

MV²: Multi-View Multi-Vehicle Driving Dataset for Novel View Synthesis

Sanjay Bhargav Dharavath¹ Hanvitha Saraswathi Mukkamala¹ Faizan Farooq Khan² Ioannis Kakogeorgiou³ Aditya Arun⁴ C.V. Jawahar¹ and Zakaria Laskar^{2*}

¹ International Institute of Information Technology, Hyderabad

² Indian Institute of Science Education and Research, Thiruvananthapuram

³ King Abdullah University of Science and Technology (KAUST)

⁴ IIT, National Centre for Scientific Research “Demokritos”

⁵ Adobe MDSR, India

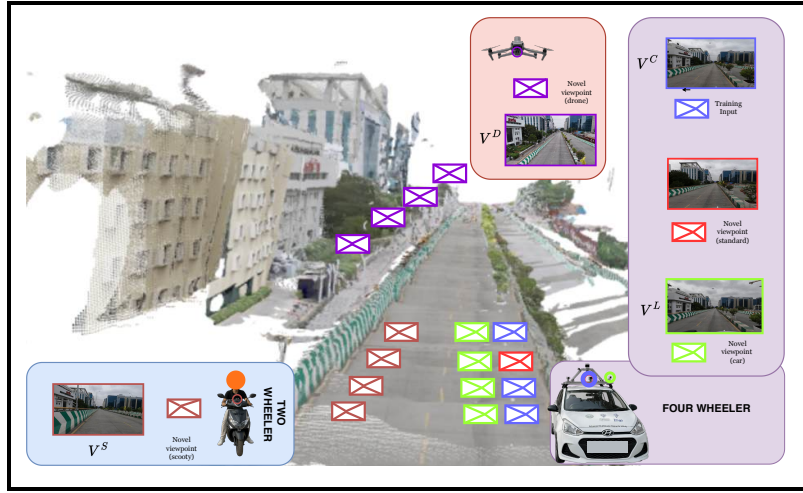


Fig. 1: Example images and camera trajectories from the proposed MV² dataset and benchmark. The dataset consists of camera trajectories from sensors placed on different vehicles - a four-wheeler, a two-wheeler and drone, capturing the same outdoor scene from diverse viewpoints. The diverse novel viewpoints (red, green, brown, violet cameras) sampled from different camera trajectories makes MV² a challenging Novel View Synthesis (NVS) benchmark, in contrast to existing works where novel views (red camera) are sampled from train trajectory (blue cameras), and only consider ground-ground or aerial-aerial NVS evaluation.

Abstract. Differentiable rendering has advanced novel view synthesis (NVS), yet applying it to real-world driving remains difficult due to sparse capture viewpoints, dynamic objects, and limited multi-trajectory data. We introduce the **Multi-View Multi-Vehicle (MV²)** dataset and benchmark for evaluating NVS models under large viewpoint changes

* Project Guide

in dynamic urban scenes. MV² features synchronized captures from a car, scooter, and drone, each following distinct yet synchronized trajectories. Training NVS methods on one vehicle’s camera stream and testing on another enables evaluation under substantially larger viewpoint variations than existing single-trajectory datasets. All sequences are registered via Structure-from-Motion and camera poses verified using manual pixel-level correspondence annotations, yielding 50 high-quality scenes with 12000 images. Benchmarking recent NVS and camera pose estimation methods shows that NVS performance degrades with increasing viewpoint disparity, and that feed-forward pose estimators notably lag behind optimization-based approaches, highlighting MV² as a rigorous testbed for NVS in driving. The dataset, benchmark protocol, and project resources are available at <https://mv2-dataset.github.io/>

1 Introduction

Differentiable rendering has driven major progress in novel view synthesis (NVS) - the task of generating unseen views of a scene from a set of posed input images. NVS holds strong potential for autonomous driving, enabling photorealistic simulations for training and evaluating perception and planning systems across diverse viewpoints. However, applying NVS to driving scenarios is challenging due to sparse camera coverage, dynamic objects, and large untextured regions like the sky. These issues are amplified when multi-sensor setups (e.g., LiDAR or multi-camera rigs [3, 12]) are unavailable. In this work, we evaluate state-of-the-art NVS methods [5, 17, 26, 40] in the demanding setting of monocular captures with large viewpoint variations in novel viewpoints.

Evaluating NVS methods in real-world driving scenarios is limited by the lack of purpose-built datasets. Most existing benchmarks [10, 33, 42, 44] repurpose standard driving datasets [2, 3, 12, 31], which capture a single camera trajectory through traffic scenes. Training and test frames are sampled from the same path, typically by taking every k -th frame, so these setups only test interpolation, not true generalization to unseen viewpoints. Concurrent efforts remain constrained: some depend on synthetic data [19], others assume static scenes by masking dynamic objects [20, 27], and Ma et al. [25] use FID as a proxy for cross-view evaluation, which measures perceptual realism rather than geometric consistency. Practically, recording identical dynamic scenes from multiple viewpoints with a single vehicle is infeasible due to traffic constraints.

To address this gap, we present the Multi-View Multi-Vehicle (MV²) dataset for benchmarking NVS methods in dynamic driving environments. The dataset is collected using a multi-platform configuration: a four-wheeler (car), a two-wheeler (scooter) and an aerial vehicle (drone), each equipped with one or multiple cameras - recording the same scenes from independent, real-world trajectories. The car and scooter follow separate lanes to provide lateral viewpoint variation, while the drone captures the same scene from above, offering occlusion-free coverage. Our setup enables a unique evaluation protocol: models trained on one vehicle’s camera stream are tested on their ability to synthesize views

from another vehicle’s perspective. The substantial viewpoint differences between the vehicle trajectories allow us to directly evaluate extrapolation quality, a key capability missing from existing benchmarks that only measure interpolation [10, 33, 42, 44].

MV² defines two evaluation setups based on the camera platform used for training NVS models. In the first setup **Eval-Car-Train**, training images come from one of the car’s onboard camera, while the images from scooter, drone and the car’s alternate cameras form the test set. In the second setup **Eval-Drone-Train**, we consider an aerial-to-ground NVS task, where training uses drone imagery, and testing is performed on images from car and scooter cameras. In both setups, we also consider an additional test set consisting of every k^{th} image from the respective training sequence in line with existing works [24, 38, 39]. This diversity in test images from multiple vehicles enables a comprehensive evaluation of NVS methods in synthesizing images from novel viewpoints - a critical requirement in driving simulation [24, 38, 39].

For each scene, images from all camera sensors are registered to a unified coordinate frame using Structure-from-Motion (SfM). After rigorous annotation and filtering to remove pose errors, the final dataset comprises 50 scenes with 12,000 images at 1080×1920 resolution. We evaluate per-scene optimization-based and feed-forward NVS methods, including static [17, 26], dynamic [6, 29], and learning-based 3D Gaussian Splatting (3DGS) approaches [5, 23, 40]. Key findings include:

- NVS performance consistently drops with an increase in camera baseline between train and test images, demonstrating the effectiveness of MV²’s viewpoint diversity.
- Aerial-to-ground NVS is a challenging setting, where both aerial-specific [32] and general NVS models struggle.
- Optimization-based methods consistently outperform feed-forward 3DGS methods for novel view synthesis (NVS) under both small and large viewpoint changes between training and test views.
- Optimization-based methods benefit from depth priors obtained using off-the-shelf depth estimation models [21] with appropriate filtering, providing an effective alternative to LiDAR. Additionally, accurate dynamic object segmentation masks further improve performance.
- Recent feed-forward camera pose estimators [7, 8, 16, 36] lag behind optimization-based methods [30], highlighting the value of MV² as a camera pose estimation benchmark.

2 Related Work

In this section, we review existing NVS datasets and methods in the context of driving scenes.

Driving Datasets for NVS. Datasets for novel view synthesis primarily consist of densely collected multi-view images of small-scale scenes [1, 26]. These datasets



Fig. 2: Sample multi-view scenes captured across diverse environments. Each column represents a distinct scene, while each row corresponds to a different acquisition viewpoint V^C , V^D , and V^S . The colored borders highlight the viewpoint source of each image, helping visualize cross-view consistency across sensors.

consist of small-scale in-bounded scenes captured with a circular camera trajectory around a given object or part of the scene. In contrast, outdoor scenes face challenges such as unbounded large-scale scenes, the presence of distant objects such as sky, and dynamic objects such as vehicles and pedestrians. Several driving datasets, *e.g.* KITTI [12], KITTI-360, vKITTI [2], Waymo Open Dataset [31], nuScenes [3] and Argoverse 2 [11, 37] have been repurposed for evaluating novel view synthesis of driving scenes. These datasets consist of multi-view images of the scene captured from a forward moving camera. Additional sensors such as GPS, IMU and importantly LiDAR are also used to estimate camera trajectory (pose) and an initial 3D model of the scene. Unlike existing NVS datasets with dynamic objects [28] captured from diverse viewpoints, existing driving datasets are captured from a single vehicle trajectory. This limitation is also addressed in recent concurrent works [19, 20]. While XLD [19] uses asynthetic dataset based on CARLA, MTGS [20] and Para-lane [27] collect multiple traversals through the same scene, failing to *capture dynamic objects from multiple viewpoints*. In contrast, the proposed MV^2 dataset consists of real-world images capturing the *dynamic* driving scene from the viewpoints of on-road and aerial vehicles. Statistics and example images of dynamic objects are presented in Section 10 in the Supplementary and Fig. 2.

Aerial-to-ground datasets of outdoor driving scenes are limited. Existing aerial datasets do not provide pose information. Recently, the proposed co-registered aerial-to-ground image dataset [35] is restricted to scenes from MegaDepth [35] and mainly functions as a training set. Similarly, the dataset introduced in drone-Splat [32] only considers the aerial-to-aerial NVS problem. In contrast to existing aerial-to-ground datasets for NVS, proposed MV^2 provides complementarity by covering a large number of outdoor driving scenes, and provides co-registered aerial-to-ground camera poses.

NVS Methods for Autonomous Driving. Recent advances in neural rendering have been driven by NeRF-based volumetric methods and point-based differentiable rendering frameworks [1, 26, 41]. Building on these ideas, 3D Gaussian Splatting (3DGS) [17] introduced an explicit, continuous scene representation that enables real-time radiance field rendering through tile-based rasterization. Following its success, numerous approaches have adapted 3DGS for large-scale driving scenes and dynamic urban environments [6, 15, 29, 42, 47]. For example, PVG [6] models temporal dynamics via periodic vibration fields; Street Gaussians [42] jointly represent static and dynamic components using semantic logits; and DrivingGaussian [47] introduces composite dynamic graphs for multi-object motion modeling. More recent self-supervised extensions, such as PVG [6], DeSiRe-GS [29] and S³Gaussians [15], further improve static-dynamic decomposition and surface fidelity without dynamic object mask supervision.

Beyond scene-specific optimization, feed-forward generalizable 3DGS networks aim to predict Gaussian parameters directly from multi-view inputs. Methods such as PixelSplat [4], MVSplat [5], MonoSplat [23], and DepthSplat [40] enable efficient novel-view synthesis by learning scene-agnostic priors, significantly reducing optimization time compared to per-scene training. Recent approaches, such as STORM [43] extend feed-forward NVS methods to driving scenarios but require LiDAR input.

Benchmarking all driving-specific NVS methods is challenging due to the high computational requirements - for example, DesireGS [29], PVG [6], Street-Gaussians [42] and similar optimization-based methods require up to 4 hours training time on a single scene. As such, we benchmark only a subset of these representative approaches [5, 6, 17, 23, 26, 29, 40] based on their prominence and recency under a unified evaluation protocol for real-world, wide-baseline ground-to-ground and aerial-to-ground settings.

3 Multi-View Multi-Vehicle Dataset

In this section, we describe the collection and composition process of our proposed MV² dataset. We adopt a multi-vehicle setup (Section 3.1) to collect a set of driving videos following specific protocols (Section 3.2). Thereafter, the collected set of videos are divided into train and test sets (Section 3.3). Each image in MV² is registered to a single global coordinate system using a state-of-the-art camera pose estimation pipeline (Section 3.4). The estimated camera poses are then verified using epipolar geometry (Section 3.5).

MV² follows the standard novel-view synthesis setup, where train images and train camera poses are used to obtain a 3D scene representation [17, 26, 40] of the scene. The scene representation is then rendered from the viewpoint of the test images using the test camera poses. Finally, these rendered images are compared with ground-truth test images to obtain NVS performance metrics.

3.1 Sensor Setup

All the vehicles are equipped with GoPro 10 camera sensors. Two cameras, {L(-eft), C(-entral)} are fixed to the camera rig mounted on the car. We denote the set of images collected using these three cameras as V^L, V^C . Similarly, we denote the images collected using sensors, S and D placed on the scooty and the drone as V^S and drone as V^D respectively. All the sensors on the vehicles are forward-facing. Each camera captures images at a resolution of 1080×1980 at 60 FPS. The images collected from the above sensors are synchronized using the wall-clock.

3.2 Dataset Collection

The proposed dataset is collected by driving the two vehicles through various urban and semi-urban roads for two hours over a period of 5 days. Lateral and aerial adjacency between the vehicles was maintained during data collection. The video data collected from each sensor is sampled at 2 FPS and divided into smaller video segments consisting of 100 frames. Video segments of roads in high traffic congestion, under-tunnel, misaligned lateral adjacency between vehicles and red-light traffic points are excluded. This results in a total of 200 scenes, with 5 camera sequences each.

3.3 Evaluation Setup

We consider two evaluation setups: **Eval-Car-Train** and **Eval-Drone-Train**, based on which vehicle camera stream is used for training the NVS models. For each setup, different test sets are formed by sampling images from the train sequence and image sequences from other vehicles. The test sets are referred to a $T_{X \rightarrow Y}$, where X and Y are the train and test cameras. For the **Eval-Car-Train** setup, we consider the following train and test splits.

Train: Images collected from the center camera sensor, V^C .

Test $T_{C \rightarrow C}$: Existing works [10][11][34][42][44] generate training and test sets by subsampling the train trajectory C , typically selecting every k -th frame for testing (e.g., $k=10$ for Argoverse [11] and Waymo (NOTR) [44], $k=4$ for KITTI and Virtual KITTI2 [34]). We adopt this protocol with $k=5$, forming the test set, $T_{C \rightarrow C}$, with the highest viewpoint overlap with the training images. The train set introduced above excludes these frames.

Test $T_{C \rightarrow L}$: Images from the left car-mounted camera sequence, V^L , sampled at indices $\{0, 5, 10, \dots, 95\}$, forms the test set, **Test $T_{C \rightarrow L}$** . This set introduces a small lateral baseline relative to the train sequence, resulting in moderate viewpoint variation compared to the test set, $T_{C \rightarrow C}$.

Test $T_{C \rightarrow S}$: To introduce a more challenging setup, MV² adds images from scooty trajectory V^S as the cross-vehicle test set, **Test $T_{C \rightarrow S}$** . The camera sequence, V_S is uniformly sampled at $\{0, 5, 10, \dots, 95\}$. Unlike the previous test sets, this cross-vehicle test set has a significant camera baseline with the training views, providing a rigorous assessment of cross-trajectory generalization.

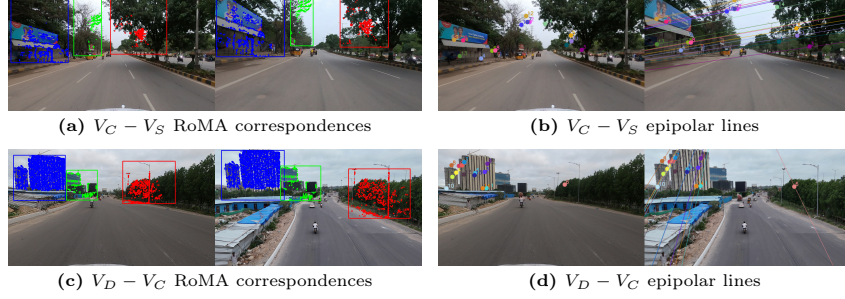


Fig. 3: Annotation and Pose Evaluation. Representative image pairs from $V_C - V_S$ (top row) $V_C - V_D$ (bottom row) with pixel correspondences obtained from annotated region correspondence and RoMA matches and epipolar lines (of randomly sampled RoMA correspondences for visualization purpose). The alignment of correspondences along epipolar lines demonstrates strong geometric consistency and reliable cross-view registration even under wide baselines.

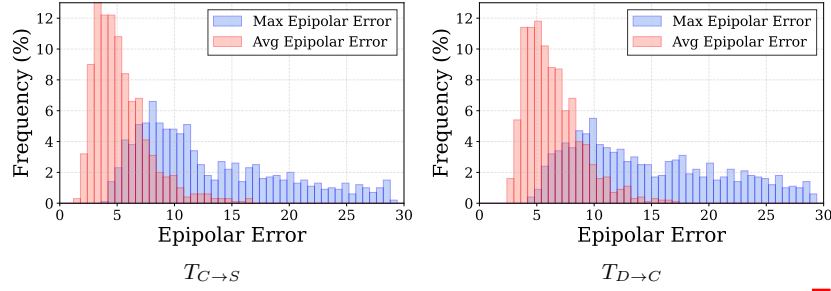


Fig. 4: Epipolar error distribution of mean and maximum epipolar errors (2) over 50 scenes. Left: $T_{C \rightarrow S}$, Right: $T_{D \rightarrow C}$.

For the **Eval-Drone-Train** setup, the training set consists of images captured from the drone camera V^D . The test sets are composed using images from intra-vehicle and cross-vehicle sensors: $T_{D \rightarrow D}$, $T_{D \rightarrow C}$ and $T_{D \rightarrow S}$, obtained by sampling V^D , V^C and V^S , respectively, at the 5-frame interval $(0, 5, 10, \dots, 95)$. The following pairs of test sets, $\{(T_{C \rightarrow C}, T_{D \rightarrow C}), (T_{C \rightarrow S}, T_{D \rightarrow S}), (T_{C \rightarrow D}, T_{D \rightarrow C})\}$ seem equivalent but are not due to the pose verification step in Section 3.5

3.4 Camera Parameters Estimation

We describe the process used to estimate both the intrinsic and camera poses (extrinsic) parameters of the images in the train and three test sets.

Intrinsic Calibration. We use a standard checkerboard calibration target to estimate the intrinsic parameters of the cameras, including focal lengths (f_x , f_y) and principal points (c_x , c_y). This calibration is performed prior to any pose estimation. The intrinsics are further refined in the following camera pose estimation steps.

Camera Pose Estimation. We estimate camera poses using a two-step pipeline. First, training images for each setup (**Eval-Car-Train** and **Eval-Drone-Train**) are registered with COLMAP using its standard configuration. Next, test images are localized with COLMAP’s localization module, using the 15 nearest training images as neighbors. While single-sequence (per-sensor) camera pose estimation produces accurate poses, jointly estimating train and test poses leads to degraded results many images from *both the train and test set* fail to register or yield incorrect poses highlighting the large viewpoint diversity in MV². The two-step strategy thus ensures (i) accurate, independently estimated training poses and (ii) the ability to filter out test images with failed or inconsistent localization without affecting the train poses. Details of this filtering process are provided next.

3.5 Pose Verification

Camera poses estimated by COLMAP are subject to errors arising from key-point mismatches and inaccuracies in Structure-from-Motion (SfM) optimization. Since ground-truth poses are unavailable, absolute pose accuracy cannot be directly verified. Instead, we assess the accuracy of each train and test image’s absolute pose by evaluating its *relative pose* with respect to its nearest training image. However, relative pose evaluation alone is insufficient to guarantee the correctness of an image’s absolute pose if the nearest training image has an inaccurate absolute pose, its relative pose with another image can still appear correct. To address this ambiguity, we first verify the accuracy of training image poses, as described below.

Train Pose Verification. We model the training image sets, V_C and V_D , as sequential pose graphs, where each node corresponds to a training image camera pose and each edge represents the relative pose between consecutive images. Our hypothesis is that if every edge (relative pose) is accurate, then the corresponding nodes (absolute poses) can also be considered accurate. We evaluate each relative pose using manually annotated pixel correspondences as described next. To address the issue of camera pose drift, we also evaluate the relative poses between the first and last or the 50th image based on covisibility.

Given a correspondence $(\mathbf{x}_1, \mathbf{x}_2)$ between two cameras with intrinsics $\mathbf{K}_1, \mathbf{K}_2$ and extrinsics $(\mathbf{R}_1, \mathbf{t}_1)$ and $(\mathbf{R}_2, \mathbf{t}_2)$, the *epipolar error* is defined as

$$e = |\mathbf{x}_2^\top \mathbf{F} \mathbf{x}_1|, \quad (1)$$

, where $\mathbf{F} = \mathbf{K}_2^{-\top} [\mathbf{t}]_{\times} \mathbf{R} \mathbf{K}_1^{-1}$, $\mathbf{R} = \mathbf{R}_2 \mathbf{R}_1^\top$, $\mathbf{t} = \mathbf{t}_2 - \mathbf{R} \mathbf{t}_1$, $[\mathbf{t}]_{\times}$ is skew-symmetric matrix [13].

For each candidate relative pose between an image pair, we compute the mean and maximum epipolar errors using N manually annotated correspondences as

$$e_a = \frac{1}{N} \sum_i e_i, \quad e_m = \max_{i \in \{1, \dots, N\}} e_i. \quad (2)$$

A relative pose is deemed inaccurate if $e_m > 30$. For each of the 200 sequences, we verify all consecutive training image pairs in V_C and V_D ; a sequence is retained only if all relative poses satisfy $e_m \leq 30$ in both training sets.

Test Pose Verification. For sequences with validated training camera poses, we then evaluate the relative pose between each test image and its nearest training image using the same epipolar error criterion, based on annotated correspondences. The nearest training image is selected from either V_C or V_D , depending on the evaluation setup (**Eval-Car-Train** or **Eval-Drone-Train**). A test image is retained if its relative pose satisfies $e_m \leq 30$.

Correspondence Annotation is performed in two stages. First, human annotators identify corresponding bounding boxes that observe the same 3D region in an image pair. These regions are cropped, resized, and processed using a dense neural matcher (e.g., RoMA [9]) to obtain dense pixel correspondences within each bounding box. The resulting correspondence set is denoted as $\mathcal{X}_{\text{RoMA}}$ and shown in Fig. 3 (a) and (c). All candidate image pairs in both the training and test sets are subsequently validated using epipolar consistency checks computed independently over correspondences in $\mathcal{X}_{\text{RoMA}}$. An example image showing epipolar lines for a subset of correspondences in $\mathcal{X}_{\text{RoMA}}$ is shown in Fig. 3 (b) and (d).

Statistics. Out of 200 recorded sequences, 50 satisfy the training-set threshold (maximum epipolar error < 30). Test images meeting the same criterion relative to validated train poses are retained. Example annotated correspondences with their epipolar lines are shown in Figure 3 and the distribution of mean and maximum epipolar errors for the selected sequences are shown in Figure 4. Statistics of selected test images in each test set are presented in Section 8 in the Supplementary.

Time Synchronization between sensors is validated using epipolar consistency checks on pixel correspondences annotated on dynamic objects that are simultaneously observed by multiple sensors. For each sequence, we randomly select two image pairs from the beginning and end segments and manually annotate four pixel correspondences on the dynamic object in each pair. Epipolar errors computed from these correspondences using the estimated camera parameters are used to verify synchronization. Example pixel annotations, epipolar lines and epipolar error distributions are shown in the Supplementary.

4 Benchmark Methods

Static NVS Methods. We evaluate classic novel-view synthesis methods, Nerf [26] and 3DGS [17]. In particular, we evaluate the NerfStudio variants, *nerfacto* and *splatfacto* respectively. For 3DGS, we initialize the centers of 3D Gaussians using the sparse 3D point cloud of the scene computed using SfM [30] (cf. Section 3.4).

Dynamic NVS Methods. Standard NVS methods are designed for static scenes, whereas driving scenes contain dynamic objects. We evaluate recent 3DGS-based methods for such settings: droneSplat [32] and PVG [6]. The latter two require LiDAR supervision and dynamic object masks. For segmentation

Model	PSNR $\uparrow$			SSIM $\uparrow$			LPIPS $\downarrow$		
	$T_{C \rightarrow C}$	$T_{C \rightarrow L}$	$T_{C \rightarrow S}$	$T_{C \rightarrow C}$	$T_{C \rightarrow L}$	$T_{C \rightarrow S}$	$T_{C \rightarrow C}$	$T_{C \rightarrow L}$	$T_{C \rightarrow S}$
Depthsplat (12v)	18.05	17.13	14.39	0.33	0.31	0.25	0.28	0.36	0.40
Monosplat (12v)	18.43	18.24	15.71	0.42	0.38	0.28	0.26	0.29	0.33
Mvsplat (12v)	16.33	14.08	12.88	0.42	0.38	0.35	0.44	0.45	0.54
Dronesplat	21.25	19.50	17.19	0.74	0.70	0.67	0.45	0.45	0.48
3DGS	24.22	22.52	16.78	0.71	0.69	0.64	0.38	0.39	0.47
NeRF	20.93	19.38	15.73	0.64	0.58	0.52	0.42	0.47	0.59
DesireGS	25.96	24.31	19.48	0.69	0.61	0.55	0.23	0.31	0.34
PVG	27.01	25.44	20.23	0.72	0.64	0.58	0.20	0.28	0.31
Model	PSNR $\uparrow$			SSIM $\uparrow$			LPIPS $\downarrow$		
	$T_{D \rightarrow D}$	$T_{D \rightarrow C}$	$T_{D \rightarrow S}$	$T_{D \rightarrow D}$	$T_{D \rightarrow C}$	$T_{D \rightarrow S}$	$T_{D \rightarrow D}$	$T_{D \rightarrow C}$	$T_{D \rightarrow S}$
Depthsplat (12v)	16.78	9.09	7.87	0.49	0.16	0.14	0.45	0.52	0.56
Monosplat (12v)	18.54	10.21	8.32	0.38	0.25	0.23	0.47	0.49	0.52
Mvsplat (12v)	16.50	9.20	7.06	0.37	0.20	0.18	0.54	0.54	0.56
Dronesplat	16.43	12.16	12.33	0.53	0.51	0.50	0.64	0.71	0.72
3DGS	29.13	10.36	11.42	0.86	0.44	0.43	0.22	0.52	0.46
NeRF	21.30	11.01	11.54	0.65	0.34	0.35	0.67	0.56	0.61
DesireGS	26.84	12.01	13.05	0.76	0.44	0.43	0.40	0.60	0.55
PVG	28.12	12.65	13.82	0.80	0.47	0.46	0.37	0.57	0.52

Table 1: NVS methods are compared on different test sets in evaluation setups **Eval-Car-Train** (top) and **Eval-Drone-Train** (bottom). 12v indicates the number of input views for feed-forward 3DGS methods. Remaining views {2,6} are presented in Section 9 in Supplementary.

masks, we intersect outputs from droneSplat (which may contain outliers) and SAM [18] prompted for vehicles and pedestrians, ensuring only dynamic regions are retained (*cf.* Fig. 14).

To replace LiDAR supervision, we use depth from the off-the-shelf estimator DaV3 [21]. The predicted depth maps are resized to the training-image resolution and aligned to COLMAP SfM depth using an affine scale-and-shift transformation:

$$\mathcal{L}_{\text{depth}} = \sum_{p \in \mathcal{P}} \|\alpha D_{\text{DaV3}}(p) + \beta - D_{\text{SfM}}(p)\|_1, \quad (3)$$

where $\mathcal{P}$ denotes pixels with valid SfM depth, and α, β are estimated per scene.

Feed-Forward 3DGS Methods directly predict pixel-aligned 3D Gaussian Splats from input context images using a neural network trained on thousands of image sequences [22, 46]. We consider three methods in particular, Depthsplat [40], Monosplat [23] and Mvsplat [5]. We consider three setups of 2,6 and 12 context views for each of these feed-forward 3DGS methods.

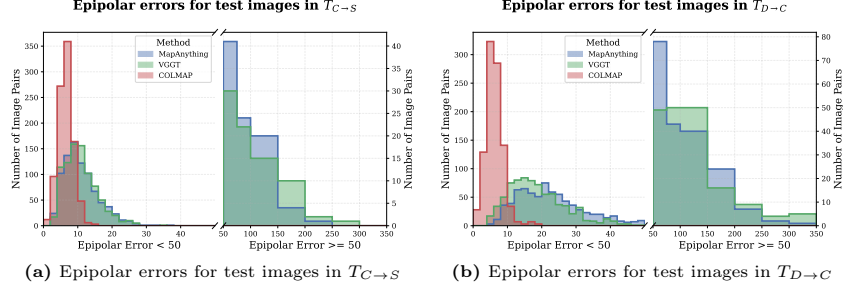


Fig. 5: Camera Pose Estimation. COLMAP produces low average epipolar errors (< 30), whereas feed-forward pose estimators yield significantly larger errors (> 50). The number of high-error cases is substantially greater in $T_{D \rightarrow C}$ due to the wide-baseline aerial-ground viewpoint gap.

5 Experiments

5.1 Evaluation Protocol

We follow standard evaluation protocol measuring three metrics to assess the quality of rendered images - PSNR, SSIM, LPIPS [45] that evaluates the quality at pixel-level, patch-level and feature-level respectively. We compute these metrics for each image in the test sets in respective evaluation setups **Eval-Car-Train** and **Eval-Drone-Train**.

5.2 Method Comparison

We evaluate multiple NVS methods and summarize the results for both evaluation setups in Table 1. On the standard test set, $T_{C \rightarrow C}$ in the **Eval-Car-Train** setup—where training and test views are sampled from the same car sequence V_C - PVG [6] achieves the highest performance with PSNR | SSIM | LPIPS = 27.01|0.72|0.20. In contrast, PVG exhibits a significant performance drop on the proposed cross-vehicle test sets, $T_{C \rightarrow S}$, where training and test images originate from car and scooty sensors, respectively. This degradation is consistent across all evaluated baselines. A similar trend is observed between $T_{C \rightarrow L}$ and $T_{C \rightarrow S}$, demonstrating that having an alternate camera sensor on the car is insufficient to assess the extrapolation capability of NVS models.

The same observation holds in the **Eval-Drone-Train** setup: 3DGS and PVG achieve the best results on the standard test set $T_{D \rightarrow D}$, but their performance declines on the cross-vehicle test sets $T_{D \rightarrow S}$ and $T_{D \rightarrow C}$. Interestingly, the aerial-specific method droneSplat [32] is outperformed by the general-purpose NVS methods 3DGS and PVG. The results on dynamic objects NVS (Section 10 in Supplementary) follow the same trend as the full-image evaluation: performance consistently degrades with increasing viewpoint variation.

Accurate NVS is essential for driving simulation, which requires smooth and continuous camera trajectories (e.g., lane changes). Such simulations are only

meaningful for downstream applications, including autonomous driving evaluation, if synthesized views are photorealistic and geometrically consistent. The poor NVS performance on $T_{C \rightarrow S}$ suggests that even simple scenarios, such as simulating a lane change from the car lane to the scooty lane, would result in unrealistic renderings due to poor synthesized scooty-view quality.

5.3 Dataset Comparison and Depth Filtering

We compare PVG on MV^2 and Waymo Open Dataset (WOD) [31] to analyze whether MV^2 remains comparable to standard driving NVS benchmarks under the same-trajectory $T_{C \rightarrow C}$ setting. On WOD [31], PVG [6] is commonly evaluated with LiDAR supervision and train/test views sampled from the same vehicle trajectory. Despite using single-camera input and monocular DaV3 depth instead of LiDAR, PVG obtains comparable $T_{C \rightarrow C}$ performance on MV^2 (Table 2), indicating that our benchmark remains consistent with existing driving NVS settings in the standard interpolation regime.

Since WOD does not provide a true cross-vehicle $T_{C \rightarrow S}$ setup, we also perform a qualitative cross-lane study using PVG. Fig. 6 shows that cross-lane renderings on WOD exhibit degradation similar to the $T_{C \rightarrow S}$ renderings on MV^2 . This supports our main claim that wide-baseline cross-lane synthesis remains challenging even when evaluated on existing driving datasets.

We further study the role of depth supervision and filtering. For this analysis, all WOD subset experiments use a matched single-camera PVG setup. We compare LiDAR depth, COLMAP depth (no DaV3), DaV3 with IQR filtering, and DaV3 with [10, 30]m filtering. As shown in Table 2, restricting DaV3 supervision to the [10, 30]m range improves results on both WOD and MV^2 . On MV^2 , this improves both $T_{C \rightarrow C}$ and $T_{C \rightarrow S}$, showing that depth filtering is important for monocular-depth-based PVG. However, the gap between $T_{C \rightarrow C}$ and $T_{C \rightarrow S}$ remains, confirming that improved depth filtering strengthens the baseline but does not remove the cross-vehicle wide-baseline NVS challenge. Additional details on dataset splits for WOD and MV^2 datasets used in this ablation alongwith the motivation for depth filtering are provided in the Supplementary Section 11.1.



Fig. 6: Lane-change scenario rendering comparison. Columns 1–2 show renderings from the MV^2 dataset ($T_{C \rightarrow C}$ and $T_{C \rightarrow S}$ views), while Columns 3–4 show renderings from the Waymo dataset ($T_{C \rightarrow C}$ and cross-lane views). Both examples illustrate a vehicle performing a lane-change maneuver.

Setup	Depth Supervision	WOD			MV ²		
		PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
$T_{C \rightarrow C}$	LiDAR multi-view	28.45	0.912	0.145	–	–	–
	LiDAR single-view	28.66	0.792	0.121	–	–	–
	COLMAP	27.84	0.765	0.145	24.72	0.655	0.238
	DaV3 + IQR	26.95	0.735	0.205	25.88	0.680	0.210
	DaV3 [10, 30]m	29.12	0.815	0.108	27.01	0.720	0.200
$T_{C \rightarrow S}$	DaV3 + IQR	–	–	–	19.43	0.550	0.330
	DaV3 [10, 30]m	–	–	–	20.23	0.580	0.310

Table 2: PVG comparison on WOD and MV² under different depth-supervision and filtering settings. LiDAR multi-view is the standard WOD PVG setup, while LiDAR single-view uses matched single-camera supervision. DaV3 + IQR uses monocular depth with IQR-based filtering, while DaV3 [10, 30]m retains only valid depths in the 10–30m range. Since WOD lacks a true cross-vehicle setup, $T_{C \rightarrow S}$ is reported only on MV².

Mask	Dynamic Region			Full Image		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
None	13.98	0.32	0.62	14.72	0.34	0.63
DS+SAM	19.93	0.52	0.38	21.24	0.58	0.30
GT	21.70	0.62	0.24	24.29	0.72	0.18

Table 3: PVG mask ablation on $T_{C \rightarrow S}$. DS+SAM denotes pseudo dynamic masks obtained from the DroneSplat+SAM pipeline, while GT denotes ground-truth dynamic-object masks.

5.4 Dynamic Mask Ablation

We evaluate the DS+SAM pseudo masks used for PVG dynamic-object supervision. These masks are obtained by intersecting DroneSplat motion masks with SAM-based vehicle and pedestrian masks; additional details of the dynamic-object annotation and mask construction are provided in the Supplementary Section 10. Against GT dynamic-object masks, DS+SAM pseudo masks achieve 63.69% IoU, 70.93% Precision, and 86.19% Recall.

We further evaluate PVG on the challenging $T_{C \rightarrow S}$ split under three settings: no masks, DS+SAM pseudo masks, and GT dynamic masks. Table 3 shows that DS+SAM pseudo masks substantially improve over no masking and approach the GT-mask upper bound, validating their use for dynamic-object supervision.

5.5 Camera Pose Estimation.

We use the proposed benchmark to evaluate recent feed-forward camera pose estimation methods [8, 16, 36] alongside the classic optimization-based approach, COLMAP. Specifically, we assess how accurately these methods estimate test image poses. As described in Section 3.4 COLMAP estimates test image poses

GT	3DGS	NeRF	DepthSplat	PVG
				
	21.49 0.73 0.33	17.89 0.51 0.61	14.33 0.39 0.53	25.91 0.72 0.19
				
	19.00 0.74 0.34	14.39 0.42 0.61	12.99 0.33 0.59	24.56 0.64 0.29
				
	13.96 0.42 0.63	18.05 0.58 0.46	9.03 0.28 0.71	20.02 0.75 0.41
				
	20.08 0.66 0.39	14.88 0.46 0.62	10.61 0.27 0.69	19.86 0.71 0.42
				
	19.89 0.58 0.31	17.08 0.54 0.51	19.65 0.59 0.26	28.34 0.86 0.10
				
	18.12 0.50 0.48	16.43 0.50 0.63	19.01 0.53 0.29	28.41 0.88 0.13
				
	15.37 0.61 0.47	15.51 0.55 0.45	15.21 0.59 0.43	19.64 0.66 0.26
				
	15.24 0.55 0.53	15.39 0.55 0.44	15.67 0.56 0.39	18.53 0.58 0.31
				
	13.21 0.44 0.52	13.08 0.49 0.47	13.87 0.52 0.44	16.61 0.53 0.47
				
	13.42 0.51 0.39	15.23 0.54 0.30	14.3 0.46 0.41	16.98 0.50 0.35

Fig. 7: Qualitative comparison of novel view synthesis across methods. Columns show GT, 3DGS, NeRF, DepthSplat, and PVG. Rows correspond to rendering settings (top to bottom): $T_{C \rightarrow C} \times 2$, $T_{C \rightarrow S} \times 2$, $T_{D \rightarrow D} \times 2$, $T_{D \rightarrow C} \times 2$, and $T_{D \rightarrow S} \times 2$. Metrics (PSNR|SSIM|LPIPS) are shown below each rendering.

through camera localization with respect to the 15 nearest training images. Similarly, each feed-forward method receives the test image and its 15 nearest training images as input to predict corresponding train and test camera poses.

For evaluation, we pair each test image with its nearest training image and compute the average epipolar error over RoMA [9] pixel correspondences through the bounding box annotation (*cf.* Fig. 3, Section 3.5). The distribution of average epipolar errors (e_a) for test sets $T_{C \rightarrow S}$ and $T_{D \rightarrow C}$ is shown in Fig. 5. Feed-forward methods exhibit significantly higher epipolar errors, indicating less accurate pose estimation; larger epipolar errors correspond directly to larger pose errors. Detailed results are presented in Section 12 in the Supplementary.

Overall, the results reveal that (i) feed-forward pose estimation methods struggle under substantial viewpoint variation, and (ii) MV² also serves as a strong benchmark for evaluating camera pose estimation. While other optimization-based methods such as VIPE [14] exist, a comprehensive evaluation of such approaches is beyond the scope of this work and is left for future work.

5.6 Qualitative Analysis.

We present qualitative examples of rendered images by different NVS methods in Fig. 7 along with their respective NVS performance metrics. For the standard $T_{C \rightarrow C}$ setting, PVG achieves the best visual quality and metrics among the evaluated methods, with improved reconstruction of both static and dynamic scene elements. However, rendering quality drops substantially in cross-vehicle and aerial-to-ground setups, including $T_{C \rightarrow S}$, $T_{D \rightarrow C}$, and $T_{D \rightarrow S}$. These qualitative results further support our main observation that large viewpoint changes across vehicles remain challenging for current NVS methods. More qualitative examples are in Section 11 in the Supplementary.

6 Conclusions

We introduce MV², a challenging NVS dataset and benchmark for driving scenes with viewpoints more diverse than existing ground and aerial driving datasets. Our results show that driving-scene NVS remains unsolved and needs major improvements for applications such as simulation. While state-of-the-art dynamic NVS methods perform well in standard settings, training-free baselines often match them in challenging cross-vehicle setups. We also demonstrate MV²'s utility for camera pose estimation, showing that feed-forward state-of-the-art methods struggle under large viewpoint variations.

7 Acknowledgements

This work was supported by the Ministry of Electronics and Information Technology (MeitY), Government of India, through the NLTM-Bhashini project, and by the Anusandhan National Research Foundation (ANRF) under Grant ANRF/ECRG/2025/005736/ENS. AWS resources were provided by the National Infrastructures for Research and Technology (GRNET) and funded by the EU Recovery and Resilience Facility.

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR (2022)
2. Cabon, Y., Murray, N., Humenberger, M.: Virtual kitti 2. arXiv preprint arXiv:2001.10773 (2020)
3. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR (2020)
4. Charatan, D., Li, S.L., Tagliasacchi, A., Sitzmann, V.: pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In: CVPR (2024)
5. Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.J., Cai, J.: Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In: ECCV (2024)
6. Chen, Y., Gu, C., Jiang, J., Zhu, X., Zhang, L.: Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. arXiv preprint arXiv:2311.18561 (2023)
7. Dong, S., Wang, S., Liu, S., Cai, L., Fan, Q., Kannala, J., Yang, Y.: Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization. In: CVPR. pp. 16739–16752 (2025)
8. Duisterhof, B.P., Züst, L., Weinzaepfel, P., Leroy, V., Cabon, Y., Revaud, J.: MAST3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. In: 3DV (2025)
9. Edstedt, J., Nordström, D., Zhang, Y., Bökman, G., Astermark, J., Larsson, V., Heyden, A., Kahl, F., Wadenbäck, M., Felsberg, M.: Roma v2: Harder better faster denser feature matching (2025), <https://arxiv.org/abs/2511.15706>
10. Fischer, T., Kulhanek, J., Bulo, S.R., Porzi, L., Pollefeys, M., Kotschieder, P.: Dynamic 3d gaussian fields for urban areas. In: NeurIPS (2024)
11. Fischer, T., Porzi, L., Bulo, S.R., Pollefeys, M., Kotschieder, P.: Multi-level neural scene graphs for dynamic urban environments. In: CVPR (2024)
12. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012)
13. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
14. Huang, J., Zhou, Q., Rabeti, H., Korovko, A., Ling, H., Ren, X., Shen, T., Gao, J., Slepichev, D., Lin, C.H., Ren, J., Xie, K., Biswas, J., Leal-Taixe, L., Fidler, S.: Vipe: Video pose engine for 3d geometric perception. In: NVIDIA Research Whitepapers arXiv:2508.10934 (2025)
15. Huang, N., Wei, X., Zheng, W., An, P., Lu, M., Zhan, W., Tomizuka, M., Keutzer, K., Zhang, S.: S³Gaussian: Self-supervised street gaussians for autonomous driving. arXiv preprint arXiv:2405.20323 (2024)
16. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., et al.: MapAnything: Universal feed-forward metric 3d reconstruction. arXiv preprint arXiv:2509.13414 (2025)
17. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. TOG (2023)
18. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: ICCV (2023)
19. Li, H., Yuan, M., Zhang, Y., Wu, C., Zhao, C., Song, C., Feng, H., Ding, E., Zhang, D., Wang, J.: Xld: a cross-lane dataset for benchmarking novel driving view synthesis. In: arXiv preprint arXiv:2406.18360 (2024)

20. Li, T., Qiu, Y., Wu, Z., Lindström, C., Su, P., Nießner, M., Li, H.: Mtgs: Multi-traversal gaussian splatting. In: arXiv preprint arXiv:2503.12552 (2025)
21. Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views (2025), <https://arxiv.org/abs/2511.10647>
22. Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: D13dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In: CVPR (2024)
23. Liu, Y., Fan, K., Yu, W., Li, C., Lu, H., Yuan, Y.: Monosplat: Generalizable 3d gaussian splatting from monocular depth foundation models. In: CVPR (2025)
24. Ljungbergh, W., Tonderski, A., Johnander, J., Caesar, H., Åström, K., Felsberg, M., Petersson, C.: Neuroncap: Photorealistic closed-loop safety testing for autonomous driving. In: ECCV (2024)
25. Ma, X., Zhang, J., Lu, P., Xu, S., Pan, C.: Novel view synthesis under large-deviation viewpoint for autonomous driving. In: AAAI (2025)
26. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM (2021)
27. Ni, Z., Du, S., Hou, Z., Wu, C., Yang, S.: Para-lane: Multi-lane dataset registering parallel scans for benchmarking novel view synthesis. arXiv preprint arXiv:2502.15635 (2025)
28. Ni, Z., Du, S., Hou, Z., Wu, C., Yang, S.: Para-lane: Multi-lane dataset registering parallel scans for benchmarking novel view synthesis. In: 3DV (2025)
29. Peng, C., Zhang, C., Wang, Y., Xu, C., Xie, Y., Zheng, W., Keutzer, K., Tomizuka, M., Zhan, W.: Desire-gs: 4d street gaussians for static-dynamic decomposition and surface reconstruction for urban driving scenes. In: CVPR (2025)
30. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016)
31. Sun, P., Kretschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR (2020)
32. Tang, J., Gao, Y., Yang, D., Yan, L., Yue, Y., Yang, Y.: Dronesplat: 3d gaussian splatting for robust 3d reconstruction from in-the-wild drone imagery. In: CVPR (2025)
33. Tonderski, A., Lindström, C., Hess, G., Ljungbergh, W., Svensson, L., Petersson, C.: Neurad: Neural rendering for autonomous driving. In: CVPR (2024)
34. Turki, H., Zhang, J.Y., Ferroni, F., Ramanan, D.: Suds: Scalable urban dynamic scenes. In: CVPR (2023)
35. Vuong, K., Ghosh, A., Ramanan, D., Narasimhan, S., Tulsiani, S.: Aerialmegadepth: Learning aerial-ground reconstruction and view synthesis. In: CVPR (2025)
36. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: CVPR (2025)
37. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., Ramanan, D., Carr, P., Hays, J.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. In: NeurIPS (2021)
38. Wu, W., He, H., Zhang, C., He, J., Zhao, S.Z., Gong, R., Li, Q., Zhou, B.: Towards autonomous micromobility through scalable urban simulation. In: CVPR (2025)
39. Xie, Z., Liu, Z., Peng, Z., Wu, W., Zhou, B.: Vid2sim: Realistic and interactive simulation from video for urban navigation. In: CVPR (2025)

40. Xu, H., Peng, S., Wang, F., Blum, H., Barath, D., Geiger, A., Pollefeys, M.: Depth-splat: Connecting gaussian splatting and depth. In: CVPR (2024)
41. Xu, Q., Xu, Z., Philip, J., Bi, S., Shu, Z., Sunkavalli, K., Neumann, U.: Point-nerf: Point-based neural radiance fields. In: CVPR (2022)
42. Yan, Y., Lin, H., Zhou, C., Wang, W., Sun, H., Zhan, K., Lang, X., Zhou, X., Peng, S.: Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In: ECCV (2024)
43. Yang, J., Huang, J., Ivanovic, B., Chen, Y., Wang, Y., Li, B., You, Y., Sharma, A., Igl, M., Karkus, P., Xu, D., Wang, Y., Pavone, M.: STORM: Spatio-temporal reconstruction model for large-scale outdoor scenes. In: ICLR (2025)
44. Yang, J., Ivanovic, B., Litany, O., Weng, X., Kim, S.W., Li, B., Che, T., Xu, D., Fidler, S., Pavone, M., Wang, Y.: EmerneRF: Emergent spatial-temporal scene decomposition via self-supervision. In: ICLR (2024)
45. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
46. Zhou, T., Tucker, R., Flynn, J., Fyfe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. TOG (2018)
47. Zhou, X., Lin, Z., Shan, X., Wang, Y., Sun, D., Yang, M.H.: Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In: CVPR (2024)