DrivingRecon: Large 4D Gaussian Reconstruction Model For Autonomous Driving

Hao LU[1,2,3*], **Tianshuo XU**[1,2], **Wenzhao ZHENG**[3], **Yunpeng ZHANG**[4], **Wei ZHAN**[3], **Dalong DU**[4], **Masayoshi Tomizuka**[3], **Kurt Keutzer**[3], **Yingcong CHEN**[1,2,†]
[1]The Hong Kong University of Science & Technology (Guangzhou),
[2]The Hong Kong University of Science & Technology,
[3]University of California, Berkeley, [4]PhiGent Robotics

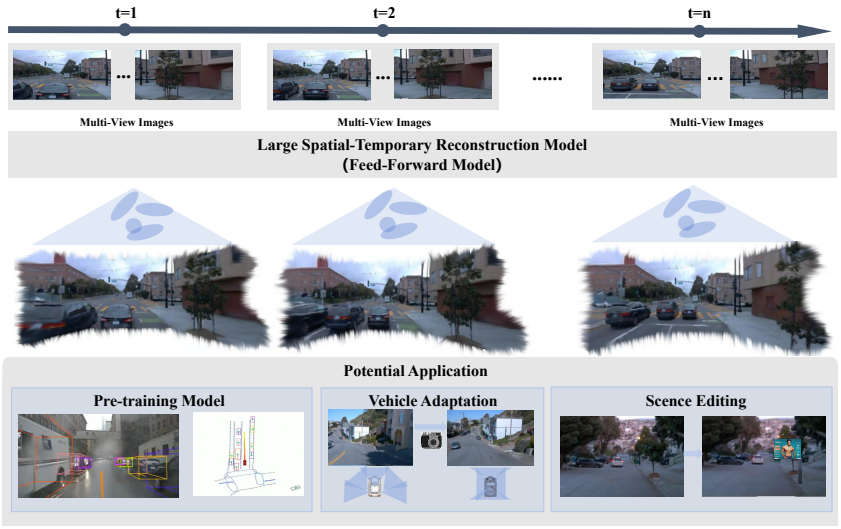

Figure 1: Leveraging temporal multi-view images, the Large 4D Gaussian Reconstruction Model (DrivingRecon) can predict 4D driving scenes. DrivingRecon serves as a pre-trained model that effectively captures geometric and motion information. Additionally, DrivingRecon can synthesize novel views based on specific camera parameters, ensuring adaptability to various vehicle models. DrivingRecon further facilitates editing of designated 4D scenes.

Abstract

Large reconstruction model has remarkable progress, which can directly predict 3D or 4D representations for unseen scenes and objects. However, current work has not systematically explored the potential of large reconstruction models in the field of autonomous driving. To achieve this, we introduce the Large 4D Gaussian Reconstruction Model (DrivingRecon). With an elaborate and simple framework design, it not only ensures efficient and high-quality reconstruction, but also provides potential for downstream tasks. There are two core contributions: firstly, the Prune and Dilate Block (PD-Block) is proposed to prune redundant and overlapping Gaussian points and dilate Gaussian points for complex objects. Then, dynamic and static decoupling is tailored to better learn the temporary-consistent geometry across different time. Experimental results demonstrate that DrivingRecon significantly improves scene reconstruction quality compared to existing methods. Furthermore, we explore applications of DrivingRecon in model pre-training, vehicle type adaptation, and scene editing. Our code is available at DriveRecon.

*Work done while visiting UC Berkeley.
†Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

Autonomous driving has made remarkable advancements in recent years, particularly in the areas of perception [25, 76, 21, 58, 54], prediction [17, 14, 26, 71, 61], and planning [10, 8, 9, 19, 73, 72, 33]. With the emergence of end-to-end autonomous driving systems that directly derive control signals from sensor data [18, 19, 22], conventional open-loop evaluations have become less effective [74]. Real-world closed-loop evaluations offer a promising solution, where the key lies in the development of high-quality scene reconstruction [50, 62, 30, 29, 31, 27, 28, 39, 69, 34].

Despite numerous advances in photorealistic reconstruction of small-scale scenes [37, 38, 4, 23, 57, 34], modeling large-scale and dynamic driving environments remains challenging. Most existing methods tackle these challenges by using 3D bounding boxes to differentiate static from dynamic components [64, 60, 50]. Subsequent methods learn the dynamics in a self-supervised manner with a 4D NeRF [66] or 3D Gaussian [20]. The aforementioned methods require numerous and time-consuming iterations for reconstruction and cannot generalize to new scenes.

Some recent methods are capable of reconstructing 3D objects [16, 75] or 3D indoor scenes [3, 6, 46] in a forward way. Many feed-forward approaches have also been tried in the autonomous driving [48, 42, 65, 56, 55]. However, all feed-forward methods predict a Gaussian point for each pixel, and the pixels from multi-view images are fused together. This paradigm has the following disadvantages: (1) The multi-view images have overlaps, and the Gaussian points corresponding to the overlapping regions will cause artifact. (2) Complex objects require more Gaussian points to represent, and the fixed number of predicted Gaussian points per region limits the quality of the image. Besides, only same moment images are used to supervise the rendered dynamic object, which limits the geometric and apparent quality of the dynamic object.

To this end, we introduce a Large Spatial-Temporal Reconstruction Model (DrivingRecon) for autonomous driving. Our framework uses the shared 2D encoder and range-view decoder, which ensures that the image encoder can serve multiple autonomous downstream tasks. The most important innovation to solve pain points is that the decoder is consisting of Prune and Dilate Blocks (PD-Blocks). The PD-Block effectively prunes overlapping Gaussian points between adjacent views and redundant background points. The pruned Gaussian points can be replaced by dilated Gaussian points of complex object. Finally, static and dynamic decoupling is proposed to improve the quality of dynamic objects. The experimental results show that our method is significantly improved compared with the existing algorithms. Our main contributions are as follows:

- We propose the PD-Block, which learns to prune redundant Gaussian points from different views and background regions. It also learns to dilate Gaussian points for complex objects, enhancing the quality of reconstruction.
- We design rendering strategies for both static and dynamic components, allowing rendered images to be efficiently supervised across temporal sequences.
- We explore the effectiveness of DrivingRecon in reconstruction, pre-training, vehicle adaptation, and scene editing tasks.

2 Related Work

2.1 Driving Scene Reconstruction

Numerous efforts have been put into reconstructing scenes from autonomous driving data captured in real scenes. Existing self-driving simulation engines such as CARLA [12] or AirSim [43] suffer from costly manual effort to create virtual environments and the lack of realism in the generated data. Many studies have investigated the application of these methods for reconstructing street scenes. Block-NeRF [70] and Mega-NeRF [49] propose segmenting scenes into distinct blocks for individual modeling. Urban Radiance Field [40] enhances NeRF training with geometric information from LiDAR, while DNMP [32] utilizes a pre-trained deformable mesh primitive to represent the scene. Streetsurf [15] divides scenes into close-range, distant-view, and sky categories, yielding superior reconstruction results for urban street surfaces. MARS [60] employs separate networks for modeling background and vehicles, establishing an instance-aware simulation framework. With the introduction of 3DGS [23], DrivingGaussian [77] introduces Composite Dynamic Gaussian Graphs and incremental static Gaussians, while StreetGaussian [64] optimizes the tracked pose of dynamic

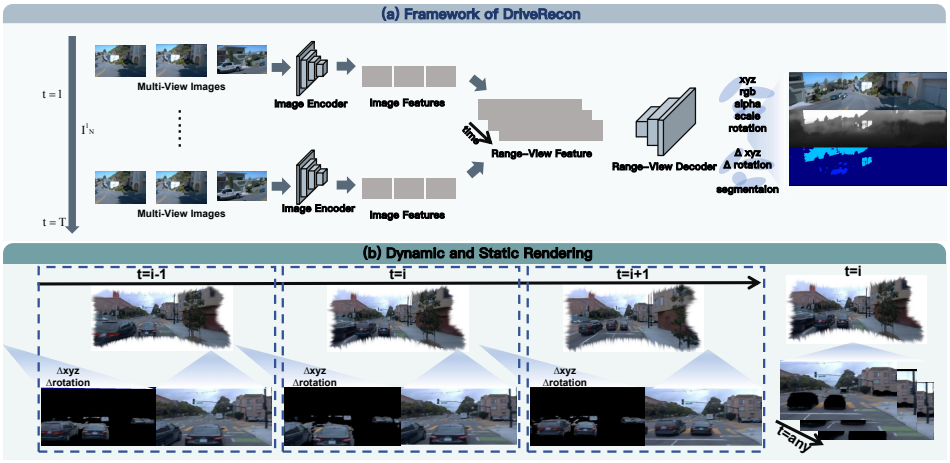

Figure 2: The overview of DrivingRecon. (a) Framework: multi-view images are in turn sent to the shared image encoder, and range view decoder to directly predict 4D Gaussians. (b) For dynamic objects, we only use next time-step images to supervise the current Gaussian parameters. For static scenes, rendering supervision is used across timestamps. In addition, reconstruction loss is also applied.

Gaussians and introduces 4D spherical harmonics for varying vehicle appearances across frames. Omnire [7] further focus on the modeling of non-rigid objects in driving scenarios. However, these reconstruction algorithms requires time-consuming iterations to build a new scene.

2.2 Large Reconstruction Models

Some works have proposed to greatly speed this up by training neural networks to directly learn the full reconstruction task in a way that generalizes to novel scenes [70, 52, 51, 59]. Recently, LRM [16] was among the first to utilize large-scale multiview datasets including Objaverse [11] to train a transformer-based model for NeRF reconstruction. The resulting model exhibits better generalization and higher quality reconstruction of object-centric 3D shapes from sparse posed images in a single model forward pass. Similar works have investigated changing the representation to Gaussian splatting [47, 65], introducing architectural changes to support higher resolution [63, 44], and extending the approach to 3D scenes [2, 6]. Recently, L4GM utilize temporal cross attention to fuses multiple frame information to predict the Gaussian representation of a dynamic object [41]. Many feed-forward approaches have also been tried in the autonomous driving [48, 42, 65, 56]. However, all feed-forward methods predict a Gaussian point for each pixel. This paradigm will cause artifacts in the overlapping areas of multiple view.

3 Method

In this section, we present the Large 4D Reconstruction Model (DrivingRecon), which generates 4D scenes from surround-view video inputs in a single feed-forward pass. Section 3.1 details the overview of DrivingRecon. In Section 3.2, we provide an in-depth examination of the Prune and Dilate Block (PD-Block). Finally, Section 3.3 discusses our training, which includes static and dynamic decoupling strategy.

3.1 Overall framework

Problem Definition. DrivingRecon utilizes temporal multi-view images D to train a feed-forward model $\mathbf{G} = f(D, \mathcal{E}, \mathcal{R}, \mathcal{V})$. For the i-th sample, $D^i = \{X^t, K^t, E^t \mid t = 1, \ldots, T\}$ includes N multi-view images $X^t = \{I_1, \ldots, I_j, \ldots, I_N\}$ at each timestep t, with corresponding intrinsic parameters $\mathcal{E}^t = \{E_1, \ldots, E_j, \ldots, E_N\}$, extrinsic rotation $\mathcal{R}^t = \{R_1, \ldots, R_j, \ldots, R_N\}$, and extrinsic translation $\mathcal{V}^t = \{V_1, \ldots, V_j, \ldots, V_N\}$. The extrinsic parameter is to project the camera coordinate system directly into the world coordinate system. We take the video start frame as the

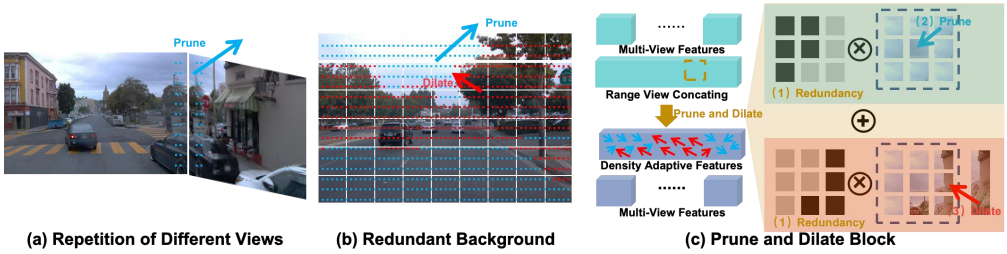

Figure 3: The motivation and details of Prune and Dilate Block (PD-Block). (a) Different views predict repeated Gaussian points, causing the model collapse. (b) Simple backgrounds (blue dots) do not need a large number of Gaussian dots to be represented, while complex objects (red dots) need more Gaussian dots to be represented. (c) PD-Block fuse the multi-view image features into a range view form. Then PD-Block prune and dilate the Gaussian points according to the complexity of the scene.

origin of the world coordinate system. This feed-forward model predicts Gaussians $\mathcal{G} = \{\mathbf{G} \in \mathbb{R}^d\}$ in the structure of $(\mathbf{xyz} \in \mathbb{R}^3, \mathbf{rgb} \in \mathbb{R}^3, \mathbf{a} \in \mathbb{R}^1, \mathbf{s} \in \mathbb{R}^3, \mathbf{c} \in \mathbb{R}^{|\mathcal{C}|}, \mathbf{r} \in \mathbb{R}^4, \mathbf{\Delta xyz} \in \mathbb{R}^3, \mathbf{\Delta r} \in \mathbb{R}^4)$. These elements represent position, RGB color, scale, rotation vectors, semantic logits, position change and rotation change, respectively.

Motivation. DriveRecon uses the shared 2D encoder and range-view decoder. The framework was designed with two principles in mind: (1) Pre-training. Driving tasks such as end-to-end and object detection typically use a shared image encoder [25, 76, 53, 22]. For 4D reconstruction tasks, the shared image encoder can learn geometry, motion, and appearance information. (2) Efficient fusion. Unlike perception and planning tasks, 4D reconstruction requires decoding high resolution features. The range-view decoder can splice adjacent image features together, so that features with similar space can be efficiently fused [24, 13].

Pipeline. Specifically, the temporal multi-view images D are processed through a shared image encoder F_{img} to extract image features e_{img} as shown in Fig. 2. The 3D position code is stitched into the channel of the image. Adjacent image features are concatenated together in the range view form. Then, different time features are stitched together in different channels. These features are fed into the range-view decoder. The range-view decoder mades up of the Prune and Dilate block (PD-Block). Finally, the Gaussian adapter in range-view decoder transforms the decoded features into Gaussian points.

3.2 Learn to Prune and Dilate

The feed-forward method predict a Gaussian point for each pixel, which causes the following disadvantages: (1) The Gaussian points corresponding to the overlapping regions will cause artifact as shown in Fig 3 (a). (2) Complex objects require more Gaussian points to represent, and the fixed number of predicted Gaussian points per region limits the quality of the image as shown in Fig 3 (b). So, we propose the Prune and Dilate Block (PD-Block) that adaptively predicts the number of Gaussian points at different areas. It can prune Gaussian points on overlapping parts and dilate the number of Gaussian points on complex objects to improve rendering quality.

Prune and Dilate Block. We propose the Prune and Dilate Block (PD-Block), which can dilate the Gaussian point of complex instances and prune the redundant Gaussian as shown in Figure 2 (c). The core steps of PD-Block are: (1) **Redundancy definition**. Given range-view features, we evenly propose K centers in space, and the center feature is computed by averaging its Z nearest points [36]. We then calculate the pair-wise cosine similarity matrix S between the region feature and the center points. (2) **Redundancy Pruning.** We set a threshold τ to generate a mask M that is considered 0 if it is below this threshold and 1 if it is above this threshold. In addition, the point most similar to the center has always been retained. (3) **Complexity Dilating.** Based on mask, we can aggregate the long-term features e_{lt} and the local features e_{lc}, $e = M * e_{lt} + (1 - M) * e_{lc}$. Here, the long-term features e_{lt} is extracted by a large kernal convolution, and the local features e_{lc} is the original range view features.

Unaligned Gaussian Adapter. The Gaussian adapter employs two convolutional blocks to convert features into segmentation $\mathbf{c} \in \mathbb{R}^C$, depth categories $\mathbf{d_c} \in \mathbb{R}^L$, depth regression refinement $\mathbf{d_r} \in \mathbb{R}^1$, RGB color $\mathbf{rgb} \in \mathbb{R}^3$, alpha $\mathbf{a} \in \mathbb{R}^1$, scale $\mathbf{r} \in \mathbb{R}^3$, rotation $\mathbf{r} \in \mathbb{R}^3$, UV-coordinate shifts $[\Delta u, \Delta v]$, and optical flow $[\Delta x, \Delta y, \Delta z]$. The activation functions for RGB color, alpha, scale, and rotation are consistent with those in [47]. The final depth per pixel is computed as $\mathbf{d_f} = \sum_{l=1}^{L} l \times \text{softmax}(\mathbf{d_c}) + \mathbf{d_r}$.

However, PD-Blocks effectively manage spatial computational redundancy by reallocating resources from simple scenes to more complex objects, allowing for Gaussian points that are not strictly pixel-aligned. For this reason, our Gaussian Adapter also predicts the offset of the uv coordinate $[\Delta u, \Delta v]$. The world coordinate $[x, y, z] = RE^{-1}\mathbf{d_f} * [u + \Delta u, v + \Delta v, 1] + V$. The above operations are universal for any time and view, so we did not label the time and views for simplicity. Gaussian points of different viewing angles are all fused to render. In addition, we can use the world coordinates at time t and the predicted optical flow to get the world coordinates at time t+1, $[x_{t+1}, y_{t+1}, z_{t+1}] = [x_t + \Delta x_t, y_t + \Delta y_t, z_t + \Delta z_t]$. Rotational changes in an object are interpreted as positional changes.

3.3 Static and Dynamic Decoupling

To learn geometry and motion information, DrivingRecon carefully designed the static and dynamic decoupling as shown in Fig. 2, including both unsupervised and supervised manner.

Unsupervised manner. The views of the driving scene are very sparse, meaning that only a limited number of cameras capture the same scene simultaneously. Hence, cross-temporal view supervision is essential. For dynamic objects, our algorithm predicts not only the current Gaussian of dynamic objects at time t but also predicts the motion flow of each Gaussian point. Therefore, we will also use the next frame to supervise the predicted Gaussian points, i.e., dynamic reconstruction loss $\mathcal{L}_{dy}$. Additionally, we have the reconstruction constraint $\mathcal{L}_{re}$, which involves rendering the image as the same as the input. The dynamic and reconstruction constraint all use the L1 constraint to constitute an unsupervised constraint $\mathcal{L}_{re}$. We also employ cross-entropy loss $\mathcal{L}_c$ for the depth categories $_c$ predicted by the Gaussian Adapter and L1 loss $\mathcal{L}_r$ for the refined depth $_r$. In the field of autonomous driving, depth supervision during training is considered unsupervised [65, 20, 66].

Supervised manner. For autonomous driving, we also want to know the semantic properties of each Gaussian point in 4D space. So, cross-entropy loss is used to constrain the segmentation results predicted by Gaussian Adapter, i.e., $\mathcal{L}_{seg}$. Segmentation labels can be provided by some existing segmentation models, but we still consider them supervised for fair comparison. With the segmentation model, we can render the scene with camera parameters of adjacent timestamps and supervise only the static part, i.e., static reconstruction loss $\mathcal{L}_{dy}$. It is important to note that when supervising the rendering across the time sequence, we will not supervise the rendered image where the threshold value is less than α, as these pixels often do not overlap across the time sequence. In summary, the overall constraints for training DrivingRecon are:

$$\mathcal{L}_{\text{total}} = \underbrace{\lambda_c\mathcal{L}_c + \lambda_r\mathcal{L}_r + \lambda_{re}\mathcal{L}_{re} + \lambda_{dr}\mathcal{L}_{dr}}_{\text{Unsupervised}} + \underbrace{\lambda_{sr}\mathcal{L}_{sr} + \lambda_{seg}\mathcal{L}_{seg}}_{\text{Supervised}}$$

where each λ term balances the contribution of the respective loss component. $\mathcal{L}_{sr}$ and $\mathcal{L}_{seg}$ used segmentation labels, which is not used for pre-training experiment. Other loss are considered unsupervised, which also allows DrivingRecon to achieve good performance. These collective regulations and constraints enable DrivingRecon to effectively integrate geometry and motion information, enhancing its capacity for accurate scene reconstruction across time and perspectives.

4 Experiment

In this section, we evaluate the performance of DrivingRecon in terms of reconstruction and novel view synthesis, as well as explore its potential applications. We also provide detailed information on the dataset setup, baseline methods, and implementation details.

Table 1: **Comparison to state-of-the-art methods on the Waymo Open Dataset.** PSNR, SSIM, and Depth RMSE (D-RMSE) are reported. Speed metrics are estimated on a single A100 GPU. *: reproduced by us. †: Non-sky region. $^-$: not using segmentation supervision.

Methods	Dynamic-only			Full image†			Inference speed	Real-time rendering
	PSNR↑	SSIM↑	D-RMSE↓	PSNR↑	SSIM↑	D-RMSE↓	Time↓	(>200FPS)
Per-Scene Optimization methods								
EmerNeRF [66]	17.79	0.255	40.88	24.51	0.738	33.99	14min	×
3DGS [23]	17.13	0.267	13.88	25.13	0.741	19.68	23min	✓
PVG [5]	15.51	0.128	15.91	22.38	0.661	13.01	27min	✓
DeformableGS [68]	17.10	0.266	12.14	25.29	0.761	14.79	29min	✓
Generalizable feed-forward methods								
LGM [47]	19.58	0.443	9.43	23.59	0.691	8.02	0.06s	✓
GS-LRM [75]	20.02	0.520	9.95	25.18	0.753	7.94	**0.02s**	✓
SCube* [42]	20.51	0.532	8.82	25.72	0.783	5.62	0.42s	✓
DrivingForward* [48]	21.97	0.622	7.42	26.32	0.774	5.79	0.21s	✓
STORM [65]	22.10	0.624	7.50	26.38	0.794	5.48	0.18s	✓
Ours								
DriveRecon$^-$	23.11	0.641	7.32	27.23	0.819	5.23	0.08s	✓
DriveRecon	**24.01**	**0.662**	**7.03**	**27.52**	**0.833**	**5.14**	0.08s	✓

4.1 Datasets.

Following [48, 42, 65, 56], we use both the Waymo Open dataset [45] and the nuScenes [1] to test the algorithm's performance. The rendered image resolution of DrivingRecon is 160×240 as same as [65]. For nuScenes dataset, the rendered image resolution of DrivingRecon is 224×400 as same as [56]. All training and testing validation are consistent with the official ones.

4.2 Training Details.

The model is trained on 24 NVIDIA A100 (80G) GPUs for 50000 iterations, all about 24 gpu days. A batch size of 2 for each GPU is used under bfloat16 precision, resulting in an effective batch size of 48. The AdamW optimizer is employed with a learning rate of $4 * 10^{-4}$ and a weight decay of 0.05. $\lambda_{re}, \lambda_c, \lambda_r, \lambda_{dr}, \lambda_{sr}, \lambda_{seg}$ are set as 1.0, 0.1, 0.1, 0.1, 0.1, 0.1, respectively. Following [65], The input to our model consists of 4 context timesteps, evenly spaced at $t + 0$s, $t + 0.5$s, $t + 1.0$s, and $t + 1.5$s, where t is a randomly chosen starting timestep. These balance parameters are based on our experience.

4.3 Main Results

We compare our method against both two categories of approaches: per-scene optimization methods and feed-forward models. For per-scene optimization, we evaluate against both NeRF-based and 3DGS-based approach, including EmerNeRF [66], 3DGS [23], PVG [5], and DeformableGS [68]. Since LiDAR data is not provided at test time in our setup, these baselines is also without LiDAR supervision to ensure a fair comparison. In the second category, we compare against recent large reconstruction models, including LGM [47], GS-LRM [75], SCube [42], DrivingForward [48], STORM [65]. SCube and DrivingForward were modified to the same resolution and test protocol using their officially provided code.

We present the quantitative results in Tab. 1. Compared to per-scene optimization methods, the large reconstruction models achieves comparable performance in both dynamic regions and full images in terms of photorealism, geometry, and inference speed. Notably, compared to other generalizable feed-forward models, DriveRecon demonstrates a robust ability to model scene dynamics and process multi-timestep, multi-view images. DriveRecon also achieves the best performance when it has no segmentation supervision, that is, pure unsupervised. In fact, both Scube and STORM use different forms of segmentation supervision. Besides, DriveRecon achieves these improvements while reducing inference time to just 0.08 second. The sota method STORM has 100.60M parameters, and DriveRecon only has 53.37M parameters. This is because we used the shared image encoder and the range view decoder. It ensures that our model parameters and reasoning are significantly efficient.

To better clarify the effectiveness of our algorithm. We compare Scube and OmniScene, two new generalizable Gaussian methods. STORM is not open source so it is not compared. As shown in Fig. 4,

Figure 4: Reconstruction quantitative comparison. The red box indicates the presence of artifacts in the overlap area of multi-view images, due to Gaussian prediction at per-pixel level **(zoom in for the best view.)**

Scube and OmniScene have serious artifacts in overlapping areas. This is because these method predict a Gaussian point for each pixel, and the same location in the overlapping region may correspond to multiple Gaussian points. Our PD-Block can dynamically remove these redundant points, and the range-view decoder can better integrate different views. This responds to the motivation of our paper.

Table 2: Comparison to state-of-the-art methods on the nuScenes. *: reproduced by us. Pearson Correlation Coefficient (PCC) quantifies the statistical relationship between the predicted depth and ground-truth depth as the scale-invariant metric [56].

Method	PSNR↑	SSIM↑	LPIPS↓	PCC↑
pixelSplat [3]	21.51	0.616	0.372	0.001
MVSplat [6]	21.61	0.658	0.295	0.181
DrivingForward* [48]	24.32	0.732	0.229	0.766
Omni-Scene [56]	24.27	0.736	0.237	0.804
DriveRecon	**25.42**	**0.782**	**0.201**	**0.825**

Further, we compare recent DrivingForward [48] and Omni-Scene [56] simultaneously on nuScenes. The DrivingForward is reproduced using the same protocol as [56]. STORM is not open source, and Scube relies heavily on more fine-grained Occ. So, we can't reproduce these method on nuScenes. As shown in Tab. 2, we have also demonstrated the advance of our approach. This proves that our PD-Block and dynamic and static decoupling are very efficient.

4.4 Ablation study

To assess the effectiveness of our proposed algorithm, we conducted a series of ablation experiments. The key components under evaluation include the PD-Block and Dynamic and Static Decoupling(DS).

Table 3: Ablation of DrivingRecon.

	PSNR	SSIM	LPIPS
all	27.52	0.83	0.16
w/o PD-Block	26.21	0.78	0.24
w/o DS	26.44	0.79	0.20

As shown in Tab. 3, each module contributes significant performance improvements. Notably, the PD-Block achieves the highest enhancement. This improvement stems from two primary factors: (1) an optimized distribution of computational resources based on spatial complexity, where more Gaussian points are allocated to complex regions while simpler backgrounds receive fewer points; (2) enhanced multi-view integration within a broad field of view. The DS-R mechanism also led to marked improvements, largely attributed to the use of cross-temporal supervision for better dynamic and static object differentiation. Qualitative visualization of the new view can be seen in Fig. 5. It proved that we can render geometrically consistent new perspectives in different time.

4.5 Potential application

Vehicle adaptation. The introduction of a new car model may result in changes in camera parameters, such as camera type (intrinsic parameters) and camera placement (extrinsic parameters) [53, 35].

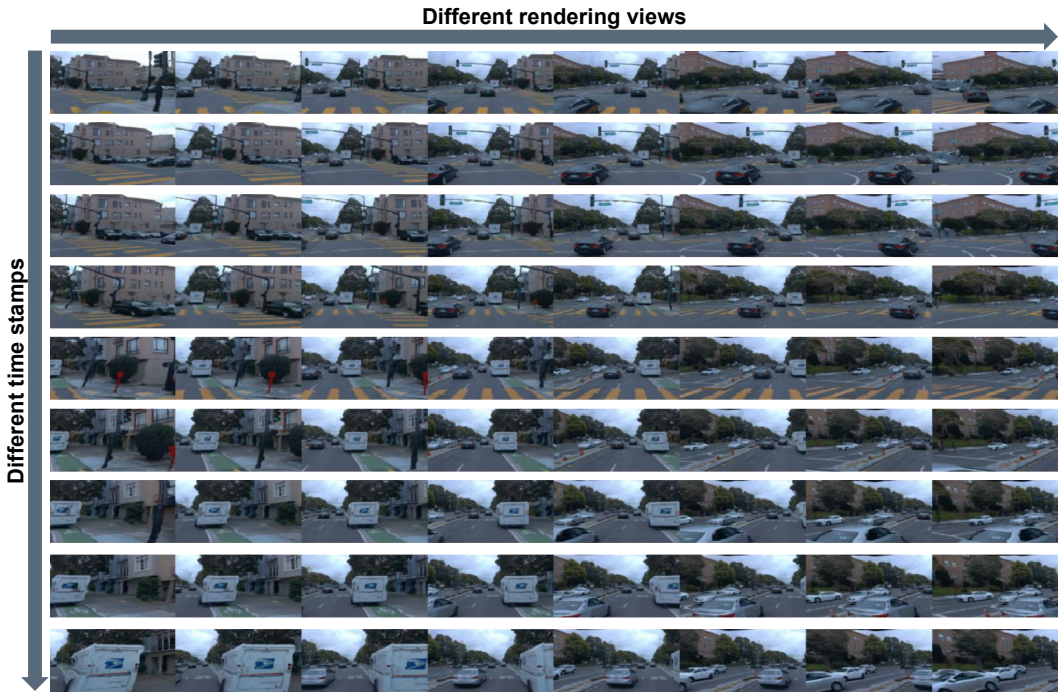

Figure 5: Novel view rendering. Based on the predicted Gaussians, we render different views at different times. The novel views are of very high quality and very high spatio-temporal consistency **(zoom in for the best view.)**

Figure 6: Scene editing. We can insert the new object in the scene, and ensure time consistency.

Table 4: Comparison of different approaches on domain generalization protocols, where * stands for using aligned intrinsic parameters, + stands for randomly augmenting camera extrinsic parameters.

Waymo → nuScenes	Target Domain (nuScenes)				
Method	mAP↑	mATE↓	mASE↓	mAOE↓	NDS* ↑
Oracle	0.475	0.577	0.177	0.147	0.587
DG-BEV	0.303	0.689	0.218	0.171	0.472
PD-BEV	0.311	0.686	0.216	0.170	0.478
Ours*	0.305	0.690	0.219	0.167	0.471
Ours*+	**0.323**	**0.675**	**0.212**	**0.166**	**0.490**

The 4D reconstruction model is capable of rendering images with different camera parameters to mitigate the potential overfitting of these parameters. To achieve this, we rendered images on Waymo with random intrinsic parameters and performed random rendering of novel views as a form of data augmentation. Based on PD-BEV[3], we used this jointly rendered and original data to train the BEVDepth on Waymo, following the approach of [53, 35]. It is important to note that our rendered images also undergo an augmentation pipeline as part of the detection algorithm, including resizing and cropping.

[3] `https://github.com/EnVision-Research/Generalizable-BEV`

Table 5: Performance gain of our method for joint perception, prediction, and planning.

Method	Detection		Tracking			Future Occupancy Prediction			
	NDS ↑	mAP ↑	AMOTA↑	AMOTP↓	IDS↓	IoU-n.↑	IoU-f.↑	VPQ-n.↑	VPQ-f.↑
UniAD	49.36	37.96	38.3	1.32	1054	62.8	40.1	54.6	33.9
ViDAR	52.57	42.33	42.0	1.25	991	65.4	42.1	57.3	36.4
Ours+	**53.21**	**43.21**	**42.9**	**1.18**	**948**	**66.5**	**43.3**	**58.2**	**37.3**

Method	Mapping		Motion Forecasting			Planning	
	IoU-lane↑	IoU-road↑	minADE↓	minFDE↓	MR↓	avg.L2↓	avg.Col.↓
UniAD	31.3	69.1	0.75	1.08	0.158	1.12	0.27
ViDAR	33.2	71.4	0.67	0.99	0.149	0.91	0.23
Ours+	**33.9**	**72.1**	**0.60**	**0.89**	**0.138**	**0.84**	**0.19**

As demonstrated in Tab. 4, when we employ both camera intrinsic and extrinsic parameter augmentation, we observe a significant improvement in performance. However, the use of only camera intrinsic parameter augmentation did not yield good results, due to the superior ability of virtual depth in addressing the issue of camera intrinsic parameters. The utilization of multiple extrinsic parameters helps the algorithm learn the stereo relationship between cameras more effectively.

Pre-training model. The 4D reconstruction network is capable of understanding the geometric information of the scene, the motion trajectory of dynamic objects, and the semantic information. These capabilities are reflected in the encoding of images, where the weights of these encoders are shared. We replaced our encoder with the ResNet-50, which is a commonly used base network for many algorithms as same as [67]. Following VIDAR [4], we then retrained the 4D reconstruction network using this configuration, without using any segmentation annotations, resulting in completely unsupervised pre-training. Subsequently, we replaced the encoder of UniAD [19] with our pre-trained model and fine-tuned it on the nuScenes dataset. The results, as presented in Tab. 5, demonstrate that our pre-trained model achieved better performance compared to ViDAR [67], highlighting the ability of our algorithm to leverage large-scale unsupervised data for pre-training and improving multiple downstream tasks.

Scene editing. The 4D scene reconstruction model enables us to obtain comprehensive 4D geometry information of a scene, which allows for the removal, insertion, and control of objects within the scene. As shown in Fig. 6, we added billboards with people's faces to fixed positions in the scene, representing a corner case where cars come to a stop. As can be seen from the figure, the scenario we created exhibits a high level of temporal consistency.

5 Conclusion

The paper introduces DrivingRecon, a novel 4D Gaussian Reconstruction Model for fast 4D reconstruction of driving scenes using surround-view video inputs. The entire framework uses a shared image encoder and range view decoder. A key innovation is the Prune and Dilate Block (PD-Block), which prunes redundant Gaussian points from adjacent views and dilates points around complex edges, enhancing the reconstruction of dynamic and static objects. Additionally, a dynamic-static rendering approach using optical flow prediction allows for better supervision of moving objects across time sequences. DrivingRecon shows superior performance in scene reconstruction and novel view synthesis compared to existing methods. It is particularly effective for tasks such as model pre-training, vehicle adaptation, and scene editing. We quantitatively evaluated pre-training and vehicle adaptation and made significant improvements.

6 Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 62206068), HKUST-HKUST(GZ) Cross-Campus Collaborative Research Scheme (Project No. C036) and Guangdong Provincial Department of Science and Technology's '1+1+1' Joint Funding Program for Guangdong-Hong Kong Universities.

[4] https://github.com/OpenDriveLab/ViDAR

References

[1] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset. In *Proc. IEEE CVPR*, 2020.

[2] D. Charatan, S. Li, A. Tagliasacchi, and V. Sitzmann. pixelsplat. In *arXiv preprint arXiv:2312.12337*, 2023.

[3] D. Charatan, S. L. Li, A. Tagliasacchi, and V. Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proc. CVPR*, pages 19457–19467, 2024.

[4] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su. Tensorf: Tensorial radiance fields. In *Proc. ECCV*, pages 333–350, 2022.

[5] Y. Chen, C. Gu, J. Jiang, X. Zhu, and L. Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. In *ArXiv*, 2023.

[6] Y. Chen, H. Xu, C. Zheng, B. Zhuang, M. Pollefeys, A. Geiger, T.-J. Cham, and J. Cai. Mvsplat. In *arXiv preprint arXiv:2403.14627*, 2024.

[7] Z. Chen, J. Yang, J. Huang, R. de Lutio, J. M. Esturo, B. Ivanovic, O. Litany, Z. Gojcic, S. Fidler, M. Pavone, et al. Omnire. In *arXiv preprint arXiv:2408.16760*, 2024.

[8] J. Cheng, Y. Chen, Q. Zhang, L. Gan, C. Liu, and M. Liu. Real-time trajectory planning for autonomous driving with gaussian process and incremental refinement. In *Proc. ICRA*, pages 8999–9005, 2022.

[9] J. Cheng, X. Mei, and M. Liu. Forecast-MAE: Self-supervised pre-training for motion forecasting with masked autoencoders. In *Proc. ICCV*, 2023.

[10] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Proc. CoRL*, 2023.

[11] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse. In *Proc. IEEE/CVF CVPR*, pages 13142–13153, 2023.

[12] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun. Carla: An open urban driving simulator. In *Proc. CoRL*, 2017.

[13] L. Fan, X. Xiong, F. Wang, N. Wang, and Z. Zhang. Rangedet: In defense of range view for lidar-based 3d object detection. In *Proc. IEEE ICCV*, pages 2918–2927, 2021.

[14] J. Gu, C. Hu, T. Zhang, X. Chen, Y. Wang, Y. Wang, and H. Zhao. Vip3d. In *arXiv preprint arXiv:2208.01582*, 2022.

[15] J. Guo, N. Deng, X. Li, Y. Bai, B. Shi, C. Wang, C. Ding, D. Wang, and Y. Li. Streetsurf. In *ArXiv*, 2023.

[16] Y. Hong, K. Zhang, J. Gu, S. Bi, Y. Zhou, D. Liu, F. Liu, K. Sunkavalli, T. Bui, and H. Tan. Lrm: Large reconstruction model for single image to 3d. In *arXiv preprint arXiv:2311.04400*, 2023.

[17] A. Hu, Z. Murez, N. Mohan, S. Dudas, J. Hawke, V. Badrinarayanan, R. Cipolla, and A. Kendall. Fiery. In *Proc. IEEE ICCV*, 2021.

[18] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao. St-p3. In *Proc. ECCV*, 2022.

[19] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, et al. Planning-oriented autonomous driving. In *Proc. IEEE CVPR*, pages 17853–17862, 2023.

[20] N. Huang, X. Wei, W. Zheng, P. An, M. Lu, W. Zhan, M. Tomizuka, K. Keutzer, and S. Zhang. *S^3 Gaussian: Self-Supervised Street Gaussians for Autonomous Driving*. In *arXiv preprint arXiv:2405.20323*, 2024.

[21] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu. Tri-perspective view. In *Proc. IEEE CVPR*, pages 9223–9232, 2023.

[22] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang. Vad. In *arXiv preprint arXiv:2303.12077*, 2023.

[23] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. In *ACM Trans. Graph.*, volume 42, pages 1–14, 2023.

[24] L. Kong, Y. Liu, R. Chen, Y. Ma, X. Zhu, Y. Li, Y. Hou, Y. Qiao, and Z. Liu. Rethinking range view representation for lidar segmentation. In *Proc. IEEE ICCV*, pages 228–240, 2023.

[25] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai. Bevformer. In *Proc. ECCV*, 2022.

[26] M. Liang, B. Yang, W. Zeng, Y. Chen, R. Hu, S. Casas, and R. Urtasun. Pnpnet. In *Proc. IEEE CVPR*, 2020.

[27] J. Liu, J. Han, L. Liu, A. I. Aviles-Rivero, C. Jiang, Z. Liu, and H. Wang. Mamba4d: Efficient 4d point cloud video understanding with disentangled spatial-temporal state space models. In *CVPR*, pages 17626–17636, 2025.

[28] J. Liu, Z. Huang, M. Liu, T. Deng, F. Nex, H. Cheng, and H. Wang. Topolidm: Topology-aware lidar diffusion models for interpretable and realistic lidar point cloud generation. *arXiv preprint arXiv:2507.22454*, 2025.

[29] J. Liu, G. Wang, W. Ye, C. Jiang, J. Han, Z. Liu, G. Zhang, D. Du, and H. Wang. Difflow3d: Toward robust uncertainty-aware scene flow estimation with iterative diffusion-based refinement. In *CVPR*, pages 15109–15119, 2024.

[30] J. Liu, W. Ye, G. Wang, C. Jiang, L. Pan, J. Han, Z. Liu, G. Zhang, and H. Wang. Difflow3d: Hierarchical diffusion models for uncertainty-aware 3d scene flow estimation. *IEEE TPMAI*, 2025.

[31] J. Liu, D. Zhuo, Z. Feng, S. Zhu, C. Peng, Z. Liu, and H. Wang. Dvlo: Deep visual-lidar odometry with local-to-global feature fusion and bi-directional structure alignment. In *ECCV*, pages 475–493. Springer, 2025.

[32] F. Lu, Y. Xu, G.-S. Chen, H. Li, K.-Y. Lin, and C. Jiang. Urban radiance field representation with deformable neural mesh primitives. In *Proc. IEEE ICCV*, pages 465–476, 2023.

[33] H. Lu, Z. Liu, G. Jiang, Y. Luo, S. Chen, Y. Zhang, and Y.-C. Chen. Uniugp: Unifying understanding, generation, and planing for end-to-end autonomous driving. *arXiv preprint arXiv:2512.09864*, 2025.

[34] H. Lu, Z. Ma, G. Jiang, W. Ge, B. Li, Y. Cai, W. Zheng, Y. Zhang, and Y. Chen. 4d driving scene generation with stereo forcing. *arXiv preprint arXiv:2509.20251*, 2025.

[35] H. Lu, Y. Zhang, Q. Lian, D. Du, and Y. Chen. Towards generalizable multi-camera 3d object detection via perspective debiasing. In *AAAI*, 2025.

[36] X. Ma, Y. Zhou, H. Wang, C. Qin, B. Sun, C. Liu, and Y. Fu. Image as set of points. In *ICLR*, 2023.

[37] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf. In *Communications of the ACM*, pages 99–106, 2021.

[38] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. In *ACM transactions on graphics (TOG)*, pages 1–15, 2022.

[39] X. Pan, J. Yao, H. Kou, T. Wu, and C. Xiao. Harmonicnerf: Geometry-informed synthetic view augmentation for 3d scene reconstruction in driving scenarios. In *ACM MM*, pages 5987–5996, 2024.

[40] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. A. Funkhouser, and V. Ferrari. Urban radiance fields. In *Proc. IEEE CVPR*, pages 12922–12932, 2021.

[41] J. Ren, K. Xie, A. Mirzaei, H. Liang, X. Zeng, K. Kreis, Z. Liu, A. Torralba, S. Fidler, S. W. Kim, et al. L4gm: Large 4d gaussian reconstruction model. In *arXiv preprint arXiv:2406.10324*, 2024.

[42] X. Ren, Y. Lu, J. Z. Wu, H. Ling, M. Chen, S. Fidler, F. Williams, J. Huang, et al. Scube: Instant large-scale scene reconstruction using voxsplats. In *NeurIPS*, 2024.

[43] S. Shah, D. Dey, C. Lovett, and A. Kapoor. Airsim. In *International Symposium on Field and Service Robotics*, 2017.

[44] Q. Shen, X. Yi, Z. Wu, P. Zhou, H. Zhang, S. Yan, and X. Wang. Gamba: Marry gaussian splatting with mamba for single view 3d reconstruction. In *arXiv preprint arXiv:2403.18795*, 2024.

[45] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, S. Zhao, S. Cheng, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov. Scalability in perception for autonomous driving. In *Waymo Open Dataset*, 2020.

[46] S. Szymanowicz, E. Insafutdinov, C. Zheng, D. Campbell, J. F. Henriques, C. Rupprecht, and A. Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. In *arXiv preprint arXiv:2406.04343*, 2024.

[47] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *Proc. ECCV*, pages 1–18. Springer, 2024.

[48] Q. Tian, X. Tan, Y. Xie, and L. Ma. Drivingforward: Feed-forward 3d gaussian splatting for driving scene reconstruction from flexible surround-view input. *AAAI*, 2025.

[49] H. Turki, D. Ramanan, and M. Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs. In *Proc. IEEE/CVF CVPR*, pages 12912–12921, 2021.

[50] H. Turki, J. Y. Zhang, F. Ferroni, and D. Ramanan. Suds: Scalable urban dynamic scenes. In *Proc. IEEE/CVF CVPR*, pages 12375–12385, 2023.

[51] P. Wang, X. Chen, T. Chen, S. Venugopalan, Z. Wang, et al. Is attention all that nerf needs? In *arXiv preprint arXiv:2207.13298*, 2022.

[52] Q. Wang, Z. Wang, K. Genova, P. Srinivasan, H. Zhou, J. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser. Ibrnet. In *Proc. IEEE CVPR*, pages 4690–4699, 2021.

[53] S. Wang, X. Zhao, H.-M. Xu, Z. Chen, D. Yu, J. Chang, Z. Yang, and F. Zhao. Towards domain generalization for multi-view 3d object detection in bird-eye-view. In *Proc. CVPR*, pages 13333–13342, 2023.

[54] Z. Wang, B. Li, C. Wang, and S. Scherer. Airshot: Efficient few-shot detection for autonomous exploration. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11654–11661. IEEE, 2024.

[55] Z. Wang, J. Tan, T. Khurana, N. Peri, and D. Ramanan. Monofusion: Sparse-view 4d reconstruction via monocular fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8252–8263, 2025.

[56] D. Wei, Z. Li, and P. Liu. Omni-scene: omni-gaussian representation for ego-centric sparse-view scene reconstruction. In *Proc. IEEE CVPR*, 2025.

[57] X. Wei, R. Zhang, J. Wu, J. Liu, M. Lu, Y. Guo, and S. Zhang. Noc: High-quality neural object cloning. In *arXiv preprint arXiv:2309.12790*, 2023.

[58] Y. Wei, L. Zhao, W. Zheng, Z. Zhu, J. Zhou, and J. Lu. Surroundocc: Multi-camera 3d occupancy prediction. In *ICCV*, pages 21729–21740, 2023.

[59] C.-Y. Wu, J. Johnson, J. Malik, C. Feichtenhofer, and G. Gkioxari. Multiview compressive coding for 3d reconstruction. In *Proc. IEEE CVPR*, pages 9065–9075, 2023.

[60] Z. Wu, T. Liu, L. Luo, Z. Zhong, J. Chen, H. Xiao, C. Hou, H. Lou, Y. Chen, R. Yang, Y. Huang, X. Ye, Z. Yan, Y. Shi, Y. Liao, and H. Zhao. Mars: An instance-aware, modular and realistic simulator. In *CICAI*, 2023.

[61] M. Xie, S. Zeng, X. Chang, X. Liu, Z. Pan, M. Xu, and X. Wei. Seqgrowgraph: Learning lane topology as a chain of graph expansions. In *ICCV*, pages 27166–27175, October 2025.

[62] Z. Xie, J. Zhang, W. Li, F. Zhang, and L. Zhang. S-nerf: Neural radiance fields for street views. In *arXiv preprint arXiv:2303.00749*, 2023.

[63] Y. Xu, Z. Shi, W. Yifan, H. Chen, C. Yang, S. Peng, Y. Shen, and G. Wetzstein. Grm: Large gaussian reconstruction model. In *arXiv preprint arXiv:2403.14621*, 2024.

[64] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng. Street gaussians for modeling dynamic urban scenes. In *ArXiv*, volume abs/2401.01339, 2024.

[65] J. Yang, J. Huang, Y. Chen, Y. Wang, B. Li, Y. You, A. Sharma, M. Igl, P. Karkus, D. Xu, et al. Storm: Spatio-temporal reconstruction model for large-scale outdoor scenes. *Proc. ICLR*, 2024.

[66] J. Yang, B. Ivanovic, O. Litany, X. Weng, S. W. Kim, B. Li, T. Che, D. Xu, S. Fidler, M. Pavone, and Y. Wang. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. 2023.

[67] Z. Yang, L. Chen, Y. Sun, and H. Li. Visual point cloud forecasting. In *Proc. IEEE CVPR*, 2024.

[68] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *ArXiv*, volume abs/2309.13101, 2023.

[69] J. Yao, J. Zhang, X. Pan, T. Wu, and C. Xiao. Depthssc: Monocular 3d semantic scene completion via depth-spatial alignment and voxel adaptation. In *WACV*, pages 2154–2163. IEEE, 2025.

[70] A. Yu, V. Ye, M. Tancik, and A. Kanazawa. pixelnerf: Neural radiance fields. In *Proc. IEEE/CVF CVPR*, pages 4578–4587, 2021.

[71] S. Zeng, X. Chang, X. Liu, Z. Pan, and X. Wei. Driving with prior maps: Unified vector prior encoding for autonomous vehicle mapping. *arXiv preprint arXiv:2409.05352*, 2024.

[72] S. Zeng, X. Chang, M. Xie, X. Liu, Y. Bai, Z. Pan, M. Xu, and X. Wei. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving. *arXiv preprint arXiv:2505.17685*, 2025.

[73] S. Zeng, D. Qi, X. Chang, F. Xiong, S. Xie, X. Wu, S. Liang, M. Xu, and X. Wei. Janusvln: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation. *arXiv preprint arXiv:2509.22548*, 2025.

[74] J.-T. Zhai, Z. Feng, J. Du, Y. Mao, J.-J. Liu, Z. Tan, Y. Zhang, X. Ye, and J. Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. In *arXiv preprint arXiv:2305.10430*, 2023.

[75] K. Zhang, S. Bi, H. Tan, Y. Xiangli, N. Zhao, K. Sunkavalli, and Z. Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *arXiv preprint arXiv:2404.19702*, 2024.

[76] Y. Zhang, Z. Zhu, W. Zheng, J. Huang, G. Huang, J. Zhou, and J. Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. In *arXiv preprint arXiv:2205.09743*, 2022.

[77] X. Zhou, Z. Lin, X. Shan, Y. Wang, D. Sun, and M.-H. Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *ArXiv*, volume abs/2312.07920, 2023.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: On Paper

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Show on Appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Application-driven

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our code will be made publicly available soon.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Dataset is open.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Show on main paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Show on main paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Show on main paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have reviewed the NeurIPS Code of Ethics and confirm that the research described in this paper complies with the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA] .

    Justification: NA

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: On appendix.

    Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA] .

    Justification: Sure

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA] .

Justification: NA

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA] .

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA] .

    Justification: NA

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.