DetRF: Detachable Novel Views Synthesis of Dynamic Scenes Using Backdrop-Driven Neural Radiance Fields

Boyu Zhang[1], **Zheng Zhu**[2*], **Wenbo Xu**[3]
[1]SenseTime Research
[2]GigaAI
[3]Waytous
zhangboyu1@sensetime.com, zhengzhu@ieee.org, rick0xffffff@gmail.com

Abstract

Representing and synthesizing novel views in real-world dynamic scenes from casual monocular videos is a long-standing problem. Existing solutions typically approach dynamic scenes by applying geometry techniques or utilizing temporal information between several adjacent frames without considering the underlying background distribution in the entire scene or the transmittance over the ray dimension, limiting their performance on static and occlusion areas. Our approach backdrop-driven neural radiance fields offers high-quality view synthesis and a 3D solution to detach the background from the entire dynamic scene, which is called DetRF. Specifically, it employs a neural representation to capture the scene distribution in the static background and a 6D-input NeRF to represent dynamic objects, respectively. Each ray sample is given an additional occlusion weight to indicate the transmittance lying in the static and dynamic components. We evaluate DetRF on public dynamic scenes and our urban driving scenes acquired from an autonomous-driving dataset. Extensive experiments demonstrate that our approach outperforms previous methods in rendering texture details and motion areas while also producing a clean static background. Our code will be available soon.

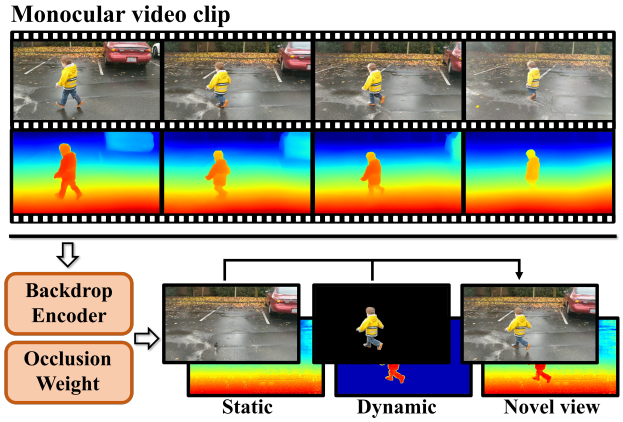


Figure 1: Our method takes a casual monocular video clip captured from real-world dynamic scenes as input and generates a static representation driven by the underlying backdrop code and a dynamic representation. The target novel view rendering can be obtained by blending them with occlusion weight.

Introduction

Novel view synthesis (Avidan and Shashua 1997) from dynamic scenes aims to generate photorealistic views at arbitrary viewpoints and any time step with given images from one or more cameras as input. It provides the possibility to generate free-view rendering using finite-view input and brings a lifelike representation. This can have vast and varied applications, *e.g.* switching views in cinematic/games special effects (Tewari et al. 2020), rendering visuals in AR/VR world (Cai et al. 2022; Li et al. 2022a), assisting camera imaging (Mildenhall et al. 2022), and enabling interactive exploration in robot/autonomous-driving perception and navigation (Li, Li, and Zhu 2023).

In essence, novel view synthesis of dynamic scenes is an extension of representing static 3D scenes from discrete 2D images, which is highly ill-posed (Ivanov, Vasin, and Tanana 2013) since there are infinite solutions that can render the input video appropriately while only a single 2D image observation is available at each view. Compared to static scenes (Kar, Hne, and Malik 2017; Niemeyer et al. 2019), dynamic novel view synthesis is more challenging since approaches for dynamic scenes need to capture spatial-temporal information using a 6D plenoptic representation ($\mathbf{I}(t) = \Phi(\mathbf{x}, \mathbf{d}, t; \boldsymbol{\theta}) : \mathbb{R}^3 \times \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}^3$), and the occlusion between different objects might also be tricky.

Traditionally, novel view synthesis in real-world dynamic scenes (Yoon et al. 2020; Lei, Xing, and Chen 2020; Luo et al. 2020; Oswald, Stühmer, and Cremers 2014) requires images from multiple camera views to estimate the geometry and occlusion of the entire scene. However, capturing such multi-view inputs is laborious and expensive. Neural Radiance Fields (NeRF) (Mildenhall et al. 2020) offers a new circumvention to this problem when just given monocular views. Rather than explicit shape representation, NeRF encodes 3D location and viewing angle to color and density using a neural network and generates arbitrary novel views through volume rendering. Recent NeRF-based methods (Park et al. 2021; Kundu et al. 2022; Jayasundara et al.

*Corresponding author.

2023) show promising results. Using one or more NeRFs, these methods can implicitly represent dynamic scenes for novel view synthesis and time interpolation.

Several methods (Kundu et al. 2022; Jiang et al. 2022; Li et al. 2022a; Tancik et al. 2022) use supervision networks (*e.g.* semantic segmentation) to produce dynamic view synthesis and can achieve editable results, but they are limited to specific domains. Other methods (Gao et al. 2021; Park et al. 2021; Li et al. 2021; Tretschk et al. 2021) focus on adjacent temporal information in a finite number of frames to capture time-dependent components, such as optical flow and depth, while overlooking the underlying distribution existing in the entire video clip. For scenes containing a single object, such as Lego and Fern (Mildenhall et al. 2020), the 2D scene projection on the camera plane will alter drastically with a switch in view angle. Unlike these scenes, real-world locomotor scenes typically encompass dynamic objects against a predominantly static background. These static backgrounds, such as blocks, buildings are distant from the camera lens. When photographing these scenes from various perspectives, the 2D projection on the camera plane will be minimally disturbed by changes in viewpoint. This implies that the bulk of similar projection pixels in a real-world dynamic scene are shared by the static background across multiple frames. Assuming that identical regions are the observations for each frame and follow the same scene representation, then the camera projections from different angles can be obtained with tiny shifts based on this representation. Moreover, all previous approaches determine occlusion only from RGB space by merging rendering color and neglect transmittance weight existing on each ray sample, resulting in a sub-optimal performance in occlusion areas between the static backdrop and dynamic objects.

To tackle the aforementioned distribution representation problem and provide occlusion weight over the ray dimension, we introduce DetRF, a novel method that *captures the latent code in the entire scene and adds transmittance weight to each ray*. It can generate high-quality novel views at arbitrary viewpoints and any interpolation time steps for view synthesis of real-world dynamic scenes, and applied to general domains. Specifically, we extend NeRF to a parallel structure, as shown in Fig. 1. A background pipeline presenting the underlying distribution and a 6D-input NeRF generating time-varying fields are used to express static and dynamic components, respectively. Then we weight transmittance matrices on their rendering corresponding to each ray to learn the occlusion relationship. To optimize the pipeline effectively, we use multiple regularization losses to drive training in different modules. We evaluate DetRF on NVIDIA dynamic scenes (Yoon et al. 2020) and our urban driving scenes obtained from an autonomous-driving dataset, Argoverse(Wilson et al. 2021). Our contributions can be summarized as follows:

- To the best of our knowledge, we are the first to exploit the underlying scene representation and produce transmittance weight across each ray sample in order to depict dynamic real-world scenes.
- We present a novel attention-based structure and an occlusion weight neural representation that offers a 3D pattern for decoupling the clean static background from the entire scene.
- Substantial quantitative and qualitative experiments suggest that DetRF operates better than existing methods. Additionally, an urban driving scenes dataset is built for dynamic novel view synthesis. We will release this dataset for research purpose.

Related Work

Novel View Synthesis

Novel view synthesis is to generate the geometry of scenes from a set of limited views. Intuitively, we need to create an explicit 3D representation of scene geometry, such as point clouds or meshes (Debevec, Taylor, and Malik 1996; Hedman et al. 2017; Snavely, Seitz, and Szeliski 2006), and render novel views by transporting pixels among views via this representation. There are many works that concentrate on using various graphic techniques, such as light field (Mildenhall et al. 2019) and multi-plane based methods (Zhou et al. 2018; Tucker and Snavely 2020), to obtain 3D representation. In addition, learning-based methods (Chen and Zhang 2019; Trevithick and Yang 2020) attempt to utilize neural networks to learn view interpolation to obtain implicit representation. More recently, Neural Radiance Fields (NeRF) (Mildenhall et al. 2020) have attained photo-realistic synthesis performance and applied in many scenes (Ma et al. 2023; Fang et al. 2023; Wang et al. 2024). NeRFw (Martin-Brualla et al. 2021) is suitable for novel view synthesis in outdoor scenes. Mip-NeRF (Barron et al. 2022) enhances NeRF for unbounded scenes. And many methods (Rematas et al. 2022; Zhang et al. 2024) put NeRF to large-scale scenes. Nevertheless, these methods are only applicable to static scenes, as a 6D-input of dynamic scenes will lead to radiance ambiguity (Zhang et al. 2021).

Dynamic Scene Synthesis by Neural Rendering

Since the promising performance of neural radiance field, many works consider extending NeRF as a spatial-temporal representation to address view synthesis in dynamic scenes. (Pumarola et al. 2021; Wu et al. 2022) consider time as an additional input to handle single dynamic objects indoors. (Li et al. 2021; Du et al. 2021; Yuan et al. 2021) model the dynamic scene as a time-dependent continuous function with scene flow. Gao et al. (Gao et al. 2021) introduce regularization losses to encourage plausible reconstruction. Xian et al. (Xian et al. 2021) employ video depth estimation to supervise a space-time radiance field. Using a ray-blending network, Tretschk et al. (Tretschk et al. 2021) model the occlusion parts in dynamic scenes. TöRF (Attal et al. 2021) utilizes prior information from the time-of-flight camera to enhance reconstruction quality. Park et al. (Park et al. 2021) optimize an additional continuous volumetric deformation field to capture dynamic humans. (Li et al. 2022a; Wang et al. 2023; Kim et al. 2024; Attal et al. 2023) aim to represent scenes in multi-view videos effectively. Some methods (Yan, Li, and Lee 2023; Li et al. 2024) seek to im-

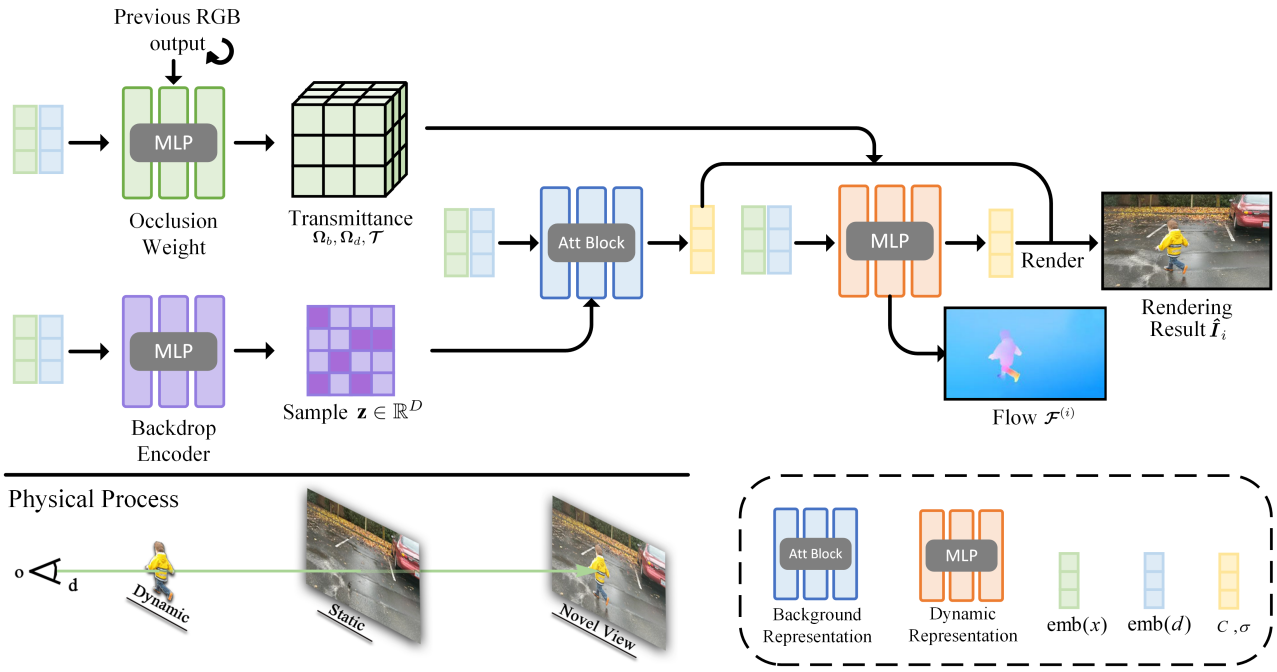


Figure 2: **Network pipeline of DetRF.** Our approach trains two neural representations jointly: a backdrop-driven background representation and a flow-predicting dynamic representation, producing RGB color C and volume density σ. A backdrop encoder is constructed to learn the implicit distribution from the background. And an occlusion weight module is created to generate transmittance weight. At i-th time instance, the final rendering result $\hat{\boldsymbol{I}}_i$ is made by mixing occlusion weight Ω and $\boldsymbol{\mathcal{T}}$ with the outputs of two networks, as described in the physical process (bottom).

prove dynamic rendering speed and accuracy. These NeRF-based approaches disregard the global distribution over all frames. More recently, NeRF-based methods (Jiang et al. 2022; Kundu et al. 2022; Jayasundara et al. 2023; Liu et al. 2024; Song et al. 2023) have provided a more accurate representation of dynamic scenes and achieved editable effects, but they are limited to specific domains.

Method

In this section, we introduce the proposed *Backdrop-Driven Neural Radiance Fields for Detachable Novel Views Synthesis of Dynamic Scenes, DetRF.* As shown in Fig. 2, DetRF consists of a background distribution driving module and a 6D-input NeRF. An occlusion weight module is designed to decouple the dynamic and static components. We first describe the background pipeline and the temporal-input NeRF. Then the occlusion weight module will be discussed. Lastly, we will explain a group of loss functions for optimizing the proposed network.

Backdrop Representation

We intend to represent the static background in terms of latent distribution such that camera projections from different views can be queried by small shifts based on this background representation. Assuming that the background obeys $P(\mathbf{z} \in \mathbb{R}^D|\Theta)$, it can be formulated to:

$$\Phi_b : (\mathbf{x}, \mathbf{d}, \mathbf{z}) \rightarrow (\sigma_b, \mathbf{c}_b). \tag{1}$$

The latent code technique has already demonstrated effectiveness (Van Den Oord et al. 2017; Park et al. 2021; Peng, Zhang et al. 2021). To obtain the implicit distribution from the static background, we construct a backdrop encoder, as shown in Fig. 2. Then a latent code is sampled from the output posterior $\mathbf{z} \sim P_\Theta(\mathbf{z})$, with dimensions corresponding to the number of rays sampled each time. The sample is utilized to drive the rendering of the background component. As we need to obtain the similarity between different views and scene distribution, attention mechanisms (Vaswani et al. 2017; Dosovitskiy et al. 2020; Liu et al. 2021) is highly suitable for this task. We construct an attention-based structure to exploit the latent variable, as shown in Fig. 3.

Specifically, we apply attention to embedded inputs with querying sample z. By mapping embedded features and the latent distribution sample into a low-dimensional space, attention elements of 3D location $\mathbf{Q}_l, \mathbf{K}$,and $\mathbf{V}$ can be derived (see supplemental material). $\mathbf{F}_\uparrow$ denotes re-projecting inputs to the original-dimension space and extending elements along the ray dimension with interleaving repetition. Then the attention feature $\boldsymbol{\mathcal{C}}_l$ of 3D location can be interpreted as:

$$\begin{aligned} \boldsymbol{\mathcal{C}}_l &= \mathbf{F}_\uparrow(\mathrm{Att}(\mathbf{Q}_l)) + emb(\mathbf{x}), \\ \text{where} \quad \mathrm{Att}(\mathbf{Q}_l) &= \mathrm{softmax}\left(\frac{\mathbf{Q}_l\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}. \end{aligned} \tag{2}$$

Attention feature of view direction $\boldsymbol{\mathcal{C}}_d$ can be calculated in the same pattern as Eq. (2), with $\boldsymbol{\mathcal{C}}_l$ concatenated to the

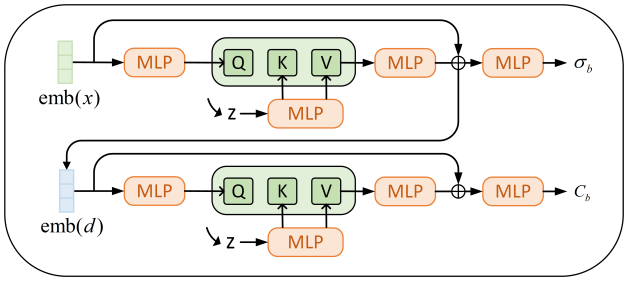


Figure 3: **Attention block.** We use the latent variable z sampled from the background distribution as **K**ey and **V**alue, and embedded 3D position and direction as **Q**uery. $\oplus$ denotes element-wise addition.

input. Then the volumetric density and RGB color will be generated by subsequent networks $\sigma_b = \textbf{MLP}(\boldsymbol{\mathcal{C}}_l)$ and $\mathbf{c}_b = \textbf{MLP}(\boldsymbol{\mathcal{C}}_d)$. Same to NeRF, the density part is solely determined by the position $\mathbf{x}$. The rendered RGB color $\hat{\mathbf{C}}_b(\mathbf{r})$ is denoted as:

$$\hat{\mathbf{C}}_b(\mathbf{r}) = \int_{s_n}^{s_f} T(s)\sigma_b(\mathbf{r}(s),\mathbf{z})\mathbf{c}_b(\mathbf{r}(s),\mathbf{d},\mathbf{z})ds$$
$$\text{where } T(s) = \exp\left(-\int_{s_n}^{s}\sigma_b(\mathbf{r}(p),\mathbf{z})dp\right). \quad (3)$$

Motion Representation

We employ a dynamic NeRF to describe the time-varying motion in real-world dynamic scenes. At i-th time instance, we have:

$$\Phi_d : (\mathbf{x},\mathbf{d},i) \rightarrow (\sigma_d, \mathbf{c}_d, \boldsymbol{\mathcal{F}}^{(i)}) \quad (4)$$

The rendered color $\hat{\mathbf{C}}^i(\mathbf{r})$ of the pixel corresponding to i-th time step is an integral over the radiance weighted by accumulated opacity $T(s) = \exp\left(-\int_{s_n}^{s}\sigma_d(\mathbf{r}(p),i)dp\right)$:

$$\hat{\mathbf{C}}_d^{(i)}(\mathbf{r}) = \int_{s_n}^{s_f} T(s)\sigma_d\left(\mathbf{r}(s),i\right)\mathbf{c}_d\left(\mathbf{r}(s),\mathbf{d},i\right)ds \quad (5)$$

Dynamic NeRF also predicts scene flow in neighboring time instance to the current time step, $\boldsymbol{\mathcal{F}}^{(i)} = (\boldsymbol{f}_{\text{fw}}^{(i)}, \boldsymbol{f}_{\text{bw}}^{(i)})$. Instead of rendering, this flow prediction is used to create region masks and geometric constraints, as explained in the loss section.

At each time step, only one observation view is available. Consequently, generating the implicit representation of dynamic objects from a single video is an ill-posed problem. To effectively optimize our dynamic NeRF and maximize the usage of temporal information, we introduce multiple regularization losses.

Occlusion Weight

Our background pipeline models the time-invariant component, and our dynamic NeRF generates a per-time-step deformation field. To combine them at pixel-level (ray-level), here we introduce our occlusion weight module, which targets rendering transmittance weight between static and dynamic components.

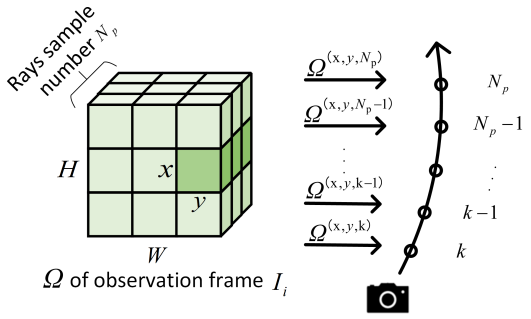


Figure 4: **Transmittance on each ray.** Each ray sample for a ray at observation time step i at location x, y is weighted by transmittance $\boldsymbol{\Omega}$ over the ray dimension.

As shown in Fig. 4, at i-th frame, we calculate the transmittance $\boldsymbol{\Omega}_b, \boldsymbol{\Omega}_d \in \mathbb{R}^{H\times W\times N_p}$ corresponding to each sample for each ray and the transmittance $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{H\times W}$ [1] corresponding to each ray using the warped previous rendering result $\boldsymbol{I}_{i-1}$ concatenated as input:

$$\mathbf{F}_w : (\mathbf{x},\mathbf{d},\boldsymbol{I}_{i-1}) \rightarrow (\boldsymbol{\Omega}_b, \boldsymbol{\Omega}_d, \boldsymbol{\mathcal{T}}). \quad (6)$$

Warped $\boldsymbol{I}_{i-1}$ is concatenated to aid in learning the location of motion pixels. We divide the transmittance weight into two parts. $\boldsymbol{\Omega}$ acts on the rendering process of two NeRFs and leads the generation of each light point. $\mathcal{T}$ controls the mixing of the dynamic and background rendering results. Two different losses are designed to fit their respective optimization goals. The discrete integral formulation of rendered color $\hat{\mathbf{C}}_b(\mathbf{r})$ and $\hat{\mathbf{C}}_d^{(i)}(\mathbf{r})$ are weighted along the ray dimension (see supplemental material), giving us the final rendered pixel at time step i for location x, y:

$$\begin{aligned}\hat{\mathbf{C}}^{(i)}(\mathbf{r}) =& \boldsymbol{\mathcal{T}}^{(x,y)}\boldsymbol{\Omega}_b^{(x,y)}\hat{\mathbf{C}}_b(\mathbf{r}) \\ &+ (1-\boldsymbol{\mathcal{T}}^{(x,y)})\boldsymbol{\Omega}_d^{(x,y)}\hat{\mathbf{C}}_d^{(i)}(\mathbf{r})\end{aligned} \quad (7)$$

Given the entire rays set $\mathcal{R} \in \mathbb{R}^{H\times W}$, the final rendering image can be calculated from $\hat{\boldsymbol{I}}(x,y,i) = \{\hat{\mathbf{C}}^{(i)}(\mathbf{r})|\mathbf{r}\in\mathcal{R}\}$. Our experiments show that the occlusion weight helps decouple static and dynamic scene components.

Loss

The lack of input views makes modeling time-dependent scenes ill-posed and difficult to optimize. To this end, several losses are designed to control pipeline training.

Reconstruction loss. To learn the implicit representation from a N-frames input video clip $\mathscr{D} = \left\{\left(\mathbf{x}^i,\mathbf{d}^i,\boldsymbol{I}^i\right)|i=1,\ldots,N\right\}$, we apply a reconstruction loss penalizing the difference between the target

[1] In practice, NeRF-based methods are constrained by GPU memory space and therefore randomly sample N_{rand} rays instead of rendering all $H \times W$ rays for an image at each training iteration.

(a) Ours (b) NeRF (c) Gao et al. (d) NSFF (e) Ours (f) Reference

Figure 5: **Comparison of novel view synthesis details.** Our method shows a more effective way of recovering texture in both static and motion regions.

video frame $\mathbf{C}^{(i)}(\mathbf{r})$ and our yielding volume-rendered image $\hat{\mathbf{C}}^{(i)}(\mathbf{r})$:

$$\mathcal{L}_{\text{rec}} = \sum_{i=1}^{N} \sum_{\mathbf{r}\in\mathcal{R}} \left\| \hat{\mathbf{C}}^{(i)}(\mathbf{r}) - \mathbf{C}^{(i)}(\mathbf{r}) \right\|_2^2 \tag{8}$$

Transmittance loss. Our occlusion weight module mitigates the ambiguous problem in occluded components between frames by rendering transmittance weight, targeting at predicting the overlay relationship for rendering results between the backdrop and dynamic foreground. Ideally, each ray should be rendered from either dynamic or static parts, hence the corresponding transmittance should be regularized to 0 or 1. Ray samples need to exist initially, and this encourages transmittance for each sample to be close to one:

$$\begin{aligned} \mathcal{L}_{\text{w}} = &\sum_{\mathbf{r}\in\mathcal{R}} -\boldsymbol{\mathcal{T}}^{(x,y)} \log(-\boldsymbol{\mathcal{T}}^{(x,y)} + \epsilon) \\ &+ \sum_{\mathbf{r}\in\mathcal{R}} \sum_{k=1}^{N_p} \left\| 1 - \boldsymbol{\omega}_b^{(k)} \right\|_1 + \left\| 1 - \boldsymbol{\omega}_d^{(k)} \right\|_1 . \end{aligned} \tag{9}$$

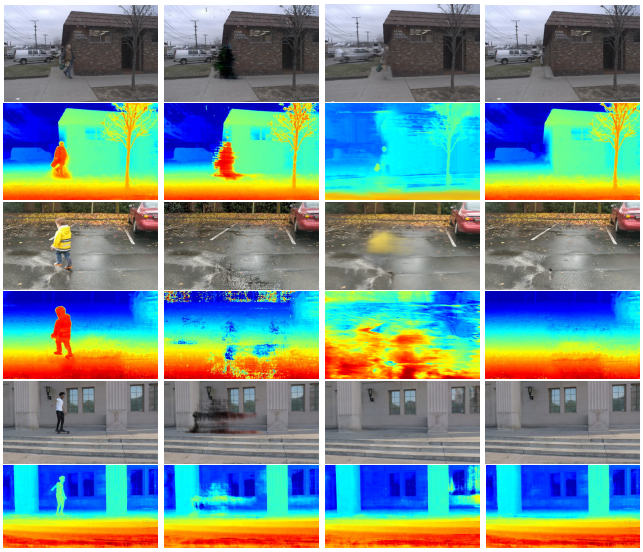
Depth and optical flow serve as geometry supervision priors in many NeRF-based methods, aiding resolve motion–appearance ambiguity and accelerating convergence (Deng et al. 2022; Xian et al. 2021; Li et al. 2021). Here we use depth loss $\mathcal{L}_{\text{depth}}$ and flow consistence losses $\mathcal{L}_{\text{cons}}, \mathcal{L}_{\text{flow}}$ for regularization, which will be discussed detailedly in the supplemental material. With using λ coefficients weight each term, the overall loss can be interpreted as:

$$\begin{aligned} \mathcal{L}_{\text{overall}} = &\lambda_{\text{rec}} \mathcal{L}_{\text{rec}} + \lambda_{\text{depth}} \mathcal{L}_{\text{depth}} + \lambda_{\text{cons}} \mathcal{L}_{\text{cons}} \\ &+ \lambda_{\text{flow}} \mathcal{L}_{\text{flow}} + \lambda_{\text{w}} \mathcal{L}_{\text{w}} \end{aligned} \tag{10}$$

Experiments

Datasets and Metrics

Urban driving scenes. To evaluate the performance in outdoor autonomous driving scenes, we built a dataset of real-world street scenes captured from an autonomous-driving collection Argoverse (Wilson et al. 2021). All scenes are taken in urban street, using one of seven high-resolution surrounded monocular cameras recorded at 30 Hz. Each gathered video clip depicts an outdoor scene with vehicles or people performing dynamic actions, spans 20-50 frames , and is scaled to a resolution of 436×272. As there is only



(a) Entire Scene (b) NSFF (c) Gao et al. (d) Ours

Figure 6: **Comparison on decoupling static background from entire scenes.** (a) shows the entire scene, (b-d) shows the separation results of the static background from NSFF, Gao et al., and ours. DetRF obtains an uncontaminated background in both RGB and depth space.

one monocular camera deployed, we carry out time interpolation for evaluation in each scene. The recording frequency is decreased to 15 Hz for training. For more information, please refer to the supplementary material.

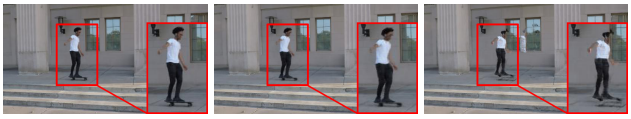
NVIDIA dynamic scenes. We then evaluate the dynamic scenes from NVIDIA (Yoon et al. 2020), which comprise 4 scenes captured by a hand-held monocular moving camera and 8 scenes captured by 12 stationary multi-view cameras evenly distributed and manually synchronized. 7 multi-view scenes are selected as our evaluation target for their correct camera pose estimation. For each scene, 20 to 30 frames are extracted from the source video for training. At each time instance, just one of these cameras is utilized as input, and other 11 camera views per time step are used for evaluation. Selected clips are downsized to a resolution of 510×272.

Implementation Details

All experiments are carried out on 8 NVIDIA 3090 GPUs in the Pytorch framework. We use 48 threads of Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz to expedite data loading. The weight coefficients for $\mathcal{L}_{\text{rec}}$, $\mathcal{L}_{\text{depth}}$, $\mathcal{L}_{\text{cons}}$, $\mathcal{L}_{\text{flow}}$, and $\mathcal{L}_{\text{w}}$ are 1, 3e-2, 3e-2, 1e-2, and 5e-1, respectively. Adam optimizer (Kingma and Ba 2014) with $\beta_1 = 0.9, \beta_2 = 0.999$, and ϵ=1e-8 is used with learning rate of 3e-4. We generate $N_{\text{rand}} = 1024$ rays randomly in a batch for each training iteration, and we sample $N_p = 128$ ray points along each ray. We initially warm up our occlusion weight module to get the right flow estimation for 2e5 iterations supervised by truth flow estimated from RAFT (Teed and Deng 2020). Thereafter, all losses are involved to train another 5e5 iterations. To get truth depth, we leverage the state-of-the-art monocular image depth estimation method (Ranftl et al. 2020). The above estimation methods are the same as those used in (Li et al. 2021; Gao et al. 2021), in order to eliminate any unfair bias towards either depth or flow. COLMAP SfM technique (Schonberger and Frahm 2016) is applied to estimate the camera poses, scene bounds, and camera params with the premise that intrinsics and extrinsics are fixed.

(a) Entire Scene (b) Ours (c) w/o RW

Figure 7: **Visual ablation on the occlusion weight module.** RW ensures proper separating capacity.



(a) Ours (b) Ours w/o $\mathcal{L}_{\text{depth}}$ (c) Ours w/o $\mathcal{L}_{\text{flow}}$

Figure 8: **Depth and flow regularization.** Depth loss reduces the occurrence of ghosting artifacts. And flow loss helps rendering motion areas accurately.

Quantitative Evaluation

To quantitatively validate the performance of our method, we carry out experiments on NVIDIA dynamic scenes and urban driving scenes. We compare DetRF with four baselines. The first is a classical learning-based method (Yoon et al. 2020). The second is vanilla NeRF (Mildenhall et al. 2020). The third and the fourth are recent state-of-the-art methods (Li et al. 2021; Gao et al. 2021) that also use multiple NeRFs to perform dynamic novel view synthesis. All baseline methods are implemented using the author's provided official codebase and default hyper-parameters.

Tab. 1 and Tab. 3 show the performance of our method on NVIDIA dynamic scenes and Urban driving scenes compared to other baseline methods. DetRF achieves the best performance on average and in most scenes. Due to space limitations, we only present the results of PSNR and LPIPS on NVIDIA dynamic scenes, and the average comparision on Urban driving scenes. Please refer to our supplemental material for complete comparison results.

Ablation study. We verify the effectiveness of different modules in our approach, including the backdrop encoder (BE), the ray occlusion weight module (RW), and two regularization losses $\mathcal{L}_{\text{depth}}$ and $\mathcal{L}_{\text{flow}}$. We conduct our ablation study on NVIDIA dynamic scenes shown in Tab. 2. The first row shows the performance of only two naive NeRFs, while the second row proves the effectiveness of BE, which brings $0.79 \uparrow$ and $.0245 \downarrow$ improvement on PSNR and LPIPS. Besides, RW also shows an increase of 5.5% on PSNR and a decrease of 27.1% on LPIPS. The depth loss and the flow loss further show their efficacy in the fourth and fifth rows, respectively. Fig. 8 provides their visual ablation results.

	Skating		Balloon1		Balloon2		Truck		Umbrella		Jumping		Playground		Ave	
	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS
Yoon et al.	24.33	.1563	19.33	.1625	19.78	.1797	28.78	.0835	20.37	.1696	21.51	.2294	17.43	.2124	21.65	.1705
NeRF	25.40	.0906	21.41	.1412	23.30	.0705	27.93	.0975	21.23	.2341	22.21	.1511	20.75	.1566	23.18	.1345
NSFF	33.42	.0235	23.39	.0762	27.85	.0490	**32.09**	.0372	23.67	.1153	26.67	.0624	23.77	.0857	27.26	.0642
Gao et al.	33.83	.0225	23.60	.0736	**27.90**	.0457	31.00	.0321	**24.25**	.1071	26.72	.0584	23.82	.0823	27.30	.0603
Ours	**34.52**	**.0171**	**23.87**	**.0666**	27.60	**.0411**	31.75	**.0232**	24.20	**.1037**	**27.00**	**.0463**	**23.94**	**.0734**	**27.55**	**.0530**

Table 1: **Quantitative comparison results on NVIDIA dynamic scenes.** The **best** result is in bold, and the second-best is underlined in each column.

Method	BE	RW	$\mathcal{L}_{\text{flow}}$	$\mathcal{L}_{\text{depth}}$	PSNR/LPIPS
Base					25.23/.1144
Base+BE	✓				26.02/.0899
Base+BE+RW	✓	✓			27.44/.0655
Ours+$\mathcal{L}_{\text{flow}}$	✓	✓	✓		27.50/.0601
Ours+$\mathcal{L}_{\text{flow}}$+$\mathcal{L}_{\text{depth}}$	✓	✓	✓	✓	**27.55/.0530**

Table 2: **Ablation study.** We evaluate different modules on NVIDIA dynamic scenes dataset. PSNR and LPIPS are shown on average.

	Urban Driving Scenes		
	PSNR	SSIM	LPIPS
Yoon et al.	29.19	.9198	.1036
NeRF	30.22	.9260	.0859
NSFF	32.81	.9569	.0413
Gao et al.	32.85	.9571	.0415
Ours	**33.16**	**.9586**	**.0359**

Table 3: **Quantitative comparison results (average) on Urban driving scenes.** The **best** result is in bold, and the second-best is underlined in each column.

Qualitative Evaluation

While this section has many qualitative evaluations, we recommend the reader watches the supplementary video for a more comprehensive understanding. In Fig. 5, we compare novel view rendering visually. DetRF surpasses all other methods on dynamic scenes. NSFF recovers dynamic objects properly but produces imprecise details, such as the wall in the first scene and the lamp in the second scene. Gao et al. capture finer details but does poorly in recovering exact motion. Our method presents the highest qualitative results in recovering static texture and motion details.

Background disentangling. We further show that our approach could disentangle static backgrounds from entire dynamic scenes self-supervised. DetRF can restore the unoccluded clean background simply by observing the occluded regions in different frames of a video clip. Other NeRF-based methods also tend to infer occlusion relationships, but get subpar performance. As seen in Fig. 6, compared to other multi-NeRF-based methods, DetRF obtains the most accurate separation results in RGB-D space with fewer artifacts and blurs. In Fig. 7, we also show ablation that without our occlusion weight, the decoupled synthesized results are fuzzy and noisy. We include more comparison results with SOTA decoupled method D^{2}NeRF (Wu et al. 2022) in the supplementary material.

(a) Ghosting in motion areas. (b) Half-baked disentangle results.

Figure 9: **Limitations.** Our method fails to recover the motion area (a) and is unable to produce a complete decoupling result (b) with incorrect flow or depth estimation.

Though the separation of static components from dynamic scenes has been addressed in 2D geometric and deep learning algorithms (Ebdelli, Le Meur, and Guillemot 2015; Li et al. 2022b), the geometry technique of inpainting lack 3D comprehension, leading to restricted results in arbitrary view synthesis and time interpolation. Our approach gives a 3D pattern for dealing with this problem and allows the reconstruction of a decoupled implicit 3D scene representation. This makes it possible to generate novel views at arbitrary viewpoints and any input time step.

Conclusion

We presented DetRF, a method for modeling dynamic scenes in the real world and performing novel view synthesis from casual monocular videos. DetRF provides a 3D pattern that could decouple the occlusion area into static and dynamic components and recovers a clean background. We empirically demonstrate superior quantitative and qualitative performance on both urban driving scenes and NVIDIA dynamic scenes. Our work suggests various directions for future research, such as autonomous driving dynamic scene simulation and editor models for outdoor dynamic scenes.

Limitations. Same as other NeRF-based methods, our approach relies on accurate estimation, *e.g.* camera poses and optical flow, to enable the proper representation of dynamic and static parts. Inaccurate ones may lead to motion artifacts and incorrect decoupling results (See Fig. 9).

References

Attal, B.; Huang, J.-B.; Richardt, C.; Zollhoefer, M.; Kopf, J.; O'Toole, M.; and Kim, C. 2023. Hyperreel: High-fidelity

6-dof video with ray-conditioned sampling. In *Proceedings of the CVPR*.

Attal, B.; Laidlaw, E.; Gokaslan, A.; Kim, C.; and O'Toole, M. 2021. T"oRF: Time-of-Flight Radiance Fields for Dynamic Scene View Synthesis. In *NeurIPS*.

Avidan, S.; and Shashua, A. 1997. Novel view synthesis in tensor space. In *Proceedings of the CVPR*.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; and Hedman, P. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the CVPR*.

Cai, G.; Yan, K.; Dong, Z.; Gkioulekas, I.; and Zhao, S. 2022. Physics-Based Inverse Rendering using Combined Implicit and Explicit Geometries. In *Proceedings of the CVPR*.

Chen, Z.; and Zhang, H. 2019. Learning implicit fields for generative shape modeling. In *Proceedings of the CVPR*.

Debevec, P. E.; Taylor, C. J.; and Malik, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *CGIT*, 11–20.

Deng, K.; Liu, A.; Zhu, J.-Y.; and Ramanan, D. 2022. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the CVPR*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Du, Y.; Zhang, Y.; Yu, H. X.; Tenenbaum, J. B.; and Wu, J. 2021. Neural Radiance Flow for 4D View Synthesis and Video Processing. In *Proceedings of the ICCV*.

Ebdelli, M.; Le Meur, O.; and Guillemot, C. 2015. Video inpainting with short-term windows: application to object removal and error concealment. *IEEE Transactions on Image Processing*, 24(10): 3034–3047.

Fang, S.; Xu, W.; Wang, H.; Yang, Y.; Wang, Y.; and Zhou, S. 2023. One is all: Bridging the gap between neural radiance fields architectures with progressive volume distillation. In *Proceedings of the AAAI*.

Gao, C.; Saraf, A.; Kopf, J.; and Huang, J. B. 2021. Dynamic View Synthesis from Dynamic Monocular Video. In *Proceedings of the ICCV*.

Hedman, P.; Alsisan, S.; Szeliski, R.; and Kopf, J. 2017. Casual 3D photography. *TOG*, 36(6): 1–15.

Ivanov, V. K.; Vasin, V. V.; and Tanana, V. P. 2013. *Theory of Linear Ill-Posed Problems and its Applications*, volume 36. Walter de Gruyter.

Jayasundara, V.; Agrawal, A.; Heron, N.; Shrivastava, A.; and Davis, L. S. 2023. FlexNeRF: Photorealistic free-viewpoint rendering of moving humans from sparse views. In *Proceedings of the CVPR*.

Jiang, W.; Yi, K. M.; Samei, G.; Tuzel, O.; and Ranjan, A. 2022. NeuMan: Neural Human Radiance Field from a Single Video. In *Proceedings of the ECCV*.

Kar, A.; Hne, C.; and Malik, J. 2017. Learning a Multi-View Stereo Machine. In *NeurIPS*.

Kim, S.; Bae, J.; Yun, Y.; Lee, H.; Bang, G.; and Uh, Y. 2024. Sync-NeRF: Generalizing Dynamic NeRFs to Unsynchronized Videos. In *AAAI*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kundu, A.; Genova, K.; Yin, X.; Fathi, A.; Pantofaru, C.; Guibas, L.; Tagliasacchi, A.; Dellaert, F.; and Funkhouser, T. 2022. Panoptic Neural Fields: A Semantic Object-Aware Neural Scene Representation. In *Proceedings of the CVPR*.

Lei, C.; Xing, Y.; and Chen, Q. 2020. Blind Video Temporal Consistency via Deep Video Prior. In *NeurIPS*.

Li, T.; Slavcheva, M.; Zollhoefer, M.; Green, S.; Lassner, C.; Kim, C.; Schmidt, T.; Lovegrove, S.; Goesele, M.; Newcombe, R.; and Lv, Z. 2022a. Neural 3D Video Synthesis from Multi-view Video. In *Proceedings of the CVPR*.

Li, X.; Cao, Z.; Wu, Y.; Wang, K.; Xian, K.; Wang, Z.; and Lin, G. 2024. S-DyRF: Reference-Based Stylized Radiance Fields for Dynamic Scenes. In *Proceedings of the CVPR*.

Li, Z.; Li, L.; and Zhu, J. 2023. Read: Large-scale neural scene rendering for autonomous driving. In *Proceedings of the AAAI*.

Li, Z.; Lu, C.-Z.; Qin, J.; Guo, C.-L.; and Cheng, M.-M. 2022b. Towards an end-to-end framework for flow-guided video inpainting. In *Proceedings of the CVPR*.

Li, Z.; Niklaus, S.; Snavely, N.; and Wang, O. 2021. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. In *Proceedings of the CVPR*.

Liu, J.-W.; Cao, Y.-P.; Wu, J. Z.; Mao, W.; Gu, Y.; Zhao, R.; Keppo, J.; Shan, Y.; and Shou, M. Z. 2024. Dynvideo-e: Harnessing dynamic nerf for large-scale motion-and view-change human-centric video editing. In *Proceedings of the CVPR*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the ICCV*.

Luo, X.; Huang, J.; Szeliski, R.; Matzen, K.; and Kopf, J. 2020. Consistent Video Depth Estimation. In *Proceedings of ACM SIGGRAPH*, volume 39. ACM.

Ma, T.; Li, B.; He, Q.; Dong, J.; and Tan, T. 2023. Semantic 3D-aware Portrait Synthesis and Manipulation Based on Compositional Neural Radiance Field.

Martin-Brualla, R.; Radwan, N.; Sajjadi, M. S. M.; Barron, J. T.; Dosovitskiy, A.; and Duckworth, D. 2021. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *Proceedings of the CVPR*.

Mildenhall, B.; Hedman, P.; Martin-Brualla, R.; Srinivasan, P.; and Barron, J. T. 2022. NeRF in the Dark: High Dynamic Range View Synthesis from Noisy Raw Images. In *Proceedings of the CVPR*.

Mildenhall, B.; Srinivasan, P. P.; Ortiz-Cayon, R.; Kalantari, N. K.; Ramamoorthi, R.; Ng, R.; and Kar, A. 2019. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *TOG*, 38(4): 1–14.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the ECCV*.

Niemeyer, M.; Mescheder, L.; Oechsle, M.; and Geiger, A. 2019. Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision. In *Proceedings of the CVPR*.

Oswald, M. R.; Stühmer, J.; and Cremers, D. 2014. Generalized connectivity constraints for spatio-temporal 3d reconstruction. In *Proceedings of the ECCV*.

Park, K.; Sinha, U.; Barron, J. T.; Bouaziz, S.; Goldman, D. B.; Seitz, S. M.; and Martin-Brualla, R. 2021. Nerfies: Deformable Neural Radiance Fields. In *Proceedings of the ICCV*.

Peng, S.; Zhang, Y.; et al. 2021. Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans. In *Proceedings of the CVPR*.

Pumarola, A.; Corona, E.; Pons-Moll, G.; and Moreno-Noguer, F. 2021. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the CVPR*.

Ranftl, R.; Lasinger, K.; Hafner, D.; Schindler, K.; and Koltun, V. 2020. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Rematas, K.; Liu, A.; Srinivasan, P. P.; Barron, J. T.; Tagliasacchi, A.; Funkhouser, T.; and Ferrari, V. 2022. Urban Radiance Fields. In *Proceedings of the CVPR*.

Schonberger, J. L.; and Frahm, J.-M. 2016. Structure-from-motion revisited. In *Proceedings of the CVPR*.

Snavely, N.; Seitz, S. M.; and Szeliski, R. 2006. Photo tourism: exploring photo collections in 3D. In *Proceedings of ACM SIGGRAPH*, 835–846. ACM.

Song, L.; Chen, A.; Li, Z.; Chen, Z.; Chen, L.; Yuan, J.; Xu, Y.; and Geiger, A. 2023. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 29(5): 2732–2742.

Tancik, M.; Casser, V.; Yan, X.; Pradhan, S.; Mildenhall, B.; Srinivasan, P. P.; Barron, J. T.; and Kretzschmar, H. 2022. Block-NeRF: Scalable Large Scene Neural View Synthesis. In *Proceedings of the CVPR*.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the ECCV*.

Tewari, A.; Fried, O.; Thies, J.; Sitzmann, V.; Lombardi, S.; Sunkavalli, K.; Martin-Brualla, R.; Simon, T.; Saragih, J.; Nießner, M.; et al. 2020. State of the art on neural rendering. In *Computer Graphics Forum (CGF)*, volume 39, 701–727. Wiley Online Library.

Tretschk, E.; Tewari, A.; Golyanik, V.; Zollhfer, M.; Lassner, C.; and Theobalt, C. 2021. Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video. In *Proceedings of the ICCV*.

Trevithick, A.; and Yang, B. 2020. GRF: Learning a General Radiance Field for 3D Representation and Rendering. *arXiv preprint arXiv:2010.04595*.

Tucker, R.; and Snavely, N. 2020. Single-view view synthesis with multiplane images. In *Proceedings of the CVPR*.

Van Den Oord, A.; et al. 2017. Neural discrete representation learning. In *NeurIPS*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

Wang, F.; Tan, S.; Li, X.; Tian, Z.; Song, Y.; and Liu, H. 2023. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the ICCV*.

Wang, P.; Fan, Z.; Wang, Z.; Su, H.; Ramamoorthi, R.; et al. 2024. Lift3D: Zero-Shot Lifting of Any 2D Vision Model to 3D. In *Proceedings of the CVPR*.

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.; and Hays, J. 2021. Argoverse 2: Next Generation Datasets for Self-driving Perception and Forecasting. In *NeurIPS Datasets and Benchmarks 2021*.

Wu, T.; Zhong, F.; Tagliasacchi, A.; Cole, F.; and Oztireli, C. 2022. Dˆ 2NeRF: Self-Supervised Decoupling of Dynamic and Static Objects from a Monocular Video.

Xian, W.; Huang, J. B.; Kopf, J.; and Kim, C. 2021. Space-time Neural Irradiance Fields for Free-Viewpoint Video. In *Proceedings of the CVPR*.

Yan, Z.; Li, C.; and Lee, G. H. 2023. Nerf-ds: Neural radiance fields for dynamic specular objects. In *Proceedings of the CVPR*.

Yoon, J. S.; Kim, K.; Gallo, O.; Park, H. S.; and Kautz, J. 2020. Novel View Synthesis of Dynamic Scenes with Globally Coherent Depths from a Monocular Camera. In *Proceedings of the CVPR*.

Yuan, W.; Lv, Z.; Schmidt, T.; and Lovegrove, S. 2021. Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering. In *Proceedings of the CVPR*.

Zhang, K.; Riegler, G.; Snavely, N.; and Koltun, V. 2021. NeRF++: Analyzing and Improving Neural Radiance Fields. *arXiv:2010.07492*.

Zhang, Y.; Chen, G.; Chen, J.; and Cui, S. 2024. Aerial Lifting: Neural Urban Semantic and Building Instance Lifting from Aerial Imagery. In *Proceedings of the CVPR*.

Zhou, T.; Tucker, R.; Flynn, J.; Fyffe, G.; and Snavely, N. 2018. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*.