CTNeRF: Cross-Time Transformer for Dynamic Neural Radiance Field from Monocular Video

Xingyu Miao[a], Yang Bai[b], Haoran Duan[a], Fan Wan[a], Yawen Huang[c], Yang Long[a,*], Yefeng Zheng[c]

[a]*Department of Computer Science, Durham University, UK.*
[b]*Institute of High Performance Computing (IHPC), ASTAR, Singapore*
[c]*Jarvis Research Center, Tencent YouTu Lab, China.*

Abstract

The goal of our work is to generate high-quality novel views from monocular videos of complex and dynamic scenes. Prior methods, such as DynamicNeRF, have shown impressive performance by leveraging time-varying dynamic radiation fields. However, these methods have limitations when it comes to accurately modeling the motion of complex objects, which can lead to inaccurate and blurry renderings of details. To address this limitation, we propose a novel approach that builds upon a recent generalization NeRF, which aggregates nearby views onto new viewpoints. However, such methods are typically only effective for static scenes. To overcome this challenge, we introduce a module that operates in both the time and frequency domains to aggregate the features of object motion. This allows us to learn the relationship between frames and generate higher-quality images. Our exper-

*Corresponding author
Email addresses: xingyu.miao@@durham.ac.uk (Xingyu Miao), bai_yang@ihpc.a-star.edu.sg (Yang Bai), haoran.duan@ieee.org (Haoran Duan), fan.wan@durham.ac.uk (Fan Wan), yawenhuang@tencent.com (Yawen Huang), yang.long@ieee.org (Yang Long), yefengzheng@tencent.com (Yefeng Zheng)

iments demonstrate significant improvements over state-of-the-art methods on dynamic scene datasets. Specifically, our approach outperforms existing methods in terms of both the accuracy and visual quality of the synthesized views. Our code is available on https://github.com/xingy038/CTNeRF.
Keywords: Dynamic neural radiance field, monocular video, scene flow, transformer.

1. Introduction

Realistically rendering and presenting dynamic real-world scenes is a highly challenging research topic, with diverse applications in fields such as film production and virtual reality [1, 2, 3]. However, accurately modeling these scenes using traditional mesh-based methods can be difficult due to the complex movements of multiple objects and changes in factors like mirroring and transparency that occur during these movements. While multi-view-based methods have shown better results, they come with their own limitations. These methods require a large number of cameras, resulting in high costs and technical challenges like synchronization and data processing [4, 5, 6, 7]. Additionally, they are not easily applicable in daily life scenarios. Although reconstruction from monocular videos is a promising approach for scene reconstruction, novel view synthesis for monocular videos of dynamic scenes is more challenging.

Recent advancements in deep learning have made significant breakthroughs in novel view synthesis, with Neural Radiative Fields (NeRF) [8, 9] being one of the most notable contributions to this area. NeRF employs the position and viewing direction of a given image as a query, and employs volume render-

ing to generate the color of each pixel. However, these methods are primarily designed for static scenes and do not perform optimally when dealing with dynamic objects or scenes. To address this limitation, recent research has explored the application of this approach to monocular dynamic video [10, 11]. For example, some studies have focused on learning a deformable warp field [12] or a neural scene flow between adjacent frames [13, 14, 11, 15]. These efforts aim to extend the utility of NeRF and enable more robust and accurate synthesis of dynamic scenes. Despite the success achieved by NeRF-based methods for dynamic scenes, they still have some limitations. For instance, deformable warp field methods such as Nerfies [16] can handle long sequences but may not perform well for dynamic scenes with complex object motion. On the other hand, neural scene flow or neural trajectory methods like NSFF [13] can handle large movements in dynamic scenes, but their effectiveness is highly dependent on the accuracy of the predicted scene flow or trajectory.

We propose a novel approach that can be applied to dynamic scenes, enabling the handling of more complex motions and improving the rendering results. Our method draws inspiration from recent research on rendering static scenes [17, 18, 19, 20], where local image features are synthesized by aggregating them along epipolar lines from nearby views to enhance the rendering process. However, the apparent limitations assumed by these methods are violated by scenes in motion, making them unsuitable for direct application to dynamic scenes. To overcome this challenge, we have designed a module that aggregates changes in ray due to motion in the ray space, along with the obtained multi-view features [21]. This enables us to accurately consider both temporal and spatial changes in geometry and appearance,

resulting in better rendering of dynamic scenes. More specifically, we first input the extracted feature vector into a cross-time transformer. Next, we input the feature aggregated with time information into a ray transformer to find the relationship between the sampling points on the ray and obtain the aggregated feature. In addition, to strengthen the spatial-temporal relationship of feature vectors, we use a 2D fast Fourier transform frequency-domain feature aggregation module to obtain the aggregated features. Finally, we feed the fused feature vectors of these two features together with the queried rays into residual-based MLPs to output color and density. Experimental results demonstrate that our method can synthesize new views with high quality. Furthermore, compared to previous methods, our approach can render higher-quality ground-truth details of ground truth in dynamic regions. In summary, the contributions of our work are as follows:

1) A novel dynamic neural rendering field for dynamic monocular video, which can aggregate multi-view feature vectors to improve rendering novel view quality.

2) The aggregation of multi-frame feature vectors may lead to the potential loss or merging of intricate details into other features, thereby compromising the retention of crucial characteristics from the original data. To address this issue, we introduce a Ray-based cross-time transformer.

3) To mitigate potential blurring during feature aggregation, we introduce a Global Spatio-Temporal Filter.

4) Extensive experiments show that our method achieves superior novel view synthesis of dynamic scenes.

2. Related Work

2.1. Novel view synthesis

Recently, neural implicit representation methods like NeRF [22, 8, 9, 23, 24, 25, 26] have demonstrated significant potential for achieving high-quality rendering. NeRF employs multi-layer perceptrons (MLPs) to implicitly represent continuous scenes, yielding impressive view synthesis results. Despite their progress, NeRF-based methods necessitate training separate models for each scene, with optimization demanding varying training times. Applying these methods faces some other challenges, including unknown camera poses, boundary blur, and observation noise. For unknown camera poses, Li et al. [27] proposed a novel online scene representation method that can simultaneously learn to represent the target scene and estimate the camera pose from the RGB-D stream. For boundary-blurring, Barron et al. proposed Mip-NeRF [28], which uses sampling of cones instead of rays and considers scale information by integrating position encoding, so that the scene is represented in a scale of continuous values, and the rendering result is anti-aliased. In addition, variations of NeRF-based methods, such as PixelNeRF [26], MVS-NeRF [19], and IBRNet [17], exhibit promise in incorporating feature information to generalize to unseen scenes. However, their primary focus is on static scenarios, neglecting dynamic scenes with objects or cameras in motion. These methods estimate a 3D representation of a scene using multiple input images, which they then leverage for rendering novel views. Neverthe-

less, their applicability is limited in dynamic scenes, where assumptions of scene stability may lead to inaccuracies or artifacts in rendered images. Our work addresses this limitation by extending the approach to more challenging dynamic scenarios, concentrating on modeling complex object motion and synthesizing higher-quality novel views.

2.2. Dynamic region view synthesis

With NeRF [22, 8, 9] demonstrating impressive results in view synthesis tasks, several works have attempted to extend NeRF to tackle dynamic new view synthesis challenges [11, 14, 13, 29]. These methods can be classified into two main directions. The first direction involves using deformation fields to represent scenes [29, 12, 16, 30]. While this approach can handle long sequences of videos, its primary challenge is dealing with large motions in the scene. As this method typically warps the scene from the same frame, it can result in a lack of continuity throughout the entire sequence, such as Nerfies [16] and HyperNeRF [12]. The second approach is based on the time-varying 4D radiance fields approach [31, 13, 32]. These methods model dynamic scenes as time-varying continuous functions of appearance, geometry, and 3D scene motion by predicting the scene flow field. Although such methods can capture fast and complex motion in the scene, they usually require more accurate scene flow or trajectory guidance and cannot handle non-rigid deformation well. Our proposed method aggregates feature from nearby views to effectively handle this situation and improve rendering results.

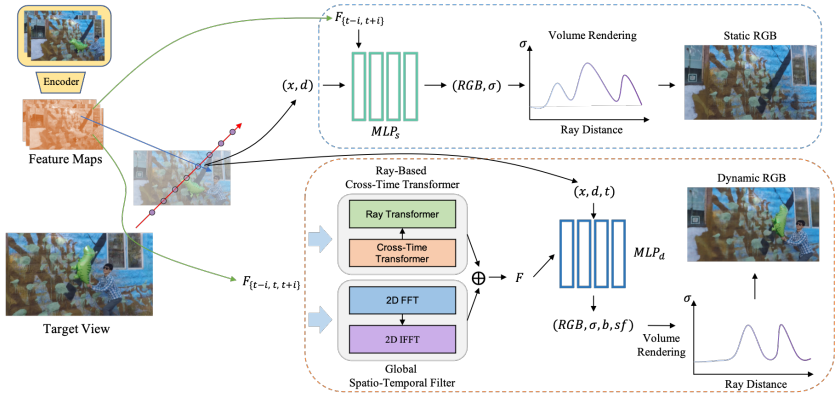

Figure 1: The pipeline of our model. Our model is composed of two main parts, each responsible for handling a different aspect of the input data. One component focuses on the static background, while the other deals with the dynamic foreground. These two sets of values are then blended together to obtain the final novel view.

3. Proposed Method

In this section, we present the proposed methods with the goal of enabling the trained model to query new viewpoints at any time and angle within a monocular video of a dynamic scene. Our system pipeline (Figure 1) can be divided into two parts: one part focuses on the static background, while the other part handles the dynamic foreground, and finally blends the two through blending to obtain the reconstructed video. Furthermore, similar to other time-varying NeRF-based techniques, we first optimize the model to reconstruct the input frame, before being utilized for rendering novel views. Instead of directly encoding 3D color and density in the weights of the MLPs like recent dynamic NeRF methods [11, 13], we borrow the idea of a recent generalized NeRF to aggregate features from views near the target view to enhance rendering. Below we describe our approach to multi-feature aggre-

gation, a ray-based cross-time aggregation module, and a boost module via frequency-domain effects.

3.1. Multi-view aggregation

We leverage two models to reconstruct the static and the dynamic area respectively and finally blend the color and density of the two through a blending value predicted by the dynamic model in the interval $[0, 1]$ to obtain the final reconstructed image.

3.1.1. Static Region Feature Aggregation

For the static region, we simply adopt the projection methods to query the position of the camera ray projected on the image coordinates at a certain point in the space. And then the corresponding feature vectors can be obtained by the method of bilinear difference. More specifically, the feature vectors of two adjacent frames are queried using the camera ray from the target viewpoint, which can be expressed as:

$$x_{t\pm i} = P_{\pm i} x_t \in \mathbb{R}^3 \tag{1}$$

where x_t is a point in space on the camera ray on the target view, the $x_{t\pm i} \in \mathbb{R}^3$ is the adjacent view, and the $P_{\pm i} = [R_{\pm i}, T_{\pm i}] \in \mathbb{R}^{4\times4}$ is the camera parameters, note that we set $i = 1$. And the queried feature can be expressed as:

$$F_{t\pm i} = E(proj\ \langle x_{t\pm i} \rangle) \in \mathbb{R}^d \tag{2}$$

$proj\ \langle \cdot \rangle$ represents the coordinates of the point projection image in space, and then uses a feature extractor $E(\cdot)$ to extract the features of the image, and

Figure 2: Aggregating feature vectors in an epipolar-aligned manner will cause errors in the rendering of the model, resulting in artifacts that degrade the quality of the model rendering novel views.

finally obtains the query feature vector $F_{t\pm i} \in \mathbb{R}^d$. The RGB c and density σ of the static region can use an MLP to query, which can be expressed as:

$$MLP_{\theta s}(x, y, z, t, d_t, F_{t\pm i}) = (c_s, \sigma_s) \tag{3}$$

the inputs include extracted feature vectors $F_{t\pm i} \in \mathbb{R}^d$, target view direction $d_t \in \mathbb{R}^3$, and space coordinates x, y, z.

3.1.2. Dynamic Region Feature Aggregation

For the dynamic region, we cannot use the same method as the static region to aggregate features. The object movement violates the static hypothesis, computing adjacent frames only with camera parameters cannot handle this change. Therefore, inspired by recent neural scene flow work [11, 13], we first use predicted scene flow to warp the camera rays to describe the motion of a point in space in the scene, which can be expressed as:

$$x_{t+1} = x_t + s_{fw} \in \mathbb{R}^3 \tag{4}$$

$$x_{t-1} = x_t + s_{bw} \in \mathbb{R}^3 \tag{5}$$

where, $s_{fw} \in \mathbb{R}^3$ and $s_{bw} \in \mathbb{R}^3$ are the predicted scene flow. And then, we can obtain the corresponding feature vectors using Equation (2). Note that

we use a four-layer MLP to predict this scene flow, while also outputting a blend value to weigh the color and density of static and dynamic regions. Thus, it can be expressed as:

$$MLP_{\theta d}(x, y, z, t, d_t, F_{t\pm i}) = (c_d, \sigma_d, s_{fw}, s_{bw}, b) \tag{6}$$

$s_{fw} \in \mathbb{R}^3$ and $s_{bw} \in \mathbb{R}^3$ are the predicted scene flow, b is the predicted blending value and the $i = \{0, 1\}$.

3.1.3. Combining static and dynamic models

Time-varying-based dynamic models typically undergo very much deformation to reliably infer correspondences over larger time intervals, however, static regions should be consistent. In order to render complete and high-quality content in the static area of the new view composition, we follow the idea of NSFF [13] and use two separate models (static and dynamic) to model the entire scene. Through the above methods, we can obtain static and dynamic colors c_s, c_d and densities σ_s, σ_d respectively, then the volumetric radiance field can then be rendered into a 2D image via:

$$\begin{aligned} C_{full}(r) = \int_{t_n}^{t_f} (T_d(t)\sigma_d(t)c_d(t)b \\ + T_s(t)\sigma_s(t)c_s(t)(1-b))dt \end{aligned} \tag{7}$$

$$T_{\{s,d\}}(t) = exp\left(-\int_{t_n}^{t} \sigma_{\{s,d\}}(s)ds\right) \tag{8}$$

The rendered pixel values for camera ray r can then be compared to the corresponding ground truth pixel values:

$$\mathcal{L}_{pho} = \sum_r \left\| \hat{C}(r) - C_{gt}(r) \right\|_2^2 \tag{9}$$

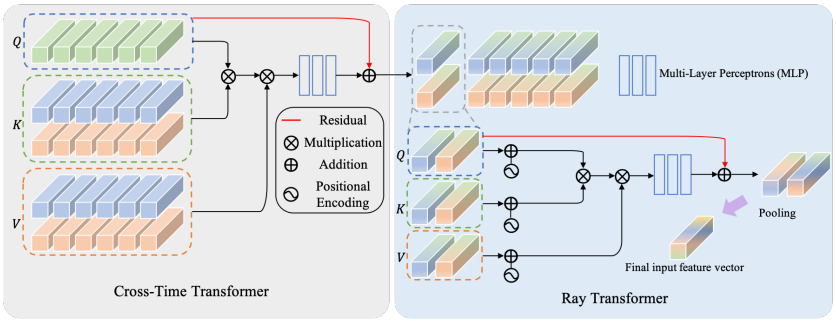

Figure 3: The pipeline of the RBCT module. The model consists of two main components: the cross-time transformer on the left and the ray transformer on the right. The left component takes a set of feature vectors from consecutive frames as input and applies cross-time attention to aggregate these vectors with the current frame. The resulting feature vector is then passed to the right component, which uses ray attention to aggregate feature vectors from multiple sampling points along each ray. Finally, a pooling operation is applied to these vectors to obtain the final aggregated feature vector.

where $C(r)$ includes the static, dynamic, and blended regions. Directly aggregating these features can enhance the representation of target feature maps, and the effect of improving quality can be obtained in the reconstruction stage. However, through our observations during the rendering process of the novel view, artifacts of adjacent frames (see Figure 2) manifest in the novel view, significantly compromising the performance of our model. In addition, we observed that our methods will produce some non-rigid deforms when rendering novel views of dynamic scenes, it also will affect the quality of novel views synthesis. Thus, we propose a ray-based cross-time (RBCT) aggregation module to handle this issue.

3.2. Ray-Based Cross-Time Aggregation Module

3.2.1. Cross-Time Aggregation

By means of the aforementioned method, the correlation between the frontal and posterior frames can be integrated using the camera rays from the current viewing angle. However, it should be noted that there exists a temporal relationship among the input feature vectors of adjacent frames, and a direct aggregation would overlook this temporal variation. Therefore, we design a cross-time converter to enable the feature vector of the target frame to attentively focus on the variations in the feature vector of its adjacent frame (as illustrated in the left figure, see Figure 3). The cross-time transformer leverages a classic cross-attention mechanism [33, 34, 35, 36] that allows for a variable number of inputs, given the frame-to-frame changes involved:

$$\hat{F}_{t\pm i} = \text{Cross-Time Transformer}(F_{t\pm i}, F_t) \tag{10}$$

For a given target frame F_t, the query vector Q_t is derived by linearly transforming the original feature vector:

$$Q_t = W_Q \cdot F_t \tag{11}$$

Similarly, key ($K_{t\pm i}$) and value ($V_{t\pm i}$) vectors for adjacent frames ($F_{t\pm i}$) are obtained through linear transformations:

$$K_{t\pm i} = W_K \cdot F_{t\pm i}, \quad V_{t\pm i} = W_V \cdot F_{t\pm i} \tag{12}$$

The attention scores are computed using the dot product of the query and key vectors, scaled by a factor $\sqrt{d_k}$:

$$Attn = \frac{Q_t \cdot K_{t\pm i}^T}{\sqrt{d_k}} \tag{13}$$

Finally, the target frame's feature vector $\hat{F}_{t\pm i}$ is updated by computing a weighted sum of the value vectors:

$$\hat{F}_{t\pm i} = Softmax(Attn) \cdot V_{t\pm i} \tag{14}$$

Our ablation studies demonstrate that this approach effectively enhances the quality of synthesized images.

3.2.2. Ray Aggregation

While aggregating camera views, we have successfully explored the relationship between the camera ray's point in space and its corresponding feature vector. However, one crucial aspect that has been overlooked is the relationship between feature vectors from adjacent frames. To address this limitation and improve the overall representation, we introduced a cross-time transformer mechanism, allowing the target frame to focus on the inter-frame relationships and enhancing the global correlation. Despite the progress achieved with the cross-time transformer, we encountered an issue. Specifically, this approach failed to accurately associate the local relationship between the camera ray's sampling point and its corresponding per-frame feature vector. To overcome this limitation, we propose a method similar to the depth bin utilized in stereo-matching algorithms when calculating the cost volume [37]. This involves considering each sampling point along the entire ray to match a specific pixel in the form of a matching score.

More specifically, we introduce a new ray transformer that enables the mutual focus of feature vectors corresponding to samples on a ray. The ray transformer is composed of two core components of the classical transformer

[33]: position encoding and self-attention. It can be expressed as:

$$\mathcal{F} = \text{Ray Transformer}(\hat{F}_{t\pm i}) \tag{15}$$

Specifically, we define:

$$Q = W_Q \cdot \hat{F}_{t\pm i}, K = W_K \cdot \hat{F}_{t\pm i}, V = W_V \cdot \hat{F}_{t\pm i} \tag{16}$$

Here, W_Q, W_K, and W_V are learned weight matrices. Subsequently, we compute attention scores and utilize these attention weights to calculate a weighted average, yielding a new feature representation:

$$\mathcal{F} = Softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V \tag{17}$$

Given M (we set M to 64) samples along a ray, our ray transformer transforms the output of its input cross-time transformer, resulting in an aggregated feature vector. The introduction of this ray transformer allows the model to focus on the feature vectors corresponding to samples along the ray in adjacent frames, capturing finer local relationships and addressing the limitations of the previous cross-time transformer in this regard. Our ablation experiments demonstrate that the proposed ray transformer significantly improves the quality of the final synthesized image.

3.3. Frequency Domain Aggregation Module

We introduce a novel spatio-temporal feature learning module termed the Global Spatio-Temporal Filter (GSTF), inspired by recent advancements in frequency-domain-based methodologies [38] aimed at enhancing the rendering quality of new views [39]. The primary objective of GSTF is to elevate the representation of feature vectors by capturing both spatial and temporal

relationships through specialized frequency filters. In our approach, GSTF is meticulously crafted to discern and learn distinct frequency filters at each spatial location. This enables the modeling of temporal variations within feature vectors across different spatial positions. The core mechanism involves the transformation of both temporal and spatial features at each location into frequency feature spectra. This transformation is achieved through a two-dimensional Fast Fourier Transform (FFT) [40]. The frequency filter, learned through GSTF, acts as a modulator on this transformed spectrum. Subsequently, we revert this modulated spectrum back to the time domain using an inverse FFT. This comprehensive process allows GSTF to effectively encode the intricate interplay between time and space in the feature vectors, contributing to the improvement of new view rendering. To gain a better understanding of our GSTF design, let's first review the convolution theorem in the field of digital signal processing [41]. Given a sequence of feature signals with T points $(f[t], 0 \leq t \leq T-1)$, we can calculate its discrete spectrum $S[k]$ using Discrete Fourier Transform (DFT) via:

$$S[k] = \sum_{t=0}^{T-1} f[t] e^{-j(2\pi/T)kt}, 0 \leq k \leq T-1 \tag{18}$$

In the equation above, j represents the imaginary unit. The Discrete Fourier Transform (DFT) is a one-to-one orthogonality decomposition. Moreover, we can use the DFT outputs to reconstruct input signals using Inverse Discrete Fourier Transform (IDFT) via:

$$f[t] = \frac{1}{T} \sum_{k=0}^{T-1} S[k] e^{j(2\pi/T)kt}, 0 \leq t \leq T-1 \tag{19}$$

Specifically, we first convert the features into frequency domain signals. Next, these frequency domain signals are filtered, and then the filtered signals are

Algorithm 1 Global Spatio-Temporal Filter

```
Initialization: learnable weight w
x = torch.fft.rfft2(x, dim=(0,1))
x = x * w
x = torch.fft.irfft2(x, dim=(0,1))
```

reconstructed into time domain features. Our GSTF can be easily used in modern deep-learning frameworks such as PyTorch [42], The pseudocode of PyTorch is shown in Algorithm 1. Finally, these time-domain features are merged with the RBCT-processed features to achieve effective aggregation of time-domain and frequency-domain features. It can be expressed as:

$$F = f[t] + \mathcal{F} \tag{20}$$

Where $f[t]$ is the frequency-domain features, $\mathcal{F}$ is the time-domain features, and F is the aggregated features. Through ablation experiments, we demonstrate that our global filtering mechanism is an effective spatial information mixing method. The GSTF module improves the texture quality of new views and alleviates blurring and artifacts often observed in new view synthesis.

3.4. Regularization

It is well known that monocular video reconstruction of complex dynamic scenes is an ill-posed problem, and using only a photometric error for supervision cannot avoid local minima. Therefore, many regularization strategies have been used in previous work [11, 13]. We continue to use the previous strategy and add several regularization items. Specifically, it includes the

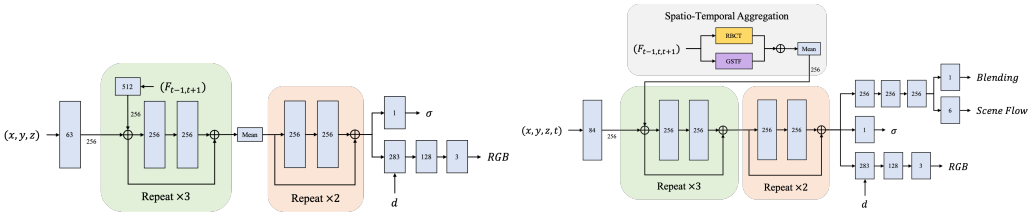

Figure 4: Network architectures of our static and dynamic representations.

following three parts:

$$\mathcal{L} = \mathcal{L}_{data} + \mathcal{L}_{small} + \mathcal{L}_{pho} \tag{21}$$

The $\mathcal{L}_{data}$ is a data-driven prior regularization term, composed of pre-trained monocular depth estimation network and optical flow estimation network consistency prior [11]. The $\mathcal{L}_{pho}$ is provided by Equation (9). Our model is highly dependent on the accuracy of scene flow, thus we provide an additional regularization term for scene flow. The $\mathcal{L}_{small} = \|s_{fw}\|_1 + \|s_{bw}\|_1 + \|s_{fw} + s_{bw}\|_1$ is a regularization term that minimizes scene flow.

3.5. Architecture

Our approach utilizes two distinct types of models: static and dynamic. The architecture of static and dynamic model is depicted in Figure 4.

4. Experiment

4.1. Implementation details

Our model uses ResNet34 [43] as the encoder to extract feature maps. There are some differences between the static and dynamic models. For the static model, we use ResNet-based MLPs block, while for the dynamic model, we add four additional layers of MLPs to predict scene flow and mixed values.

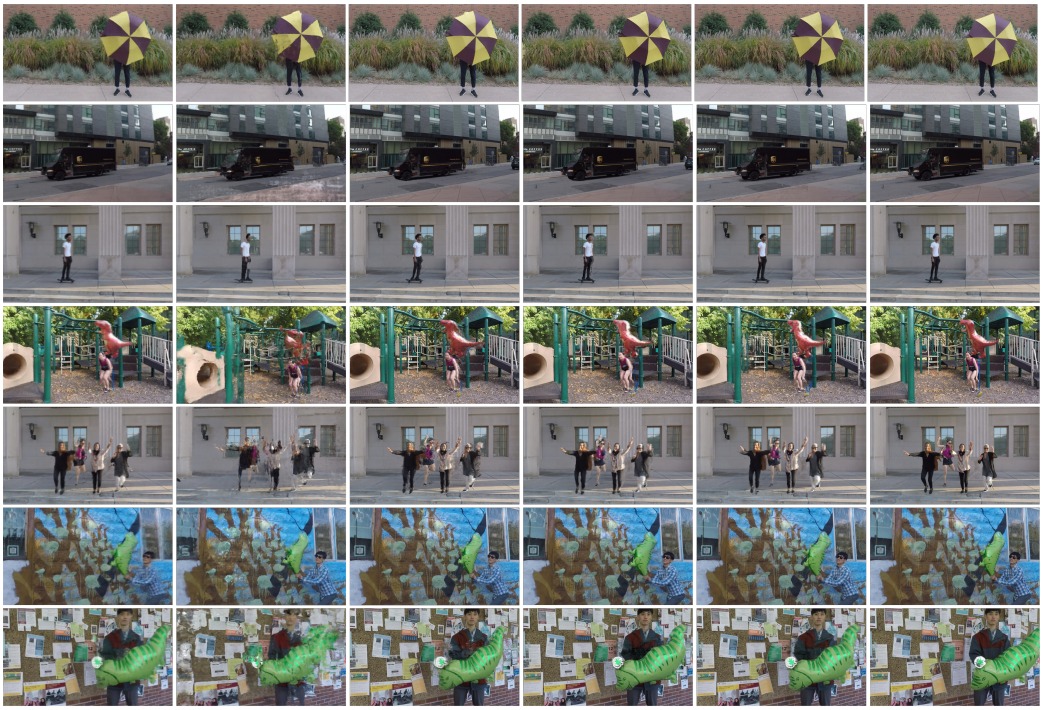

Figure 5: Novel view synthetic qualitative results on Nvidia Dynamic Scene Dataset [44]. In contrast to other NeRF-based approaches, our outcomes exhibit enhanced clarity, capturing finer details that closely approximate ground truth, particularly in dynamic regions.

We found that the ResNet-based MLP block is difficult to train for accurate scene flow. More details about our model can be found in the Figure 4. We first train the static model for 300K steps and then fix it to train the dynamic model for 200K steps. We use frames $t-1, t$, and $t+1$ as input to extract feature vectors. Note that we only choose $t-1, t$ and $t, t+1$ when selecting the first and last frames as input.

4.2. Model Parameters and Inference Time

We present our model's Parameters, inference speed, and training time. The amount of our speed parameters has increased compared to our baseline, but in more complex scenes, our baseline cannot correctly obtain dynamic scenes (As shown in Figure 6).

Method	Parameters	Inference speed	Training Time
DynamicNeRF	4.59M	8 s/Frame	about 21 hours
Our	12.21M	10 s/Frame	about 24 hours

Table 1: Parameters, inference speed, and training time.

4.3. Dataset

Our method is assessed on the Nvidia Dynamic Scene Dataset [44], DAVIS Dataset [45], and iPhone dataset [31].

Nvidia Dynamic Scene Dataset comprises nine video sequences captured using a static camera rig of 12 cameras. All cameras capture images simultaneously at 12 different time steps $\{t_0, t_1, \ldots, t_{11}\}$, and we obtain a twelve-frame monocular video $\{I_0, I_1, \ldots, I_{11}\}$ by sampling the image taken by the $i-th$ camera at time t_i. It is worth mentioning that we use a different camera for each video frame to simulate camera motion. The video frames consist of a background that remains stationary throughout the video and a dynamic object that changes over time. We adopt COLMAP [46] similar to NeRF [23], to estimate the camera poses, near and far boundaries of the scene, and assume that all cameras share the same intrinsic parameter. We exclude the DynamicFace sequence from our evaluation since COLMAP fails to estimate camera poses for this sequence. Lastly, we resize all video sequences to a resolution of 480×270. The DAVIS Dataset [45] consists of fifty sequences featuring dynamic moving objects, like animals and cars. However, due to limitations in camera movement, COLMAP could only estimate camera poses

for six out of the fifty sequences, all of which include ground truth object masks. Finally, for the iPhone Dataset [31], we conducted a quantitative evaluation using seven dynamic scenes, each accompanied by ground truth images of novel views.

4.4. Comparison with State-of-the-Art Methods

4.4.1. Qualitative Results

We present some visual comparisons on the Nvidia Dynamic Scene Dataset [44] in Figure 5 and the DAVIS dataset [45] in Figure 6. The camera poses of most sequences in the DAVIS dataset cannot be estimated by COLMAP. By aggregating the features of adjacent frames, our method generates frames with fewer visual artifacts and obtains results that are closer to ground truth. In contrast, our method exploits feature associations across frames, which yields better visual quality results.

Figure 6: Novel view synthetic qualitative results on DAVIS Dataset [45]. Compared to our baseline, our method obtains sharper results and fewer artifacts.

4.4.2. Quantitative Results

Table 2 presents the quantitative results obtained from the Nvidia Dynamic Scene Dataset [44]. We adopted the evaluation methodology from DynamicNeRF [11] to synthesize views using the first camera while varying

Table 2: Novel view synthesis quantitative results on Nvidia Dynamic Scene Dataset [44]. We report average PSNR and LPIPS [47] results by comparison with existing methods. The best performance is bold and the next best is underline.

PSNR↑ /LPIPS↓	Jumping	Skating	Truck	Umbrella	Balloon1	Balloon2	Playground	Average
NeRF [23]	20.58 / 0.305	23.05 / 0.316	22.61 / 0.225	21.08 / 0.441	19.07 / 0.214	24.08 / 0.098	20.86 / 0.164	21.62 / 0.252
NeRF [23] + time	16.72 / 0.489	19.23 / 0.542	17.17 / 0.403	17.17 / 0.752	17.33 / 0.304	19.67 / 0.236	13.80 / 0.444	17.30 / 0.453
D-NeRF [30]	22.36 / 0.193	22.48 / 0.323	24.10 / 0.145	21.47 / 0.264	19.06 / 0.259	20.76 / 0.277	20.18 / 0.164	21.48 / 0.232
HyperNeRF [12]	18.34 / 0.302	21.97 / 0.183	20.61 / 0.205	18.59 / 0.443	13.96 / 0.530	16.57 / 0.411	13.17 / 0.495	17.60 / 0.367
TiNeuVox [48]	20.81 / 0.247	23.32 / 0.152	23.86 / 0.173	20.00 / 0.355	17.30 / 0.353	19.06 / 0.279	13.84 / 0.437	19.74 / 0.285
Yoon et al. [44]	20.16 / 0.148	21.75 / 0.135	23.93 / 0.109	20.35 / 0.179	18.76 / 0.178	19.89 / 0.138	15.09 / 0.183	19.99 / 0.153
Tretschk et al. [29]	19.38 / 0.295	23.29 / 0.234	19.02 / 0.453	19.26 / 0.427	16.98 / 0.353	22.23 / 0.212	14.24 / 0.336	19.20 / 0.330
NSFF [13]	24.12 / 0.146	28.91 / 0.135	25.94 / 0.171	22.58 / 0.302	21.40 / 0.225	24.09 / 0.228	20.91 / 0.220	23.99 / 0.205
RoDynRF [49]	24.27 / 0.100	28.71 / 0.046	**28.85** / **0.066**	23.25 / **0.104**	21.81 / 0.122	25.58 / 0.064	**25.20** / **0.052**	25.38 / **0.079**
DynamicNeRF [11]	24.61 / 0.144	28.90 / 0.124	25.78 / 0.134	23.15 / 0.146	21.47 / 0.125	25.97 / 0.059	23.65 / 0.093	24.74 / 0.118
Our	**24.35** / **0.094**	**33.51** / **0.034**	28.27 / 0.084	**23.48** / 0.129	**22.19** / **0.111**	**26.86** / **0.048**	24.28 / 0.077	**26.17** / 0.082

Table 3: Assessing novel view synthesis outcomes, we measure performance using the mPSNR and mSSIM metrics, benchmarked against established methods. The evaluation is conducted on the iPhone dataset [31].

Method	Apple	Block	Paper-windmill	Space-out	Spin	Teddy	Wheel	Average
NSFF [13]	17.54 / 0.750	16.61 / 0.639	17.34 / 0.378	17.79 / 0.622	18.38 / 0.585	13.65 / 0.557	13.82 / 0.458	15.46 / 0.569
Nerfies [16]	17.64 / 0.743	17.54 / 0.670	17.38 / 0.382	17.93 / 0.605	19.20 / 0.561	13.97 / 0.568	13.99 / 0.455	16.45 / 0.569
HyperNeRF [12]	16.47 / 0.754	14.71 / 0.606	14.94 / 0.272	17.65 / 0.636	17.26 / 0.540	12.59 / 0.537	14.59 / 0.511	16.81 / 0.550
T-NeRF [31]	17.43 / 0.728	17.52 / 0.669	17.55 / 0.367	17.71 / 0.591	19.16 / 0.567	13.71 / 0.570	15.65 / 0.548	16.96 / 0.577
RoDynRF [49]	18.73 / 0.722	18.73 / 0.634	16.71 / 0.321	18.56 / 0.594	17.41 / 0.484	14.33 / 0.536	15.20 / 0.449	17.09 / 0.534
Our	19.53 / 0.691	19.74 / 0.626	17.66 / 0.346	18.11 / 0.601	19.79 / 0.516	14.51 / 0.509	14.48 / 0.430	17.69 / 0.531

the time on the Nvidia Dynamic Scene Dataset. To evaluate the rendering quality of each method, we employed two widely recognized error metrics: peak signal-to-noise ratio (PSNR) and perceptual similarity (LPIPS) as defined by [47]. Additionally, due to minor differences observed in our ablation study's results, we incorporated the structural similarity index (SSIM) for a more thorough assessment.

Our method demonstrates significant advancements, outperforming existing state-of-the-art techniques in five of the seven tested scenarios. This

Table 4: Evaluation of the whole module, RBCT module, and temporal module on the Nvidia Dynamic Scene Dataset [44] (Balloon 2 scene).

	PSNR↑	SSIM↑	LPIPS↓
A) w/o MV	26.01	0.8330	0.061
B) w/o Four-layer	25.32	0.8205	0.070
C) w/o GSTF	26.73	0.8537	0.051
D) w/o RBCT	26.40	0.8434	0.057
E) w/o $\mathcal{L}_{small}$	26.86	0.8597	0.048
Full	**26.86**	**0.8602**	**0.048**

(a) Whole Module

	PSNR↑	SSIM↑	LPIPS↓
A) w/o CTT	26.66	0.8491	0.060
B) w/o RT	26.16	0.8360	0.061
C) w/o GRSPE	26.30	0.8410	0.056
D) RT to CTT	26.69	0.8564	0.048
Full	**26.86**	**0.8602**	**0.048**

(b) RBCT Module

	PSNR↑	SSIM↑	LPIPS↓
A) w/o CTT	26.66	0.8491	0.060
B) w/o GSTF	26.76	**0.8625**	0.050
Full	**26.86**	0.8602	**0.048**

(c) Temporal Module

improvement is particularly evident in the average PSNR increase of 1dB and a notable 20% reduction in LPIPS error, underscoring a substantial enhancement in perceptual quality compared to real images. Moreover, we extended our evaluation following DyCheck's methodology [31] for the iPhone dataset, detailed in Table Table 3, where we report masked PSNR and SSIM scores. Given that a significant number of scenes within this dataset are long sequences, and considering our method's limitations in effectively modeling such sequences, notable improvements are limited. Nevertheless, our method demonstrates comparable performance to established methods and even exhibits slight enhancements in select scenarios. These outcomes serve as a compelling demonstration of the superior effectiveness of our framework in restoring intricate scene content.

4.5. Ablation Study

To validate the effectiveness of our proposed system components, we conduct an ablation study on the Dynamic Scene Dataset [44].

4.5.1. Evaluate the whole module

Figure 7: In the absence of global ray sampling point coordinate embedding, the synthesized view displays stripes.

In Table 4, we present a detailed comparison between our complete system and its variants, each lacking a specific module: A) multi-view aggregation, B) an additional four-layer MLP, C) Global Spatio-Temporal Filter module, D) Ray-Based Cross-Time Aggregation Module, and E) regularization scene flow loss. As indicated in the first two rows of Table 4, the absence of multi-view aggregation and the additional four-layer MLP markedly diminish the quality of view synthesis, with a decrease in PSNR by 0.75% and 2.61% respectively, and SSIM by 1.29% and 2.81% respectively. Additionally, LPIPS scores increased

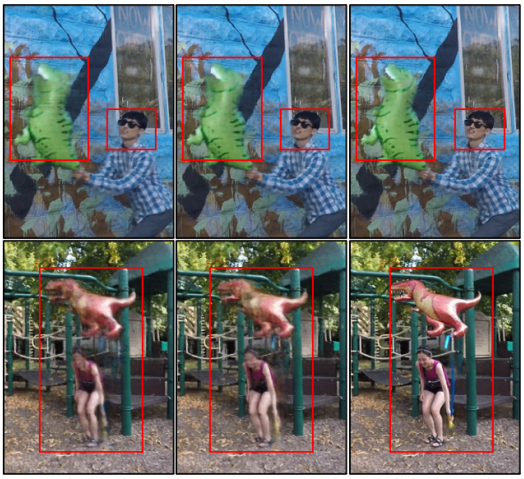

Figure 8: The qualitative results of the ablation experiments on the Nvidia Dynamic Scene Dataset [44] (Balloon 1 and Balloon 2 scene). From right to left, showcasing unused GSTF module, unused RBCT module, and the complete model.

by 25.00% and 27.08%, signaling a noticeable degradation in image quality and accuracy. These two components, therefore, are critical in enhancing the fidelity and precision of the synthesized views. We observe that removing the Global Spatio-Temporal Filter module and the Ray-Based Cross-Time Aggregation Module also impacts the system's performance, though to a slightly lesser extent compared to the first two components, with PSNR decreasing by 2.08% and 0.63%, and SSIM by 2.23% and 0.44% respectively. Additionally, the removal of the regularization scene flow loss demonstrates a relatively minor impact on the quality of view synthesis, with a less pronounced decrease in performance metrics compared to the other modules. This suggests that while this component aids in fine-tuning the system, its absence does not drastically compromise the overall effectiveness.

4.5.2. Evaluate the RBCT module

Furthermore, we investigate the impact of the internal structure of the RBCT module on the model-view synthesis performance, as summarized in Table 4. Specifically, we examine the effects of the following variations: A) without using Cross-time Transformer, B) without using Ray Transformer, C) without using global ray sampling point coordinate embedding, and D) using Ray Transformer first, followed by Cross-time Transformer. Excluding the Cross-time Transformer led to a moderate decline in synthesis quality, as indicated by a 0.74% decrease in PSNR and a 1.29% drop in SSIM, coupled with a notable 25% increase in LPIPS. The omission of the Ray Transformer had a more pronounced impact on performance, with a 2.61% reduction in PSNR, a 2.82% decrease in SSIM, and a 27.08% rise in LPIPS. This highlights the Ray Transformer's critical role in maintaining high-quality synthesis.

Furthermore, removing the global ray sampling point coordinate embedding also negatively affected the results, leading to a 2.08% reduction in PSNR, a 2.23% decrease in SSIM, and a 16.67% increase in LPIPS. Sequentially applying the Ray Transformer and the Cross-time Transformer slightly improved some metrics compared to using the full module configuration, with a 0.11% increase in PSNR and a 0.53% rise in SSIM, while maintaining stable LPIPS. Therefore, our experiments demonstrated that the absence of key components, especially the Ray Transformer and the global ray sampling point coordinate embedding, significantly compromises view synthesis quality. In Figure 7, the absence of global ray sampling point coordinate embedding resulted in stripes in the dynamic synthesis region. Although the other two experiments have a minimal impact on the model, the differences are discernible.

4.5.3. Evaluate the temporal module

To demonstrate the effectiveness of our proposed frequency-domain timing module, as summarized in Table 4. Specifically, A) without using Cross-time Transformer, B) Global Spatio-Temporal Filter module. The model without CTT shows a 1.11% improvement in PSNR, a 0.41% improvement in SSIM, and a 33.33% reduction in LPIPS compared to the full model. Conversely, without GSTF, although the PSNR improves by 0.1%, SSIM increases by 2.34%, and LPIPS decreases by 16.67%. In the Full

	PSNR↑	SSIM↑	LPIPS↓
w/ GSTF	**21.36**	**0.6597**	**0.260**
w/o GSTF	20.76	0.6129	0.350

Table 5: Evaluation of GSTF module for Dynamic region on the Nvidia Dynamic Scene Dataset [44] (Balloon 2 scene).

model, we observe more substantial improvements. Compared to A), the Full model shows a 1.2% increase in PSNR, a 1.11% improvement in SSIM, and a 20.00% reduction in LPIPS. In comparison to B), the Full model exhibits a 0.1% increase in PSNR, a 0.23% decrease in SSIM, and a 4.00% reduction in LPIPS. This indicates that simultaneous utilization of CTT and GSTF significantly enhances the quality of the novel view, with a more pronounced improvement in perceptual quality. From the perspective of the error metric, there may not be a significant disparity between the two approaches, but utilizing them simultaneously can be complementary and enhance quality of novel view.

4.5.4. Evaluate the GSTF module

In Figure 9, we present a visual comparison between baseline features extracted from Resnet34 and the features refined by our GSTF module. Our GSTF is designed with the specific goal of capturing detailed contours and high-frequency texture information, ensuring the preservation of sharp textures in the reconstructed view. A quantitative evaluation in Table 5 further underscores the impact of the GSTF module, particularly in the context of dynamic scenes. The results reveal a

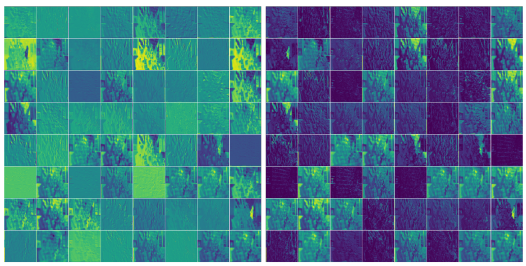

Figure 9: The left is the vanilla feature maps extracted from Resnet34 [43]. In contrast, the right displays a feature map refined using our GSTF module. This comparison clearly demonstrates that our GSTF module enhances the extraction of high-frequency details, such as texture and contour, while effectively filtering out low-frequency information.

substantial 2.1% increase in PSNR,
a noteworthy 4.68% improvement in SSIM, and a significant 25.38% reduction in LPIPS when utilizing the GSTF module (*w/ GSTF*) compared to the configuration without GSTF (*w/o GSTF*). In Table 4, the GSTF module makes a marginal contribution to the overall improvement. This is primarily because our GSTF is applied to dynamic areas, which represent just one-fifth of the total in the Balloon 2 evaluation scene.

4.5.5. Ablation study qualitative results

The Figure 8 illustrates our primary contributions, the Global Spatio-Temporal Filter (GSTF) and Ray-Based Cross-Time (RBCT) modules. These modules play pivotal roles in enhancing the quality of synthesized views. In the absence of the RBCT module, the resulting synthesized view lacks intricate surface details and may exhibit noticeable artifacts. Conversely, in the absence of the GSTF module, the synthesized view experiences a loss of edge information, resulting in a perceptible blurring effect.

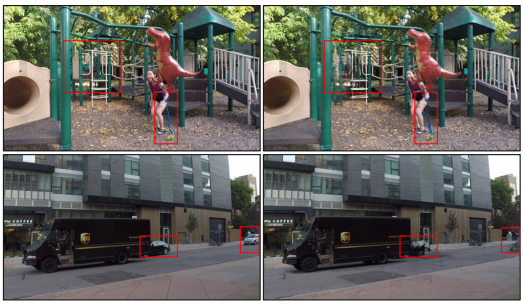

Figure 10: Limitations of our model, take the Nvidia Dynamic Scene Dataset [44] (Balloon 1 and truck scene) as an example.

4.6. Limitations

As shown in Figure 10, compare with the ground truth, we can observe that in the playground scene, the rendering of the railings and balloon ropes

appears to be blurred. This is because we are using multi-frame aggregation of feature vectors, which enables us to aggregate more information but also results in some details that may be discarded or merged into other features, making it impossible to fully express the original data. Moreover, due to the small non-rigid deformations of these parts, our method cannot handle them well, resulting in blurring when rendering new views. In the truck scene, we only input the adjacent 2 frames of the current frame. Therefore, when the time is changed for rendering after the $1st$ frame with a fixed sequence number, the synthesis effect of the long sequence of unseen frames is not optimal. For example, the synthesis result of the $11th$ frame in the figure shows that the hidden car behind the truck is very blurred due to the lack of feature vector information provided by adjacent frames.

5. Conclusion

In this work, we aim to introduce a novel dynamic neural render field framework for dynamic monocular videos, which enables high-quality rendering of novel views. To achieve this goal, we extend recent ideas in multi-view aggregation to time-varying NeRF, enabling the modeling of complex motion. Specifically, we introduce RBCT and GSTF modules to model motion from the time domain and frequency domain, respectively. Our experimental results show that these proposed modules significantly improve the performance of time-varying NeRF with multi-view aggregation when rendering new views. While our work represents a promising exploration of time-varying NeRF for multi-view aggregation, there are still some limitations. It is worth noting that our current method may not perform well when

rendering novel views of long sequences of videos. One potential solution to improve performance is to increase the length of the aggregate view, but this approach requires significant computing resources. Fortunately, recent developments such as TensoRF and 3D Gaussian splatting offer potential solutions to these challenges.

Acknowledgments

This work is supported by the UK Medical Research Council (MRC) Innovation Fellowship under Grant MR/S003916/2, International Exchanges 2022 IEC\NSFC\223523 and Securing the Energy/Transport Interface EP/X037401/1.

References

[1] G. Miller, A. Hilton, J. Starck, Interactive free-viewpoint video, in: IEEE European Conf. on Visual Media Production, 2005, pp. 50–59.

[2] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, S. Sullivan, High-quality streamable free-viewpoint video, ACM Transactions on Graphics (ToG) 34 (2015) 1–13.

[3] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, T. Wiegand, 3d video and free viewpoint video-technologies, applications and mpeg standards, in: 2006 IEEE International Conference on Multimedia and Expo, IEEE, 2006, pp. 2161–2164.

[4] J. Carranza, C. Theobalt, M. A. Magnor, H.-P. Seidel, Free-viewpoint video of human actors, ACM transactions on graphics (TOG) 22 (2003) 569–577.

[5] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, R. Szeliski, High-quality video view interpolation using a layered representation, ACM transactions on graphics (TOG) 23 (2004) 600–608.

[6] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, et al., Holoportation: Virtual 3d teleportation in real-time, in: Proceedings of the 29th annual symposium on user interface software and technology, 2016, pp. 741–754.

[7] M. Broxton, J. Flynn, R. Overbeck, D. Erickson, P. Hedman, M. Duvall, J. Dourgarian, J. Busch, M. Whalen, P. Debevec, Immersive light field video with a layered mesh representation, ACM Transactions on Graphics (TOG) 39 (2020) 86–1.

[8] B. Mildenhall, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, J. T. Barron, Nerf in the dark: High dynamic range view synthesis from noisy raw images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16190–16199.

[9] W. Xian, J.-B. Huang, J. Kopf, C. Kim, Space-time neural irradiance fields for free-viewpoint video, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9421–9431.

[10] Y. Du, Y. Zhang, H.-X. Yu, J. B. Tenenbaum, J. Wu, Neural radiance flow for 4d view synthesis and video processing, in: 2021 IEEE/CVF

International Conference on Computer Vision (ICCV), IEEE Computer Society, 2021, pp. 14304–14314.

[11] C. Gao, A. Saraf, J. Kopf, J.-B. Huang, Dynamic view synthesis from dynamic monocular video, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5712–5721.

[12] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, S. M. Seitz, Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields, arXiv preprint arXiv:2106.13228 (2021).

[13] Z. Li, S. Niklaus, N. Snavely, 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, pp. 6498–6508.

[14] Z. Li, Q. Wang, F. Cole, R. Tucker, N. Snavely, Dynibar: Neural dynamic image-based rendering, arXiv preprint arXiv:2211.11082 (2022).

[15] X. Miao, Y. Bai, H. Duan, Y. Huang, F. Wan, X. Xu, Y. Long, Y. Zheng, Ds-depth: Dynamic and static depth estimation via a fusion cost volume, IEEE Transactions on Circuits and Systems for Video Technology (2023).

[16] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, R. Martin-Brualla, Nerfies: Deformable neural radiance fields, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5865–5874.

[17] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, T. Funkhouser, Ibrnet: Learning multi-view image-based rendering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4690–4699.

[18] P. Wang, X. Chen, T. Chen, S. Venugopalan, Z. Wang, et al., Is attention all nerf needs?, arXiv preprint arXiv:2207.13298 (2022).

[19] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, H. Su, Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 14124–14133.

[20] Y. Liu, S. Peng, L. Liu, Q. Wang, P. Wang, C. Theobalt, X. Zhou, W. Wang, Neural rays for occlusion-aware image-based rendering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 7824–7833.

[21] Z. Yang, H. Zhang, Y. Wei, Z. Wang, F. Nie, D. Hu, Geometric-inspired graph-based incomplete multi-view clustering, Pattern Recognition 147 (2024) 110082.

[22] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, C. Theobalt, Neural sparse voxel fields, Advances in Neural Information Processing Systems 33 (2020) 15651–15663.

[23] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoor-

thi, R. Ng, Nerf: Representing scenes as neural radiance fields for view synthesis, Communications of the ACM 65 (2021) 99–106.

[24] Y. Xiangli, L. Xu, X. Pan, N. Zhao, A. Rao, C. Theobalt, B. Dai, D. Lin, Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII, Springer, 2022, pp. 106–122.

[25] Q. Xu, Z. Xu, J. Philip, S. Bi, Z. Shu, K. Sunkavalli, U. Neumann, Point-nerf: Point-based neural radiance fields, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5438–5448.

[26] A. Yu, V. Ye, M. Tancik, A. Kanazawa, pixelnerf: Neural radiance fields from one or few images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4578–4587.

[27] S. Li, Z. Xia, Q. Zhao, Representing boundary-ambiguous scene online with scale-encoded cascaded grids and radiance field deblurring, IEEE Transactions on Circuits and Systems for Video Technology (2023) 1–1. doi:10.1109/TCSVT.2023.3300170.

[28] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5855–5864.

[29] E. Tretschk, A. Tewari, V. Golyanik, M. Zollhöfer, C. Lassner, C. Theobalt, Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 12959–12970.

[30] A. Pumarola, E. Corona, G. Pons-Moll, 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, pp. 10318–10327.

[31] H. Gao, R. Li, S. Tulsiani, B. Russell, A. Kanazawa, Monocular dynamic view synthesis: A reality check, in: Advances in Neural Information Processing Systems, 2022.

[32] C. Wang, B. Eckart, S. Lucey, O. Gallo, Neural trajectory fields for dynamic novel view synthesis, arXiv preprint arXiv:2105.05994 (2021).

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[34] H. Duan, Y. Long, S. Wang, H. Zhang, C. G. Willcocks, L. Shao, Dynamic unary convolution in transformers, IEEE Transactions on Pattern Analysis and Machine Intelligence (2023).

[35] Z. Wu, N. Ma, C. Wang, C. Xu, G. Xu, M. Li, Spatial–temporal hypergraph based on dual-stage attention network for multi-view data lightweight action recognition, Pattern Recognition 151 (2024) 110427.

[36] J. Cao, L. Yu, B. W.-K. Ling, Z. Yao, Q. Dai, Mhsan: Multi-view hierarchical self-attention network for 3d shape recognition, Pattern Recognition (2024) 110315.

[37] Y. Yao, Z. Luo, S. Li, T. Fang, L. Quan, Mvsnet: Depth inference for unstructured multi-view stereo, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 767–783.

[38] H. Zhou, C. Tian, Z. Zhang, C. Li, Y. Xie, Z. Li, Frequency-aware feature aggregation network with dual-task consistency for rgb-t salient object detection, Pattern Recognition 146 (2024) 110043.

[39] Y. Rao, W. Zhao, Z. Zhu, J. Lu, J. Zhou, Global filter networks for image classification, Advances in neural information processing systems 34 (2021) 980–993.

[40] J. W. Cooley, J. W. Tukey, An algorithm for the machine calculation of complex fourier series, Mathematics of computation 19 (1965) 297–301.

[41] A. V. Oppenheim, A. S. Willsky, S. H. Nawab, J.-J. Ding, Signals and systems, volume 2, Prentice hall Upper Saddle River, NJ, 1997.

[42] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).

[43] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[44] J. S. Yoon, K. Kim, O. Gallo, H. S. Park, J. Kautz, Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5336–5345.

[45] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, A. Sorkine-Hornung, A benchmark dataset and evaluation methodology for video object segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 724–732.

[46] J. L. Schonberger, J.-M. Frahm, Structure-from-motion revisited, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4104–4113.

[47] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, 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, pp. 586–595.

[48] J. Fang, T. Yi, X. Wang, L. Xie, X. Zhang, W. Liu, M. Nießner, Q. Tian, Fast dynamic radiance fields with time-aware neural voxels, in: SIGGRAPH Asia 2022 Conference Papers, 2022, pp. 1–9.

[49] Y.-L. Liu, C. Gao, A. Meuleman, H.-Y. Tseng, A. Saraf, C. Kim, Y.-Y. Chuang, J. Kopf, J.-B. Huang, Robust dynamic radiance fields, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 13–23.