DenoiseGS: Delta-Based 3D Gaussian Splatting with B-spline Trajectory Optimization for Dynamic Driving Scene Reconstruction

Junjie Linghu, Qiang Ling*

Dept. of Automation, University of Science and Technology of China, Hefei 230027, China
lhjj@mail.ustc.edu.cn, qling@ustc.edu.cn

Abstract

3D street scene reconstruction is a challenging yet crucial task for autonomous driving. Many reconstruction methods often overlook two key limitations for high-quality driving scene reconstruction, sensitivity to camera parameter noise from high-speed vehicles and heavy reliance on precise dynamic object annotations of datasets. To resolve these issues, we propose **DenoiseGS**, a simple yet effective approach based on explicit 3D Gaussian splatting. Specifically, we propose a novel learnable ***Delta*** attribute per Gaussian primitive that operates on the image plane during rasterization to mitigate the impact of noisy camera parameters through modulating the inputs of the α-blending process. To enhance the representation of this ***Delta*** attribute, we propose a **DeltaEstimator** that encodes viewing direction and contextual cues to facilitate view dependence. We also extend additional CUDA operations to enable efficient gradient update for the delta attribute. Furthermore, to overcome the limitation of inaccurate annotations for dynamic objects, we propose a learnable **B-spline trajectory optimization** with few control points to model the trajectory of a moving object. Comprehensive experiments conducted on nuScenes and Waymo Open Dataset demonstrate that our DenoiseGS outperforms some state-of-the-art methods across all metrics of both reconstruction quality and novel view synthesis.

Introduction

Autonomous driving has garnered increasing attention owing to its great potential for future development (Caesar et al. 2021; Hu et al. 2023). As a data-driven task, it requires diverse driving scene data to support downstream perception tasks (Liu et al. 2022; Li et al. 2022). High-fidelity scene reconstruction offers an effective avenue for closed-loop evaluation, which enables the simulations of critical corner cases to enhance the safety of end-to-end autonomous driving systems.

Over the past few years, Neural Radiance Fields (NeRF) technology (Mildenhall et al. 2021), employing Multi-Layer Perceptrons (MLPs) for implicit scene reconstruction and novel view synthesis, has achieved remarkable success (Yang et al. 2024a; Wu et al. 2023). Compared to NeRF-based methods, 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) can significantly improve rendering quality and speed by modeling scenes with explicit Gaussian primitives instead of implicit neural networks. Original 3DGS is proficient in modeling bounded static scenes, but encounters difficulty in handling dynamic objects for dynamic driving scenes. This challenge is particularly serious when reconstructing dynamic scenes from sparse views acquired by high-velocity vehicles (Sun et al. 2020; Caesar et al. 2020). To address this challenge, several notable methods have been proposed to decompose autonomous driving scenes into dynamic objects and a static background based on calibrated camera parameters and manually annotated bounding boxes. Among them, DrivingGaussian (Zhou et al. 2024b) utilizes a composite dynamic gaussian graph to model dynamic objects with explicit scene decomposition, HUGS (Zhou et al. 2024a) proposes physical constraints to regularize the pose of dynamic objects and introduces additional semantic and optical flow supervision.

However, the above mentioned methods require precise and static camera parameters. They often overlook the impact of common sensor noise, such as camera shakeness caused by car bumps or camera offset due to external environmental forces (Liu et al. 2023). Although Gaussian primitives can adjust their centers via learnable attributes, their projection onto the image plane during the rasterization process still suffers from incorrect deviation due to perturbed camera parameters. Moreover, they may be hindered by manual pose annotations of dynamic objects during the training stage. This issue injects errors into the transformation of dynamic objects from the canonical space to the world space for background composition, leading to significant degradation of reconstruction and editing performance within dynamic driving scenes.

To address the aforementioned challenges, we propose **DenoiseGS**, a novel Delta-based 3D Gaussian Splatting framework with B-spline trajectory optimization to simulate autonomous driving scenes. We utilize distinct strategies tailored to the specific noise sources of the static background and dynamic objects. To mitigate camera parameter perturbation induced by high-speed vehicle motion, we propose a novel learnable delta attribute $\boldsymbol{\mathcal{D}}_i \in \mathbb{R}^2$ for each Gaussian primitive. Then it is coupled with a proposed **DeltaEstimator** to generate positional offsets $\boldsymbol{\mathcal{D}}_i^{\pi} \in \mathbb{R}^2$ of the projected center on the image plane. They can enable effective

*Corresponding author

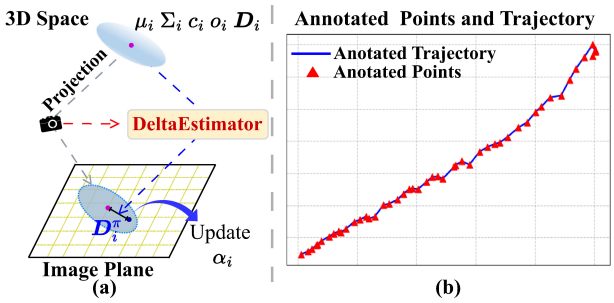

Figure 1: (a). Illustration of the working mechanism of the positional offsets $\boldsymbol{D}_i^\pi$. (b). An annotated dynamic vehicle trajectory of nuScenes dataset. We can observe that manual annotations are often subject to errors.

compensation for noisy disturbances by adaptively modulating the α-blending process during the rasterization, as illustrated in Figure 1 (a). Specifically, the DeltaEstimator encodes viewing direction vector from the camera's optical center to the center of the Gaussian primitive. It also contains the Gaussian primitive's color information to establish the implicit semantic consistency. Furthermore, to enable efficient gradient updates for delta attribute $\boldsymbol{\mathcal{D}}_i$ and $\boldsymbol{\mathcal{D}}_i^\pi$, we extend additional CUDA operations for rasterization to facilitate both forward and backward propagation.

For reconstruction of dynamic objects of the driving scene, noise mainly results from inaccurate manual annotations. We can observe that annotated trajectories are often discontinuous and suffer from annotation errors, as illustrated in Figure 1 (b). To overcome this difficulty, we propose a learnable **B-spline trajectory optimization** strategy to model the motion trajectories of dynamic objects. The high flexibility of learnable B-spline curves enables vehicle trajectory fitting with fewer control points, significantly mitigating the impact of annotation errors on dynamic object reconstruction.

Our main contributions can be summarized as follows.

- We propose DenoiseGS, a framework designed to effectively reduce noise in dynamic scene reconstruction and enable high-fidelity reconstruction and novel view synthesis.
- We introduce a novel learnable delta attribute and a dedicated DeltaEstimator to alleviate the impact of noisy camera parameters by adaptively modulating the α-blending process.
- We propose a learnable B-spline trajectory optimization strategy for moving dynamic objects, significantly mitigating manual annotation errors during training.
- Comprehensive experiments conducted on the nuScenes and Waymo Open Dataset demonstrate that our DenoiseGS outperforms previous methods in reconstructing dynamic driving scenes and synthesizing novel views.

Related Work

NeRF for Urban Scene Reconstruction

Reconstructing static or dynamic scenes is a crucial computer vision task (Liu et al. 2024; Sun et al. 2024; Li et al. 2023). To accomplish that task, many methods have been proposed. Neural Radiance Fields (NeRF) (Mildenhall et al. 2021) has garnered much attention in the field of neural rendering, introducing Multi-Layer Perceptrons (MLPs) to reconstruct 3D environments and generate novel perspectives from collections of 2D images and associated camera pose data. Plenoxels (Fridovich-Keil et al. 2022) introduces the concept of the panoptic voxel and proposes to store spherical harmonics representations of the scene within a 3D sparse grid. This representation (Fridovich-Keil et al. 2022) can be optimized by calibrated images without any neural components, and significantly improve the rendering speed. Guo (Guo et al. 2024) proposes the integration of NeRF volumetric rendering with point clouds and can effectively solve the region missing problem of single views and mitigate the impact of noise. EmerNeRF (Yang et al. 2024a) parametrizes a flow field induced by dynamic scenes and utilizes this flow field to aggregate multi-frame features and improve the rendering accuracy of dynamic objects.

3D Gaussian Splatting for Driving Scenes

As a breakthrough technique, 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) explicitly models static 3D scenes using numerous 3D Gaussian ellipsoids, achieving rapid reconstruction via CUDA parallel computing. Real-time splat-based rasterization and the efficient optimization of anisotropic 3D Gaussian ellipsoids are crucial for the high performance of 3DGS. However, the original 3DGS is limited to static scenes. It has been further advanced to accommodate dynamic scenes. SplineGS (Yoon et al. 2025) utilizes a new motion adaptive spline to model the deformation of each dynamic 3D Gaussian. Nevertheless, these methods are still inapplicable for autonomous driving scenarios, as they are incapable of handling urban scenes with multiple dynamic objects when the camera perspective continuously changes.

Recently, several 3DGS-based methods have emerged to deal with the complex task of reconstructing dynamic urban scenes. PVG (Chen et al. 2023) seamlessly integrates both static and dynamic elements of a scene using a single formula, capturing explicit motion attributes through periodic vibration Gaussian. HUGS (Zhou et al. 2024a) not only utilizes 2D optical flow and semantic information to guide the optimization of 3D Gaussian ellipsoids, but also employs a unicycle model to simulate the motion of dynamic objects, thereby achieving remarkable performance improvement. DrivingGaussian (Zhou et al. 2024b), OmniRe(Chen et al. 2024), StreetGaussian (Yan et al. 2024), Recondreamer++ (Zhao et al. 2025) all decompose urban scenes into static background and dynamic foreground. Furthermore, they utilize LiDAR data for the initialization of the background and foreground.

However, these methods rarely pay attention to the impact of noise caused by camera parameters and manual annota-

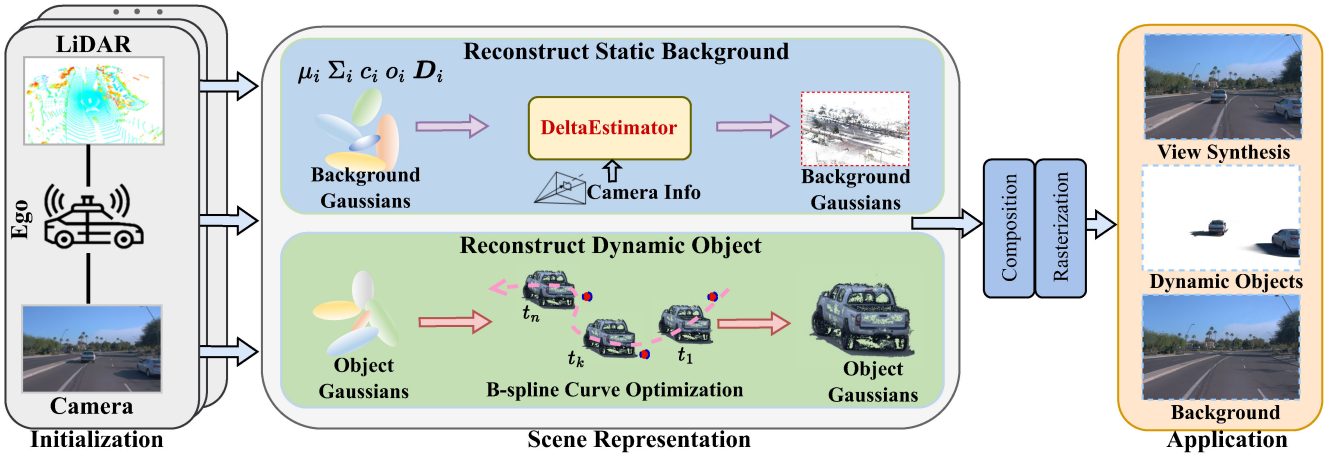

Figure 2: **The overview of the proposed DenoiseGS. Left:** DenoiseGS integrates a series of multi-modal data, including multi-camera images and LiDAR point clouds. **Middle:** We decompose dynamic driving scenes into two components: dynamic objects and the static background. The reconstructed multiple dynamic objects are optimized by learnable B-spline curves. The static background is refined by the proposed delta attribute $\mathcal{D}_i$ and DeltaEstimator. Then the static background and dynamic objects are integrated with a sky model for composition and rasterization. **Right:** Our DenoiseGS achieves outstanding reconstruction and scene simulation performance.

tion errors. To resolve those issue, our method proposes a novel attribute $\mathcal{D}_i$ of Gaussian primitive to model the positional offset $\mathcal{D}_i^{\pi}$ on the image plane, which can effectively reduce the impact of camera parameter noise. Besides, we propose a learnable B-spline trajectory optimization to fit the trajectories of moving objects, which can attenuate the manual annotation errors.

Methodology

Preliminaries

3D Gaussian Splatting 3DGS (Kerbl et al. 2023) introduces an explicit rendering method to represent 3D scenes based on a set of 3D points. It achieves rapid and efficient rendering through tile-based rasterization. Each 3D Gaussian primitive is characterized by a set of attributes, $\{\boldsymbol{\mu}_i, \boldsymbol{s}_i, \boldsymbol{q}_i, \boldsymbol{c}_i, o_i\}$. Anisotropic 3D Gaussian is defined as

$$G_i(x) = e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)}, \tag{1}$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^3$ is the mean position of Gaussian primitives and $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$ is the covariance matrix. To simplify the optimization, the covariance matrix $\boldsymbol{\Sigma}_i$ is calculated from learnable scaling $\boldsymbol{s}_i \in \mathbb{R}^{3\times1}$ and rotation $\boldsymbol{q}_i \in \mathbb{R}^{4\times1}$ as

$$\boldsymbol{\Sigma}_i = \boldsymbol{q}_i \boldsymbol{s}_i \boldsymbol{s}_i^T \boldsymbol{q}_i^T. \tag{2}$$

The final color C of a pixel in the rendered image is computed by α-blending based on the contributions of N overlapping Gaussian primitives,

$$C = \sum_{i=1}^{N} \mathbf{c}_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \tag{3}$$

where $\mathbf{c}_i \in \mathbb{R}^{3\times(k+1)^2}$ is the color attribute of the i-th Gaussian, and k is the predefined maximum order of spherical harmonic functions. α_i is calculated with the corresponding pixel coordinates $\boldsymbol{x}_i^{\mathbf{2d}}$ in the image domain and the 2D covariance matrix $\boldsymbol{\Sigma}^{\mathbf{2d}}{}_i$ as

$$\begin{aligned}\alpha_i &= o_i G(\boldsymbol{\Delta}_i) \\ &= o_i \exp\left(-\frac{1}{2}(\boldsymbol{x}_i^{\mathbf{2d}} - \boldsymbol{\mu}_i^{\mathbf{2d}})^T \left(\boldsymbol{\Sigma}^{\mathbf{2d}}{}_i\right)^{-1} (\boldsymbol{x}_i^{\mathbf{2d}} - \boldsymbol{\mu}_i^{\mathbf{2d}})\right),\end{aligned} \tag{4}$$

where $\boldsymbol{\mu}_i^{\mathbf{2d}}$ is the mean of 2D Gaussian primitives projected by $\boldsymbol{\mu}_i$ in the image domain. The 2D covariance matrix $\boldsymbol{\Sigma}^{\mathbf{2d}}{}_i$ is calculated as (Zwicker et al. 2001),

$$\boldsymbol{\Sigma}^{\mathbf{2d}}{}_i = JW\boldsymbol{\Sigma}_i W^T J^T, \tag{5}$$

where J and W are the Jacobian matrices of affine approximation and viewing transformation, respectively.

The learnable attributes $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \mathbf{c}_i, o_i\}$ are optimized according to the reconstruction loss between rendered and ground-truth images and other effective losses.

B-spline curves As a fundamental concept in computer graphics, B-spline curves (Mortenson 1999) are widely applied in vehicle trajectory fitting (Berglund et al. 2009). Given $n+1$ control points $\{\boldsymbol{p_i}\}_{i=0}^{n}$ and a series of curve segments $\{\boldsymbol{t_j}\}_{j=0}^{n+k+1}$, k-degree B-spline curve can be defined as

$$\boldsymbol{p}(t) = \sum_{i=0}^{n} B_{i,k}(t)\boldsymbol{p_i}, \tag{6}$$

where $\boldsymbol{p}(t)$ represents the position of the curve at a given parameter t, and $\boldsymbol{p_i}$ is the position of the i-th control point. $B_{i,k}(t)$ are the B-spline basis polynomial functions of degree k and defined as

$$B_{i,0}(t) := \begin{cases} 1 & \text{if } t_i \le t < t_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

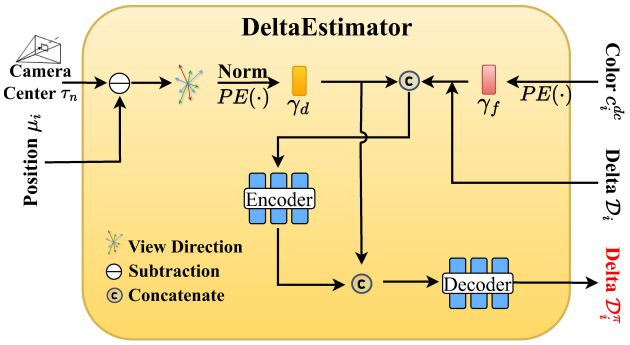

Figure 3: The architecture of the DeltaEstimator

$$B_{i,k}(t) := \frac{t - t_i}{t_{i+k} - t_i} B_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(t) \quad (8)$$

For fitting typical vehicle trajectories, we opt for a cubic B-spline, i.e., $k = 3$. We optimize learnable control points to accurately model the trajectories of dynamic objects in the real world. Given a specific timestamp t, we normalize it to the interval $[0, 1]$ to predict the moving object's position in the world coordinate system.

Delta-based Gaussian Splatting

In the real world, camera parameter noise is a frequent challenge, often stemming from vehicle vibrations on uneven roads or camera shifts due to collisions. These inaccurate parameters directly degrade the precise geometric correspondence between explicit Gaussian primitives and image-plane pixel coordinates. To address this challenge, we introduce a learnable delta attribute $\boldsymbol{\mathcal{D}}_i$ per Gaussian primitive to compensate for color deviation induced by noise. Nevertheless, maintaining constant $\boldsymbol{\mathcal{D}}_i$ for a same Gaussian primitive across different frames after the training is physically implausible. To resolve this issue, we propose the DeltaEstimator, a module that encodes both current camera pose and color features to generate a view-specific positional offset $\boldsymbol{\mathcal{D}}_i^\pi$. Specifically, for each Gaussian primitive, we compute the direction vector from its center $\boldsymbol{\mu_i}$ to the current camera's optical center $\boldsymbol{\tau_n}$. This direction vector is normalized $Norm(\cdot)$ and sinusoidally encoded $PE(\cdot)$ to effectively decouple view-dependent appearance variations, which can be formulated as

$$\gamma_d = PE\left(Norm\left(\boldsymbol{\tau_n} - \boldsymbol{\mu_i}\right)\right). \quad (9)$$

Moreover, it is critical that slight camera parameter noise may not lead to an overly large offset from the projection center, as this would compromise semantic consistency. Inspired by this observation, we introduce color information as an implicit prior to guide the learning of position offsets,

$$\gamma_f = \left[PE\left(\boldsymbol{c}_i^{dc}\right), \boldsymbol{\mathcal{D}}_i\right], \quad (10)$$

where $\boldsymbol{c}_i^{dc}$ is a basic component of spherical harmonics and $[:,:]$ stands for the concatenation. Ultimately, for each Gaussian primitive, we calculate a center offset $\boldsymbol{\mathcal{D}}_i^\pi$ on the image plane with respect to its projection center. $\boldsymbol{\mathcal{D}}_i^\pi$ incorporates the current camera's position and the primitive's inherent color content as

$$\boldsymbol{\mathcal{D}}_i^\pi = \phi_{de}\left(\left[\gamma_d, \phi_{en}\left(\left[\gamma_f, \gamma_d\right]\right)\right]\right), \quad (11)$$

where ϕ_{de} and ϕ_{en} are MLPs. The detailed architecture of DeltaEstimator is provided in Figure. 3.

Our method is significantly different from existing approaches like DeformGS(Yang et al. 2024b), which still correct Gaussian primitive centers within the world coordinate system. Instead, our proposed $\boldsymbol{\mathcal{D}}_i^\pi$ directly operates within the pixel coordinate system during rasterization. It modulates the contribution weights of different Gaussian primitives in the α-blending process, which alleviates rendering artifacts attributed to camera parameters noise, as detailed in **FORWARD PROCESS** of Algorithm 1. As a learnable attribute, $\boldsymbol{\mathcal{D}}_i$ necessitates effective supervision and optimization. We extend additional CUDA operations to adopt the gradient optimization strategy from the original 3DGS(Kerbl et al. 2023), enhancing the back-propagation from $\boldsymbol{\mathcal{D}}_i^\pi$ to further optimize $\boldsymbol{\mathcal{D}}_i$. The specifics of this process are presented in **BACKWARD PROCESS** of Algorithm 1.

Algorithm 1: Overview of CUDA operations for forward and backward passes.

FORWARD PROCESS:
Input: Gaussian $\gamma_k := (\boldsymbol{\mu}_k, \boldsymbol{c}_k, o_k, \boldsymbol{\mathcal{D}}_k^\pi)$, camera π
Output: The color of pixel $C^{'}$.
$\boldsymbol{\mu}_k^\pi, \Sigma_k^\pi, \boldsymbol{p}_k, N \leftarrow$ **splatting**(γ_k, π) ▷ Project γ_k onto image
$\boldsymbol{\mu}_k^{\pi^{'}} \leftarrow \boldsymbol{\mu}_k^\pi + \boldsymbol{\mathcal{D}}_k^\pi$ ▷ Updated center of γ_k
$\boldsymbol{\Delta}_k^{'} \leftarrow \boldsymbol{\mu}_k^{\pi^{'}} - \boldsymbol{p}_k$ ▷ Center-pixel distance
$\alpha_k^{'} \leftarrow o_k G(\boldsymbol{\Delta}_k^{'})$ ▷ Updated α-blending
$C^{'} \leftarrow \sum_{i=1}^{N} \prod_{k=1}^{i-1} (1 - \alpha_k^{'}) \alpha_i^{'} \boldsymbol{c}_i$ ▷ Updated color

BACKWARD PROCESS:
Input: Gaussian γ_k, Loss L, camera π
Output: The gradients of $\boldsymbol{\mu}_k, \boldsymbol{c}_k, o_k, \boldsymbol{\mathcal{D}}_k^\pi$
$\frac{\partial L}{\partial \boldsymbol{c}_k} \leftarrow \frac{\partial L}{\partial C^{'}} \cdot \prod_{j=1}^{k-1} (1 - \alpha_j^{'}) \alpha_k^{'}$ ▷ The gradient of color
$\frac{\partial L}{\partial o_k} \leftarrow \frac{\partial L}{\partial \alpha_k^{'}} \cdot G(\boldsymbol{\Delta}_k^{'})$ ▷ The gradient of opacity
$\frac{\partial L}{\partial \boldsymbol{\mu}_k} \leftarrow \frac{\partial L}{\partial G} \cdot \frac{\partial G}{\partial \boldsymbol{\Delta}_k^{'}} \cdot \frac{\partial \boldsymbol{\Delta}_k^{'}}{\partial \boldsymbol{\mu}_k}$ ▷ The gradient of position
$\frac{\partial L}{\partial \boldsymbol{\mathcal{D}}_k^\pi} \leftarrow \frac{\partial L}{\partial G} \cdot \frac{\partial G}{\partial \boldsymbol{\mu}_k^{\pi^{'}}} \cdot \boldsymbol{I}$ ▷ The gradient of Delta attribute

B-spline Trajectory Optimization

A key limitation of current methods (Yan et al. 2024; Zhou et al. 2024b), which are mainly built upon dynamic object modeling, is their exclusive reliance on manually annotated bounding boxes throughout the entire training stage. To overcome this limitation, we leverage learnable B-spline curves to accurately fit and optimize dynamic object trajectories, which facilitates high-fidelity reconstruction.

Dataset	Methods	Publication	Image Reconstruction PSNR↑	SSIM↑	LPIPS↓	PSNR*↑	Novel View Synthesis PSNR↑	SSIM↑	LPIPS↓	PSNR*↑
nuScenes	DeformGS (Yang et al. 2024b)	CVPR2024	34.55	0.891	0.146	27.70	29.72	0.806	0.162	23.04
	OmniRe (Chen et al. 2024)	ICLR2025	34.37	0.891	0.128	28.11	29.09	0.779	0.174	22.89
	PVG (Chen et al. 2023)	arXiv2023	33.60	0.888	0.175	28.57	29.05	0.804	0.208	24.28
	StreetGS (Yan et al. 2024)	ECCV2024	34.08	0.889	0.114	27.87	30.62	0.841	0.144	24.27
	DenoiseGS (Ours)	-	**34.85**	**0.896**	**0.101**	**29.66**	**32.28**	**0.866**	**0.121**	**26.86**
Waymo	DeformGS (Yang et al. 2024b)	CVPR2024	35.21	0.947	0.087	25.53	31.46	0.913	0.102	22.19
	OmniRe (Chen et al. 2024)	ICLR2025	35.52	0.949	0.078	30.57	31.94	0.909	0.091	26.98
	PVG (Chen et al. 2023)	arXiv2023	34.75	0.947	0.097	31.53	31.35	0.907	0.118	25.49
	StreetGS (Yan et al. 2024)	ECCV2024	37.31	0.958	0.044	34.13	35.88	0.951	0.048	30.56
	DenoiseGS (Ours)	-	**38.04**	**0.960**	**0.041**	**34.90**	**36.36**	**0.953**	**0.045**	**31.04**

Table 1: Quantitative results of novel view synthesis on the Waymo dataset (Sun et al. 2020) and nuScenes dataset (Caesar et al. 2020).We highlight the best results in bold and the second-best results with underlining. The images resolution are 640×960 in Waymo dataset and 450×800 in nuScenes dataset. "PSNR*" denotes the PSNR of dynamic objects.

In order to model the trajectory of an object σ, we utilize a series of learnable control points $\{\boldsymbol{p}_i^\sigma, i = 0, 1, \cdots, m\}$ to control the B-spline curve, which is defined as

$$\boldsymbol{\zeta}(t, \sigma) = \sum_{i=0}^{m} B_{i,k}(t) \boldsymbol{p}_i^\sigma, \tag{12}$$

where $\boldsymbol{\zeta}(t, \sigma)$ represents the center of object σ at normalized timestamp $t \in [0, 1]$ and $B_{i,k}(t)$ is the B-spline basis polynomial function of degree k. To enhance the efficiency of optimization, we utilize the annotated trajectory in the world coordinate system to initialize these control points. Furthermore, we adaptively assign the number of control points for each object σ, which is determined by the number of visible frames of the camera, N. The number of control points m is given as

$$m = \max\left(k + 1, \min\left(N, \left\lfloor \frac{N}{u} \right\rfloor + 1\right)\right) \tag{13}$$

where u is a hyper-parameter that presents the interval of frames.

Using only sparse control points, we achieve physically consistent B-spline trajectories for dynamic objects. A key advantage of B-spline trajectories is their inherent C^{k-1} continuity, which can effectively mitigate noise-induced discontinuity. Furthermore, the object's center at any given timestamp t can be directly computed, eliminating the need for naive interpolation based on the recorded timestamps, as illustrated in Figure 4. We compute the object's center in the world coordinate system at any given time t using the optimized B-spline trajectory. It is then utilized to transform object Gaussian primitives in the object's local coordinate system into the background's coordinate system, which is essential for the subsequent composition and rendering of foreground and background elements.

Loss Function

The final rendering result C is calculated as

$$C = C_G + (1 - O_G)\, C_{sky}, \tag{14}$$

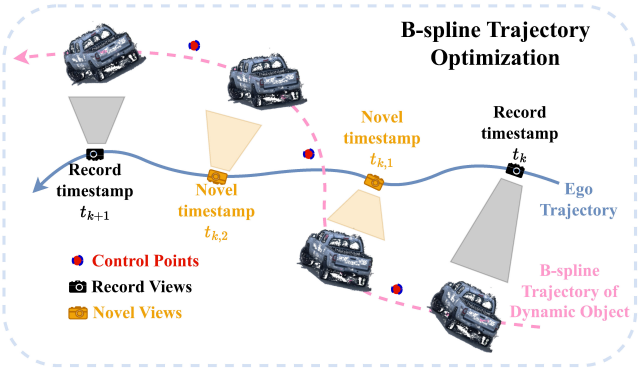

Figure 4: The illustration of the optimized B-spline trajectory of a dynamic object.

where C_G is the rendering result of the static background and dynamic objects, C_{sky} is the sky color calculated by a learnable cubemap and O_G is the rendered opacity mask.

We jointly optimize our scene representation using the following loss function,

$$\begin{aligned}\mathcal{L} =& (1 - \lambda_r)\mathcal{L}_1 + \lambda_r \mathcal{L}_{ssim} \\ &+ \lambda_{sky}\mathcal{L}_{sky} + \lambda_d \mathcal{L}_d + \lambda_\delta \mathcal{L}_\delta,\end{aligned} \tag{15}$$

where $\mathcal{L}_1$ and $\mathcal{L}_{ssim}$ are the L1 and SSIM losses, respectively, $\mathcal{L}_{sky}$ is a binary cross-entropy loss for sky region obtained from Segformer (Xie et al. 2021) and the depth loss $\mathcal{L}_d$ is supervised by sparse ground truth depth value projected from LiDAR points.

Delta Constraint Loss. For efficient convergence during training, we introduce a constraint loss $\mathcal{L}_\delta$ to encourage $\boldsymbol{\mathcal{D}}_i \in \mathbb{R}^2$ to remain within an effective range, which can be formulated as

$$\mathcal{L}_\delta = \frac{1}{2N} \sum_{i=1}^{N} \max\left(0, |\boldsymbol{\mathcal{D}}_i| - \delta\right), \tag{16}$$

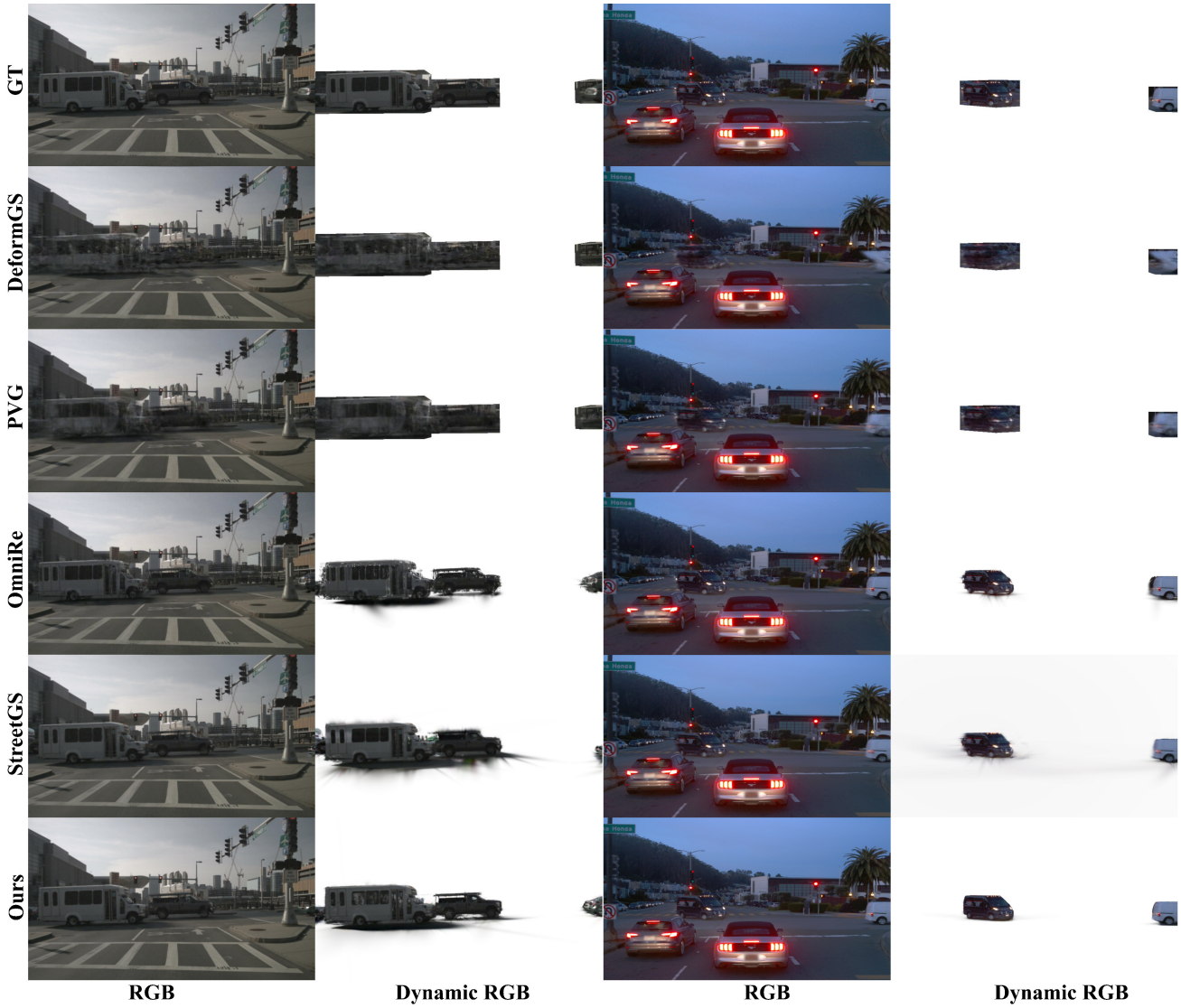

Figure 5: Novel View Synthesis Comparison on nuScenes (left two columns) and Waymo Open Dataset (right two columns). Our DenoiseGS outperforms other methods in terms of visual quality and dynamic object reconstruction.

where N is the number of total Gaussian primitives, $|\cdot|$ represents the absolute function and δ is set as a hyper-parameter.

Experiments

Experimental Settings

Datasets. Our experiments are conducted on public datasets, including nuScenes (Caesar et al. 2020) and Waymo Open Dataset (Sun et al. 2020). For Waymo dataset, we select eight sequences chosen by StreetGS (Yan et al. 2024) for a fair comparison. For a comprehensive evaluation, we also pick four sequences from the nuScenes dataset. All chosen sequences are dynamic scenes with the length of approximately 100 frames. Since nuScenes only provides 2Hz annotations, we have to obtain the annotations and camera parameters of non-keyframe through interpolation, introducing additional inaccuracy. To keep consistent with StreetGS and OmniRe, every 4th image is used as the test set, while the residual images constitute the training set. The image resolutions are 640×960 (Waymo) and 450×800 (nuScenes).

Evaluation Metrics and Baselines. The reconstruction and novel view synthesis performance is evaluated in terms of PSNR, SSIM, and LPIPS. We also report the PSNR for the dynamic objects (PSNR*). We compare our method against several state-of-the-art Gaussian Splatting approaches, including DeformGS (Yang et al. 2024b), OmniRe (Chen et al. 2024), PVG (Chen et al. 2023) and StreetGS (Yan et al. 2024).

Settings	PSNR↑	SSIM↑	LPIPS↓	PSNR*↑
(i).w/o $\mathcal{L}_\delta$	31.70	0.855	0.128	26.46
(ii).w/o delta attribute	30.62	0.844	0.146	25.52
(iii).w/o DeltaEstimator	30.60	0.840	0.136	24.73
(iv).w/o B-spline curve	31.64	0.854	0.135	25.93
DenoiseGS (Ours)	**32.28**	**0.866**	**0.121**	**26.86**

Table 2: Ablation study results for novel view synthesis on the nuScenes dataset.

Implementation details We conduct all experiments on a single NVIDIA RTX 4090D GPU for 30000 iterations. To balance the the simplicity and efficiency, we employ cubic B-spline curves. Learning rates for most parameters align with 3DGS default settings. For regularization, we choose $\lambda_r = 0.15$, $\lambda_{sky} = 0.03$, $\lambda_d = 0.002$, $\delta = 1.0$ and $\lambda_\delta = 1.0$.

Comparisons with Some State-of-the-art Methods

Results on nuScenes. We present the quantitative results of our method on the nuScenes dataset in Table 1. As the nuScenes dataset provides annotations exclusively at 2Hz intervals, interpolated bounding boxes and calibration parameters inherently contain noise and errors, which leads to a significant rendering quality drop under other state-of-the-art methods. In contrast, our DenoiseGS not only models an adaptive projection offset to alleviate noisy camera parameters, but also employs B-spline curves to precisely fit dynamic object trajectories, which results in significant performance improvement, particularly for novel view synthesis. Specifically, in the novel view synthesis task, our approach achieves the best performance with a 1.66dB improvement in PSNR, a 0.025 increase in SSIM, and a 16.0% reduction in LPIPS compared to the second best method. Especially, our DenoiseGS achieves a significant 2.58dB gain in PSNR* for dynamic object reconstruction, which mainly benefits from our proposed B-spline trajectory optimization.

Qualitative comparisons with other methods are provided in Figure 5. Notably, as DeformGS (Yang et al. 2024b) and PVG (Chen et al. 2023) cannot inherently separate dynamic objects from the static background, their dynamic regions are derived from ground truth bounding boxes. Figure 5 reveals prominent reconstruction artifacts and distortions under both methods, particularly for dynamic objects. OmniRe (Chen et al. 2024) and StreetGS (Yan et al. 2024) both struggle to reconstruct dynamic elements with high clarity and fine details. In contrast, our DenoiseGS consistently achieves high-fidelity reconstruction, which accurately represents static elements (e.g., buildings, roads) and faithfully reconstructs dynamic objects.

Results on Waymo. We also extend our experiments to the Waymo Open Dataset and present the results and qualitative comparisons in Table 1 and Figure 5. Our DenoiseGS also outperforms other state-of-the-art approaches in both reconstruction and novel view synthesis across all evaluated metrics. Specifically, in image reconstruction, our DenoiseGS surpasses StreetGS by 9.1% LPIPS and 0.77 dB PSNR*. For novel view synthesis, we also achieve a 0.48 dB gain in both PSNR and PSNR*.

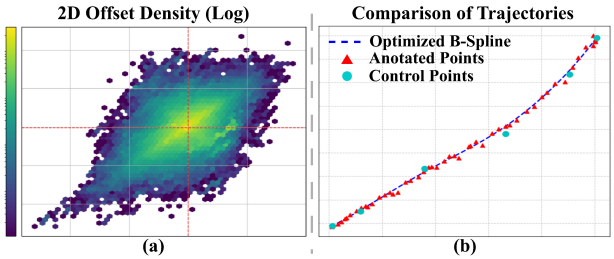

Figure 6: (a) Density map of $\mathcal{D}_i$ of total Gaussian primitives in nuScenes-mini "007" scene. (b) The optimized B-spline trajectory with annotated centers of dynamic objects.

Ablations and Analysis

We conduct ablation studies to validate the effectiveness of key components of our DenoiseGS. We report the novel view synthesis metrics on nuScenes dataset in Table 2. We can observe that **(i)** the delta constraint loss $\mathcal{L}_d$ is effective because it constrains $\mathcal{D}_i$ to remain within a valid range by setting a maximum threshold to prevent excessive drift; **(ii)** when we remove the delta attribute $\mathcal{D}_i$, the performance drops significantly, which indicates that recomputing pixel colors by offsetting the projection center to address camera parameter noise is effective; **(iii)** the DeltaEstimator enhance the representation of delta attribute $\mathcal{D}_i$ by implicitly encoding the current camera center and color information. **(iv)** the B-spline trajectory optimization can provide a smooth control and correct inaccurate annotations of b-boxes, which well improves the metrics, especially for dynamic objects.

We further visualize the spatial distribution of $\mathcal{D}_i$ using a density map in Figure 6(a), which indicates that subtle delta attribute can provide a significant improvement. In Figure 6(b), we compare the optimized trajectory derived by a learnable B-spline curve, against the original annotations. The results demonstrate that our optimized trajectory exhibits superior accuracy and smoothness.

Conclusion

This paper proposes DenoiseGS, an explicit Gaussian-based method for high-quality driving scene reconstruction. By introducing a novel learnable delta attribute, we model the projection center's offset to modulate the α-blending process and alleviate camera parameter noise. To enhance the delta attribute's representation, we then propose a DeltaEstimator encoded with viewing direction and color information. In addition, we extend CUDA operations to enable efficient gradient updates for forward and backward propagation, facilitating the optimization process. Furthermore, we propose a B-spline trajectory optimization strategy to accurately capture dynamic object motion, alleviating the reliance on inaccurate manual annotations during training. Extensive experiments demonstrate that our method achieves significant performance improvement over some state-of-the-art methods on both nuScenes dataset and Waymo Open Dataset.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62473351, in part by the University-Enterprise Joint Scientific Research and Talent Training Project of the Advanced Research Institute of Anhui Province(Research on online map construction algorithms under the guidance of light maps), in part by Open Research Fund of Anhui Provincial Key Laboratory of Intelligent Low-Carbon Information Technology and Equipment and in part by USTC-JAC Smart Electric Vehicle Joint Lab.

References

Berglund, T.; Brodnik, A.; Jonsson, H.; Staffanson, M.; and Soderkvist, I. 2009. Planning smooth and obstacle-avoiding B-spline paths for autonomous mining vehicles. *IEEE transactions on automation science and engineering*, 7(1): 167–172.

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*, 11621–11631.

Caesar, H.; Kabzan, J.; Tan, K. S.; Fong, W. K.; Wolff, E.; Lang, A.; Fletcher, L.; Beijbom, O.; and Omari, S. 2021. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*.

Chen, Y.; Gu, C.; Jiang, J.; Zhu, X.; and Zhang, L. 2023. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering.

Chen, Z.; Yang, J.; Huang, J.; Lutio, R. d.; Esturo, J. M.; Ivanovic, B.; Litany, O.; Gojcic, Z.; Fidler, S.; Pavone, M.; Song, L.; and Wang, Y. 2024. OmniRe: Omni Urban Scene Reconstruction. *arXiv preprint arXiv:2408.16760*.

Fridovich-Keil, S.; Yu, A.; Tancik, M.; Chen, Q.; Recht, B.; and Kanazawa, A. 2022. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5501–5510.

Guo, S.; Wang, Q.; Gao, Y.; Xie, R.; Li, L.; Zhu, F.; and Song, L. 2024. Depth-Guided Robust Point Cloud Fusion NeRF for Sparse Input Views. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(9): 8093–8106.

Hu, Y.; Yang, J.; Chen, L.; Li, K.; Sima, C.; Zhu, X.; Chai, S.; Du, S.; Lin, T.; Wang, W.; et al. 2023. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17853–17862.

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.

Li, C.; Zhou, L.; Jiang, H.; Zhang, Z.; Xiang, X.; Sun, H.; Luan, Q.; Bao, H.; and Zhang, G. 2023. Hybrid-MVS: Robust Multi-View Reconstruction With Hybrid Optimization of Visual and Depth Cues. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(12): 7630–7644.

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

Liu, Y.; Wang, T.; Zhang, X.; and Sun, J. 2022. Petr: Position embedding transformation for multi-view 3d object detection. In *European Conference on Computer Vision*, 531–548. Springer.

Liu, Y.; Yan, J.; Jia, F.; Li, S.; Gao, A.; Wang, T.; and Zhang, X. 2023. Petrv2: A unified framework for 3d perception from multi-camera images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3262–3272.

Liu, Z.; Su, J.; Cai, G.; Chen, Y.; Zeng, B.; and Wang, Z. 2024. GeoRGS: Geometric Regularization for Real-Time Novel View Synthesis from Sparse Inputs. *IEEE Transactions on Circuits and Systems for Video Technology*, 1–1.

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.

Mortenson, M. E. 1999. *Mathematics for computer graphics applications*. Industrial Press Inc.

Sun, H.; Zheng, X.; Ren, P.; Wang, J.; Qi, Q.; and Liao, J. 2024. SMR: Spatial-Guided Model-Based Regression for 3D Hand Pose and Mesh Reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(1): 299–314.

Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2446–2454.

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.

Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; and Luo, P. 2021. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In *Neural Information Processing Systems (NeurIPS)*.

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 *ECCV*.

Yang, J.; Ivanovic, B.; Litany, O.; Weng, X.; Kim, S. W.; Li, B.; Che, T.; Xu, D.; Fidler, S.; Pavone, M.; and Wang, Y. 2024a. EmerNeRF: Emergent Spatial-Temporal Scene Decomposition via Self-Supervision. In *The Twelfth International Conference on Learning Representations*.

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.

Yoon, J.; Han, S.; Oh, J.; and Lee, M. 2025. SplineGS: Learning Smooth Trajectories in Gaussian Splatting for Dynamic Scene Reconstruction. In *The Thirteenth International Conference on Learning Representations*.

Zhao, G.; Wang, X.; Ni, C.; Zhu, Z.; Qin, W.; Huang, G.; and Wang, X. 2025. Recondreamer++: Harmonizing generative and reconstructive models for driving scene representation. *arXiv preprint arXiv:2503.18438*.

Zhou, H.; Shao, J.; Xu, L.; Bai, D.; Qiu, W.; Liu, B.; Wang, Y.; Geiger, A.; and Liao, Y. 2024a. Hugs: Holistic urban 3d scene understanding via gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21336–21345.

Zhou, X.; Lin, Z.; Shan, X.; Wang, Y.; Sun, D.; and Yang, M.-H. 2024b. 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.

Zwicker, M.; Pfister, H.; Van Baar, J.; and Gross, M. 2001. Surface splatting. In *Proceedings of the 28th annual Conference on Computer graphics and interactive techniques*, 371–378.