CAAI Transactions on Intelligence Technology

WILEY

ORIGINAL RESEARCH OPEN ACCESS

VDNeRF: Vision-Only Dynamic Neural Radiance Field for Urban Scenes

Zhengyu Zou[1] | Jingfeng Li[1] | Hao Li[1] | Xiaolei Hou[1] | Jinwen Hu[1] | Jingkun Chen[2] | Lechao Cheng[3] | Dingwen Zhang[1]

[1]Northwestern Polytechnical University, Xi'an, China | [2]Department of Engineering Science, University of Oxford, Oxford, UK | [3]Hefei University of Technology, Hefei, China

Correspondence: Lechao Cheng (chenglc@hfut.edu.cn) | Dingwen Zhang (zdw2006yyy@nwpu.edu.cn)

Received: 23 April 2025 | **Revised:** 19 July 2025 | **Accepted:** 29 July 2025

Keywords: computer vision | deep neural networks | image processing | scene understanding

ABSTRACT

Neural radiance fields (NeRFs) implicitly model continuous three-dimensional scenes using a set of images with known camera poses, enabling the rendering of photorealistic novel views. However, existing NeRF-based methods encounter challenges in applications such as autonomous driving and robotic perception, primarily due to the difficulty of capturing accurate camera poses and limitations in handling large-scale dynamic environments. To address these issues, we propose vision-only dynamic NeRF (VDNeRF), a method that accurately recovers camera trajectories and learns spatiotemporal representations for dynamic urban scenes without requiring additional camera pose information or expensive sensor data. VDNeRF employs two separate NeRF models to jointly reconstruct the scene. The static NeRF model optimises camera poses and static background, whereas the dynamic NeRF model incorporates the 3D scene flow to ensure accurate and consistent reconstruction of dynamic objects. To address the ambiguity between camera motion and independent object motion, we design an effective and powerful training framework to achieve robust camera pose estimation and self-supervised decomposition of static and dynamic elements in a scene. Extensive evaluations on mainstream urban driving datasets demonstrate that VDNeRF surpasses state-of-the-art NeRF-based pose-free methods in both camera pose estimation and dynamic novel view synthesis.

1 | Introduction

Dynamic urban scene reconstruction using Neural Radiance Field (NeRF) [1] has become a pivotal research area due to its ability to synthesise photorealistic images from arbitrary viewpoints. This capability is essential for applications such as robotic perception [2], augmented reality (AR) [3], virtual reality (VR) [4], scene understanding [5, 6] and autonomous driving [7]. However, traditional NeRF methods typically require precise camera pose information for each image, which is often challenging to obtain accurately in real-world scenarios [8].

To estimate camera poses from images, structure from motion (SfM) [9] techniques are commonly employed. Unfortunately, these methods can be time-consuming and may lack robustness, particularly in urban environments where dynamic elements like vehicles and pedestrians introduce variability. This variability can lead to multi-view inconsistencies across input images, making SfM less effective.

To eliminate the dependency on SfM, methods such as Nope-NeRF [10] and LocalRF [11] have been developed to jointly optimise both camera poses and NeRF parameters by parameterising camera poses. However, these approaches assume strictly static scenes, an assumption that is often violated in dynamic urban environments. The presence of moving objects introduces challenges in distinguishing between camera-induced motion and object-induced motion, complicating the

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2025 The Author(s). *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

optimisation process and limiting the applicability of these methods in real-world urban scenarios.

RoDynRF [12] extends NeRF-based pose-free reconstruction to dynamic scenes by employing object segmentation to generate 2D motion masks, thereby decoupling static and dynamic components. It utilises a static radiance field to estimate camera poses and reconstruct the static background, whereas a dynamic radiance field, combined with a deformation field, reconstructs dynamic objects. However, RoDynRF faces limitations in effectively representing complex urban scenes, often resulting in visual artefacts and low-resolution outputs. Moreover, its reliance on precise 2D motion masks, which can be unreliable in urban settings, may lead to incorrect identification of stationary objects as dynamic, resulting in inaccurate scene representations.

Beyond these limitations, two key challenges must be addressed to achieve NeRF-based vision-only urban scene reconstruction and perception: (I) *Robust and Accurate Camera Pose Estimation*: In long and narrow urban scenes with dynamic objects, achieving precise camera pose estimation is crucial for modelling object motion and representing spatiotemporal scenes. However, existing methods often fall into local minima in such complex settings, leading to failures in recovering extensive camera trajectories. (II) *Self-Supervised Decomposition of Static and Dynamic Elements*: Effectively separating static backgrounds from dynamic foregrounds without precise annotations is vital for applications like identifying moving vehicles in autonomous driving. The joint optimization of camera poses and NeRF introduces challenges, as uncertainties in camera poses and the presence of dynamic objects can interfere with the static-dynamic decomposition process.

In this paper, we propose VD-NeRF (vision-only dynamic NeRF), a method that achieves novel view synthesis and spatiotemporal reconstruction for urban scenes using only a sequence of input images (see Figure 1). VD-NeRF utilises two separate NeRF models to represent dynamic scenes: *(a) Static NeRF Model*: Responsible for optimising camera poses and reconstructing the static background. *(b) Dynamic NeRF Model*: Reconstructs dynamic objects by introducing temporal encoding and employs a flow field to predict 3D scene flow, ensuring consistency and accuracy in modelling dynamic objects.

Additionally, VD-NeRF incorporates a carefully designed training framework that enables robust camera pose estimation and self-supervised static-dynamic decomposition, allowing for more accurate reconstruction and perception of urban spatiotemporal scenes. The scene is divided into overlapping subscenes based on spatial dimensions, and training begins with a limited number of images, progressively incorporating more as training advances. Motion masks are utilised to minimise the interference of dynamic objects during camera pose estimation. Once accurate camera poses are established, the dynamic NeRF model reconstructs dynamic foregrounds and performs self-supervised decoupling of dynamic objects without relying on motion masks. By leveraging 3D scene flow, the model aggregates information across multiple time steps, enhancing the distinction between dynamic and static objects and reducing the likelihood of incorrect decoupling.

In summary, our core contributions are

1. We present a method that uses only image sequences from dynamic urban scenes to recover accurate camera trajectories, learn spatiotemporal scene representations and achieve photorealistic dynamic novel view synthesis.
2. We propose a novel training framework that enhances the robustness and accuracy of camera pose estimation

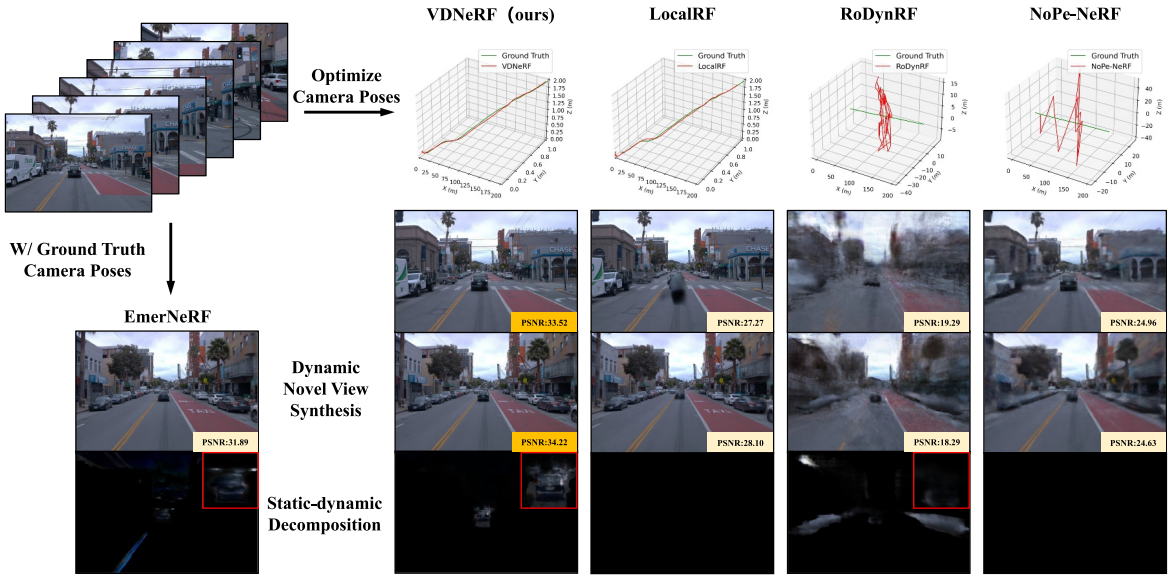


FIGURE 1 | Novel view synthesis for dynamic urban scenes. Given only a set of images without known camera poses, VDNeRF can recover the camera trajectory and reconstruct the spatiotemporal scene. Compared with existing methods, VDNeRF demonstrates more robust camera pose estimation, higher-quality dynamic novel view synthesis and more precise static-dynamic decomposition. We zoom in on the dynamic objects within the red box in the upper right corner to provide a more detailed comparison.

and urban scene reconstruction, enabling self-supervised static-dynamic decomposition without requiring precise annotations.

3. Our method is extensively validated on mainstream urban scene datasets, demonstrating state-of-the-art performance in both camera pose estimation and dynamic novel view synthesis.

2 | Related Work

2.1 | NeRF With Camera Pose Estimation

Eliminating camera pre-processing to achieve joint optimization of camera poses and NeRF is a hot research area. The first to utilise NeRF for optimising camera poses is iNeRF [13] which formulated camera pose estimation as an inverse problem. Leveraging a pre-trained NeRF model, it estimates poses by sampling key points. Since then, many NeRF-based pose-free methods have been proposed. NeRF– [14] treats camera poses as learnable parameters, proposing a framework for the joint optimisation of camera poses and NeRF. GNeRF [15] introduces generative adversarial networks (GAN) to optimise camera poses. BARF [16]extends the classical image alignment theory to NeRF, proposing a coarse-to-fine registration strategy for camera pose recovery. While recovering the camera poses, SC-NeRF [17] also takes camera intrinsic and distortion as optimisation parameters. L2G-NeRF [18] introduces a local-to-global registration method to avoid falling into local minima. Nope-NeRF [10] utilises monocular depth estimation to provide a prior to constrain the relative pose between consecutive images, recovering the large camera motion trajectory. LocalRF [11] progressively jointly optimises camera poses and NeRF, improving robustness in long camera trajectories and large-scale scenes. CF-NeRF [19] and CT-NeRF [20] use an incremental optimisation approach to handle complex camera motion trajectories effectively. Although these methods have demonstrated remarkable performance in static scenes, they often face challenges in representing spatiotemporal urban scenes. This is because urban scenes typically feature long camera motion trajectories and dynamic objects, and these methods struggle to accurately estimate camera poses and reconstruct dynamic objects.

Some NeRF-based methods estimate camera poses and reconstruct dynamic scenes without requiring input camera poses. RoDynRF [12] leverages 2D motion masks to jointly optimise camera poses and static and dynamic radiance fields. However, its accuracy is highly dependent on the quality of the motion masks and not be suitable for large-scale urban scenes. For omnidirectional dynamic images, OmniLocalRF [21] proposed a bidirectional optimisation strategy along with a motion mask prediction module to enable camera pose estimation and scene reconstruction in large-scale dynamic scenes. Although it performs well in large-scale dynamic scenes, it struggles to effectively render or decouple dynamic objects. OmniLocalRF primarily focuses on removing and inpainting dynamic objects. In addition, its optimisation strategy is specifically designed for omnidirectional images. In contrast, VDNeRF can simultaneously achieve accurate camera pose estimation and high-quality scene reconstruction in large-scale dynamic scenes such as urban scenes. In the absence of precise motion masks, it can also self-supervise static-dynamic decomposition and dynamic object rendering, thereby improving its performance for downstream applications.

2.2 | NeRF for Dynamic or Large-Scale Scene

A fundamental limitation of NeRF is the assumption that the scene is strictly static, with constant volume density and emitted radiance (colour). To address this, some works relax the static scene assumption of NeRF and extend it to dynamic scenes [22, 23]. NeRF-W [24] learns a latent embedding for each image to mitigate the effects of illumination variations and transient objects in the scene. D-NeRF [25] introduces a temporal variable to learn the deformation field over different time instants. Nerfies [26] proposes a continuous volumetric deformation field and a coarse-to-fine regularisation method to ensure robust optimisation. HyperNeRF [27] extends NeRF into a high-dimensional space, using slices of this space to reconstruct topologically variation scenes with discontinuous deformations. NSFF [28] introduces forward and backward scene flow, along with multi-view constraints to achieve spatiotemporal view synthesis of in-the-wild scenes. DynamicNeRF [29] introduces multiple regularisation losses to address the ambiguity in dynamic scene reconstruction. D^2NeRF [30] achieves the decoupling of dynamic and static objects in scenes by employing two separate NeRF models to represent the scene. It also introduces a shadow field network specifically for handling situations like shadows, improving the rendering and decomposition quality of dynamic objects. DyBluRF [31] achieves dynamic novel view synthesis from blurred dynamic videos by predicting the object's global discrete cosine transform (DCT) trajectory.

Beyond the challenge of handling dynamic scenes, another critical limitation of NeRF is its insufficient model capacity. Several methods have been proposed to extend NeRF to large-scale scenes. BungeeNeRF [32] reconstructs scenes at multi-scale by progressively growing the model and inclusive multi-level supervision strategies. Block-NeRF [33] and Mega-NeRF [34] employ multiple compact NeRF models for large-scale scene representation. S-NeRF [35] and StreetSurf [36] can render images from novel views in nonobject-centric street-view scenes. In PreSight [37], a city-scale NeRF framework compatible with online perception models is constructed.

Some works attempt to simultaneously address the limitations of NeRF in both static and small-scale settings to achieve urban scene reconstruction. SUDS [38] employs three scalable branches: static, dynamic and far-field to represent city-scale dynamic scenes. Similar to SUDS, EmerNeRF [39] and RoDUS [40] reconstruct urban scenes using multiple branches and decompose the scenes into dynamic and static components in a self-supervised manner. However, these methods heavily depend on accurate camera poses or even expensive LiDAR data, which are often difficult to obtain in real-world applications. Unlike methods such as Block-NeRF and SUDS, which focus on city-scale reconstruction by partitioning entire city scenes into blocks under known camera poses, VDNeRF is not

specifically designed for city-scale scenes. However, it can jointly optimise camera poses and NeRF in smaller-scale urban scenes, such as a street. By creating sub-scenes based on the spatial extent of the estimated camera trajectory, VDNeRF can be scaled to arbitrarily long streets. Furthermore, using only a sequence of RGB images, VDNeRF can robustly reconstruct spatiotemporal scenes and decouple dynamic elements.

3 | Method

VDNeRF jointly optimises camera poses and NeRF from dynamic urban scenes in a self-supervised manner without camera poses and ground-truth depths. Given a sequence of images $\mathcal{I} = \{I_i \in \mathbb{R}^{H\times W\times 3} | i = 0\cdots N-1\}$ and focal length f, VDNeRF estimates camera poses $\mathcal{P} = \{P_i = [\mathbf{R}_i|\mathbf{t}_i] \in \mathrm{SE}(3) | i = 0\cdots N-1\}$, generates images $\mathcal{Y}_{rgb} = \{Y \in \mathbb{R}^{H\times W\times 3}\}$ from novel viewpoints and decouples dynamic objects and static background.

In the following sections, we detail the workings of VDNeRF (as illustrated in Figure 2). Firstly, we briefly introduce NeRF in Section 3.1. Next, we describe the architecture of our NeRF models in Section 3.2, followed by an explanation of the regularisation constraints for dynamic urban scenes in Section 3.3. Then, we propose our training framework in Section 3.4, demonstrating how it enables robust camera pose estimation and dynamic object reconstruction in urban scenes. Finally, we discuss the loss functions that are minimised throughout the training process in Section 3.5.

3.1 | NeRF Preliminary

Neural radiance fields (NeRF) [1] can implicitly model a static 3D scene and synthesise photorealistic novel views. It encodes the points in 3D scene as volume density σ and emitted colour $\mathbf{c} = (r, g, b)$ with an MLP network $F_\Theta : (\mathbf{x}, \mathbf{d}) \to (\mathbf{c}, \sigma)$, where $\mathbf{x} = (x, y, z)$ is a location of the 3D point and $\mathbf{d} = (\theta, \phi)$ is a 2D viewing direction, indicating the perspective from which the 3D point is observed.

Specifically, given a posed image, along a camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ emitted from the camera optical centre $\mathbf{o}$, NeRF samples N 3D discrete points. Through the MLP network F_Θ, volume density σ and emitted colour $\mathbf{c}$ of each sampling point can be obtained. Then, using volume rendering [41], NeRF can render one pixel colour $\hat{C}(\mathbf{r})$:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_i,\ T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right), \tag{1}$$

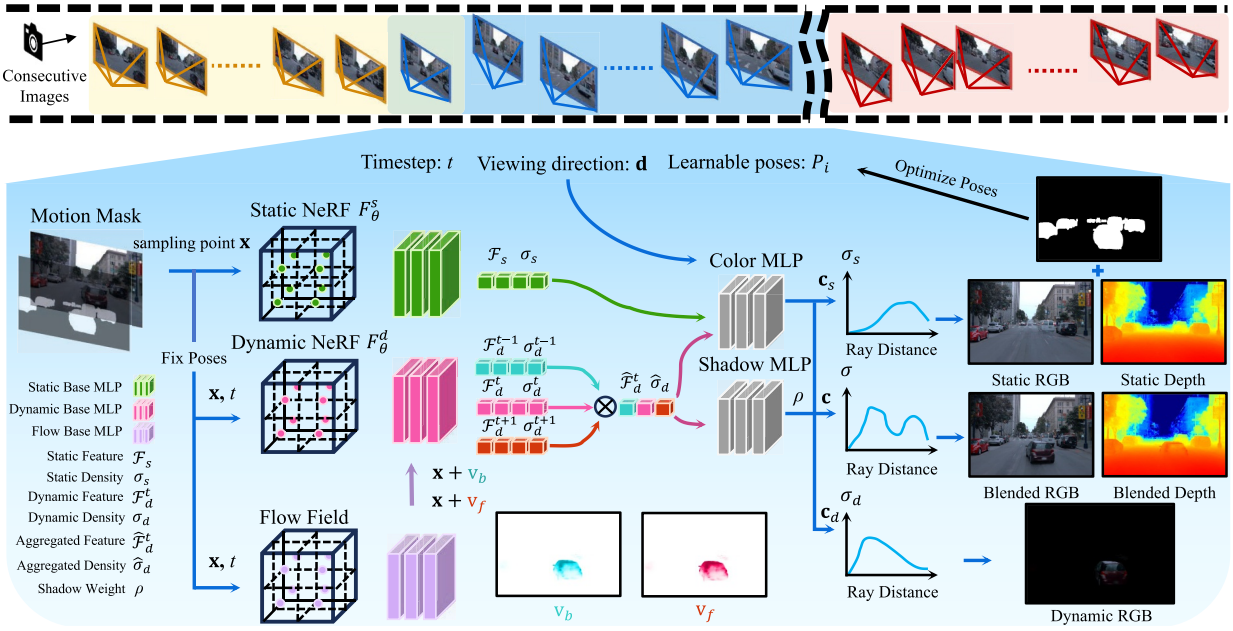


FIGURE 2 | VDNeRF Overview. VDNeRF is composed of a static NeRF F_Θ^s and a dynamic NeRF F_Θ^d with a flow field. Each model is equipped with a hash grid and a base MLP. The static NeRF F_Θ^s takes the 3D location of the sampling points $\mathbf{x} = (x, y, z)$ as input, whereas dynamic NeRF F_Θ^d additionally incorporates a timestep t as input. The static NeRF F_Θ^s query feature $\mathcal{F}_s$ and density σ_s from the base MLP. For the dynamic NeRF F_Θ^d, it leverages the 3D forward flow $\mathbf{v}_f$ and backward flow $\mathbf{v}_b$–with the 2D optical flow visualised in the figure–predicted by the flow field to aggregate the dynamic feature $\hat{\mathcal{F}}_d^t$ and dynamic density $\hat{\sigma}_d^t$ across multiple timesteps and queries the shadow weight $\rho \in [0, 1]$ from the shadow MLP. These features, along with the viewing direction $\mathbf{d} = (\theta, \phi)$, are passed to the colour MLP to obtain the colour $\mathbf{c} = (r, g, b)$. Volume rendering is applied to all sampling points along the camera ray to render static and dynamic pixel colours $\hat{C}_s(\mathbf{r})$ and $\hat{C}_d(\mathbf{r})$. Finally, the blended representation of spatiotemporal scenes is achieved using shadow weights. During training, VDNeRF partitions the scene into multiple overlapping sub-scenes. For each sub-scene, it starts with a small set of images, progressively optimising the static NeRF F_Θ^s and camera poses P_i with the help of motion masks. Once accurate camera poses are established, they are fixed, and the dynamic NeRF F_Θ^d is activated to reconstruct dynamic objects, achieving self-supervised static-dynamic decomposition.

where $\alpha_i = 1 - \exp(-\sigma_i\delta_i)$ and $\delta_j = t_{i+1} - t_i$ is the distance between two sampling points. NeRF optimises the MLP network F_Θ by minimising the photometric error between synthesised pixels and ground truth-observed images, enabling accurate scene representation.

3.2 | Static and Dynamic NeRF Models

Given the large number of objects and the extensive motion of dynamic elements in urban scenes, it is difficult to effectively represent such complex scenes using a single NeRF model. Therefore, following previous excellent works [30, 38–40], VDNeRF uses two separate NeRF models (static NeRF F^s_Θ and dynamic NeRF F^d_Θ) to represent the spatiotemporal scene. Each NeRF model employs learnable multi-resolution hash encoding as proposed by Instant-NGP [42] to enhance the speed of training and rendering. Static NeRF F^s_Θ reconstructs the background composed of nonmoving objects. Dynamic NeRF F^d_Θ combines the 3D scene flow predicted by the flow field to reconstruct the foreground composed of moving objects. The two NeRF models query feature $\mathcal{F}$ and volume density σ from the base MLPs. The feature $\mathcal{F}$ is then passed to the colour MLP to obtain the corresponding colour $\mathbf{c}$. Note that to achieve the self-supervised decomposition of the scene, both NeRF models are trained for the entire scene. As a result, two NeRF models share a colour MLP and collectively represent the spatiotemporal scene with the assistance of a shadow MLP.

Static NeRF F^s_Θ is responsible for reconstructing the static background in the scene and optimising camera poses $\mathcal{P} = \{P_i = [\mathbf{R}_i|\mathbf{t}_i] \in \mathrm{SE}(3) | i = 0 \cdots N-1\}$:

$$(\mathbf{c}_s, \sigma_s) = F^s_\Theta(\mathbf{x}, \mathbf{d}), \tag{2}$$

where $\mathbf{x}$ represents the location of a sampling point along a camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, and $\mathbf{d}$ denotes the direction of the camera ray $\mathbf{r}(t)$. The camera optical centre $\mathbf{o}$ corresponds to the translation $\mathbf{t}_i \in \mathrm{T}(3)$, whereas the ray direction $\mathbf{d}$ is determined by the rotation matrix $\mathbf{R}_i \in \mathrm{SO}(3)$. Therefore, as static NeRF $F\Theta^s$ reconstructs the static background, it can simultaneously optimise the camera poses $\mathcal{P}$.

Dynamic NeRF F^d_Θ introduces an additional timestep t as input, extending NeRF to the dynamic scene:

$$(\mathbf{c}_d, \sigma_d) = F^d_\Theta(\mathbf{x}, \mathbf{d}, t). \tag{3}$$

However, relying solely on temporal encoding is insufficient to accurately reconstruct dynamic objects in urban scenes, such as fast-moving vehicles with long motion trajectories. This becomes especially apparent in the rendered novel views, where dynamic objects exhibit blurred shapes or even residual motion trails. To address this, dynamic NeRF F^d_Θ incorporates a flow field to predict the 3D scene flow. Given a timestep t and a 3D location $\mathbf{x} = (x, y, z)$, it outputs a 3D flow vector $\mathbf{v} \in \mathbb{R}^3$ to predict the location of $\mathbf{x}$ in the adjacent timestep: $\mathbf{x}' = \mathbf{x} + \mathbf{v}$. The flow field estimates the forward and backward 3D scene flows, $\mathbf{v}_f$ and $\mathbf{v}_b$, for each point. It ensures consistency between the forward flow at timestep t and the backward flow at the next timestep $t+1$, then inputs these 3D scene flows into the dynamic NeRF F^d_Θ. The 3D scene flows enable the dynamic NeRF F^d_Θ to better capture the motion of objects, facilitating more accurate and realistic rendering of dynamic objects from new timesteps or viewpoints. Additionally, following EmerNeRF, the dynamic NeRF F^d_Θ aggregates features $\hat{\mathcal{F}}^t_d$ and dynamic volume density $\hat{\sigma}^t_d$ from multiple timesteps:

$$\begin{aligned} (\mathbf{x} + \mathbf{v}_b, t-1) &\rightarrow (\mathcal{F}^{t-1}_d, \sigma^{t-1}_d), \\ (\mathbf{x} + \mathbf{v}_f, t+1) &\rightarrow (\mathcal{F}^{t+1}_d, \sigma^{t+1}_d), \\ \hat{\mathcal{F}}^t_d &= \lambda_1 \cdot \mathcal{F}^{t-1}_d + \lambda_2 \cdot \mathcal{F}^t_d + \lambda_3 \cdot \mathcal{F}^{t+1}_d, \\ \hat{\sigma}^t_d &= \lambda_1 \cdot \sigma^{t-1}_d + \lambda_2 \cdot \sigma^t_d + \lambda_3 \cdot \sigma^{t+1}_d, \end{aligned} \tag{4}$$

where $\lambda_1 = \lambda_3 = 0.25$ and $\lambda_2 = 0.5$. It helps reduce the noise in temporal attributes and allows information to be shared across different timesteps, improving the rendering and decoupling quality of dynamic objects.

Although static NeRF F^s_Θ and dynamic NeRF F^d_Θ are two separate NeRF models, they are fused when rendering the scene. As in the real-world environment, dynamic objects exist on a static background. However, due to the inherent complexity of dynamic scenes, such as motion blur, occlusion, or noise in dynamic object representations, we cannot directly fuse the colours emitted from the dynamic NeRF F^d_Θ and the static NeRF F^s_Θ. Inspired by D^{2}NeRF, we introduce a shadow MLP to control the proportion of the fusion of static NeRF F^s_Θ and dynamic NeRF F^d_Θ. Unlike the original design, which primarily focuses on shadows for dynamic objects, we find that this approach is equally effective for the rendering and decomposition of the entire dynamic scene. This shadow MLP outputs a shadow weight $\rho \in [0, 1]$ to downweight the static background. Finally, the emitted colour $\hat{C}(\mathbf{r})$ of the sampling point can be formulated as:

$$\mathbf{c} = (1-\rho) \cdot \frac{\sigma_s}{\sigma} \cdot \mathbf{c}_s + \frac{\sigma_d}{\sigma} \cdot \mathbf{c}_d, \tag{5}$$

where volume density $\sigma = \sigma_s + \sigma_d$. The shadow weight ρ serves as a soft blending factor between the static NeRF F^s_Θ and dynamic NeRF F^d_Θ, allowing VDNeRF to precisely represent static background and dynamic objects. Notably, the learnt values of shadow weight ρ reflect the dynamic content of a scene. When a scene contains more dynamic elements, more sampling points exhibit ρ values close to 1; conversely, in scenes with fewer dynamic elements, most sampling points tend to have ρ values close to 0. To more intuitively visualise the distribution of ρ across different scenes, we render the shadow weights into images using volume rendering of NeRF, obtaining per-pixel ρ values and plotting their histograms, as shown in Figure 3.

3.3 | Regularisation Constraints for Urban Scenes

In feed-forward and nonobject-centric urban scenes, the observations of objects are transient and sparse. Under this setting, it is easy to trigger the inherent shape-radiance ambiguity [43] of

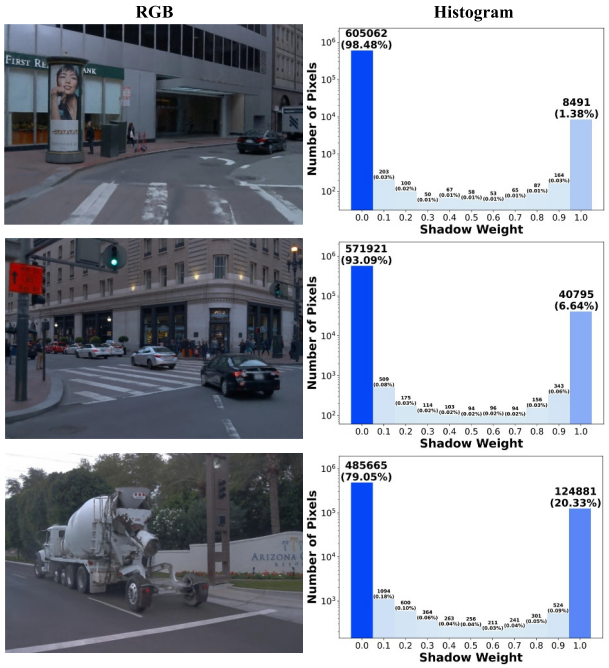


FIGURE 3 | Histograms of shadow weight distributions in different scenes. The left column shows the synthesised RGB images, whereas the right column presents histograms of the shadow weight distributions corresponding to the pixels. The *y*-axis of each histogram is plotted on a logarithmic scale to better visualise the distribution. We also annotate each histogram with the exact counts and corresponding percentages. Please consider zooming in to view the detailed numbers.

NeRF. This ambiguity resembles overfitting, preventing the scene reconstruction from generalising to novel viewpoints. Unfortunately, jointly optimising the camera poses and NeRF further exacerbates the shape-radiance ambiguity. Moreover, the reconstruction of dynamic objects is also affected by this ambiguity, as the significant variation in observed motion states across different timesteps allows for multiple possible fits. Depending exclusively on photometric error for supervision cannot mitigate this issue. To address this ambiguity, we introduce two regularisation supervisions: monocular depth and 2D optical flow.

Using DPT [44], we generate monocular depth maps from RGB images. These depth priors provide relative scale and geometric constraints that reduce convergence time and improve reconstruction quality from sparse viewpoints [45]. They also simplify the process of optimising camera poses and effectively mitigate shape-radiance ambiguity [10]. Furthermore, they can help infer the direction of object motion, leading to a better reconstruction of dynamic objects.

The 2D optical flow generated by RAFT [46] also helps mitigate this ambiguity, enabling NeRF to plausibly represent challenging scenes [28, 29]. By capturing pixel correspondences between adjacent images, optical flow imposes constraints on both the geometric structure of the scene and the motion relationships within it. This provides explicit priors for the motion trajectories of dynamic objects and the camera. Moreover, it indirectly supervises the 3D scene flow, enhancing the temporal consistency of dynamic objects.

Given that the monocular depth estimation and 2D optical flow priors generated by DPT and RAFT may contain errors, we adopt an annealing strategy, exponentially decaying their constraint weights during training.

For more details on the depth and optical flow supervision, please refer to Section 3.5.

3.4 | Training Framework

Our training framework partitions the urban scene into multiple overlapping sub-scenes, each undergoing the following three training stages: (a) progressively optimising static NeRF F_Θ^s and camera poses, (b) fixing the camera pose and activating dynamic NeRF F_Θ^d to model dynamic objects, and (c) dynamically creating a new sub-scenes. During camera pose optimization, we introduce motion masks from RoDynRF and compute the loss only in regions outside the motion masks, while keeping the dynamic NeRF F_Θ^d frozen. And the camera poses are fixed when F_Θ^d is activated. This separation is essential because distinguishing between pixel motion caused by dynamic objects and that caused by camera movement is challenging. Optimising dynamic NeRF F_Θ^d and camera poses simultaneously in an urban scene easily leads the training in the wrong direction and falls into local minima, resulting in inaccurate camera pose estimation and failed dynamic object modelling. Consequently, this hinders the accurate representation and perception of the spatiotemporal scene.

The three training stages are detailed as follows.

Progressively Optimising. Even with the motion masks excluding the interference of dynamic objects, accurately estimating camera trajectory in narrow and long urban scenes remains a challenge. To ensure the accuracy and robustness of camera pose estimation, we adopt a progressive optimization approach inspired by LocalRF. Specifically, during the joint optimization of the static NeRF and camera poses, we do not input all images at once. Instead, it starts with only a small set of images and progressively incorporates additional images as the number of training iterations increases. When the number of images reaches a certain threshold or the camera trajectory exceeds a specified distance, no further images are added. At this point, rather than immediately activating the dynamicF_Θ^d, we continue to jointly optimise static NeRF F_Θ^s and the camera poses while gradually reducing the learning rate for camera poses optimization to achieve more accurate camera pose estimation. The number of joint optimization iterations is determined by the total number of incorporated images.

Activating Dynamic NeRF. Once the accurate camera pose is established, we activate dynamic NeRF F_Θ^d and start to jointly optimise with the static NeRF F_Θ^s to achieve a spatiotemporal

representation of the urban scene. Notably, unlike RoDynRF, we do not treat the motion mask as the definitive dynamic region, as it often misclassifies stationary pedestrians and parked vehicles as dynamic objects in urban scenes (see in Figure 4). Instead, the static-dynamic decomposition of the scene relies only on the self-supervision of VDNeRF. This is made possible by the ability of our dynamic NeRF F_Θ^d to model object motion accurately and consistently. Additionally, by aggregating dynamic information across multiple timesteps, we effectively reduce noise, further highlighting the distinction between dynamic and static objects. To reinforce this decomposition process, we also impose constraints on the dynamic volume density $\hat{\sigma}_d^t$, encouraging it to be generated only when necessary. It is just that in the motion mask, we allow it to generate dynamic volume density σ_d more easily and assume that the dynamic colours $\hat{C}^d(\mathbf{r})$ is more reliable when rendering the scene. We chose motion masks from RoDynRF simply because they can be conveniently and quickly generated in a self-supervised manner. However, we believe that if there is a more accurate motion mask, VDNeRf will perform better in static-dynamic decomposition and camera pose estimation. Because the motion mask contains parked vehicles, etc, which have obvious geometric and texture features, it can improve the accuracy of camera pose estimation and the quality of dynamic scene representation to some extent.

Creating a New Sub-scenes. When the training of the current sub-scene is complete, we create a new sub-scene and a new NeRF model. However, instead of discarding all images from the current NeRF model, we pass the last few images along with their camera poses to the next sub-scene. The scene depicted by these images will be jointly reconstructed by both NeRF models. Dividing the elongated urban scene into overlapping sub-scenes along spatial dimensions ensures that the optimization starts in the correct direction, maintaining the consistency and continuity of both the camera trajectory and object motion. Additionally, the strategy of using multiple NeRF models to represent large-scale urban scenes helps overcome the capacity limitations of a single NeRF model, thereby improving reconstruction quality.

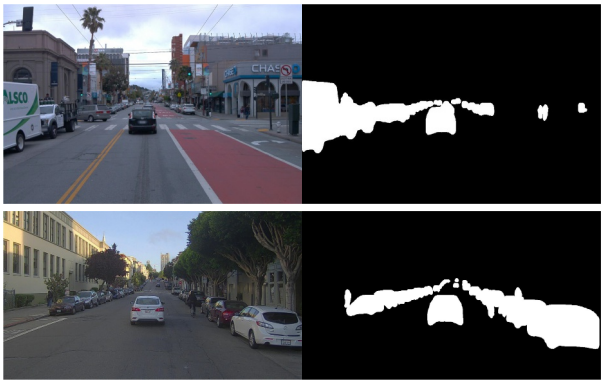


FIGURE 4 | Motion mask from RoDynRF. The motion masks generated by RoDynRF include stationary vehicles and pedestrians on the roadside. We also do not use it to supervise static-dynamic decomposition but only use it to exclude dynamic elements to achieve more accurate and robust camera pose estimation.

3.5 | Loss Functions

Total loss. The total loss for VDNeRF is:

$$\begin{aligned}\mathcal{L}_{\text{total}} = &\underbrace{\mathcal{L}_{\text{colour}} + \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{flow}}}_{\text{also for camera pose}} \\ &+ \underbrace{\mathcal{L}_{\text{cycle}} + \mathcal{L}_{\text{dynamic}} + \mathcal{L}_{\text{shadow}}}_{\text{when dynamic NeRF is activated}}.\end{aligned} \tag{6}$$

When we jointly optimise the camera poses and the static NeRF F_Θ^s, we only use $\mathcal{L}_{\text{colour}}$, $\mathcal{L}_{\text{depth}}$ and $\mathcal{L}_{\text{flow}}$. When we activate the dynamic NeRF F_Θ^d, we enable $\mathcal{L}_{\text{cycle}}$, $\mathcal{L}_{\text{dynamic}}$ and $\mathcal{L}_{\text{shadow}}$.Below, we provide a detailed explanation of each loss function.

Colour loss. We minimise the photometric error between rendered pixel colour $\hat{C}(\mathbf{r})$ and the ground truth $C(\mathbf{r})$:

$$\mathcal{L}_{\text{colour}} = \left\| \hat{C}(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2. \tag{7}$$

Depth loss. We utilise the monocular depth estimation $D(\mathbf{r})$ from DPT [44] as a prior. Similar to (1), NeRF can render the depth of pixel:

$$\hat{D}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i d_i,\ T_i = \exp\left(- \sum_{j=1}^{i-1} \sigma_j \delta_j \right), \tag{8}$$

where d_i is the distance from the sampling point to the camera origin, N is the number of sampling points on the camera ray corresponding to the pixel, $\delta_j = t_{i+1} - t_i$ represents the distance between two sampling points and $\alpha_i = 1 - \exp(-\sigma_i \delta_i)$. Since monocular depth estimation is not scaled and shift-invariant, we normalise $\hat{D}(\mathbf{r})$ and $D(\mathbf{r})$ to obtain normalised depth $\hat{D}^*(\mathbf{r})$ and $D^*(\mathbf{r})$ [47]. So the depth supervision is

$$\mathcal{L}_{\text{depth}} = \left\| \hat{D}^*(\mathbf{r}) - D^*(\mathbf{r}) \right\|_2^2. \tag{9}$$

Flow loss. We use RAFT [46] to introduce 2D optical flow supervision $\mathbf{f}_{i \to i-1}$ and $\mathbf{f}_{i \to i+1}$. Using the depth and adjacent camera pose, we project a pixel into a 3D point and then reproject it onto adjacent images to compute the expected backward optical flow $\hat{\mathbf{f}}_{i \to i-1}$ and forward optical flow $\hat{\mathbf{f}}_{i \to i+1}$. We minimise the error between them and the optical flows estimated by RAFT:

$$\mathcal{L}_{\text{flow}} = \left\| \hat{\mathbf{f}}_{i \to i-1} - \mathbf{f}_{i \to i-1} \right\|_1 + \left\| \hat{\mathbf{f}}_{i \to i+1} - \mathbf{f}_{i \to i+1} \right\|_1. \tag{10}$$

Cycle loss. As in previous work [28, 38, 39], for the 3D scene flow predicted by the flow field, we also introduce a 3D scene flow cycle consistency term to encourage cycle consistency between forward and backward 3D scene flow:

$$\begin{aligned}\mathcal{L}_{\text{cycle}} = &\left(\mathbf{v}_f(\mathbf{x}, t_i) + \mathbf{v}_b\left(\mathbf{x} + \mathbf{v}_f(\mathbf{x}, t_i), t_{i+1}\right) \right)^2 \\ &+ \left(\mathbf{v}_b(\mathbf{x}, t_i) + \mathbf{v}_f(\mathbf{x} + \mathbf{v}_b(\mathbf{x}, t_i), t_{i-1}) \right)^2.\end{aligned} \tag{11}$$

Dynamic loss. In real-world urban scenes, dynamic objects are a minority. Therefore, to better decompose dynamic objects, we

penalise the dynamic volume density σ_d to encourage them to be generated only when necessary:

$$\mathcal{L}_{\text{dynamic}} = \frac{1}{N}\sum_{i=1}^{N}\sigma_d(\mathbf{x}, t), \tag{12}$$

where N is the number of sampling points on the camera ray corresponding to the pixel.

Shadow loss. Follow D^{2}NeRF [30]; to avoid over-interpreting dark regions of the scene and ensure that the scene is composed of a static background as much as possible, we penalise the square of the shadow's ratio ρ along a camera ray:

$$\mathcal{L}_{\text{shadow}} = \sum_{i=1}^{N} T_i \alpha_i \rho_i^2. \tag{13}$$

4 | Experiments

We first introduce our experimental setup in Section 4.1. Next, in Section 4.2 and Section 4.3, we evaluate VDNeRF from two perspectives: novel view synthesis and camera pose estimation, respectively, and compare it against mainstream NeRF-based methods. Then, we conduct further ablation studies in Section 4.4. Finally, we summarise the limitations of VDNeRF in Section 4.5.

4.1 | Experimental Setup

Dataset. We selected two mainstream urban driving datasets, NeRF On-The-Road(NOTR) [39] and Pandaset [48] to evaluate our method. The scenes we selected all contain certain dynamic objects. *NOTR*: this dataset is derived from the Waymo Open Dataset [49], which can reflect the challenges of training NeRF in real-world urban driving scenes. We use four dynamic scenes, each with 199 consecutive images captured by the front camera. All images are downsampled to 960× 640. Excluding the first image, one out of every 10 images is selected for novel view synthesis. **Pandaset :** is also a challenging urban driving dataset. We use three dynamic scenes, each containing 80 consecutive images captured by the front-facing camera. All images are down-sampled to 960× 540. The image selection strategy for novel view synthesis follows that of NOTR, except for the first image, every 10th image is selected.

Comparisons with SOTA Methods. We primarily compare VDNeRF against RoDynRF [12], LocalRF [11] and NoPe-NeRF [10], as these methods are state-of-the-art NeRF-based pose-free methods. Our evaluation focuses on two key aspects: novel view synthesis and camera trajectory recovery. In addition, RoDynRF also extends NeRF to dynamic scenes. Therefore, we further compare it in terms of dynamic object rendering quality and static-dynamic decomposition effectiveness. All baseline methods are configured and executed according to their official open-source implementations. To further evaluate the performance of VDNeRF in dynamic urban scenes, we compare it against the state-of-the-art NeRF-based method EmerNeRF [39]. It is important to note that EmerNeRF inputs the ground

TABLE 1 | Quantitative results of dynamic novel view synthesis on the NOTR dataset.

		✓ pose-free									✗ pose-free		
		Ours			LocalRF			NoPe-NeRF			RoDynRF			EmerNeRF		
Scenes		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NOTR	016	33.133	0.920	0.186	28.036	0.860	0.282	25.001	0.701	0.482	23.810	0.678	0.475	30.928	0.876	0.270
	021	30.902	0.879	0.292	26.391	0.843	0.333	25.666	0.793	0.443	22.706	0.721	0.447	28.387	0.850	0.335
	022	32.250	0.913	0.219	28.145	0.862	0.298	24.362	0.711	0.485	20.406	0.624	0.546	29.814	0.880	0.269
	025	30.998	0.887	0.227	27.419	0.836	0.300	23.832	0.651	0.520	24.186	0.677	0.454	29.685	0.850	0.280
Pandaset	011	27.202	0.814	0.260	22.729	0.731	0.361	21.768	0.632	0.503	19.980	0.558	0.539	23.072	0.681	0.396
	027	30.527	0.878	0.239	26.185	0.793	0.363	24.592	0.685	0.498	16.215	0.522	0.611	26.231	0.746	0.413
	047	31.090	0.883	0.199	26.906	0.787	0.333	25.060	0.664	0.503	16.142	0.488	0.615	26.859	0.731	0.395

Note: We report PSNR, SSIM and LPIPS of novel view synthesis. ↑: higher is better, ↓: lower is better. The red, orange and yellow highlights, respectively, denote the best, the second best and the third best results. VDNeRF significantly leads other methods in all metrics.

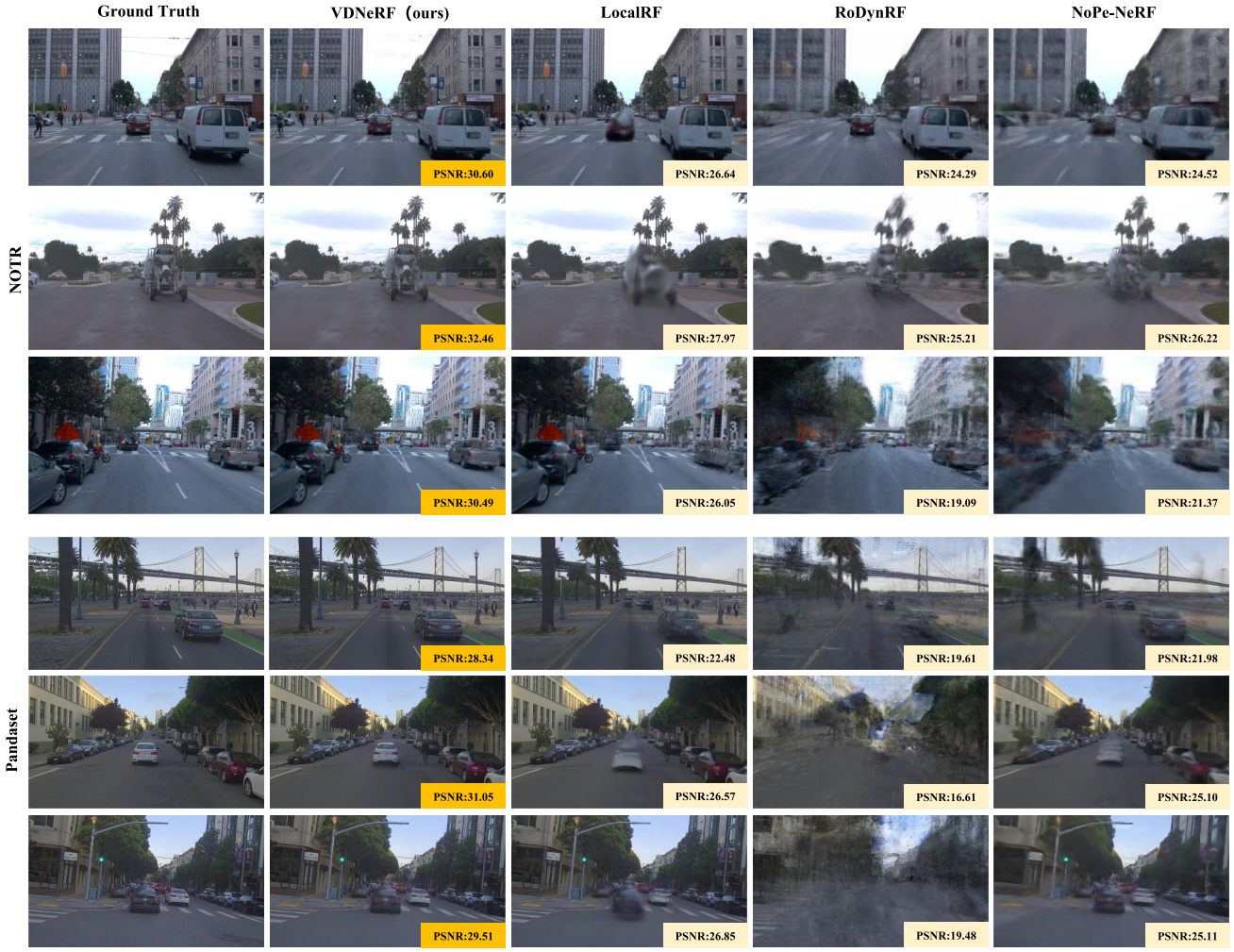


FIGURE 5 | Qualitative results of dynamic novel view synthesis on the NOTR and Pandaset datasets. We visualise the images rendered from novel viewpoints. Compared to other methods, VDNeRF accurately reconstructs dynamic objects and synthesises more detailed and photorealistic images.

truth camera poses but does not input LiDAR point clouds. In the NOTR dataset, EmerNeRF inputs the motion masks and sky masks that are provided by the dataset, whereas such masks are not available in the Pandaset dataset.

Metrics. For the rendered images from novel views, we employ three standard error metrics: peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM) [50] and learnt perceptual image patch similarity (LPIPS) [51]. PSNR intuitively reflects the errors of each pixel. SSIM takes into account the adjacent pixels, focusing on the structure of the image. LPIPS uses a pre-trained neural network model (VGG architecture) to compare the deep neural information of the image, which is more consistent with human perception. For the recovered camera trajectory, we first use Umeyama [52] to align it with the ground truth. We then evaluate it using two standard metrics: absolute trajectory error (ATE) and relative pose error (RPE). ATE measures the camera position errors of the entire camera trajectory and the ground truth. RPE measures the errors between pairs of camera poses, including relative rotation error (RPE_r) and relative translation error (RPE_t).

Implementation Details. We build VDNeRF based on Instant-NGP [42] with the tiny-cuda-nn [53] and use NerfAcc [54] to accelerate. All parameters are optimised by the Adam optimiser [55]. The train starts with five images. In each iteration, we render a total of 4096 pixels from different images. During the progressive optimisation, we incorporate the new image every 600 iterations. To parameterise the unbounded scene, we adopt the sampling strategy of Mip-NeRF 360 [56]. When the head-to-tail distance of the estimated camera trajectory exceeds 4 or the total number of images contained surpasses 70, we stop adding new images and begin refining the current sub-scene. In urban driving scenes, these thresholds are chosen to create sub-scenes of appropriate spatial scales, enabling high-quality spatiotemporal reconstruction. The number of refinement iterations N is determined by the total number of images, following a schedule of 840 iterations per incorporated image. In the first seventh of iterations N, we jointly optimise the static NeRF F^s_Θ and the camera poses. During this phase, the learning rate of camera poses and the regularisation weights of flow and depth decrease exponentially to their original 0.1 factor. Afterwards, we activate dynamic NeRF F^d_Θ, at which point the learning rate of NeRF also decreases exponentially to its original 0.1 factor. Once the current sub-scene is fully trained, we incorporate the last 20 images into a created new sub-scene and repeat the above process until all images are trained. Training one scene (199 images) of the NOTR dataset takes approximately 12 h on an

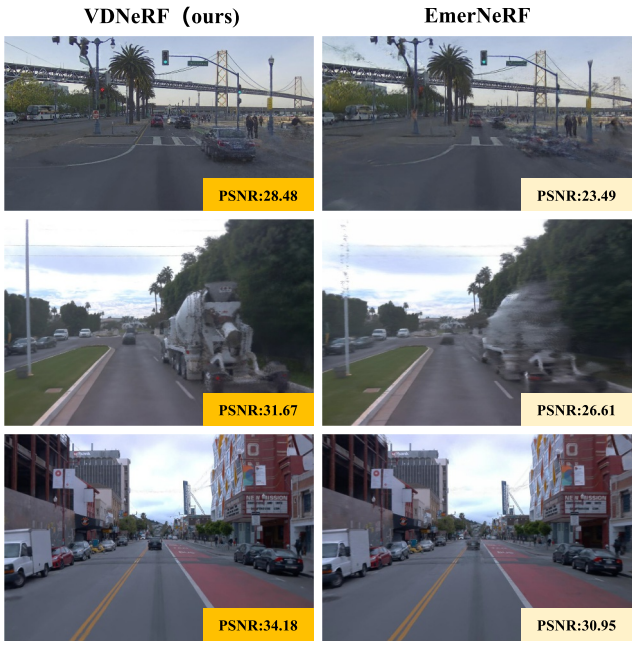


FIGURE 6 | Novel view synthesis comparison of VDNeRF and EmerNeRF. EmerNeRF takes ground-truth camera poses as inputs but does not use ground-truth depths. The qualitative results of dynamic novel view synthesis demonstrate that VDNeRF achieves superior rendering quality with finer details and more accurate dynamic object modelling.

NVIDIA RTX3090 GPU. The training proceeds at about seven iterations per second, with a maximum GPU memory usage of around 12 GB. Synthesising a blended RGB image (960× 640) takes approximately 2.7 s.

4.2 | Comparison of Novel View Synthesis

Following the post-process strategy of NeRF-based pose-free methods like Nope-NeRF and LocalRF, we obtain the camera poses for test viewpoints by freezing the NeRF model and optimising only the camera poses. Using these learnt camera poses, we synthesise novel view images for testing. We present the quantitative results of novel view synthesis in Table 1. Across two different datasets, VDNeRF obtains consistent and significant gains across all metrics. Compared to the second-best method, VDNeRF achieves average improvements of 2.117 in PSNR, 0.036 in SSIM, and a drop of 0.057 in LPIPS on the NOTR dataset. On the Pandaset dataset, VDNeRF achieves average improvements of 4.203 in PSNR, 0.088 in SSIM, and a drop of 0.120 in LPIPS.

For an intuitive comparison, Figure 5 shows the qualitative results of novel view synthesis for the evaluated methods. The novel view images synthesised by VDNeRF have clearer and more realistic geometry and texture. Notably, in the absence of ground-truth depths, VDNeRF performs even better than the state-of-the-art methods for dynamic driving scenes, EmerNeRF, which relies on ground-truth camera poses. Qualitative comparisons are shown in Figure 6.

LocalRF and NoPe-NeRF are designed for static scenes. Although they can sometimes fit a roughly correct scene by leveraging their scene representation capabilities, they fail to accurately reconstruct the motion and geometry of dynamic objects.

Both RoDynRF and EmerNeRF are capable of modelling and rendering dynamic objects to achieve static-dynamic decomposition. Therefore, VDNeRF also compares with RoDynRF and EmerNeRF for the dynamic decoupling performance in the synthesised novel view images. Qualitative results are displayed in Figure 7. Unlike VDNeRF and EmerNeRF, which are self-supervised, RoDynRF relies on generated motion masks as supervision. However, in urban scenes, these motion masks often misclassify parked vehicles as dynamic objects, causing RoDynRF to incorrectly model stationary objects as dynamic ones, leading to inaccurate scene representation. Additionally, the model capacity of RoDynRF is limited, leading to low-resolution reconstructions when handling large-scale urban scenes. Without input LiDAR point clouds, although EmerNeRF can achieve static-dynamic decomposition, its decomposition accuracy and the rendering quality of dynamic objects are not as good as VDNeRF.

We present the RGB images obtained from our static-dynamic decomposition along with the corresponding depth maps in Figure 8. Moreover, VDNeRF is capable of generating high-quality spatiotemporal scene representations even under rotational camera motion, such as intersection turns. The visualisation of turning scenarios is presented in Figure 9.

4.3 | Comparison of Camera Poses Estimation

Firstly, to obtain the complete camera trajectory we freeze the NeRF model and optimise the camera poses for the test viewpoints. Afterwards, we align the recovered camera trajectory with the ground truth using the Umeyama method (Sim (3) with 7 DoF) and evaluate its accuracy. The quantitative results of camera pose estimation are presented in Table 2. NoPe-NeRF is not designed for dynamic scenes, whereas RoDynRF performs poorly in large-scale dynamic scenes such as urban driving due to its limited model capacity and inaccurate motion masks. Although LocalRF is not suitable for dynamic scenes, it can progressively optimise camera poses and NeRF, making it robust and accurate for long camera trajectories. However, due to its inability to effectively handle dynamic objects' interference, the recovered camera trajectories are less accurate than ours, especially in terms of global trajectory accuracy and consistency, where the gap is more noticeable. We visualise the camera trajectories in Figure 10 for qualitative comparison. Note that for a more detailed comparison, we set the coordinate systems of VDNeRF and LocalRF to the same range.

4.4 | Ablation Study

We conduct ablation studies around our training framework (motion masks, fixing the camera poses, freezing the dynamic



FIGURE 7 | Qualitative results of dynamic decoupling. We visualise the dynamic RGB rendered by RoDynRF, EmerNeRF and VDNeRF. VDNeRF outperforms other methods in terms of both the accuracy of static-dynamic decomposition and the rendering quality of dynamic objects. In contrast, RoDynRF and EmerNeRF occasionally misclassify static backgrounds as dynamic objects (see the red box in the image) or struggle to accurately render and separate dynamic objects.

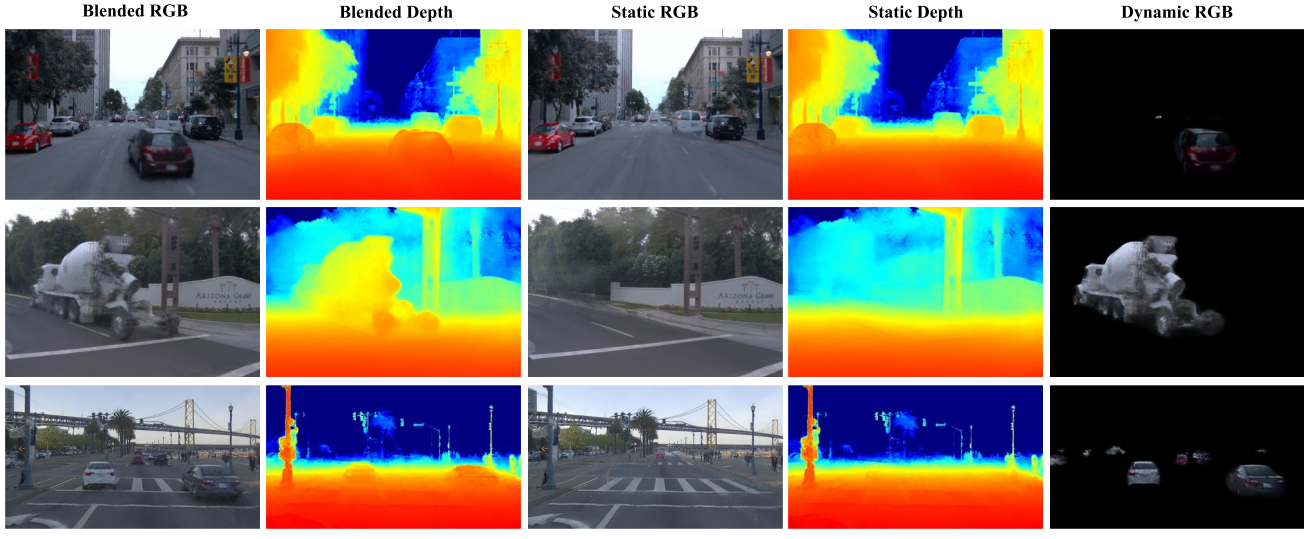


FIGURE 8 | Our results of static-dynamic decomposition. We visualise the blended RGB and depth, as well as the static RGB, static depth and dynamic RGB for a more comprehensive presentation.



FIGURE 9 | Visualisation of turning scenarios. We present synthesised images during turning manoeuvres to demonstrate that VDNeRF is capable of accurately reconstructing turning scenarios.

TABLE 2 | Quantitative comparison of camera pose estimation on the NOTR and Pandaset datasets.

		Ours			LocalRF			NoPe-NeRF			RoDynRF		
Scenes		**RPE$_t$ ↓**	**RPE$_r$ ↓**	**ATE ↓**	**RPE$_t$ ↓**	**RPE$_r$ ↓**	**ATE ↓**	**RPE$_t$ ↓**	**RPE$_r$ ↓**	**ATE ↓**	**RPE$_t$ ↓**	**RPE$_r$ ↓**	**ATE ↓**
NOTR	016	0.884	0.225	0.329	0.886	0.225	1.282	1.859	0.308	22.967	1.946	0.296	31.280
	021	0.747	0.574	1.900	0.746	0.560	2.578	1.658	0.670	19.516	1.619	0.615	21.216
	022	1.417	0.146	1.456	1.461	0.146	7.388	3.577	0.342	43.451	4.194	0.603	55.443
	025	0.720	0.289	0.568	0.739	0.287	1.356	1.758	0.478	10.201	1.600	0.374	26.494
Pandaset	011	2.176	0.182	0.813	2.147	0.178	3.162	2.629	0.522	10.983	3.410	0.363	11.170
	027	1.625	0.215	0.259	1.618	0.217	0.888	2.189	0.282	17.059	6.300	6.137	7.792
	027	1.475	0.292	0.159	1.478	0.293	0.474	1.836	0.397	15.413	5.481	7.626	6.252

Note: We report RPE$_t$, RPE$_r$ and ATE of the recovered camera trajectory. The unit of RPE$_r$ is in degrees, whereas RPE$_t$ and ATE is in the ground truth scale. ↑: higher is better, ↓: lower is better. The red, orange and yellow highlights, respectively, denote the best, the second best and the third best results. The camera trajectories recovered by VDNeRF have better accuracy and global consistency.

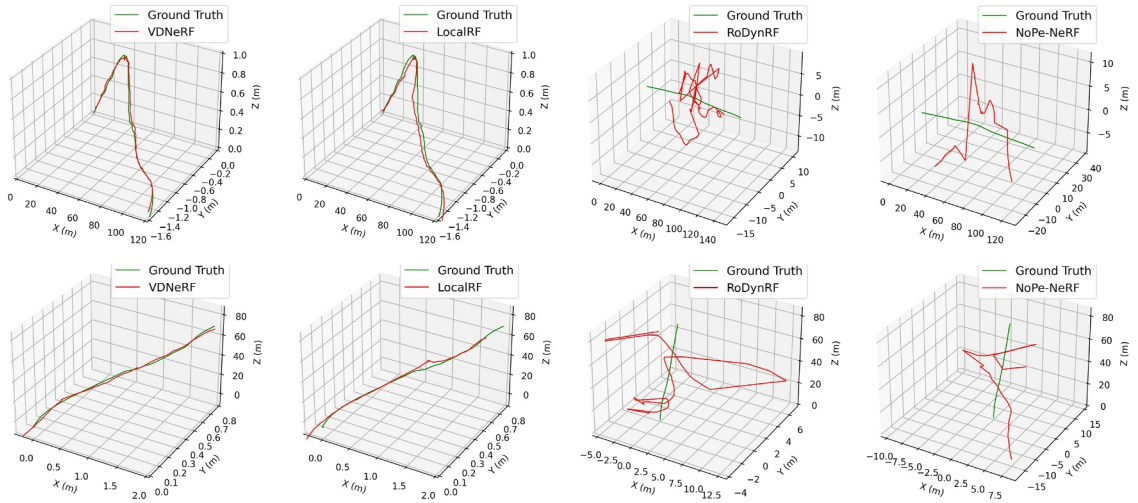


FIGURE 10 | Qualitative results of camera pose estimation. For ease of comparison, VDNeRF and LocalRF are visualised within the same coordinate system range. Compared to other NeRF-based pose-free methods, VDNeRF estimates camera trajectories that align more closely with the ground truth.

TABLE 3 | Ablation study results on the NOTR dataset.

	Novel view synthesis			Camera poses		
	PSNR ↑	SSIM ↑	LPIPS ↓	RPE_t ↓	RPE_r ↓	ATE ↓
Full	**33.133**	**0.920**	0.186	0.884	**0.225**	**0.329**
w/o motion masks	32.738	0.917	0.196	**0.883**	**0.225**	0.662
w/o fixing poses	33.115	0.919	**0.185**	0.884	0.226	0.561
w/o freezing F_Θ^d	28.834	0.825	0.310	0.925	0.518	1.740

Note: We report PSNR, SSIM and LPIPS of novel view synthesis, RPE_t, RPE_r and ATE of the camera pose estimation. ↑: higher is better, ↓: lower is better. Bold values indicate the best results for each metric.

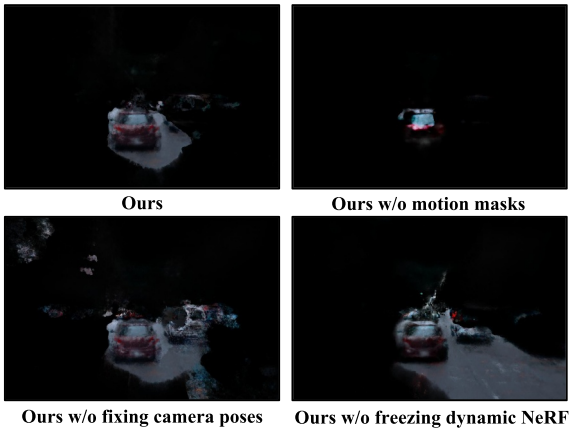


FIGURE 11 | Comparison of static and dynamic decomposition results of ablation studies. The full VDNeRF achieves the best performance in static-dynamic decomposition and dynamic object rendering quality.

NeRF). We analyse them in Table 3. We show the static-dynamic decomposition results of the ablation studies in Figure 11. Below, we analyse in detail to further verify the effectiveness and reason of our training framework.

Effect of motion masks. With the help of motion masks, VDNeRF can exclude dynamic objects in the scene when jointly optimising the camera poses and NeRF. This improves the robustness and accuracy of camera pose estimation because, for dynamic objects, the motion of pixels on the image plane is caused by both the camera motion and the independent motion of the dynamic objects. Therefore, without motion masks, the global consistency and accuracy of the camera poses are somewhat affected, which is reflected in an increase of 0.333 in ATE. The reconstruction and decomposition quality of dynamic objects also worsens. In addition, we encourage objects contained in the motion mask to be more easily regarded as dynamic objects, which will also improve the effect of static-dynamic decomposition.

Effect of fixing the camera poses. Constantly optimising the camera poses during the training process does not have a significant impact on novel view synthesis, but it does cause the accuracy of the recovered camera trajectory to deteriorate. The scene is reconstructed jointly by static NeRF and dynamic NeRF, achieving static-dynamic decomposition in a self-supervised manner. Once dynamic NeRF is activated and begins rendering dynamic objects, errors in the initial stage of decomposition can affect the rendering quality of the static background, leading to inaccuracies in camera pose optimisation. Consequently, ATE increases by 0.329. Furthermore, continuous adjustments to camera poses, even minor ones, can lead to incorrect modelling of independent object motion. This is manifested in that some static pixels are misclassified as dynamic, and dynamic objects cannot be individually decoupled.

Effect of freezing the dynamic NeRF. When optimising the camera poses, activating the dynamic NeRF F_Θ^d not only leads to inaccurate camera pose estimation but also causes poor image rendering quality. VDNeRF misinterprets the pixel motion caused by the camera movement as a dynamic object, preventing it from learning the correct scene representation and camera trajectory. As a result, for novel view synthesis, PSNR decreases by 4.299, whereas LPIPS increases by 0.124. Additionally, it tends to treat the entire motion trajectory of an object as a single dynamic for decoupling.

4.5 | Limitations

Although VDNeRF can jointly optimise the camera pose and NeRF in dynamic urban scenes and achieve static-dynamic decomposition, it still has certain limitations. First, it may struggle with complex camera motion trajectories, such as rapid in-place 360-degree rotations. In such cases, the large angular displacement between adjacent images results in a minimal overlap of observable regions, making feature matching unreliable and thereby reducing the continuity and consistency of the estimated camera trajectory. Secondly, its segmentation of dynamic objects lacks fine granularity, often leading to the over-segmentation of an entire region when multiple dynamic objects are in close proximity. Lastly, VDNeRF currently cannot meet the low-latency requirements of real-time systems such as robotic navigation. VDNeRF is better suited for offline reconstruction, simulation or as a source of priors for real-time systems.

5 | Conclusions

In this paper, we propose VDNeRF, which jointly optimises camera poses and NeRF using only a set of images as inputs, without relying on ground-truth camera poses or depths. To achieve this, we carefully design two NeRF models, one for static background and another for dynamic objects, along with a tailored training framework. Through extensive quantitative and qualitative evaluation of mainstream urban driving datasets,

VDNeRF demonstrates outstanding robustness and performance, outperforming state-of-the-art NeRF-based methods. In future work, we will incorporate continual learning capabilities [57, 58] into our framework to enhance its adaptability across diverse urban scenarios.

Funding

This study was supported by the National Natural Science Foundation of China (Grants 62293543, 62322605, and 62472139).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The datasets used to support the findings of this study include the NOTR dataset, available at https://waymo.com/open/data/motion/, and the Pandaset dataset, available at https://pandaset.org/#data-collection. Both datasets were used under licence and are subject to usage restrictions. Access to the data is granted with permission from Waymo and Pandaset, respectively.

References

1. B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NERF: Representing Scenes as Neural Radiance Fields for View Synthesis," *Communications of the ACM* 65, no. 1 (2021): 99–106, https://doi.org/10.1145/3503250.

2. T.-M. Nguyen, S. Yuan, T. H. Nguyen, et al., "Mcd: Diverse Large-Scale Multi-Campus Dataset for Robot Perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), 22304–22313.

3. J. Carmigniani and B. Furht, "Augmented Reality: An Overview," in *Handbook of Augmented Reality* (2011), 3–46.

4. G. C. Burdea and P. Coiffet, *Virtual Reality Technology* (John Wiley & Sons, 2003).

5. H. Li, D. Zhang, Y. Dai, et al., "Gp-Nerf: Generalized Perception Nerf for Context-Aware 3d Scene Understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2024), 21708–21718.

6. D. Zhang, H. Li, D. He, et al., "Unsupervised Pre-Training With Language-Vision Prompts for Low-Data Instance Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 47, no. 10 (2025): 8642–8657, https://doi.org/10.1109/tpami.2025.3579469.

7. J. Levinson, J. Askeland, J. Becker, et al., "Towards Fully Autonomous Driving: Systems and Algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)* (IEEE, 2011), 163–168.

8. Y. Jiao, Z. Jie, S. Chen, et al., "Instance-Aware Multi-Camera 3d Object Detection With Structural Priors Mining and Self-Boosting Learning," in *Proceedings of the AAAI Conference on Artificial Intelligence* 38, no. 3 (2024): 2598–2606, https://doi.org/10.1609/aaai.v38i3.28037.

9. J. L. Schonberger and J.-M. Frahm, "Structure-From-Motion Revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 4104–4113.

10. W. Bian, Z. Wang, K. Li, J.-W. Bian, and V. A. Prisacariu, "Nope-Nerf: Optimising Neural Radiance Field With No Pose Prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), 4160–4169.

11. A. Meuleman, Y.-L. Liu, C. Gao, et al., "Progressively Optimized Local Radiance Fields for Robust View Synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), 16539–16548.

12. Y.-L. Liu, C. Gao, A. Meuleman, et al., "Robust Dynamic Radiance Fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), 13–23.

13. L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "Inerf: Inverting Neural Radiance Fields for Pose Estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2021), 1323–1330.

14. Z. Wang, S. Wu, W. Xie, M. Chen, and V. A. Prisacariu, "Nerf–: Neural Radiance Fields Without Known Camera Parameters," preprint arXiv:2102.07064 (2021).

15. Q. Meng, A. Chen, H. Luo, et al., "Gnerf: Gan-Based Neural Radiance Field Without Posed Camera," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 6351–6361.

16. C.-H. Lin, W.-C. Ma, A. Torralba, and S. Lucey, "Barf: Bundle-Adjusting Neural Radiance Fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 5741–5751.

17. Y. Jeong, S. Ahn, C. Choy, A. Anandkumar, M. Cho, and J. Park, "Self-Calibrating Neural Radiance Fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 5846–5854.

18. Y. Chen, X. Chen, X. Wang, et al., "Local-To-Global Registration for Bundle-Adjusting Neural Radiance Fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), 8264–8273.

19. Q. Yan, Q. Wang, K. Zhao, et al., "Cf-nerf: Camera Parameter Free Neural Radiance Fields With Incremental Learning," in *Proceedings of the AAAI Conference on Artificial Intelligence* 38, no. 6 (2024): 6440–6448, https://doi.org/10.1609/aaai.v38i6.28464.

20. Y. Ran, Y. Li, Q. Ye, et al., "Ct-nerf: Incremental Optimizing Neural Radiance Field and Poses With Complex Trajectory," preprint arXiv: 2404.13896 (2024).

21. D. Choi, H. Jang, and M. H. Kim, "Omnilocalrf: Omnidirectional Local Radiance Fields From Dynamic Videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), 6871–6880.

22. Y. Gao, Y. Dai, H. Li, et al., "Cosurfgs: Collaborative 3d Surface Gaussian Splatting With Distributed Learning for Large Scene Reconstruction," preprint arXiv:2412.17612 (2024).

23. D. Wu, F. Yang, B. Xu, P. Liao, W. Zhao, and D. Zhang, "Hv-bev: Decoupling Horizontal and Vertical Feature Sampling for Multi-View 3d Object Detection," *IEEE Transactions on Intelligent Transportation Systems* 26, no. 11 (2025): 18734–18746, https://doi.org/10.1109/tits.2025.3590464.

24. R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. 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* (2021), 7210–7219.

25. A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer, "D-nerf: Neural Radiance Fields for Dynamic Scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), 10318–10327.

26. K. Park, U. Sinha, J. T. Barron, et al., "Nerfies: Deformable Neural Radiance Fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 5865–5874.

27. K. Park, U. Sinha, P. Hedman, et al., "Hypernerf: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields," preprint arXiv:2106.13228 40, no. 6 (2021): 1–12, https://doi.org/10.1145/3478513.3480487.

28. Z. Li, S. Niklaus, N. Snavely, and O. 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* (2021), 6498–6508.

29. C. Gao, A. Saraf, J. Kopf, and J.-B. Huang, "Dynamic View Synthesis From Dynamic Monocular Video," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 5712–5721.

30. T. Wu, F. Zhong, A. Tagliasacchi, F. Cole, and C. Oztireli, "D 2nerf: Self-Supervised Decoupling of Dynamic and Static Objects From a Monocular Video," *Advances in Neural Information Processing Systems* 35 (2022): 32653–32666, https://doi.org/10.5555/3600270.3602636.

31. H. Sun, X. Li, L. Shen, X. Ye, K. Xian, and Z. Cao, "Dyblurf: Dynamic Neural Radiance Fields From Blurry Monocular Video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), 7517–7527.

32. Y. Xiangli, L. Xu, X. Pan, et al., "Bungeenerf: Progressive Neural Radiance Field for Extreme Multi-Scale Scene Rendering," in *European Conference on Computer Vision* (Springer, 2022), 106–122.

33. M. Tancik, V. Casser, X. Yan, et al., "Block-Nerf: Scalable Large Scene Neural View Synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 8248–8258.

34. H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-Nerf: Scalable Construction of Large-Scale Nerfs for Virtual Fly-Throughs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 12922–12931.

35. Z. Xie, J. Zhang, W. Li, F. Zhang, and L. Zhang, "S-Nerf: Neural Radiance Fields for Street Views," preprint arXiv:2303.00749 (2023).

36. J. Guo, N. Deng, X. Li, et al., "Streetsurf: Extending Multi-View Implicit Surface Reconstruction to Street Views," preprint arXiv: 2306.04988 (2023).

37. T. Yuan, Y. Mao, J. Yang, Y. Liu, Y. Wang, and H. Zhao, "Presight: Enhancing Autonomous Vehicle Perception With City-Scale Nerf Priors," preprint arXiv:2403.09079 (2024): 323–339, https://doi.org/10.1007/978-3-031-72980-5_19.

38. H. Turki, J. Y. Zhang, F. Ferroni, and D. Ramanan, "Suds: Scalable Urban Dynamic Scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), 12375–12385.

39. J. Yang, B. Ivanovic, O. Litany, et al., "Emernerf: Emergent Spatial-Temporal Scene Decomposition via Self-supervision," preprint arXiv: 2311.02077 (2023).

40. T.-A.-Q. Nguyen, L. Roldão, N. Piasco, M. Bennehar, and D. Tsishkou, "Rodus: Robust Decomposition of Static and Dynamic Elements in Urban Scenes," preprint arXiv:2403.09419 (2024): 112–130, https://doi.org/10.1007/978-3-031-73220-1_7.

41. J. T. Kajiya and B. P. Von Herzen, "Ray Tracing Volume Densities," *ACM SIGGRAPH computer graphics* 18, no. 3 (1984): 165–174, https://doi.org/10.1145/800031.808594.

42. T. Müller, A. Evans, C. Schied, and A. Keller, "Instant Neural Graphics Primitives With a Multiresolution Hash Encoding," *ACM Transactions on Graphics* 41, no. 4 (2022): 1–15, https://doi.org/10.1145/3528223.3530127.

43. K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and Improving Neural Radiance Fields," preprint arXiv: 2010.07492, (2020).

44. R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision Transformers for Dense Prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 12179–12188.

45. K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-Supervised Nerf: Fewer Views and Faster Training for Free," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 12882–12891.

46. Z. Teed and J. Deng, "Raft: Recurrent All-Pairs Field Transforms for Optical Flow," in *Computer Vision–ECCV 2020: 16Th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16* (Springer, 2020), 402–419.

47. R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, no. 3 (2020): 1623–1637, https://doi.org/10.1109/tpami.2020.3019967.

48. P. Xiao, Z. Shao, S. Hao, et al., "Pandaset: Advanced Sensor Suite Dataset for Autonomous Driving," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)* (IEEE, 2021), 3095–3101.

49. P. Sun, H. Kretzschmar, X. Dotiwalla, et al., "Scalability in Perception for Autonomous Driving: Waymo Open Dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 2446–2454.

50. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing* 13, no. 4 (2004): 600–612, https://doi.org/10.1109/tip.2003.819861.

51. R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), 586–595.

52. S. Umeyama, "Least-Squares Estimation of Transformation Parameters Between Two Point Patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, no. 4 (1991): 376–380, https://doi.org/10.1109/34.88573.

53. T. Müller, Tiny Cuda Neural Network Framework. Retrieved May 18, (2021) 2023.

54. R. Li, M. Tancik, and A. Kanazawa, "Nerfacc: A General Nerf Acceleration Toolbox," preprint arXiv:2210.04847 (2022).

55. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," preprint arXiv:1412.6980 (2014).

56. J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-Nerf 360: Unbounded Anti-Aliased Neural Radiance Fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 5470–5479.

57. D. Cheng, Y. Ji, D. Gong, et al., "Continual All-in-One Adverse Weather Removal With Knowledge Replay on a Unified Network Structure," *IEEE Transactions on Multimedia* 26 (2024): 8184–8196, https://doi.org/10.1109/tmm.2024.3377136.

58. D. Cheng, Y. Hu, N. Wang, D. Zhang, and X. Gao, "Achieving Plasticity-Stability Trade-Off in Continual Learning Through Adaptive Orthogonal Projection," *IEEE Transactions on Circuits and Systems for Video Technology* 35, no. 8 (2025): 7485–7498, https://doi.org/10.1109/tcsvt.2025.3547916.