

FEEDFORWARD 4D RECONSTRUCTION FOR DYNAMIC DRIVING SCENES USING UNPOSED IMAGES

Anonymous authors

Paper under double-blind review

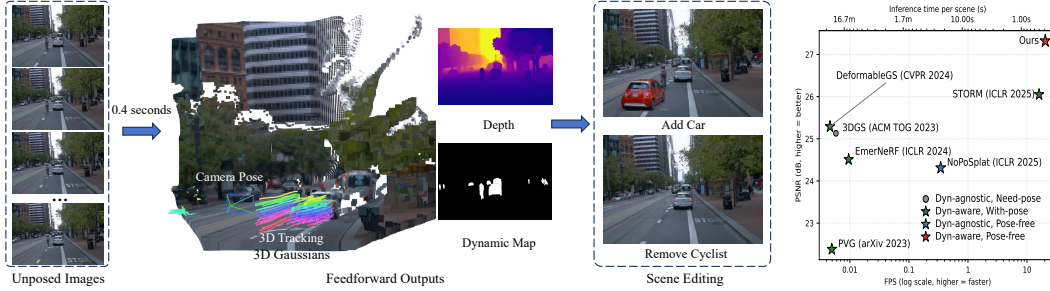


Figure 1: **Left:** Our feedforward framework reconstructs dynamic scenes from unposed images in 0.4 s, enabling editing tasks such as object addition or removal. **Right:** We achieve state-of-the-art performance with competitive speed among feedforward methods.

ABSTRACT

Autonomous vehicles require diverse dynamic scenes for robust training and evaluation, yet existing dynamic scene reconstruction methods are often limited by slow per-scene optimization and reliance on explicit annotations or camera calibration. In this paper, we introduce a pose-free, feedforward framework for 4D scene reconstruction that jointly infers camera parameters, dynamic Gaussian representations, and 3D motion directly from sparse, unposed images. Unlike prior feedforward approaches, our model accommodates an arbitrary number of input views, enabling long-sequence modeling and improved generalization. Dynamic objects are disentangled via estimated motion and aggregated into unified 3DGS representations, while a diffusion-based refinement module mitigates flow artifacts and enhances novel view synthesis under sparse inputs. Trained on the Waymo Dataset and evaluated on nuScenes and Argoverse2, our method achieves superior performance while generalizing effectively across datasets, benefiting from the pose-free design that reduces dataset-specific biases. Additionally, the framework supports instance-level scene editing and high-fidelity view synthesis, providing a scalable foundation for real-world autonomous driving simulation.

1 INTRODUCTION

Autonomous vehicles navigate complex and dynamic 3D environments, requiring diverse dynamic scenes for robust training and evaluation Yan et al. (2025); Wu et al. (2023); Tonderski et al. (2024); Li et al. (2025); Jin et al. (2024); Tian et al. (2023); Chen et al. (2024c); Yan et al. (2024). Recent advances in neural scene representations Shao et al. (2023); Park et al. (2021); Luiten et al. (2024); Wu et al. (2024) achieve impressive visual fidelity in reconstructing dynamic scenes from multi-timestep images. However, these approaches typically rely on per-scene optimization that minimizes photometric loss, requiring several minutes to hours to train on a single scene Martin-Brualla et al. (2021); Zhang et al. (2024a); Kulhanek et al. (2024); Gao et al. (2024); Wang et al. (2025b). Such workflows are computationally expensive and time-consuming, making them impractical for large-scale simulation. Moreover, many existing methods Yan et al. (2024); Zhou et al. (2024) rely on explicit 3D annotations such as bounding boxes to model motion, which are labor-intensive to obtain and restrict applicability in real-world driving scenarios.

To address these limitations, STORMYang et al. (2024a) introduces a feedforward framework for fast dynamic scene reconstruction from images. While this marks an important step toward scalable 4D reconstruction, several challenges remain. First, training is limited to sequences of only four timesteps, which hinders the ability to capture long-term dynamics. Second, novel view synthesis suffers from sparse input views, often leading to noticeable fidelity degradation under large viewpoint changes. Finally, the method still assumes access to accurate camera poses, requiring an additional calibration stage that reduces its practicality for in-the-wild video sequences.

In this paper, we present a feedforward model for 4D scene reconstruction that jointly estimates camera parameters and 3D scene representation in a single pass. Compared with STORM, our framework accommodates an arbitrary number of unposed images as input, enabling the modeling of longer sequences and demonstrating improved generalization to unseen data by mitigating dataset-specific biases in camera pose. Built on a unified vision-transformer backbone, the model predicts per-frame 3D Gaussian representations and 3D motion to capture dynamic objects. Without relying on extrinsic camera calibration or instance-level annotations for dynamic objects, this framework enables efficient and scalable reconstruction of 4D scenes directly from sparse, unposed images.

To represent dynamic objects, we leverage the estimated motion and per-frame 3DGS maps to decompose the scene into static and dynamic components, and aggregate dynamic objects across frames into a unified 3DGS representation via spatial transformations. However, imperfections in motion estimation inevitably introduce artifacts into the aggregated 3DGS. To address this, we further introduce a diffusion-based refinement module that operates on rendered images, substantially improving reconstruction fidelity. Beyond artifact reduction, this refinement process mitigates the limitations of novel view synthesis under sparse-view settings, yielding higher-quality renderings from challenging viewpoints.

Trained on the Waymo Open Dataset Sun et al. (2020) and evaluated on diverse benchmarks, including nuScenes Caesar et al. (2020) and Argoverse2 Wilson et al. (2023), our method achieves superior rendering quality and generalization ability, while also improving runtime efficiency compared to optimization-based approaches. Additionally, it supports instance-level editing, as illustrated in Fig. 1. By combining high-quality 4D reconstruction with real-time performance, our framework provides a scalable and practical foundation for future autonomous driving simulation systems.

2 RELATED WORKS

Dynamic scene reconstruction aims to recover a time-varying 3D representation of a scene from an image sequence. Recent extensions of NeRF Li et al. (2022); Pumarola et al. (2021); Shao et al. (2023); Park et al. (2021); Mildenhall et al. (2021); Wang et al. (2021); Yu et al. (2022); Liu et al. (2024); Yuan & Zhao (2024) and 3DGS Luiten et al. (2024); Wu et al. (2024); Yan et al. (2024); Zhou et al. (2024); Chen et al. (2024a); Cheng et al. (2024); Zhang et al. (2024c); Ye et al. (2025) to dynamic settings have achieved high-fidelity reconstructions. Based on their motion modeling strategies, existing approaches can be broadly categorized into two main classes. The first class employs temporally conditioned representations, where time is served as an explicit input to the model. For example, PVG Chen et al. (2023) integrates periodic vibration-based temporal signals to predict 3DGS representations. However, such methods are often impractical for downstream tasks like object editing, as they lack disentangled, object-centric representations. The second class represents the scene as a compositional scene graph, where dynamic entities are modeled independently—typically as separate NeRFs Wu et al. (2023) or 3DGS Chen et al. (2024c) components. While this structure facilitates object-level manipulation, these methods often rely on external 3D annotations such as bounding boxes, which are costly to obtain. Moreover, most existing methods rely on per-scene optimization, requiring several hours to reconstruct each scene. To address these limitations, we propose a fast, feedforward dynamic reconstruction method that eliminates the need for additional annotations, enabling efficient and generalizable 4D modeling across diverse scenes.

Feedforward reconstruction infers 3D scene representations like NeRF or 3DGS directly from input observations via a single forward pass of a trained neural network. Unlike per-scene optimization methods, feedforward approaches are designed to generalize across diverse scenes and enable real-time inference. Several methods Hong et al. (2023); Tang et al. (2024); Wang et al. (2025a); Keetha et al. (2025) adopt vision transformers to reconstruct 3D objects from multi-view images. Subsequent works such as Flash3D Szymanowicz et al. (2024), MVSplat Chen et al. (2024b), No-

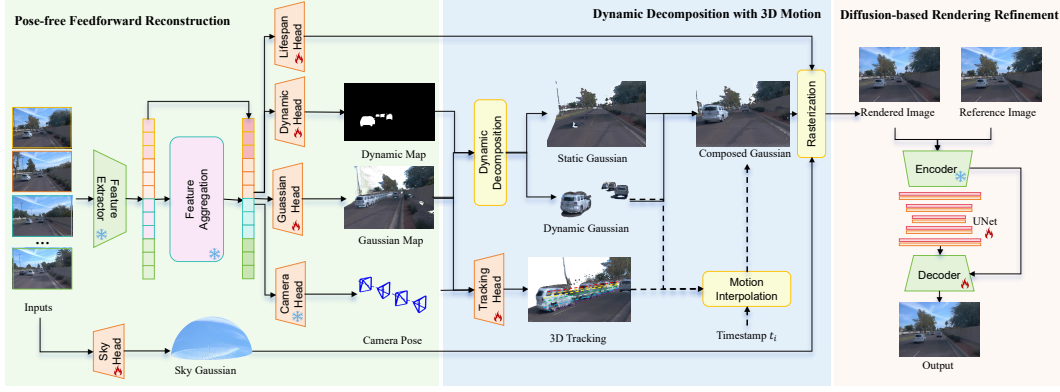


Figure 2: **Overall Architecture.** Given unposed images of dynamic scene, we estimate camera parameters, dynamic maps, and per-pixel Gaussians in a single pass. Subsequently, a motion head is employed to track dynamic objects across time, and their trajectories are interpolated to construct temporally consistent Gaussian representations. Finally, a diffusion-based rendering module refines the resulting composition, producing high-fidelity renderings.

PoSplatYe et al. (2024) and DepthSplat Xu et al. (2025) extend feedforward 3D Gaussian Splatting to the scene level, though they remain limited to static environments. L4GMRen et al. (2024) presents the first 4D reconstruction model, yet the application is mainly restricted to objects. More recently, STORM Yang et al. (2024a) introduces a feedforward framework for dynamic scenes, but it is constrained by the number of input frames and struggles with long sequences. In contrast, our method supports an arbitrary number of input frames and does not require camera poses, enabling more flexible and efficient dynamic scene reconstruction.

3 METHOD

In this paper, we introduce a pose-free, feedforward framework that reconstructs 3D structure directly from unposed images (Sec.3.1). To handle dynamic scenes, we incorporate a motion estimation module that predicts 3D motion and enables consistent fusion of moving objects (Sec. 3.2). Finally, a diffusion-based refinement module is further applied to enhance reconstruction quality and support high-fidelity view synthesis (Sec. 3.3). This fully feedforward design enables efficient and scalable reconstruction, delivering high-quality results directly from raw image sequences without any pose or 3D annotation as inputs. The overall architecture is depicted in Fig 2.

3.1 POSE-FREE FEEDFORWARD RECONSTRUCTION

Given a sequence of unposed RGB images $\{I^t \mid I^t \in \mathbb{R}^{H \times W \times 3}, t = 1, \dots, N\}$ captured over N timestamps of a dynamic scene, our objective is to reconstruct a temporally coherent 3D representation in a single forward pass, without requiring on external pose calibration or 3D annotations. To this end, we propose a feedforward model f_θ that directly maps the image sequence to a dynamic 3D scene representation. Concretely, the model jointly estimates the per-frame camera parameters Π^t and the 3D scene representation for each timestamp.

Scene Representation We represent the scene at each frame using a pixel-aligned Gaussian map $G^t \in \mathbb{R}^{H \times W \times 15}$, where each pixel-aligned primitive at (i, j) encodes RGB color $c_{i,j}^t \in \mathbb{R}^3$, 3D mean position $\mu_{i,j}^t \in \mathbb{R}^3$, rotation quaternion $r_{i,j}^t \in \mathbb{R}^4$, scale $s_{i,j}^t \in \mathbb{R}^3$, opacity $o_{i,j}^t \in \mathbb{R}$, and a lifespan parameter $\sigma_{i,j}^t \in \mathbb{R}^+$, which controls its temporal influence by modulating opacity over time. Specifically, given the predicted parameters at timestamp t , the opacity at another timestamp t' is computed as:

$$o^{t'} = o^t \cdot e^{-\frac{1}{2} \cdot \frac{(t'-t)^2}{\sigma^t}}, \quad (1)$$

σ controls the temporal spread of the Gaussian, and a larger σ leads to a longer-lasting Gaussian in time, while a smaller σ results in faster temporal fading.

The first frame I^1 is designated as the reference frame, with its camera origin serving as the world coordinate origin. All subsequent frames are aligned to this reference, ensuring consistent 3D positioning across time.

Sky Modeling To account for distant background regions such as the sky, we introduce a separate set of sky Gaussians denoted by G_{sky} . Their centers are uniformly sampled on a hemisphere of fixed radius r_{sky} , chosen to approximate an effectively infinite background. Each sky Gaussian is assigned a fixed rotation and opacity. To determine their appearance, we project their 3D coordinates onto the input images and extract the corresponding pixel colors. These initial colors, along with the Gaussian scale parameters, are then refined by a lightweight MLP $\mathcal{H}_{sky}$ to improve consistency and realism in sky modeling.

Model Architecture Following VGGT Wang et al. (2025a), we adopt a 24-layer ViT architecture as the backbone of our feedforward model f_θ . The input images are first partitioned into patches and transformed into token sequences, which are subsequently encoded by a DINO-pretrained feature extractor Zhang et al. (2022) to obtain rich visual representations, denoted as F_{dino} . These features are then refined through an alternating-attention mechanism, yielding F_{attn} . The resulting features are processed by multiple prediction heads: the camera head $\mathcal{H}_{cam}$ for estimating camera poses, the Gaussian head $\mathcal{H}_{gs}$ for generating 3D Gaussians, and the lifespan head $\mathcal{H}_{life}$ for lifespan parameters.

During training, we leverage pretrained priors by freezing the feature extractor and camera head, while training the remaining heads from scratch. Feeding F_{attn} into the prediction heads, the camera pose is obtained as $\Pi^t = \mathcal{H}_{cam}(F_{attn})$. However, we observe that F_{attn} primarily encodes high-level semantics and lacks sufficient detail for appearance reconstruction. To mitigate this issue, we fuse F_{attn} with the original DINO features, thereby enhancing spatial fidelity. The fused feature is then used by the Gaussian head to generate the Gaussian splatting map as $G^t = \mathcal{H}_{gs}(F_{dino}, F_{attn})$.

This unified framework enables reconstruction of the 3D scene structure across all timestamps. Nevertheless, in dynamic environments, naively aggregating Gaussian maps across time leads to incoherent reconstructions due to object motion. To address this, we introduce a motion field that explicitly models dynamic object trajectories, as detailed in the following section.

3.2 DYNAMIC DECOMPOSITION WITH 3D MOTION

Given the predicted Gaussian maps, directly aggregating them over time would produce ghosting artifacts due to moving objects, as illustrated in Fig. 2 (Gaussian map). Therefore, we extend the feedforward model with a dynamic head $\mathcal{H}_{dy}$ that predicts the probability of dynamic regions, formulated as $M_d^t = \mathcal{H}_{dy}(F_{attn})$. This dynamic map distinguishes moving objects from the static background, enabling us to decompose each Gaussian map G^t into a static component G_s^t and a dynamic component G_d^t :

$$G_s^t = G^t \odot (1 - M_d^t), \quad G_d^t = G^t \odot M_d^t, \quad (2)$$

where $\odot$ denotes element-wise multiplication. For each timestamp t , the full Gaussian representation is constructed by combining the sky Gaussian, the static components from all frames, and the dynamic component of the current frame as:

$$\hat{G}^t = \left(\bigcup_{t'=1}^N G_s^{t'} \right) \cup G_d^t \cup G_{sky}. \quad (3)$$

here G_d^t denotes the dynamic Gaussians at frame t , and $\bigcup_{t'=1}^N G_s^{t'}$ represents the union of static Gaussians from all frames. The rendered image at timestamp t is then obtained via the differentiable renderer, defined as:

$$\hat{I}^t = \text{Renderer}(\hat{G}^t, \Pi^t), \quad (4)$$

where Π^t denotes the predicted camera parameters.

3D Motion Estimation In practice, the input timestamps are sparse. For an intermediate time t_i with $t_i \notin \{1, \dots, N\}$, corresponding dynamic Gaussian representation is not directly available. To address this, we explicitly model object motion over time. We exploit correspondences between

input images and introduce a motion head that predicts 3D motion for a set of query pixels $\mathcal{Q} \in \mathbb{R}^{q \times 2}$, where q is the number of queries. Concretely, for any pair of timestamps $t_a, t_b \in \{1, \dots, N\}$, the motion head estimates 3D motion $F(t_a, t_b) \in \mathbb{R}^{q \times 3}$, which provides per-pixel 3D displacement vectors, enabling temporal alignment of Gaussian primitives across frames.

Specifically, we introduce a transformer-based motion head $\mathcal{H}_{\text{motion}}$, which jointly processes 2D images and 3D points extracted from Gaussian maps. The images are first encoded into multi-scale features, which are then associated with 3D points to construct a spatio-temporal feature cloud. For each timestamp t_a , we select the one-valued pixels in the dynamic map $M_d^{t_a}$ as query pixels $\mathcal{Q}$, back-project them to initialize their 3D positions, and iteratively refine their trajectories via neighborhood-to-neighborhood attention. Formally, the motion head is defined as:

$$F(t_a, t_b) = \mathcal{H}_{\text{motion}}(\mathcal{Q} \mid G^{t_a}, G^{t_b}, I^{t_a}, I^{t_b}), \quad (5)$$

where G^{t_a} and G^{t_b} denote the Gaussian maps and I^{t_a}, I^{t_b} the corresponding images at times t_a and t_b . The motion head is initialized with pretrained weights Zhang et al. (2025) and subsequently finetuned using a photometric loss on interpolated frames.

Motion Interpolation With the 3D motion, we can estimate the dynamic Gaussian representation at an intermediate time $t_i \in [t_a, t_b]$, where t_a and t_b are adjacent timestamps. Specifically, we interpolate the mean coordinates $\mu_d^{t_i}$ of the dynamic Gaussians using the motion prediction as:

$$\mu_d^{t_i} = \mu_d^{t_a} + \omega_{t_i} \cdot F(t_a, t_b), \quad \omega_{t_i} = \frac{t_i - t_a}{t_b - t_a}, \quad (6)$$

where ω_{t_i} is a linear interpolation weight and $M_d^{t_a}$ denotes the dynamic mask at t_a . This yields the dynamic Gaussians $G_d^{t_i}$ at the intermediate timestamp t_i , which are then combined as Eq. 3 to obtain the full Gaussian representation. For camera pose Π^{t_i} at timestamp t_i , we linearly interpolated the translation between Π^{t_a} and Π^{t_b} , while the rotation was interpolated using spherical linear interpolation (SLERP) on quaternions.

Notably, our model predicts all camera parameters, Gaussian representations, 3D motion, and dynamic masks in a single forward pass, enabling simultaneous 4D reconstruction and scene understanding. The full forward process is defined as:

$$G_{\text{sky}}, \{G^t, F^t, M_d^t, \Pi^t\}_{t=1}^N = f_{\theta}(\{I^t\}_{t=1}^N), \quad (7)$$

where $G_{\text{sky}}, G^t, F^t, M_d^t, \Pi^t$ are the sky Gaussian, Gaussian map, 3D motion, dynamic map and camera pose respectively, f_{θ} denotes the feedforward network, and $\{I^t\}_{t=1}^N$ is the set of input images.

Training Objectives We train the feedforward reconstruction model including the motion estimation head, in an end-to-end manner. For each training iteration, we randomly sample $N \in [4, 8]$ input images and generate $2N$ images by interpolating between the sampled frames. A reconstruction loss is applied to all $2N$ images, combining an ℓ_2 term with a perceptual LPIPS term:

$$\mathcal{L}_{\text{rgb}} = \mathcal{L}_{\ell_2} + \lambda_{\text{LPIPS}} \mathcal{L}_{\text{LPIPS}}. \quad (8)$$

To supervise opacity and dynamic maps, we adopt binary cross-entropy losses as:

$$\mathcal{L}_{\text{opacity}} = \text{BCE}(M_{\text{sky}}, \hat{M}_{\text{sky}}), \quad \mathcal{L}_{\text{dynamic}} = \text{BCE}(M_d, \hat{M}_{\text{dynamic}}), \quad (9)$$

where M_{sky} is derived from the rendered opacity with ground-truth $\hat{M}_{\text{sky}}$, and M_d denotes the predicted dynamic map with ground-truth $\hat{M}_{\text{dynamic}}$. We impose an ℓ_1 regularization for lifespan parameters under the assumption that most of the scene is static as $\mathcal{L}_{\text{lifespan}} = \left| \frac{1}{\sigma} \right|_1$.

The overall training objective is a weighted combination of these components:

$$\mathcal{L}_{\text{feedforward}} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{opacity}} \mathcal{L}_{\text{opacity}} + \lambda_{\text{dynamic}} \mathcal{L}_{\text{dynamic}} + \lambda_{\text{lifespan}} \mathcal{L}_{\text{lifespan}}. \quad (10)$$

3.3 DIFFUSION-BASED RENDERING REFINEMENT

Although the estimated motion field provides a plausible description of object dynamics, interpolation still produces artifacts such as ghosting and disocclusion gaps, due to deviations in the estimated

Method	Render Quality			Inference time (s)	Dynamic	Pose-free
	PSNR $\uparrow$	SSIM $\uparrow$	D-RMSE $\downarrow$			
EmerNeRF Yang et al. (2023)	24.51	0.738	33.99	14min	✓	✗
3DGSKerbl et al. (2023)	25.13	0.741	19.68	23min	✗	✗
PVGChen et al. (2023)	22.38	0.661	13.01	27min	✓	✗
DeformableGS Yang et al. (2024b)	25.29	0.761	14.79	29min	✓	✗
LGMTang et al. (2024)	18.53	0.447	9.07	0.06s	✗	✗
GS-LRMZhang et al. (2024b)	25.18	0.753	7.94	0.02s	✗	✗
MVSplat Chen et al. (2024b)	20.56	0.697	10.13	0.08s	✗	✗
NoPoSplat Ye et al. (2024)	24.31	0.751	9.08	23.22s	✗	✓
DepthSplat Xu et al. (2025)	23.26	0.696	10.05	<u>0.11s</u>	✗	✗
STORM* Yang et al. (2024a)	26.05	0.819	5.91	0.50s	✓	✗
STORMYang et al. (2024a)	<u>26.38</u>	<u>0.794</u>	5.48	0.18s	✓	✗
Ours	27.41	0.846	<u>5.56</u>	0.39s	✓	✓

Table 1: **Quantitative comparison on the Waymo dataset.** Higher PSNR and SSIM, and lower D-RMSE indicate better performance. (* denotes the results from our replication)

motion and the limited robustness of the 3DGS representation under large rotations and translations. To enhance realism, we further incorporate an image-space refinement stage with diffusion model. This post-render module serves two key purposes: (1) suppressing interpolation artifacts to deliver high-quality renderings, and (2) compensating for novel view gaps caused by the limited field of view in the input observations.

We build our refinement model $f_{\text{diffusion}}$ on a single-step diffusion framework Wu et al. (2025), consisting of a frozen VAE encoder, a UNet denoiser, and a LoRA fine-tuned decoder. As illustrated in Fig. 2, given a rendered image $\hat{I}^{t_i}$ and a reference image I_{ref} randomly sampled from the input sequence, we first concatenate them frame-wisely and encode the result into the latent space using the VAE encoder. The latent embedding is subsequently processed by the UNet denoiser and decoded to yield the refined image $\tilde{I}^{t_i}$. Formally, the process is formulated as:

$$\tilde{I}^{t_i} = f_{\text{diffusion}}(\hat{I}^{t_i}, I_{\text{ref}}). \quad (11)$$

The diffusion model is also supervised under an ℓ_2 reconstruction loss between the model output $\tilde{I}^{t_i}$ and the corresponding ground-truth image I^{t_i} , complemented by a perceptual LPIPS loss. To further enhance sharpness and fine details, we also incorporate a style loss based on the Gram matrices of VGG-16 features. The overall objective for fine-tuning the model is defined as

$$\mathcal{L}_{\text{diffusion}} = \mathcal{L}_{\text{Recon}} + \mathcal{L}_{\text{LPIPS}} + \lambda_{\text{Gram}} \mathcal{L}_{\text{Gram}}. \quad (12)$$

4 EXPERIMENT

4.1 EXPERIMENTAL PROTOCOL

Datasets We conduct experiments on the Waymo Open Dataset Sun et al. (2020), which contains real-world autonomous driving logs. For training, we use the full Waymo Open Dataset training split, which comprises 798 scenes, each containing 190–200 frames. For evaluation, we use the Waymo Open Dataset test split, which comprises 202 scenes and covers challenging scenarios such as multi-vehicle interactions, nighttime driving, and rainy weather. To assess the generalization ability of our method across varied driving conditions, we evaluate on two additional benchmarks: nuScenes Caesar et al. (2020) and Argoverse2 Wilson et al. (2023). For quantitative evaluation on 4D reconstruction, performance is reported using PSNR and SSIM for images, as well as root mean square error (RMSE) for depth estimation. For 3D tracking evaluation, we adopt End-Point Error in 3D (EPE3D), Acc5, Acc10, and angular error as metrics, following Yang et al. (2024a).

4.2 COMPARISON WITH SOTA METHODS

Baselines We compare our approach with a broad set of reconstruction methods, including optimization-based techniques such as EmerNeRF Yang et al. (2023), 3DGS Kerbl et al. (2023),

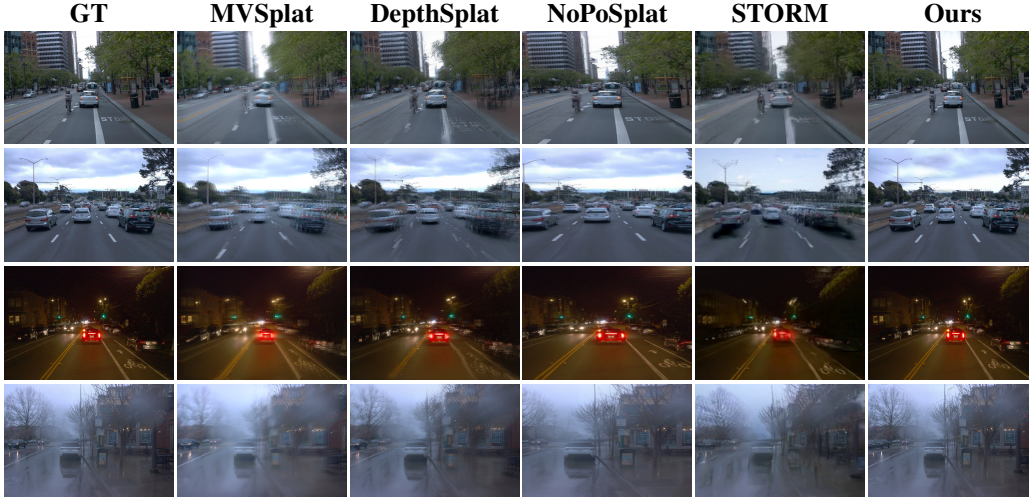


Figure 3: **Qualitative comparison of different methods on Waymo dataset.**



Figure 4: **3D Tracking Visualization.** Points with the same color correspond across frames.

PVG Chen et al. (2023), and DeformableGS Yang et al. (2024b), as well as feedforward methods including LGM Tang et al. (2024), GS-LRM Zhang et al. (2024b), MVSpLat Chen et al. (2024b), NoPoSpLat Ye et al. (2024), DepthSpLat Xu et al. (2025), and STORM Yang et al. (2024a). For 3D tracking, we compare with NSFP Li et al. (2021), NSFP++ Najibi et al. (2022), and STORM.

Novel View Synthesis To evaluate both reconstruction on input frames and novel view synthesis on interpolated frames, we use 8-frame sequences from 202 scenes as the test set, taking 4 frames as input to predict the intermediate frames and evaluating performance on all 8 frames. Tab. 1 reports results on the Waymo dataset, where our method outperforms all baselines in PSNR and SSIM. Fig. 3 further provides qualitative comparisons with feedforward approaches. While MVSpLat, NoPoSpLat, and DepthSpLat fail to capture dynamic components due to their limited architectural design, STORM also struggles to accurately model object motion. In contrast, our model successfully distinguishes moving objects from the static background and produces faithful reconstructions. To further assess generalization, we evaluate on nuScenes and Argoverse2 (Tab. 2) under both zero-shot and training-from-scratch settings. In both cases, our model achieves the best performance, demonstrating strong adaptability and robustness across diverse driving scenarios. We credit this generalization ability to the pose-free design, which helps reduce domain gaps between datasets..

3D Motion Estimation We evaluate motion estimation on the Waymo Scene Flow dataset. As summarized in Tab. 5, our approach consistently outperforms prior arts across all metrics. These results demonstrate that reliable correspondences can be effectively learned through the rendering loss. Fig. 4 presents qualitative 3D tracking results at timestamps T_0 and T_1 . We render the 3D motion vectors onto the point clouds, where color-coded tracks highlight consistent correspondences across frames. These results show our method accurately recovers dynamic trajectories for vehicles and pedestrians, validating the motion estimation model.

4.3 ABLATION STUDY AND APPLICATION

Number of Input Views One advantage of our method over previous feedforward approaches is its support for a flexible number of input views while maintaining strong performance on long se-

Method	nuScenes			Argoverse2		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
Zero-shot						
MVSplat Chen et al. (2024b)	17.84	0.563	0.451	18.67	0.647	0.304
NoPoSplat Ye et al. (2024)	19.75	0.545	0.394	22.00	0.646	0.237
DepthSplat Xu et al. (2025)	19.52	0.601	0.376	22.05	0.636	0.280
STORM Yang et al. (2024a)	17.77	0.669	0.394	20.83	0.542	0.326
Ours	25.31	0.794	0.152	26.34	0.812	0.155
Trained						
STORM Yang et al. (2024a)	24.54	0.784	0.267	24.97	0.791	0.240
Ours	26.63	0.813	0.122	26.96	0.831	0.118

Table 2: **Quantitative Comparison under Trained and Zero-Shot Settings on nuScenes and Argoverse2 datasets.** Our Waymo-trained model shows generalization ability in zero-shot NVS on other datasets, with training on each target dataset further boosting performance.

Method	#Views Input	Reconstruction			NVS		
		PSNR $\uparrow$	SSIM $\uparrow$	D-RMSE $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	D-RMSE $\downarrow$
STORM Yang et al. (2024a)	4	26.55	0.851	6.139	26.05	0.819	5.914
Ours		30.54	0.884	3.673	27.41	0.846	5.565
STORM Yang et al. (2024a)	8	25.11	0.807	6.054	25.44	0.807	5.470
Ours		31.41	0.895	3.508	27.74	0.858	5.703
STORM Yang et al. (2024a)	16	23.69	0.765	5.700	22.98	0.723	5.836
Ours		30.66	0.887	3.550	28.14	0.885	5.933

Table 3: **Ablation study on the number of input views.** Reconstruction performance shows low sensitivity to the number of input frames, demonstrating robustness with sparse inputs.

quences. Tab. 4 reports both reconstruction results on input frames and novel view synthesis (NVS) results on interpolated frames with 4, 8, and 16 input views. Our approach not only consistently outperforms STORM across all evaluation metrics, but also remains stable as the number of views increases. In contrast, STORM exhibits a sharp drop in PSNR and SSIM as views increase, demonstrating our superior scalability and robustness under varying input settings.

Lifespan Parameter We conduct an ablation study to assess the contributions of different model components, summarized in Tab. 4 and Fig. 5. Removing the lifespan parameter causes a significant performance drop, with PSNR decreasing from 27.41 to 24.21. This occurs because lifespan is crucial for capturing the dynamic appearance of static objects, such as lighting changes over time (see Fig. 5, row 3). Without it, Gaussians fail to model these subtle variations, resulting in unstable reconstructions and degraded perceptual quality.

Diffusion Refinement To evaluate the effectiveness of our diffusion refinement module, we conduct an ablation study with and without it, as summarized in Tab. 4 and visualized in Fig. 5. Quantitatively, the refinement improves PSNR and SSIM, confirming its role in enhancing rendering quality. Qualitatively, without refinement the reconstructions show noticeable artifacts and loss of fine details, particularly in the sky and dynamic objects. In contrast, incorporating diffusion refinement effectively removes these artifacts and produces more realistic renderings.

Scene Editing We further demonstrate the scene editing capability of our feedforward 4D reconstruction model. Since the method directly generates 3D Gaussian representations and decomposes them into dynamic and static components, we can flexibly manipulate the reconstructed scene by adding, removing, or repositioning dynamic objects. As illustrated in Fig. 6, the first row shows examples where cars are removed or shifted by modifying the dynamic Gaussians, while the second row demonstrates the seamless composition of novel vehicles and a cyclist by integrating the static scene with dynamic Gaussians reconstructed from other scenes. This editing process highlights the

Table 4: **Ablation study.** Removing lifespan parameters or the diffusion refinement model decreases performance.

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
w/o lifespan	24.21	0.774	0.169
w/o diffusion	<u>27.32</u>	<u>0.844</u>	0.108
Ours	27.41	0.846	<u>0.109</u>

Table 5: **Quantitative results on 3D motion estimation.** We achieve state-of-the-art performance in 3D tracking on the Waymo Open Dataset.

Method	EPE3D (m) $\downarrow$	Acc ₅ (%) $\uparrow$	Acc ₁₀ (%) $\uparrow$	θ (rad) $\downarrow$
NSFP	0.698	42.17	54.26	0.919
NSFP++	0.711	53.10	63.02	0.989
STORM	<u>0.276</u>	<u>81.12</u>	<u>85.61</u>	<u>0.658</u>
Ours	0.183	85.42	90.42	0.328

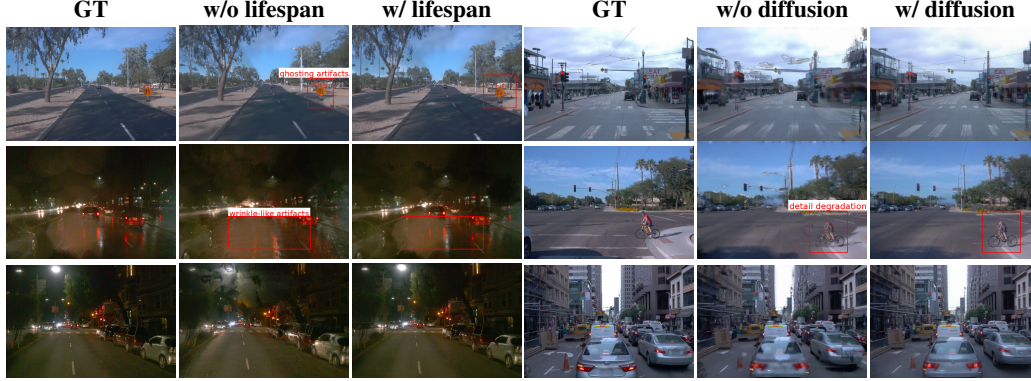


Figure 5: **Ablation study.** Removing the lifespan parameter hinders the capture of changing appearance of static scene, while the diffusion refinement reduces artifacts and improves rendering quality.

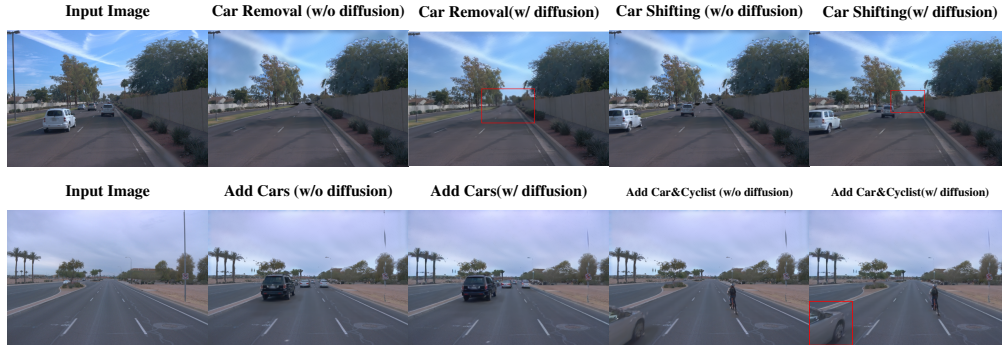


Figure 6: **Scene editing results.** Cars can be removed or shifted (row 1), and novel vehicles/cyclists inserted from other scenes (row 2). Diffusion refinement fixes artifacts such as holes (red box).

advantage of our representation: scene elements can be recombined at the Gaussian level without retraining, enabling efficient modifications of complex driving scenarios. Moreover, the edited images can be further refined using our diffusion-based model, which effectively addresses issues such as holes or artifacts introduced during Gaussian manipulation, as highlighted in the red box.

5 CONCLUSIONS

We proposed a pose-free, feedforward framework for 4D scene reconstruction from unposed images. Our method unifies camera estimation, 3D Gaussian reconstruction, and 3D motion prediction, while a diffusion-based refinement module improves fidelity under sparse inputs. Experiments on large-scale datasets show SOTA performance with fast speed. Despite these strengths, some limitations remain. In particular, failure cases can occur when dynamic masks are inaccurate or when tracking fails under heavily occluded motion. Future work will focus on improving dynamic modeling and enhancing tracking robustness to better handle complex dynamic scenes.

REFERENCES

- Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nusenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.
- Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 2024a.
- Yuedong Chen, Haoqi Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pp. 370–386. Springer, 2024b.
- Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *arXiv preprint arXiv:2311.18561*, 2023.
- Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, et al. Omnire: Omni urban scene reconstruction. *arXiv preprint arXiv:2408.16760*, 2024c.
- Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. In *Forty-first International Conference on Machine Learning*, 2024.
- Jian Gao, Chun Gu, Youtian Lin, Zhihao Li, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussians: Realistic point cloud relighting with brdf decomposition and ray tracing. In *European Conference on Computer Vision*, pp. 73–89. Springer, 2024.
- Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.
- Bu Jin, Yupeng Zheng, Pengfei Li, Weize Li, Yuhang Zheng, Sujie Hu, Xinyu Liu, Jinwei Zhu, Zhijie Yan, Haiyang Sun, et al. Tod3cap: Towards 3d dense captioning in outdoor scenes. In *European Conference on Computer Vision*, pp. 367–384. Springer, 2024.
- Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv preprint arXiv:2509.13414*, 2025.
- Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
- Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. Wildgaussians: 3d gaussian splatting in the wild. *arXiv preprint arXiv:2407.08447*, 2024.
- Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, et al. Uniscene: Unified occupancy-centric driving scene generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11971–11981, 2025.
- Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5521–5531, 2022.
- Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6498–6508, 2021.

- Junchen Liu, Wenbo Hu, Zhuo Yang, Jianteng Chen, Guoliang Wang, Xiaoxue Chen, Yantong Cai, Huan-ang Gao, and Hao Zhao. Rip-nerf: Anti-aliasing radiance fields with ripmap-encoded platonic solids. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.
- Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pp. 800–809. IEEE, 2024.
- Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7210–7219, 2021.
- Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
- Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R Qi, Xincheng Yan, Scott Ettinger, and Dragomir Anguelov. Motion inspired unsupervised perception and prediction in autonomous driving. In *European Conference on Computer Vision*, pp. 424–443. Springer, 2022.
- Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5865–5874, 2021.
- Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10318–10327, 2021.
- Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.
- Jiawei Ren, Cheng Xie, Ashkan Mirzaei, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, Huan Ling, et al. L4gm: Large 4d gaussian reconstruction model. *Advances in Neural Information Processing Systems*, 37:56828–56858, 2024.
- Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16632–16642, 2023.
- Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020.
- Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, Joao F Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. *arXiv preprint arXiv:2406.04343*, 2024.
- Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pp. 1–18. Springer, 2024.
- Beiwen Tian, Mingdao Liu, Huan-ang Gao, Pengfei Li, Hao Zhao, and Guyue Zhou. Unsupervised road anomaly detection with language anchors. In *2023 IEEE international conference on robotics and automation (ICRA)*, pp. 7778–7785. IEEE, 2023.
- Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14895–14904, 2024.

- Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025a.
- Nan Wang, Xiaohan Yan, Xiaowei Song, and Zhicheng Wang. Semantic-guided gaussian splatting with deferred rendering. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025b.
- Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.
- Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khanelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023.
- Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20310–20320, 2024.
- Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. Difx3d+: Improving 3d reconstructions with single-step diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26024–26035, 2025.
- Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, et al. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CAAI International Conference on Artificial Intelligence*, pp. 3–15. Springer, 2023.
- Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.
- Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthspat: Connecting gaussian splatting and depth. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 16453–16463, 2025.
- Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision*, pp. 156–173. Springer, 2024.
- Ziyang Yan, Wenzhen Dong, Yihua Shao, Yuhang Lu, Haiyang Liu, Jingwen Liu, Haozhe Wang, Zhe Wang, Yan Wang, Fabio Remondino, et al. Renderworld: World model with self-supervised 3d label. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6063–6070. IEEE, 2025.
- Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023.
- Jiawei Yang, Jiahui Huang, Yuxiao Chen, Yan Wang, Boyi Li, Yurong You, Apoorva Sharma, Maximilian Igl, Peter Karkus, Danfei Xu, et al. Storm: Spatio-temporal reconstruction model for large-scale outdoor scenes. *arXiv preprint arXiv:2501.00602*, 2024a.
- Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20331–20341, 2024b.
- Baijun Ye, Minghui Qin, Saining Zhang, Moonjun Gong, Shaoting Zhu, Zebang Shen, Luan Zhang, Lu Zhang, Hao Zhao, and Hang Zhao. Gs-occ3d: Scaling vision-only occupancy reconstruction for autonomous driving with gaussian splatting. *arXiv e-prints*, pp. arXiv–2507, 2025.

- Botao Ye, Sifei Liu, Haoifei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. *arXiv preprint arXiv:2410.24207*, 2024.
- Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022.
- Shiran Yuan and Hao Zhao. Slimmerf: Slimmable radiance fields. In *2024 International Conference on 3D Vision (3DV)*, pp. 64–74. IEEE, 2024.
- Bowei Zhang, Lei Ke, Adam W Harley, and Katerina Fragkiadaki. Tapip3d: Tracking any point in persistent 3d geometry. *arXiv preprint arXiv:2504.14717*, 2025.
- Dongbin Zhang, Chuming Wang, Weitao Wang, Peihao Li, Minghan Qin, and Haoqian Wang. Gaussian in the wild: 3d gaussian splatting for unconstrained image collections. In *European Conference on Computer Vision*, pp. 341–359. Springer, 2024a.
- Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.
- Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pp. 1–19. Springer, 2024b.
- Saining Zhang, Baijun Ye, Xiaoxue Chen, Yuntao Chen, Zongzheng Zhang, Cheng Peng, Yongliang Shi, and Hao Zhao. Drone-assisted road gaussian splatting with cross-view uncertainty. *arXiv preprint arXiv:2408.15242*, 2024c.
- Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21634–21643, 2024.

A IMPLEMENTATION DETAILS

APPENDIX A.1 — DATA PREPROCESSING

We construct dynamic masks based on the LiDAR-based 3D bounding-box annotations of the Waymo Open DatasetSun et al. (2020), which include tracking identifiers. Specifically, we first transform the center of each 3D box from the ego-vehicle coordinate frame to the global coordinate system, and aggregate per-object temporal trajectories using the provided timestamps. Object velocities are then estimated, and category-specific thresholds are applied to determine dynamic instances (pedestrians: > 0.2 m/s; vehicles: > 0.5 m/s). For objects identified as dynamic, we project their 3D bounding boxes onto the corresponding camera planes to obtain accurate 2D bounding boxes for each frame. These 2D boxes are subsequently used as prompts for an *off-the-shelf* instance segmentation model Ravi et al. (2024), combined with temporal information to propagate masks and obtain temporally consistent, object-level instance masks. Static background and sky masks are produced using a semantic segmentation model Xie et al. (2021), whose semantic outputs are further employed in downstream scene understanding tasks.

APPENDIX A.2 — BASELINE IMPLEMENTATIONS

Baseline selection. For all per-scene reconstruction methods shown in Table 1, we selected ten representative baselines. Among them, PVGChen et al. (2023), DeformableGSYang et al. (2024b), 3DGSKerbl et al. (2023), and STORMYang et al. (2024a) are reconstruction methods implemented on the Waymo Open DatasetSun et al. (2020); MVSplatChen et al. (2024b) and DepthSplatXu et al. (2025) are feedforward inference networks with strong empirical performance; NoPoSplatYe et al. (2024) is a *pose-free* inference framework that does not require camera pose inputs. In addition, we include EmerNeRFYang et al. (2023), LGMTang et al. (2024), and GS-LRMZhang et al. (2024b) from STORM’s reproduced results. For some baselines, reproduced outputs reported by STORM are used directly.

Task setup. The task in Tab.1 and Fig.3 is short-sequence reconstruction and prediction: given four input frames (id = 0, 2, 4, 6), reconstruct eight frames (id = 0–7). This represents a basic scene reconstruction and novel view synthesis (NVS) task. Across the ten experiments, methods that require iterative fitting are uniformly trained/fitted for 5,000 iterations; feedforward reconstruction methods are evaluated without this iterative training constraint. All evaluations are performed on the Scene Flow validation split of the Waymo Open DatasetSun et al. (2020), which contains 202 scenes.

Sky mask and depth metrics. Some methods do not explicitly reconstruct the sky; therefore depth-related metrics are computed only over non-sky regions. Sky masks are obtained from Waymo’s LiDAR data and filtered to ensure high confidence. For depth evaluation, methods without camera pose input, including NoPoSplat and ours, predict only relative depth, whose scale and offset may differ from ground truth. To ensure fair comparison, we perform a linear alignment of the predictions before computing the error, and then report the aligned depth RMSE (D-RMSE) within the valid mask regions.

Computation. During the training phase, the model was trained on the Waymo Open DatasetSun et al. (2020) using an eight-card H200 GPU configuration. The training process was completed in approximately 24 hours, with convergence achieved at around 5,000 iterations, as indicated by the stabilization of loss values and performance metrics on the validation set. In the experimental phase, to ensure direct comparability with STORMYang et al. (2024a) and its reproduced baseline methods, all evaluations in this study were exclusively conducted on NVIDIA A100 GPUs. This hardware alignment guarantees a consistent and fair comparison of computational performance and results across different models.

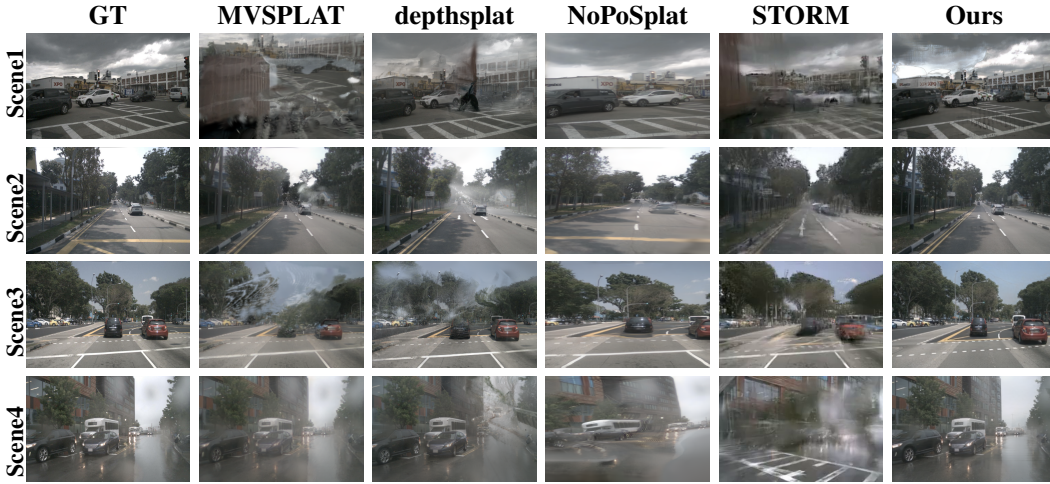


Figure 7: Zero-shot experiment on nuScenes and Argoverse2 datasets.

B ADDITIONAL EXPERIMENTAL RESULTS

APPENDIX B.1 — COMPARISON ON ADDITIONAL DATASETS

We evaluate the generalization of our model on the public nuScenes Caesar et al. (2020) and Argoverse2 Wilson et al. (2023) datasets. To provide a systematic comparison of cross-domain and in-domain performance, we design two complementary experiments: (1) **zero-shot evaluation**, where the model is trained on the Waymo Open Dataset Sun et al. (2020) and tested directly on nuScenes/Argoverse2 to assess cross-domain generalization; (2) **target-domain training evaluation**, where the model is trained and evaluated independently on nuScenes and Argoverse2 to measure upper-bound performance within each target domain.

Sampling and split details are as follows. For nuScenes (v1.0), the dataset contains approximately 1,000 driving scenes, each lasting roughly 20 s, with camera sampling at 12 Hz. We randomly sample 600 scenes for this study, using the first 500 scenes for training and the remaining 100 scenes for testing. For the Argoverse2 Sensor Dataset, which comprises roughly 1,000 annotated driving sequences and provides multi-modal sensor observations, the camera trigger frequency is approximately 20 Hz; we likewise select 600 sequences, with 500 used for training and 100 for evaluation.

To ensure comparability across datasets, all experiments adopt the same preprocessing and training configuration used for Waymo: camera images are uniformly downsampled/resampled to 518×518 , and data augmentation, optimizer settings, and training schedules are kept consistent. Models typically converge after roughly 1,000 epochs. Selected zero-shot inference examples on nuScenes and Argoverse2 are shown in Fig. 7.

APPENDIX B.2 — MORE QUALITATIVE RESULTS

Fig. 8 presents additional qualitative results of our method, showing full-image renderings, dynamic object renderings, and the predicted dynamic masks. Our approach effectively separates dynamic elements, such as vehicles and pedestrians, from the static background across diverse urban driving scenarios. The dynamic renderings align closely with ground-truth object locations, and the dynamic maps provide accurate object masks, demonstrating the effectiveness of our dynamic scene modeling and motion decomposition.

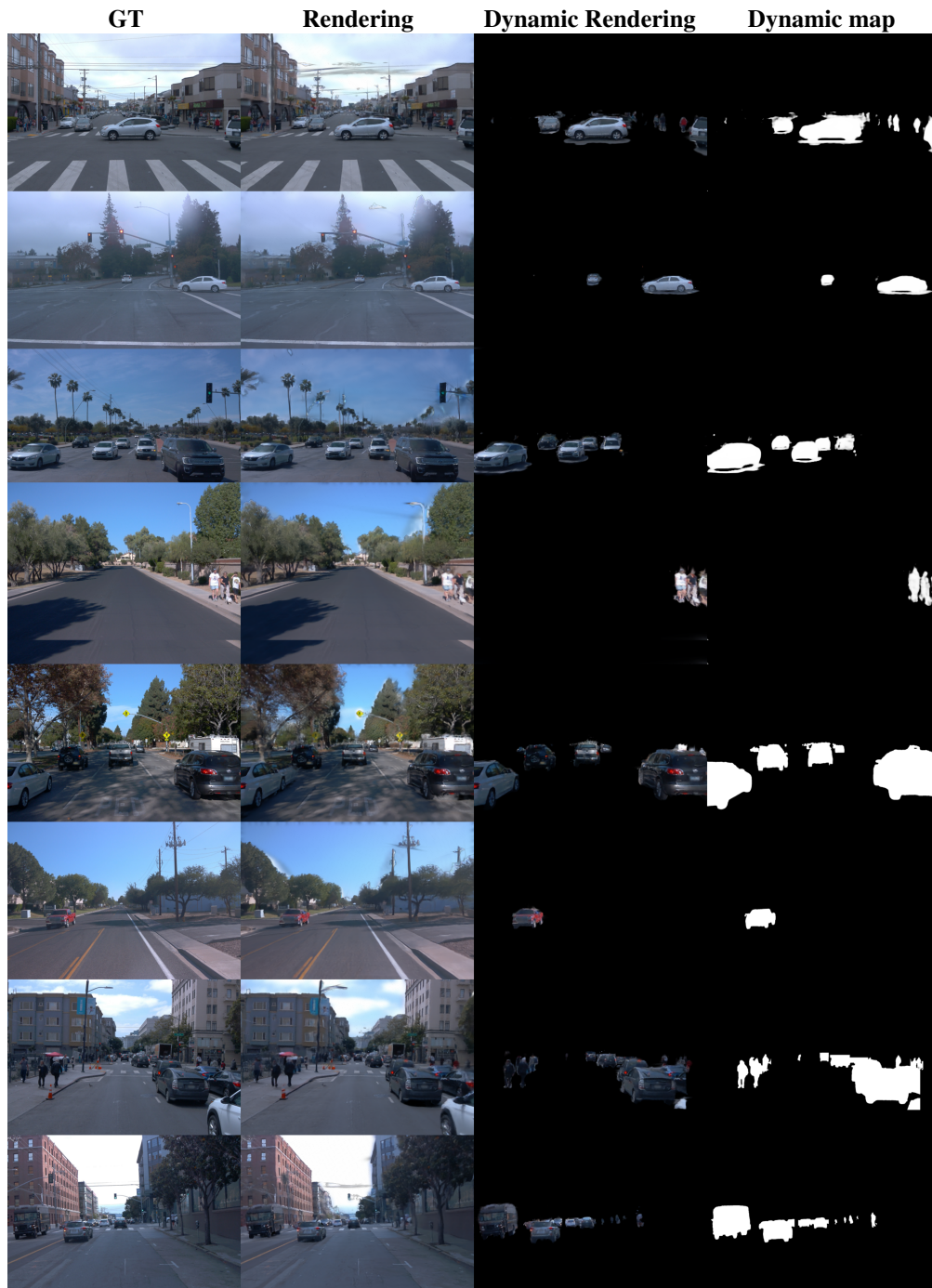


Figure 8: More Qualitative Results.