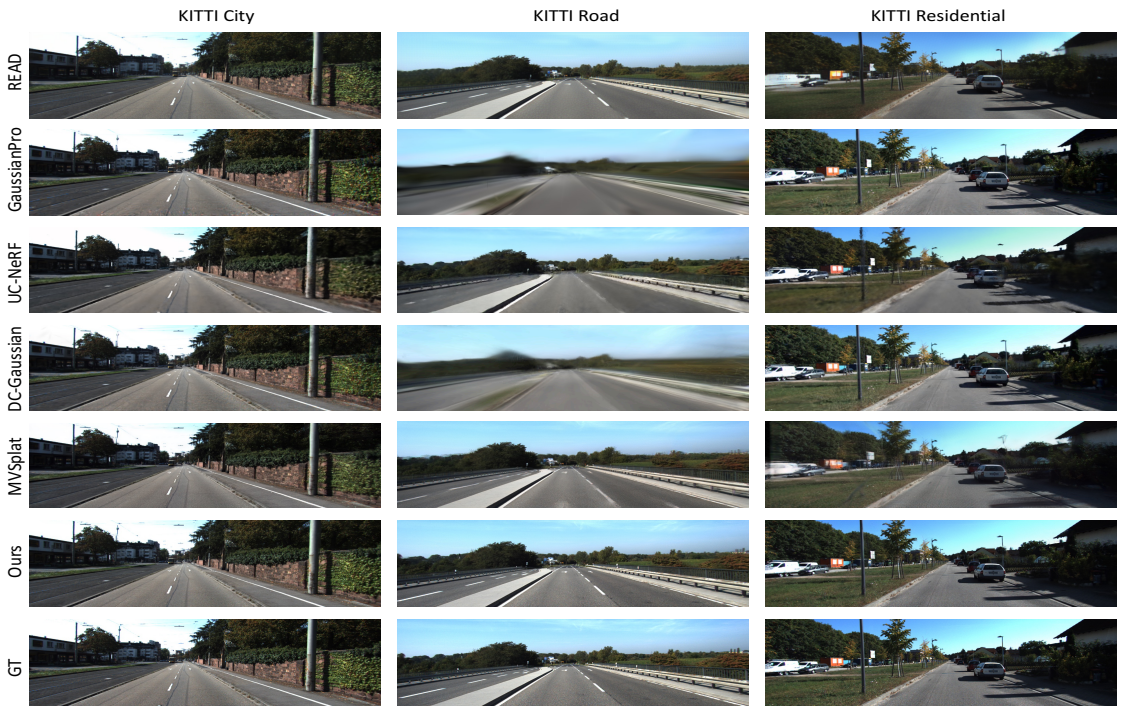

Figure 5: Comparison results of novel view synthesis based on KITTI for residential, road, and urban scenes.

approach helps ensure that the autonomous driving lane remains visible in the image following a change in viewpoint.

Specifically, we adapt the Stable Diffusion framework (Rombach et al. 2022), and use the Variational AutoEncoder (Kingma and Welling 2013) to encode the multi-lane images into latent code, including the left lane, middle lane and right lane. Then, we perform several denoising steps on the latent code as an initialization parameter for Denoising U-Net, and fix the input text as the autonomous driving label. We generated it through the CLIP (Radford et al. 2021), denoised through several steps, and then decoded into images using the Variational AutoEncoder. These images serve as supervision to guide the synthesis of novel views.

Loss Function

Our model is trained on a single lane dataset and introduces a method of constructing virtual lanes to generate unknown domains through diffusion models. Therefore, our method mainly includes reconstruction loss, depth loss, virtual lane switching loss, and diffusion loss. The overall loss function is represented as follows:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{switch}} + \mathcal{L}_{\text{diffusion}}. \quad (12)$$

Reconstruction loss. Our GGS model is a generative model for novel views on autonomous driving. During the training process, we construct a reconstruction loss function by comparing the rendered image with the ground truth using mean square error loss:

$$\mathcal{L}_{\text{recon}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \quad (13)$$

where y_i represents the color value in Ground Truth corresponding to a certain pixel, and $\hat{y}_i$ represents the color value in the rendered image corresponding to the same pixel.

Depth loss. In most autonomous driving scenarios, lanes are regular and even, so the depth of adjacent pixels should be smooth to avoid abrupt changes. Therefore, we construct the depth loss function as follows:

$$\mathcal{L}_{\text{depth}} = \frac{1}{n}\sum_{i=1}^{n}(\frac{dD_i}{dx} + \frac{dD_i}{dy} + \lambda(\frac{d^2D_i}{dx^2} + \frac{d^2D_i}{dy^2})), \quad (14)$$

where $\frac{dD_i}{dx}$, $\frac{dD_i}{dy}$, $\frac{d^2D_i}{dx^2}$ and $\frac{d^2D_i}{dy^2}$ represent the first and second derivatives of the depth in the x and y-axis directions of the image, respectively, and λ is the depth smoothing adjustment factor.

Lane switching loss. Due to the lack of lane switching data, we train the model by constructing virtual lanes and switching back, and construct a lane switching loss:

$$\mathcal{L}_{\text{switch}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \Psi(\Phi(\hat{y}_i)))^2, \quad (15)$$

where Φ represents constructing virtual lanes and Ψ represents switching from the virtual lane to the current lane.

Multi-lane diffusion loss. When we switch lanes in autonomous driving, changes in view can cause artifacts, so we use denoising methods to eliminate noise:

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{\pi,t}\left[\beta(t)\left(\|y - \hat{y}_\pi\|_1 + \mathcal{L}_{\text{lpips}}(y, \hat{y}_\pi)\right)\right], \quad (16)$$

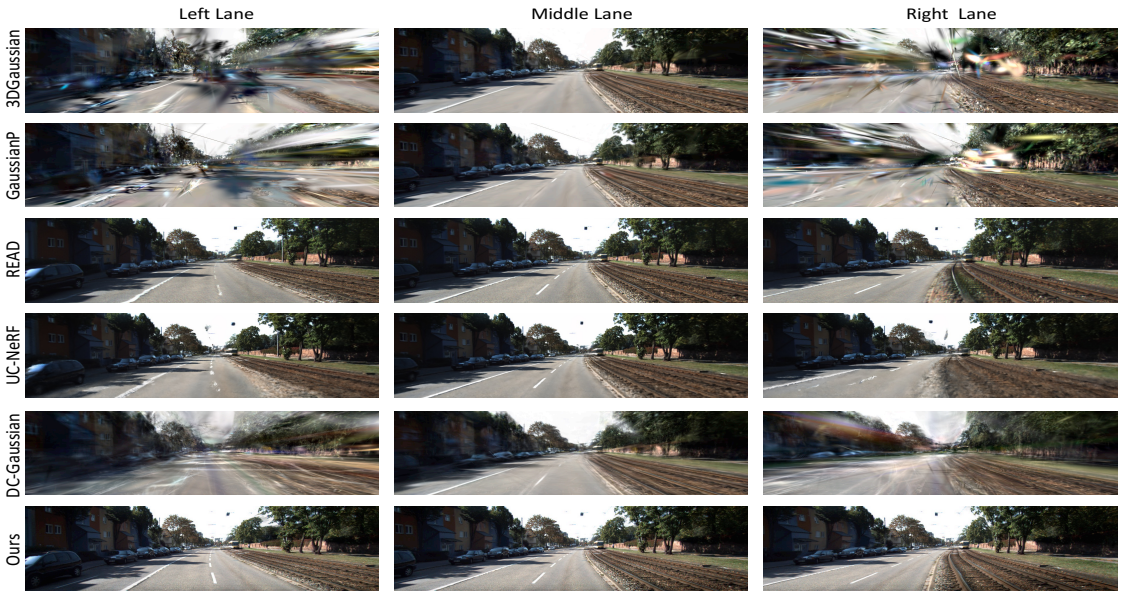

Figure 6: Comparison of lane switching between different models on KITTI dataset.

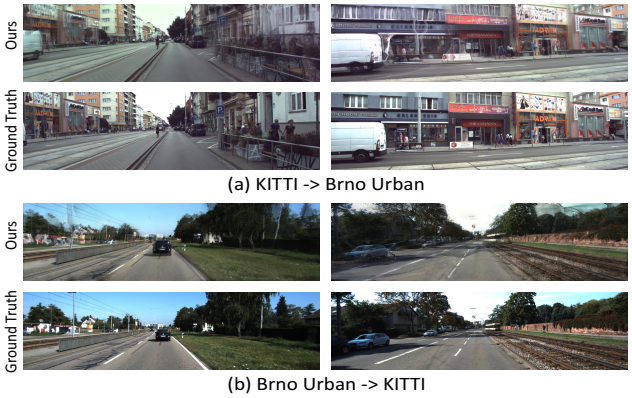

Figure 7: Cross-dataset generalization. (a) Train the model on the KITTI dataset and test it on the Brno Urban dataset. (b) Train the model on the Brno Urban dataset and test it on the KITTI dataset.

where π represents the camera pose of the selected view, y represents the multi-lane image, $\hat{y}_\pi$ represents the output image from the denoising model, $\beta(t)$ is a weight function related to the noise level, and $\mathcal{L}_{\text{lpips}}$ represents perceptual loss, which aims to emulate human perception of image similarity to better capture visual differences between images.

Experiments

We compare GGS with ADOP (Rückert, Franke, and Stamminger 2022), READ (Li, Li, and Zhu 2023), 3DGaussian (Kerbl et al. 2023), GaussianPro (Cheng et al. 2024), UC-NeRF (Cheng et al. 2023) and DC-Gaussian (Wang et al. 2024a). We use Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), perceptual loss (VGG loss), perceptual metrics, and Learned Perceptual Image Patch Similarity (LPIPS) as evaluation metrics.

Evaluation on KITTI and BrnoUrban

From Table 1, the methods based on 3D Gaussian Splatting, such as GaussianPro and DC-Gaussian, generate slightly better quality than other methods based on neural radiation fields. However, in some scenes, the rendering quality is inferior, and our model performs better.

As illustrated in Figure 5, GaussianPro and DC-Gaussian fail to capture details such as tree leaves and utility poles. The rendering quality of the READ is inadequate, and UC-NeRF does not render the white lines in the middle of the road. The comparison methods of different models for lane switching are shown in Figure 6. Compared to other models, our method demonstrates excellent overall rendering quality and lane switching quality.

Assessing Cross-dataset Generalization

Our GGS method has the advantage of generalization in extending to new scenarios outside the distribution. To evaluate the generalization of our model, we conduct two cross-dataset evaluations. Specifically, we train the model on the KITTI dataset and test it on the Brno Urban dataset (Ligocki, Jelinek, and Zalud 2020). Conversely, we train the model on the Brno Urban dataset and test it on the KITTI dataset, as shown in Figure 7.

Ablation Study

Effect of Virtual Lane Generation module. To demonstrate the effectiveness of the virtual lane generation module, we use FID (Heusel et al. 2017) to conduct lane-switching experiments on different models, as shown in Ta-

	KITTI Residential				KITTI Road				KITTI City			
	VGG↓	PSNR↑	LPIPS↓	SSIM↑	VGG↓	PSNR↑	LPIPS↓	SSIM↑	VGG↓	PSNR↑	LPIPS↓	SSIM↑
Test on KITTI dataset												
ADOP	610.8	19.07	0.2116	0.5659	577.7	19.67	0.2150	0.5554	560.9	20.08	0.1825	0.6234
READ	454.9	22.09	0.1755	0.7242	**368.2**	24.29	**0.1465**	0.7402	391.1	23.48	0.1321	0.7871
UC-NeRF	555.1	23.7	0.4229	0.7564	772.9	20.62	0.4998	0.6502	469.2	24.7	0.3453	0.7555
3DGaussian	585.8	22.66	0.3683	0.7859	760	20.92	0.4544	0.7331	372.4	24.92	0.2258	0.8566
GaussianPro	532.5	23.74	0.337	0.8006	602.5	23.46	0.3803	0.78	327.9	24.84	0.1999	0.8763
DC-Gaussian	416.6	25.63	0.2739	0.8406	707.4	21.28	0.417	0.7422	343.4	25.04	0.2115	0.8713
MVSplat	549.6	22.45	0.3008	0.6562	515.6	21.08	0.2749	0.7457	369.9	24.11	0.1667	0.7755
Ours	**259.1**	**26.26**	**0.0948**	**0.8840**	372.6	**25.01**	0.1542	**0.827**	**271.6**	**26.79**	**0.0933**	**0.8781**
Test on Brno Urban dataset												
	Left side view				Left front side view				Right side view			
ADOP	634.0	19.19	0.2414	0.5927	520.6	20.83	0.2189	0.6633	807.1	16.51	0.3636	0.5009
READ	459.8	21.79	0.1905	0.7067	**341.1**	24.85	**0.1513**	0.7836	663.6	18.44	0.3065	0.5771
UC-NeRF	640.9	23.47	0.5201	0.8318	900.7	20.28	0.6315	0.7251	431.8	27.27	0.4212	0.7977
3DGaussian	530.6	25.63	0.3583	0.828	753.5	19.05	0.5634	0.7675	400.8	28.02	0.3229	0.8629
GaussianPro	520.5	25.75	0.3501	0.8307	738.6	19.43	0.5528	0.7731	394.0	**28.27**	0.3151	0.8623
DC-Gaussian	699.1	20.86	0.4772	0.8102	493.5	25.82	0.3373	0.8404	303.4	26.95	0.2692	**0.898**
MVSplat	383.3	24.97	0.1652	0.8013	513.7	22.49	0.2712	0.7693	511.4	22.21	0.3806	0.6575
Ours	**275.3**	**27.8**	**0.0861**	**0.8829**	354.7	**25.85**	0.1679	**0.8415**	**288.9**	26.5	**0.1833**	0.8936

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

	VGG↓	PSNR↑	LPIPS↓	SSIM↑
baseline	398.1	25.14	0.1935	0.7665
w/o virtual lane generation module	257.7	26.76	0.0783	0.8808
w/o multi-lane diffusion loss	215.4	28.97	0.0681	0.9056
w/o depth refinement module	213.5	29.05	0.0670	0.9074
Ours(full model)	**210.8**	**29.12**	**0.0657**	**0.9087**

Table 2: Ablation study on KITTI dataset.

	FID@LEFT↓	FID@RIGHT↓
3DGaussian	201.13	164.01
GaussianPro	198.35	154.94
DC-Gaussian	165.13	209.39
UC-NeRF	106.89	82.98
READ	79.79	76.87
Ours	**60.34**	**55.17**

Table 3: Lane switching experiment on KITTI dataset.

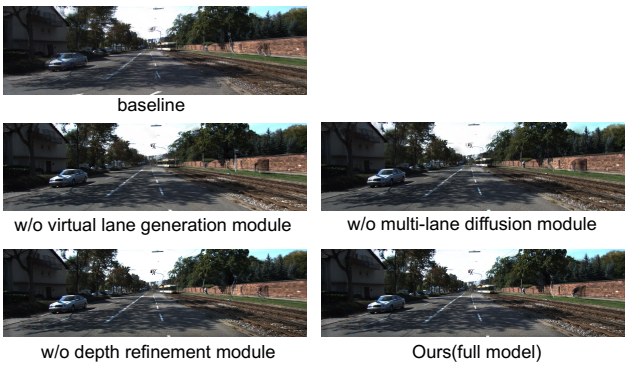

Figure 8: Qualitative ablation study on KITTI dataset.

ble 3. FID@LEFT and FID@RIGHT represent the distance between the rendered images of the left and right lanes and the GT. The qualitative experimental results are illustrated in Figure 6. Our model achieves high rendering quality while ensuring that quality remains unaffected during lane switching, with quantitative results shown in Table 2 and qualitative results shown in Figure 8.

Effect of Multi-Lane Diffusion Loss. Due to limited input view information, some unknown areas cannot be synthesized after lane switching. Therefore, a diffusion model is used to imagine the unknown areas and optimize the generation quality, as shown in Table 2.

Effect of Depth Refinement Module. The depth refinement module introduces neighboring feature information to optimize depth estimation in the presence of occluded objects, as shown in Table 2. After removing the depth refinement module, each metric is affected slightly.

Conclusions

In this paper, we have proposed a generative framework based on MVS and 3D Gaussian Splatting fusion, which can repair unknown regions to optimize generation quality. By simulating the virtual lanes, our method effectively switches driving lanes in autonomous driving scenarios, suitable for simulation testing of autonomous driving systems. Our method has some limitations, and the quality of lane switching generation needs to be improved when encountering dynamic scenes with complex road conditions, multiple people, and mixed vehicles.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (No. 62372336 and No. 61972298) and Wuhan University Huawei GeoInformatices Innovation Lab.

References

2019. Agisoft: Metashape software. *retrieved 20.05.2019 (2019)*.

Chen, Y.; Xu, H.; Zheng, C.; Zhuang, B.; Pollefeys, M.; Geiger, A.; Cham, T.-J.; and Cai, J. 2024. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. *arXiv preprint arXiv:2403.14627*.

Cheng, K.; Long, X.; Yang, K.; Yao, Y.; Yin, W.; Ma, Y.; Wang, W.; and Chen, X. 2024. Gaussianpro: 3d gaussian splatting with progressive propagation. In *Forty-first International Conference on Machine Learning*.

Cheng, K.; Long, X.; Yin, W.; Wang, J.; Wu, Z.; Ma, Y.; Wang, K.; Chen, X.; and Chen, X. 2023. UC-NeRF: Neural Radiance Field for Under-Calibrated multi-view cameras in autonomous driving. *arXiv preprint arXiv:2311.16945*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Huang, S.; Gojcic, Z.; Wang, Z.; Williams, F.; Kasten, Y.; Fidler, S.; Schindler, K.; and Litany, O. 2023. Neural lidar fields for novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18236–18246.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4): 139–1.

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

Li, X.; Zhang, Y.; and Ye, X. 2023. DrivingDiffusion: Layout-Guided multi-view driving scene video generation with latent diffusion model. *arXiv preprint arXiv:2310.07771*.

Li, Y.; Luo, F.; and Xiao, C. 2023. Monocular human depth estimation with 3D motion flow and surface normals. *The Visual Computer*, 39(8): 3701–3713.

Li, Y.; Luo, F.; and Xiao, C. 2024. Diffusion-FOF: Single-View Clothed Human Reconstruction via Diffusion-Based Fourier Occupancy Field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9525–9534.

Li, Y.-Z.; Zheng, S.-J.; Tan, Z.-X.; Cao, T.; Luo, F.; and Xiao, C.-X. 2023. Self-Supervised Monocular Depth Estimation by Digging into Uncertainty Quantification. *Journal of Computer Science and Technology*, 38(3): 510–525.

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.

Liu, J. Y.; Chen, Y.; Yang, Z.; Wang, J.; Manivasagam, S.; and Urtasun, R. 2023. Real-time neural rasterization for large scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8416–8427.

Liu, T.; Wang, G.; Hu, S.; Shen, L.; Ye, X.; Zang, Y.; Cao, Z.; Li, W.; and Liu, Z. 2024. Fast Generalizable Gaussian Splatting Reconstruction from Multi-View Stereo. *arXiv preprint arXiv:2405.12218*.

Luo, C.; Luo, F.; Wang, Y.; Zhao, E.; and Xiao, C. 2024. DLCA-Recon: Dynamic Loose Clothing Avatar Reconstruction from Monocular Videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 3963–3971.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, 8162–8171. PMLR.

Niedermayr, S.; Stumpfegger, J.; and Westermann, R. 2024. Compressed 3d gaussian splatting for accelerated novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10349–10358.

Paliwal, A.; Ye, W.; Xiong, J.; Kotovenko, D.; Ranjan, R.; Chandra, V.; and Kalantari, N. K. 2024. CoherentGS: Sparse Novel View Synthesis with Coherent 3D Gaussians. *arXiv preprint arXiv:2403.19495*.

Qin, J.; Luo, F.; Cao, T.; Xu, W.; and Xiao, C. 2024. HS-Surf: A Novel High-Frequency Surface Shell Radiance Field to Improve Large-Scale Scene Rendering. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 6006–6014.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Rückert, D.; Franke, L.; and Stamminger, M. 2022. Adop: Approximate differentiable one-pixel point rendering. *ACM Transactions on Graphics (ToG)*, 41(4): 1–14.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2256–2265. PMLR.

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

Tang, S.; Chen, J.; Wang, D.; Tang, C.; Zhang, F.; Fan, Y.; Chandra, V.; Furukawa, Y.; and Ranjan, R. 2024. Mvdiffusion++: A dense high-resolution multi-view diffusion model for single or sparse-view 3d object reconstruction. *arXiv preprint arXiv:2402.12712*.

Wang, L.; Cheng, K.; Lei, S.; Wang, S.; Yin, W.; Lei, C.; Long, X.; and Lu, C.-T. 2024a. DC-Gaussian: Improving 3D Gaussian Splatting for Reflective Dash Cam Videos. *arXiv preprint arXiv:2405.17705*.

Wang, Y.; Zhou, K.; Zhang, W.; and Xiao, C. 2024b. MegaSurf: Scalable Large Scene Neural Surface Reconstruction. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 6414–6423.

Wu, R.; Mildenhall, B.; Henzler, P.; Park, K.; Gao, R.; Watson, D.; Srinivasan, P. P.; Verbin, D.; Barron, J. T.; Poole, B.; et al. 2024. Reconfusion: 3d reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21551–21561.

Wu, Z.; Liu, T.; Luo, L.; Zhong, Z.; Chen, J.; Xiao, H.; Hou, C.; Lou, H.; Chen, Y.; Yang, R.; et al. 2023. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CAAI International Conference on Artificial Intelligence*, 3–15. Springer.

Yang, H.; Zhang, S.; Huang, D.; Wu, X.; Zhu, H.; He, T.; Tang, S.; Zhao, H.; Qiu, Q.; Lin, B.; et al. 2024a. Unipad: A universal pre-training paradigm for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15238–15250.

Yang, J.; Ivanovic, B.; Litany, O.; Weng, X.; Kim, S. W.; Li, B.; Che, T.; Xu, D.; Fidler, S.; Pavone, M.; et al. 2023a. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*.

Yang, Z.; Chai, Y.; Anguelov, D.; Zhou, Y.; Sun, P.; Erhan, D.; Rafferty, S.; and Kretzschmar, H. 2020. Surfelgan: Synthesizing realistic sensor data for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11118–11127.

Yang, Z.; Chen, Y.; Wang, J.; Manivasagam, S.; Ma, W.-C.; Yang, A. J.; and Urtasun, R. 2023b. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1389–1399.

Yang, Z.; Gao, X.; Zhou, W.; Jiao, S.; Zhang, Y.; and Jin, X. 2024b. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20331–20341.

Yu, A.; Ye, V.; Tancik, M.; and Kanazawa, A. 2021. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4578–4587.

Yu, Z.; Wang, H.; Yang, J.; Wang, H.; Xie, Z.; Cai, Y.; Cao, J.; Ji, Z.; and Sun, M. 2024. SGD: Street View Synthesis with Gaussian Splatting and Diffusion Prior. *arXiv preprint arXiv:2403.20079*.

Zhou, X.; Lin, Z.; Shan, X.; Wang, Y.; Sun, D.; and Yang, M.-H. 2024. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21634–21643.