DrivingForward: Feed-forward 3D Gaussian Splatting for Driving Scene Reconstruction from Flexible Surround-view Input

Qijian Tian[1], **Xin Tan**[2*], **Yuan Xie**[2], **Lizhuang Ma**[1 2 3*]

[1]Shanghai Jiao Tong University, Shanghai, China
[2]East China Normal University, Shanghai, China
[3]MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai, China
tianqijian@sjtu.edu.cn, xtan@cs.ecnu.edu.cn, xieyuan8589@foxmail.com, ma-lz@cs.sjtu.edu.cn

Abstract

We propose DrivingForward, a feed-forward Gaussian Splatting model that reconstructs driving scenes from flexible surround-view input. Driving scene images from vehicle-mounted cameras are typically sparse, with limited overlap, and the movement of the vehicle further complicates the acquisition of camera extrinsics. To tackle these challenges and achieve real-time reconstruction, we jointly train a pose network, a depth network, and a Gaussian network to predict the Gaussian primitives that represent the driving scenes. The pose network and depth network determine the position of the Gaussian primitives in a self-supervised manner, without using depth ground truth and camera extrinsics during training. The Gaussian network independently predicts primitive parameters from each input image, including covariance, opacity, and spherical harmonics coefficients. At the inference stage, our model can achieve feed-forward reconstruction from flexible multi-frame surround-view input. Experiments on the nuScenes dataset show that our model outperforms existing state-of-the-art feed-forward and scene-optimized reconstruction methods in terms of reconstruction.

Code — https://github.com/fangzhou2000/DrivingForward

1 Introduction

3D scene reconstruction is critical for understanding driving scenes. Modern self-driving assistance cars are usually equipped with several cameras to capture surrounding scenes. Real-time reconstruction of driving scenes from sparse vehicle-mounted cameras contributes to various downstream tasks in autonomous driving, including online mapping (Li et al. 2022a), BEV perception (Li et al. 2022b; Liang et al. 2022; Liu et al. 2023), scene understanding (Ji et al. 2024; Sun et al. 2024a; Tan et al. 2023; Sun et al. 2024b; Ma et al. 2024) and 3D detection (Chen et al. 2023; Cai et al. 2023). However, the real-time computing required by downstream tasks and the sparse surrounding views challenge the driving scene reconstruction.

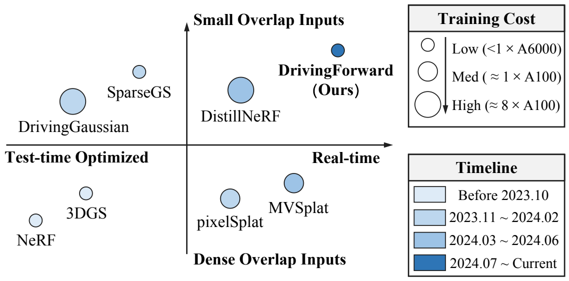

Figure 1: Comparison of our DrivingForward with the latest related works. We achieve real-time reconstruction from small overlap inputs with fewer computing resources.

Neural Radiance Fields (NeRF) (Mildenhall et al. 2021) and 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) have significantly progressed the development of 3D scene reconstruction. DrivingGaussian (Zhou et al. 2024), StreetGaussian (Yan et al. 2024), S^{3}Gaussian (Huang et al. 2024a), and AutoSplat (Khan et al. 2024) further explore the reconstruction of driving scenes. While these methods demonstrate strong capability in novel view synthesis, they are scene-optimized methods that require dozens of images and expensive computing time to reconstruct just one scene. These offline reconstruction methods are unsuitable for real-time downstream tasks in autonomous driving, thereby limiting their practicality.

Our goal is to achieve online, generalizable driving scene reconstruction from sparse surrounding views. Several attempts, such as pixelSplat (Charatan et al. 2024) and MVSplat (Chen et al. 2024), have explored the generalizable reconstruction. They learn powerful priors from large-scale datasets during training and achieve fast 3D reconstruction from sparse input views through a feed-forward inference. Unfortunately, these methods are difficult to apply in driving scenes. Since the number of vehicle-mounted cameras is limited (usually 6 cameras), the overlap of adjacent views is extremely small (as low as 10%). While these existing methods often require densely overlapping (usually over 60%) input images. Additionally, acquiring camera extrinsics for each view at various timesteps in driving scenes is costly. These methods depend on such data during training, limiting their practical applicability. A recent NeRF-based work DistillNeRF (Wang et al. 2024) attempts to develop

[*]Corresponding author.

a generalizable 3D representation for driving scenes. However, it gains a suboptimal performance and relies on LiDAR to train numerous NeRF models for distillation, which is extremely computationally expensive. Besides, previous feed-forward methods typically have a fixed mode of input views, either using stereo images (e.g., MVSplat, pixelSplat) or single-frame images of surrounding views (e.g., DistillNeRF). However, as the vehicle moves forward and captures surround-view images frame-by-frame, we aim to support flexible multi-frame inputs for reconstruction, such as predicting the next frame's views from single-frame surrounding views or synthesizing intermediate frame surrounding views from two interval frames.

In summary, online and generalizable reconstruction of driving scenes face challenges including real-time processing, sparse surrounding views with minimal overlap, and variable numbers of input frames.

To this end, we introduce DrivingForward, a novel feed-forward Gaussian Splatting model that enables real-time reconstruction of driving scenes from flexible sparse surround-view images. We train a generalizable model and achieve real-time reconstruction via a feed-forward inference. In driving scenes, the minimal overlap between sparse cameras limits the direct use of geometric relationships from multi-views. Consequently, we predict Gaussian primitives from each input image individually and aggregate them to represent the 3D driving scenes. However, reconstruction from a single image is inherently ill-posed due to the principle of scale ambiguity (Charatan et al. 2024), which may lead to inconsistent scales across multi-views. To address this issue, inspired by surround-view depth estimation (Kim et al. 2022; Guizilini et al. 2022), we propose scale-aware localization for Gaussian primitives. At the training stage, we input multi-frame surround-view images into a pose network and a depth network. The pose network predicts the camera pose, i.e., extrinsics, and the depth network estimates the dense depth map for each image. These two networks are only supervised by the photometric loss from input images and learn scale information in a self-supervised manner without ground truth depth and camera extrinsics. At the inference stage, the depth network predicts real-scale depth from single-frame images individually, ensuring consistent depth estimation across multi-frame inputs.

By unprojecting the consistent depth estimation, we obtain the position of Gaussian primitives. For other Gaussian parameters, we individually predict them from each image through a Gaussian network. The Gaussian network is jointly trained with the pose network and depth network. It takes the depth map and image feature from the depth network as input and outputs the covariance, opacity, and spherical harmonics coefficients of Gaussian primitives. Since Gaussian primitives are predicted independently from single-frame images of surrounding views, our method is not constrained by a fixed number of input frames. This allows for flexible multi-frame surround-view input, such as predicting the next frames' views from the current frame or synthesizing the intermediate frame from two interval frames.

Extensive experiments on the nuScenes dataset demonstrate that our DrivingForward outperforms other feed-forward methods on the novel view synthesis under various inputs. It also achieves higher reconstruction quality compared to scene-optimized methods with the same input. A functional comparison of our DrivingForward with the latest related works is present in Figure 1.

We summarize our main contributions as follows:

- To our knowledge, DrivingForward is the first feed-forward Gaussian Splatting model for surround-view driving scenes. It achieves real-time reconstruction from sparse vehicle-mounted cameras and supports flexible multi-frame inputs of surrounding views.
- We introduce a scale-aware localization and Gaussian parameters prediction to reconstruct driving scenes. The scale-aware localization learns real scale depth from surrounding views without using ground truth depth and extrinsics. Then we independently predict Gaussian parameters from each image, thereby supporting flexible multi-frame inputs. The full model is trained end-to-end.
- Comprehensive experiments show that DrivingForward achieves the best performances against both feed-forward methods and scene-optimized methods for driving scene reconstruction.

2 Related Work

2.1 Feed-Forward Reconstruction

Neural Radiance Fields (NeRF) (Mildenhall et al. 2021) and 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) have significantly advanced 3D scene reconstruction. Some following works (Xiong et al. 2023; Li et al. 2024) also explore the reconstruction from sparse views. However, these scene-optimized methods require training on dozens of images for each scene and lack generalizability across different scenes. In contrast, feed-forward reconstruction methods learn powerful priors from large-scale datasets and reconstruct scenes through a feed-forward inference from sparse views, which are significantly faster than scene-optimized methods. NeRF-based methods (Yu et al. 2021; Du et al. 2023; Suhail et al. 2022) pioneer the paradigm of feed-forward reconstruction. Recent 3DGS avoids NeRF's expensive volume sampling via splat-based rasterization. Therefore, 3DGS-based feed-forward reconstruction methods, such as Splatter Image (Szymanowicz, Rupprecht, and Vedaldi 2024), GPS-Gaussian (Zheng et al. 2024), pixelSplat (Charatan et al. 2024), and MVSplat (Chen et al. 2024), outperform the previous NeRF-based methods. However, they fail to apply in driving scenes. Splatter Image focuses on single-object reconstruction, while GPS-Gaussian targets human reconstruction. They are both unsuitable for much larger driving scenes. The pixelSplat involves a two-view encoder to resolve scale ambiguity and MVSplat relies on stereo images to construct cost volume, both requiring densely overlapping images as input. However, the surrounding views of driving scenes have only minimal overlap, which greatly impacts their performance of reconstruction from single-frame surround-view images. In this work, we propose a feed-forward model to reconstruct driving scenes from sparse surrounding views with minimal overlap.

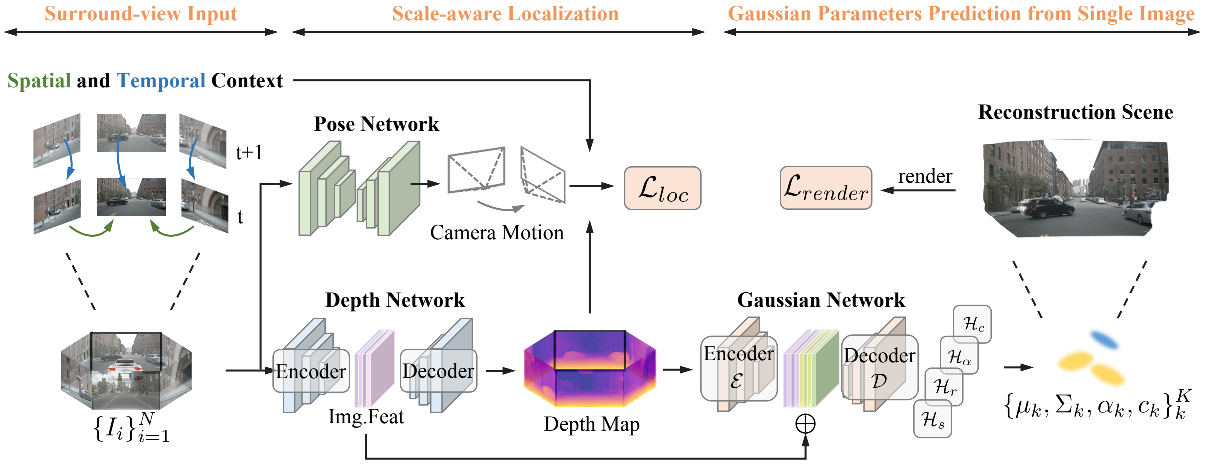

Figure 2: Overview of DrivingForward. Given sparse surround-view input from vehicle-mounted cameras, our model learns scale-aware localization for Gaussian primitives from the small overlap of spatial and temporal context views. A Gaussian network predicts other parameters from each image individually. This feed-forward pipeline enables the real-time reconstruction of driving scenes and the independent prediction from single-frame images supports flexible input modes. At the inference stage, we include only the depth network and the Gaussian network, as shown in the lower part of the figure.

2.2 Driving Scene Reconstruction

Based on NeRFs or 3DGS, a few methods (Guo et al. 2023; Yang et al. 2024b; Zhou et al. 2024; Yan et al. 2024; Khan et al. 2024) extend reconstruction specifically for driving scenes. However, most of them focus on accurate 3D or 4D reconstruction for a single scene. These offline reconstruction methods limit the application of real-time downstream tasks in driving scenes. In contrast, our method is designed for real-time reconstruction to support online downstream tasks. A few NeRFs-related methods (Huang et al. 2024b; Yang et al. 2024a) also learn 3D representations for online driving, while they mainly focus on other downstream tasks instead of reconstruction. Another related work DistillNeRF (Wang et al. 2024) proposes a generalizable model for driving scenes and achieves reasonable novel-view synthesis. However, it requires a pre-training stage that involves offline optimization of NeRFs for each scene, which is extremely costly and necessitates 8 A100 GPUs (8 × 80GB). In contrast, our DrivingForward only requires a single GPU with 48GB. DistillNeRF also relies on LiDAR for training, while our method is trained from RGB images alone. Despite DitillNeRF requiring more training resources and additional LiDAR data, it achieves only suboptimal performance compared to our DrivingForward under the same setting.

3 Method

3.1 Overview

DrivingForward learns powerful priors from large-scale driving scene datasets during training and achieves real-time driving scene reconstruction in a feed-forward manner from sparse vehicle-mounted cameras at the inference stage.

We begin with N sparse camera images $\{I_i\}_{i=1}^N$ as input and aim to predict Gaussian primitives from input view images. The overview framework is illustrated in Figure 2. A pose network $\mathcal{P}$ and a deep network $\mathcal{D}$ predict the vehicle motion and estimate the scale-aware depth from the input. We assign each pixel to one Gaussian primitive and the position is located through the estimated depth. Other parameters of the Gaussian primitives are predicted by a Gaussian network $\mathcal{G}$. We unproject the Gaussian primitives from all views into 3D space, render them to the target view in a differentiable way, and jointly train the full model end-to-end. At the inference stage, the depth network and Gaussian network are used for feed-forward reconstruction. Since the scale-aware localization and prediction of other parameters do not depend on other frames, we can flexibly input different numbers of surround-view frames during inference.

3.2 Scale-aware Localization

The original 3DGS (Kerbl et al. 2023) explicitly models a scene with a set of Gaussian primitives $\{g_k = \{\mu_k, \Sigma_k, \alpha_k, c_k\}_k^K\}$, each of which is parameterized by a position μ_k, a 3D covariance matrix Σ_k, an opacity α_k and spherical harmonics c_k that determines color. It uses Structure from Motion to initialize the Gaussians' position and optimizes them through splat-based rasterization rendering.

In contrast, to achieve feed-forward inference without test-time optimization, we directly predict Gaussian primitives from input images in a pixel-wise manner and assign each pixel to one primitive. In this way, accurately localizing the position of Gaussian primitives is the key to high-quality reconstruction as it determines the center of primitives. However, in driving scenes, the limited overlap between sparse cameras limits geometric relationships from multiple views. This presents challenges for existing methods (Chen et al. 2024; Charatan et al. 2024) that depend on multi-views with large overlapping (such as the adjacent frames of the same camera) to get the position. We instead estimate the depth map of a single frame without relying on other frames. To obtain multi-frame consistent depth, we propose a scale-aware localization inspired by self-supervised surround-view depth estimation (Guizilini et al. 2022; Kim et al. 2022). It learns scale-aware depth from multi-frame surround views during training and independently predicts the depth of the real scale from different frames of surrounding views during inference, thereby achieving consistent scale-aware Gaussian localization.

Specifically, we introduce a pose network $\mathcal{P}$ and a depth network $\mathcal{D}$. At the training stage, we input multi-frame surround view images from sparse vehicle-mounted cameras $\{C_i\}_{i=1}^N$. The pose network predicts the vehicle motion and the depth network estimates the depth map:

$$\begin{aligned} \mathcal{P}(I_i^t, I_i^{t'}) \rightarrow T_i^{t\rightarrow t'}, \\ \mathcal{D}(I_i^t) \rightarrow D_i^t, \end{aligned} \tag{1}$$

where I_i^t denotes the image from camera C_i at the timestep t, $t' \in \{t+1, t-1\}$, $T_i^{t\rightarrow t'}$ is a project matrix from timestep t to timestep t' that indicates the camera motion and the D_i^t is the depth map of image I_i^t.

To learn the camera motion and scale-aware depth map from input images in a self-supervised manner, we apply photometric loss for multi-frame surrounding views. The photometric loss is used to minimize the projecting loss between a target image I_{trg} and a synthesis image $\hat{I}_{trg}$ that is warped from a source image I_{src}, and I_{src} is usually obtained from stereo pairs or monocular videos (Godard, Mac Aodha, and Brostow 2017; Godard et al. 2019),

$$\mathcal{L}_{reproj} = \eta \frac{1 - SSIM(I_{trg}, \hat{I}_{trg})}{2} + (1-\eta)\|I_{trg} - \hat{I}_{trg}\|, \tag{2}$$

where SSIM is the structure similarity metric (Wang et al. 2004), η is 0.15. The warped operation can be depicted as:

$$\hat{I}_{trg} = I_{src}[K_{src} T^{trg\rightarrow src} D_{trg} K_{trg}^{-1}], \tag{3}$$

where K_{src} and K_{trg} are camera intrinsics of I_{src} and I_{trg}, D_{trg} is the depth map of I_{trg}, and $T^{trg\rightarrow src}$ is the cam-to-cam transformation matrix from I_{trg} to I_{src}.

In driving scenes, with multi-frame surrounding views, we take different inputs as the source image to compute the photometric loss. First, we use the images from the same camera at different frames, denoted as temporal contexts. Then, we use the images from adjacent cameras at the same frame, denoted as spatial contexts. We also combined the two ways, using images from adjacent cameras at different frames, denoted as spatial-temporal contexts. The key insight is to leverage the small overlap between spatially and temporally neighboring images for matching, which provides scale information and enables learning scale-aware camera motion and depth maps during training.

Let C_i, C_j be two adjacent cameras and I_i^t, I_j^t be their images at timestep t. For $i = 1, ..., N$:

$$I_{trg} = I_i^t, \tag{4}$$

$$I_{src} = \begin{cases} I_i^{t'}, t' \in \{t+1, t-1\} & \text{for temporal,} \\ I_j^t, & \text{for spatial,} \\ I_j^{t'}, t' \in \{t+1, t-1\} & \text{for spatial-temporal,} \end{cases} \tag{5}$$

and

$$T^{trg\rightarrow src} = \begin{cases} T_i^{t\rightarrow t'}, t' \in \{t+1, t-1\} & \text{for temporal,} \\ E_j E_i^{-1} & \text{for spatial,} \\ E_j E_i^{-1} T_i^{t\rightarrow t'}, t' \in \{t+1, t-1\} & \text{for spatial-temp,} \end{cases} \tag{6}$$

where E_i and E_j are the transformation matrix from the camera coordinate system to the vehicle coordinate system. Note that this camera-to-vehicle transformation matrix is fixed for each camera across all timesteps and is relatively easy to obtain in practice, whereas the general world-to-camera extrinsics vary at every timestep and thus are costly to collect. Leveraging the fixed camera-to-vehicle transformation matrix and the camera motion predicted by the pose network, we do not require the world-to-camera extrinsics during training, which is another advantage of our method.

Through spatial and temporal contexts, we compute three photometric losses $\mathcal{L}_{tm}$, $\mathcal{L}_{sp}$ and $\mathcal{L}_{sp-tm}$ for each camera. We also adopt a smoothness loss (Godard, Mac Aodha, and Brostow 2017) that encourages the output depth to be locally smooth. The loss function for scale-aware localization is:

$$\mathcal{L}_{loc} = \mathcal{L}_{tm} + \lambda_{sp}\mathcal{L}_{sp} + \lambda_{sp-tm}\mathcal{L}_{sp-tm} + \lambda_{smooth}\mathcal{L}_{smooth}. \tag{7}$$

By matching the small overlap between spatial and temporal contexts, the model learns scale information during training and predicts real-scale depth for single-frame input during inference, ensuring consistency across multi-frame inputs.

Utilizing the real scale depth map of each input image I_i, we obtain the position of Gaussian primitives μ_i by unprojecting the multi-view consistent depth D_i into 3D space:

$$\mu_i = \Pi^{-1}(I_i, D_i), \tag{8}$$

where Π is the projection matrix with camera intrinsics K_i and camera-to-vehicle transformation matrix E_i. Hence, the scale-aware localization for Gaussian primitives is achieved.

3.3 Gaussian Parameters Prediction from Single Image

In our DrivingForward, each Gaussian primitive is parameterized by properties $\{\mu, \Sigma, \alpha, c\}$ following the original 3DGS (Kerbl et al. 2023), where the covariance matrix Σ can be decomposed into a scaling factor s and a rotation quaternion r. We obtain the $\mu \in \mathbb{R}^3$ through scale-aware localization. Then we need to predict the other parameters, including scaling factor $s \in \mathbb{R}_+^3$, rotation quaternion $r \in \mathbb{R}^4$, opacity $\alpha \in [0, 1]$ and color $c \in \mathbb{R}^k$ that represented by k degree spherical harmonics.

Unlike pixelSplat (Charatan et al. 2024) and MVSplat (Chen et al. 2024) that rely on paired views to predict Gaussian parameters, we propose a Gaussian network to independently predict these parameters from each image and aggregate the Gaussian primitives across all images, which is more suitable for driving scenes with small overlap between sparse vehicle-mounted cameras. To ensure the predicted Gaussian parameters are consistent across all input views, we utilize the previous scale-aware depth and image feature from the depth network as input for the Gaussian network. Since the previous depth network learns scale information from spatial and temporal context images, we argue that the scale-aware depth and image feature can enhance the multi-view consistency of the remaining Gaussian parameters.

The Gaussian network $\mathcal{G}$ consists of a depth encoder $\mathcal{E}$, a feature fusion decoder $\mathcal{D}$, and four prediction heads $\mathcal{H}_s$, $\mathcal{H}_r$, $\mathcal{H}_\alpha$, $\mathcal{H}_c$ for scaling factor s, rotation quaternion r, opacity α, and color c, respectively.

Given the output estimated depth map D_i of input image I_i, the depth encoder $\mathcal{E}$ of the Gaussian network extracts the depth feature F_{depth}:

$$F_{depth} = \mathcal{E}(D_i), \tag{9}$$

where F_{depth} contains the 3D geometric information for 2D pixels. Then the fusion decoder $\mathcal{D}$ combined F_{depth} with the image feature F_{image} from depth network encoder:

$$F_{fusion} = \mathcal{D}(Concat(F_{depth}, F_{image})), \tag{10}$$

where $Concat$ indicates the concat operation at each feature level and F_{fusion} is the fusion feature for Gaussian parameters prediction. The prediction heads simply consist of two convolutions to effectively predict parameters from the fusion feature. We also apply softplus and softmax activation on $\mathcal{H}_s$ and $\mathcal{H}_\alpha$ to ensure $s \in \mathbb{R}^3_+$ and $\alpha \in [0, 1]$. The parameters are obtained by:

$$p = \mathcal{H}_p(F_{fusion}), \quad p \in \{s, r, \alpha, c\}. \tag{11}$$

Since the Gaussian network predicts Gaussian parameters from one single frame of surround-view images without relying on additional frames, the single-frame-based prediction supports flexible single-frame or multi-frame inputs of surrounding views, rather than being restricted to fixed inputs like paired images from two adjacent frames.

3.4 Joint Training Strategy

By applying scale-aware localization and Gaussian parameters prediction to each input view, we obtain the Gaussian primitives for all images. These primitives are then aggregated into 3D space to form a 3D representation. Novel view synthesis can be achieved through the splat-based rasterization rendering in 3DGS (Kerbl et al. 2023).

We jointly train the full model, including the depth network, the pose network, and the Gaussian network. For warp operation of the depth and pose network, we use spatial transformer network (Jaderberg et al. 2015) to sample the synthesis image from the source image, which is fully differentiable (Godard, Mac Aodha, and Brostow 2017). For rendering novel views after obtaining the Gaussian primitives in 3D space, the splat-based rasterization rendering is also fully differentiable. These two operations along with other differentiable parts enable the joint training end-to-end.

We fuse image features from the depth network into the Gaussian network. This shared feature connects the scale-aware positions with predictions of other Gaussian parameters, allowing the Gaussian network to leverage scale information from temporal and spatial contexts. Additionally, it promotes the convergence of the full model.

The Gaussian network is supervised on a linear combination of L2 and LPIPS loss (Zhang et al. 2018) between a novel view ground truth I_{gt} and the rendering image I_{render}:

$$\mathcal{L}_{render} = \beta L_2 + \gamma L_{LPIPS}, \tag{12}$$

with $\beta = 1$ and $\gamma = 0.05$. In practical training, we take the adjacent frame of input images as the novel view. The final loss function for the full model is:

$$\mathcal{L} = \mathcal{L}_{loc} + \lambda_{render}\mathcal{L}_{render}. \tag{13}$$

Through the joint training strategy, we achieve scale-aware localization and Gaussian parameters prediction in one stage and support flexible multi-frame inputs, as the prediction independently depends on each frame of surrounding views.

4 Experiments

4.1 Experimental Setup

Implementation Details We implement DrivingForward with PyTorch and use a pre-built 3DGS renderer (Kerbl et al. 2023). We use ResNet-18 as the encoder for both the depth and pose network and a U-Net encoder for the Gaussian network. The prediction heads of the Gaussian network consist of two convolution layers. We also integrate a feature aggregate module (Kim et al. 2022) into the depth and pose network to enhance feature representation. We use an Adam optimizer with a learning rate of 1×10^{-4}, a batch size of 1 with 6 surround-view images as one sample. During training, we use images from the spatially adjacent left and right cameras and the temporally adjacent frames for spatial and temporal contexts. We set $\lambda_{render} = 0.01$, $\lambda_{sp} = 0.03$, $\lambda_{sp-tm} = 0.1$, and $\lambda_{smooth} = 0.001$ in the loss function and train our model for 10 epochs. All experiments are conducted on a single A6000 GPU with 48GB.

Datasets and Metrics The nuScenes dataset (Caesar et al. 2020) is a public large-scale dataset for autonomous driving. It contains 1000 driving scenes from different geographic areas. Each scene comprises sequential frames of approximately 20 seconds, and the entire dataset encompasses around 40,000 keyframes. The images are captured by 6 vehicle-mounted cameras that cover the surrounding views. The overlap between adjacent camera images is minimal, at approximately 10%. In our experiments, we use the default split of 700 scenes for training and 150 scenes for validation. Unless otherwise specified, we adopt a default resolution of 352 × 640. To evaluate the quality of reconstructed scenes, we synthesize novel views for frames adjacent to the input frame images and compute the peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) (Wang et al. 2004), and perceptual distance (LPIPS) (Zhang et al. 2018).

Baselines Since our method is among the first to explore the real-time reconstruction of driving scenes, no existing benchmarks are available. Therefore, we defined two modes of novel view synthesis to accommodate different comparison methods. The first is the **single-frame (SF) mode**, where given one single frame of surrounding views at timestep t, the goal is to synthesize the next frame's surround-view images at timestep $t + 1$. The other mode is the **multi-frame**

Method	Venue&Year	Mode	Resolution	**PSNR↑**	**SSIM↑**	**LPIPS↓**
MVSplat	ECCV 2024	MF	352 × 640	22.83	0.629	0.317
pixelSplat	CVPR 2024	MF	352 × 640	25.00	0.727	0.298
Ours	AAAI 2025	MF	352 × 640	**26.06**	**0.781**	**0.215**
UniPad	CVPR 2024	SF	114 × 228	16.45	0.375	-
SelfOcc	CVPR 2024	SF	114 × 228	18.22	0.464	-
EmerNeRF	ICLR 2024	SF	114 × 228	20.95	0.585	-
DistillNeRF	NeurIPS 2024	SF	114 × 228	20.78	0.590	-
Ours	AAAI 2025	SF	352 × 640	21.67	0.727	0.259
Ours	AAAI 2025	SF	114 × 228	**21.76**	**0.767**	**0.194**

Table 1: Comparison of our method against other feed-forward methods in both MF and SF mode.

(MF) mode, where given two surround-view images of interval frames, i.e. the surround-view images at timestep $t-1$ and $t+1$, the goal is to synthesize the intermediate surround-view images at the timestep t. Using the two modes of novel view synthesis, we compare our method against both feed-forward and scene-optimized reconstruction methods.

For feed-forward methods, we compare our model with MVSplat (Chen et al. 2024), pixelSplat (Charatan et al. 2024), and DistllNeRF (Wang et al. 2024) (along with its comparison methods EmerNeRF (Yang et al. 2024b), Unipad (Yang et al. 2024a), and SelfOcc (Huang et al. 2024b)). MVSplat and pixelSplat are designed for training on datasets with densely overlapping inputs. Given that temporally adjacent frames have significantly more overlap than spatially adjacent frames, we use MF mode to meet their overlapping requirements. We retrained them by combining their publicly available codebases with our dataset and data loader. DistillNeRF does not release the code and cannot be trained. To enable a fair comparison, we align our method with the settings specified in its paper, including input, rendered novel view, image resolution, and validation scenes. Since SF mode aligns with DistillNeRF's input and novel views, we train our model using this mode.

For scene-optimized reconstruction methods, we compare our trained method to the original 3DGS (Kerbl et al. 2023) using the SF mode. Note that our method does not require test-time optimization while 3DGS needs to be optimized scene by scene. We conduct this comparison to demonstrate that our method can achieve comparable or even higher reconstruction quality without test-time optimization.

4.2 Comparison with Feed-forward Methods

We report the quantitative results of comparison with state-of-the-art feed-forward methods in Table 1. Although we adapt our method to align with the different settings of the baselines, we outperform them across all metrics under the corresponding configuration. The qualitative comparisons of MVSplat, pixelSplat, and our method are visualized in Figure 3. Our DrivingForward achieves the highest quality on novel view results even for challenging details, such as the traffic sign in the front left view and the monument with words in the back right view. Other methods show obvious artifacts in these regions, while our method synthesizes clear novel views without such artifacts.

Method	Test Time (per scene)	PSNR↑	SSIM↑	LPIPS↓
3DGS	≈ 9 min	19.57	0.599	0.465
Ours	0.29 s	21.84	0.758	0.181

Table 2: Comparison of our feed-forward method against scene-optimized 3DGS in SF mode.

Method	Time		Memory	
	Training (total)	Inference (per scene)	Training	Inference
MVSplat	≈ 4 days	1.39 s	35.3 GB	5.45 GB
pixelSplat	≈ 6 days	2.95 s	45.4 GB	11.3 GB
Ours	≈ 3 days	0.63 s	40.0 GB	7.58 GB

Table 3: Comparison of our method against MVSplat and pixelSplat on runtime and GPU memory usage in MF mode.

4.3 Comparison with Scene-optimized Methods

We compare our feed-forward method against the original 3DGS which represents the scene-optimized methods. In SF mode, we train our model and select the first three scenes from the validation set. Then we optimize the 3DGS model for each scene individually and the rendered novel view images of 3DGS models are compared with ours. The average test time pre-scene and metrics across the three scenes are reported in Table 2. 3DGS takes several minutes to synthesize the novel views of a scene (6 images). In contrast, our feed-forward method completes this within half a second and achieves higher reconstruction quality without the costly test-time optimization.

4.4 Comparison on Runtime and Memory

We compare the runtime and GPU memory usage of our method against MVSplat and pixelSplat under the MF mode. As shown in Table 3, our method requires less training time and achieves faster inference speed in synthesizing novel surrounding views that are composed of 6 images with a resolution of 352 × 640. We also report the GPU memory usage in practical training and inference. Note that our GPU memory usage is not the lowest as we use two frames of

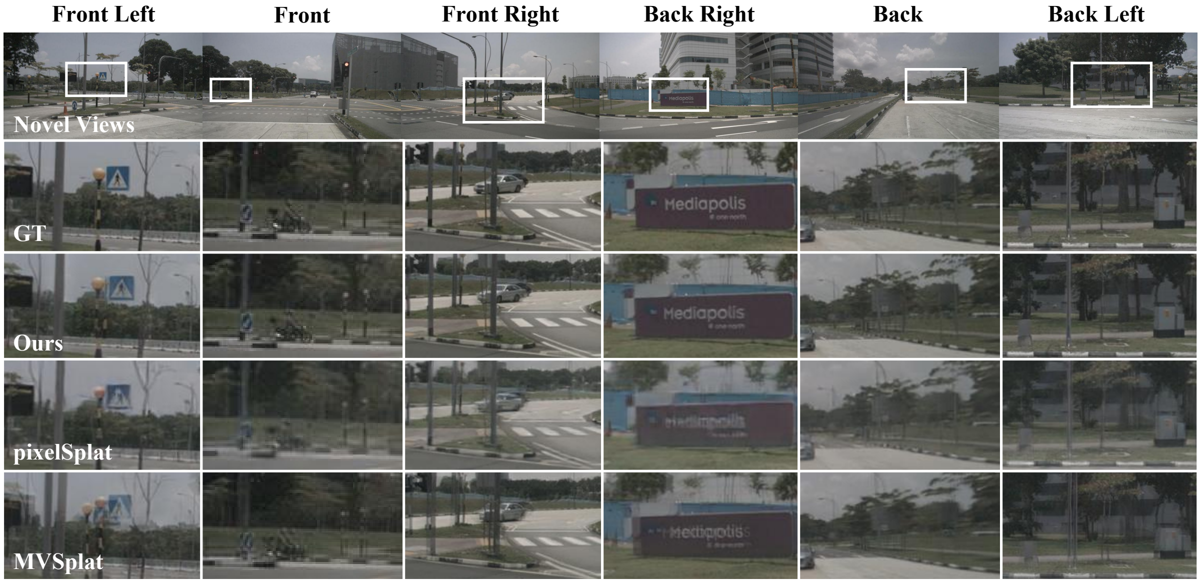

Figure 3: Qualitative results of novel surrounding views. Details from surrounding views are present for easy comparison.

Model	PSNR↑	SSIM↑	LPIPS↓
Full Model	21.67	0.727	0.259
w/o Joint Training	21.50	0.726	0.261
w/o Image Feature Share	19.48	0.699	0.314
w/o Scale-aware Loc.	11.96	0.466	0.772

Table 4: Ablation studies. All ablation models are based on our full model by removing the corresponding module.

surrounding images in one batch, while MVSplat and pixelSplat use two images in one batch. The difference is due to different training architectures. Despite this, our memory usage remains comparable to that of other methods.

4.5 Ablation Studies

We conduct ablation experiments to demonstrate the effectiveness of our method in detail. We train our model using the single-frame (SF) mode of novel view synthesis and refer to it as “Full Model” in Table 4.

To validate the effectiveness of the joint training strategy, we first train the scale-aware localization module, which includes only the depth and pose networks. We then use this pre-trained model to train the Gaussian network (2nd row, w/o Joint Training). The result in Table 4 shows that joint training enables end-to-end training without decreasing performance and even gains a better performance. This indicates that the joint training is able to facilitate model convergence and can lead to improvement.

In addition to the estimated depth map, the shared image feature from the depth network is another key component for connecting to the subsequent Gaussian network. To demonstrate the importance of this connection, we ablate the shared image feature and instead construct another feature extractor like the depth network encoder for the Gaussian network. As illustrated in Table 4, the model “w/o Image Feature Share” (3rd row) drops more than 2 dB PSNR. This highlights the importance of the shared image feature in connecting the positions and other parameters of Gaussian primitives.

To investigate whether the scale-aware localization is necessary, we replace the scale-aware localization with a monocular depth estimation model (Godard et al. 2019) (4th row, w/o Scale-aware Loc.), as shown in Table 4. Although monocular depth estimation models can estimate depth from a single image, they are scale-ambiguous, which indicates the depth maps are inconsistent across different views and lead to a large drop in performance. This demonstrates the irreplaceable role of scale-aware localization.

5 Conclusion

We introduce DrivingForward, a feed-forward Gaussian Splatting model that achieves real-time driving scene reconstruction from flexible surround-view input. To solve the problem of minimal overlap of the surrounding views, we predict the Gaussian primitives from each image independently and propose the scale-aware localization to obtain the multi-view consistent position for Gaussian primitives. A Gaussian network predicts primitives’ other parameters from each image. The individual prediction enables flexible multi-frame inputs of surrounding views. Our method does not require depth ground truth and is extrinsic-free during training. At the inference stage, our model is faster than other methods and achieves higher reconstruction quality for driving scenes compared with both existing feed-forward and scene-optimized reconstruction methods. In the future, this work is expected to integrate with human perception research (Guo et al. 2024) to develop a more intelligent human-in-scene perception system.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (62302167, U23A20343, 62222602, 62176092, 62472282, 72192821); in part by Shanghai Sailing Program (23YF1410500); in part by Chenguang Program of Shanghai Education Development Foundation and Shanghai Municipal Education Commission (23CGA34); in part by the Fundamental Research Funds for the Central Universities (YG2023QNA35); in part by YuCaiKe [2023] (231111310300).

References

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11621–11631.

Cai, Q.; Pan, Y.; Yao, T.; Ngo, C.-W.; and Mei, T. 2023. Objectfusion: Multi-modal 3d object detection with object-centric fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 18067–18076.

Charatan, D.; Li, S. L.; Tagliasacchi, A.; and Sitzmann, V. 2024. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19457–19467.

Chen, X.; Zhang, T.; Wang, Y.; Wang, Y.; and Zhao, H. 2023. Futr3d: A unified sensor fusion framework for 3d detection. In *proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 172–181.

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.*

Du, Y.; Smith, C.; Tewari, A.; and Sitzmann, V. 2023. Learning to render novel views from wide-baseline stereo pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4970–4980.

Godard, C.; Mac Aodha, O.; and Brostow, G. J. 2017. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 270–279.

Godard, C.; Mac Aodha, O.; Firman, M.; and Brostow, G. J. 2019. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3828–3838.

Guizilini, V.; Vasiljevic, I.; Ambrus, R.; Shakhnarovich, G.; and Gaidon, A. 2022. Full surround monodepth from multiple cameras. *IEEE Robotics and Automation Letters (RAL)*, 7(2): 5397–5404.

Guo, D.; Li, K.; Hu, B.; Zhang, Y.; and Wang, M. 2024. Benchmarking micro-action recognition: dataset, method, and application. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 34(7): 6238–6252.

Guo, J.; Deng, N.; Li, X.; Bai, Y.; Shi, B.; Wang, C.; Ding, C.; Wang, D.; and Li, Y. 2023. Streetsurf: Extending multi-view implicit surface reconstruction to street views. *arXiv preprint arXiv:2306.04988.*

Huang, N.; Wei, X.; Zheng, W.; An, P.; Lu, M.; Zhan, W.; Tomizuka, M.; Keutzer, K.; and Zhang, S. 2024a. S3Gaussian: Self-Supervised Street Gaussians for Autonomous Driving. *arXiv preprint arXiv:2405.20323.*

Huang, Y.; Zheng, W.; Zhang, B.; Zhou, J.; and Lu, J. 2024b. Selfocc: Self-supervised vision-based 3d occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19946–19956.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial transformer networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 28.

Ji, Y.; Zhu, H.; Tang, J.; Liu, W.; Zhang, Z.; Xie, Y.; and Tan, X. 2024. FastLGS: Speeding up Language Embedded Gaussians with Feature Grid Mapping. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4): 139:1–139:14.

Khan, M.; Fazlali, H.; Sharma, D.; Cao, T.; Bai, D.; Ren, Y.; and Liu, B. 2024. AutoSplat: Constrained Gaussian Splatting for Autonomous Driving Scene Reconstruction. *arXiv preprint arXiv:2407.02598.*

Kim, J.-H.; Hur, J.; Nguyen, T. P.; and Jeong, S.-G. 2022. Self-supervised surround-view depth estimation with volumetric feature fusion. *Advances in Neural Information Processing Systems (NeurIPS)*, 35: 4032–4045.

Li, J.; Zhang, J.; Bai, X.; Zheng, J.; Ning, X.; Zhou, J.; and Gu, L. 2024. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20775–20785.

Li, Q.; Wang, Y.; Wang, Y.; and Zhao, H. 2022a. HDMapNet: An Online HD Map Construction and Evaluation Framework. In *International Conference on Robotics and Automation (ICRA)*, 4628–4634.

Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Qiao, Y.; and Dai, J. 2022b. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European Conference on Computer Vision (ECCV)*, 1–18.

Liang, T.; Xie, H.; Yu, K.; Xia, Z.; Lin, Z.; Wang, Y.; Tang, T.; Wang, B.; and Tang, Z. 2022. Bevfusion: A simple and robust lidar-camera fusion framework. *Advances in Neural Information Processing Systems (NeurIPS)*, 35: 10421–10434.

Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D. L.; and Han, S. 2023. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2774–2781.

Ma, Q.; Tan, X.; Qu, Y.; Ma, L.; Zhang, Z.; and Xie, Y. 2024. Cotr: Compact occupancy transformer for vision-based 3d occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19936–19945.

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.

Suhail, M.; Esteves, C.; Sigal, L.; and Makadia, A. 2022. Generalizable patch-based neural rendering. In *European Conference on Computer Vision (ECCV)*, 156–174.

Sun, T.; Zhang, Z.; Tan, X.; Peng, Y.; Qu, Y.; and Xie, Y. 2024a. Uni-to-Multi Modal Knowledge Distillation for Bidirectional LiDAR-Camera Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 46(12): 11059–11072.

Sun, T.; Zhang, Z.; Tan, X.; Qu, Y.; and Xie, Y. 2024b. Image understands point cloud: Weakly supervised 3D semantic segmentation via association learning. *IEEE Transactions on Image Processing (TIP)*, 33: 1838–1852.

Szymanowicz, S.; Rupprecht, C.; and Vedaldi, A. 2024. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10208–10217.

Tan, X.; Ma, Q.; Gong, J.; Xu, J.; Zhang, Z.; Song, H.; Qu, Y.; Xie, Y.; and Ma, L. 2023. Positive-negative receptive field reasoning for omni-supervised 3d segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(12): 15328–15344.

Wang, L.; Kim, S. W.; Yang, J.; Yu, C.; Ivanovic, B.; Waslander, S. L.; Wang, Y.; Fidler, S.; Pavone, M.; and Karkus, P. 2024. DistillNeRF: Perceiving 3D Scenes from Single-Glance Images by Distilling Neural Fields and Foundation Model Features. *Advances in Neural Information Processing Systems (NeurIPS)*.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 13(4): 600–612.

Xiong, H.; Muttukuru, S.; Upadhyay, R.; Chari, P.; and Kadambi, A. 2023. Sparsegs: Real-time 360 ° sparse view synthesis using gaussian splatting. *arXiv preprint arXiv:2312.00206*.

Yan, Y.; Lin, H.; Zhou, C.; Wang, W.; Sun, H.; Zhan, K.; Lang, X.; Zhou, X.; and Peng, S. 2024. Street Gaussians: Modeling Dynamic Urban Scenes with Gaussian Splatting. In *European Conference on Computer Vision (ECCV)*, volume 15131, 156–173.

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 (CVPR)*, 15238–15250.

Yang, J.; Ivanovic, B.; Litany, O.; Weng, X.; Kim, S. W.; Li, B.; Che, T.; Xu, D.; Fidler, S.; Pavone, M.; and Wang, Y. 2024b. EmerNeRF: Emergent spatial-temporal scene decomposition via self-supervision. In *International Conference on Learning Representations (ICLR)*.

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 (CVPR)*, 4578–4587.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 586–595.

Zheng, S.; Zhou, B.; Shao, R.; Liu, B.; Zhang, S.; Nie, L.; and Liu, Y. 2024. Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19680–19690.

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 (CVPR)*, 21634–21643.