CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

ORCA – Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository:https://orca.cardiff.ac.uk/id/eprint/167438/

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Jing, Xinyi, Yu, Tao, He, Renyuan, Lai, Yukun and Li, Kun 2024. FRNeRF: Fusion and regularization fields for dynamic view synthesis. Computational Visual Media

Publishers page:

Please note:
Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See http://orca.cf.ac.uk/policies.html for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.

Computational Visual Media
https://doi.org/10.1007/s41095-0xx-xxxx-x

Research Article

FRNeRF: Fusion and Regularization Fields for Dynamic View Synthesis

Xinyi Jing[1*], Tao Yu[1*], Renyuan He[1], Yu-Kun Lai[2], and Kun Li[1](✉)

© The Author(s)

Abstract Novel space-time view synthesis for monocular video is a highly challenging task: both static and dynamic objects usually appear in the video, but only a single view of the current scene is available, resulting in inaccurate synthesis results. To address this challenge, we propose *FRNeRF*, a novel space-time view synthesis method with a fusion regularization field. Specifically, we design a 2D-3D fusion regularization field for the original dynamic neural field, which helps reduce blurring of dynamic objects in the scene. In addition, we add image prior features to the hierarchical sampling to solve the problem that the traditional hierarchical sampling strategy cannot obtain sufficient sampling points during training. We evaluate our method extensively on multiple datasets and show the results of dynamic space-time view synthesis. Our method achieves state-of-the-art performance both qualitatively and quantitatively. Code is available for research purposes at https://cic.tju.edu.cn/faculty/likun/projects/FRNerf.

Keywords neural radiance fields, space-time view synthesis, dynamic scene reconstruction, flow fields

1 Introduction

The environment we live in is a three-dimensional space, and images captured from the environment have various viewpoints. With monocular videos as input, novel space-time view synthesis aims to generate novel view images of dynamic scenes. Novel view synthesis has many applications in real life, such as achieving space-time interpolation in virtual game scenes, replaying the actions of athletes from novel viewpoints for professional sport events, and creating cinematic effects.

Fig. 1 Novel view synthesis for dynamic monocular video. Our method takes monocular video frames as input. Each frame in the video is taken from a different viewpoint at a unique time step. Existing space-time view synthesis methods such as NSFF struggle to render high-quality views from monocular videos with highly dynamic motion. Our method produces results with higher clarity.

Systems for novel view synthesis need to overcome the challenging problems associated with video capture, reconstruction, compression and rendering. Most existing methods use expensive and laborious setups, *e.g.* multi-view camera systems [1], fast-moving cameras [2], or other specialized hardware to capture and observe the scenes [3, 4]. However, such approaches are complicated in real life applications. Therefore, it is more practical to generate dynamic scenes from a monocular video captured by a single RGB camera. Few methods are able to achieve novel view synthesis from a single stereo camera or even monocular RGB camera, and they are further constrained to specific fields such as human reconstruction [5, 6]. Some methods [7–9] represent dynamic scenes as continuous neural radiance fields of space and time and generate reflectivity, density, and 3D scene motion information with multi-layer perceptrons (MLPs). Unlike static neural radiance fields (NeRFs) [10], a scene flow establishes tight relationships for frame sequences. NSFF [7] strengthens the consistency between viewpoints and the 3D scene flow and introduces a variety of prior knowledge, so it can generate more coherent novel view images. At the same time,

1 College of Intelligence and Computing, Tianjin University, Tianjin 300350, China. E-mail: Xinyi Jing, jingxinyi@tju.edu.cn; Tao Yu, xiaoyudan@tju.edu.cn; Renyuan He, renyuanhry@tju.edu.cn; Kun Li, lik@tju.edu.cn(✉).
2 School of Computer Science and Informatics, Cardiff University, Cardiff CF24 4AG, U.K. E-mail: Yu-Kun Lai, LaiY4@cardiff.ac.uk.
3 Corresponding Author: Kun Li.
∗ Contributed equally.

NSFF [7] creates separate neural radiation fields for static and dynamic regions to improve the synthesis quality of the network. However, the most significant difficulty is to reduce the artifacts caused by fast motions of multi-frame images in novel viewpoint synthesis. NSFF [7] estimates the scene flow field between corresponding 3D points in a multi-frame scene and calculates the pixel colors in adjacent frames through the scene flow field. The depth in the 3D scene flow field is essentially generated by a depth estimation network, which differs from the ground truth depth information, so is inaccurate. In addition, directly calculating a loss using color differences between adjacent frames has the problem of pixel misalignment. As a result, fast-moving objects in novel view images suffer from noticeable artifacts.

To address the problems above, in this paper, we propose *FRNeRF* for dynamic space-time view synthesis. It can generate novel view images from monocular videos with greater clarity and realism. We propose a 2D-3D fusion regularization field, which can fuse the 2D feature field with the 3D scene flow field to enhance it. Specifically, we first introduce a 2D feature field in the dynamic NeRF to simulate the real spatial offset of pixels between adjacent frames. Then we extract the high-level semantic features of the original 3D scene flow field predicted by dynamic NeRF to re-match and correct the misaligned dynamic pixels due to inaccurate depth information. The fusion regularization process of the flow field in the scene can significantly reduce the artifacts caused by the fast motion of dynamic objects, and generate a more realistic novel view image.

Dynamic space-time view synthesis is a challenging problem. Unlike static novel view synthesis, in which the input is an intensive multi-view observation, a novel view of the captured scene can be synthesized simply by a hierarchical sampling strategy. In the dynamic case, novel view synthesis requires more information about the scene as the dynamic scene changes over time. However, the sparse viewpoints cannot adequately capture the dynamic pixels in the scene, and a simple hierarchical sampling strategy cannot provide sufficient sampling points, affecting the quality of novel view synthesis.

We propose two improvements to overcome this challenge. The first improvement is to append the 2D image features extracted by the feature extractor to the input of *FRNeRF*, which provides more feature information to the implicit neural representation and can improve inference on unseen pixels, and we add a local convolution module to the pre-trained feature extractor. The local modeling property of the convolution module can utilize the two-dimensional neighborhood information during each iteration. Additional 2D priors can increase the pixel features needed for rendering. In order to further improve the quality of novel view synthesis on a global scale, we propose as a second improvement to add a global pixel alignment loss between the estimated view and the input view to enhance the global rendering perception quality, which diminishes the spatial ambiguity due to the high-speed movement of pixels. As shown in Figure 1, our method significantly improves the rendering fidelity of dynamic space-time view synthesis.

Our main contributions can be summarized as:

- a joint 2D-3D fusion regularization field, which contains both 2D feature field and 3D scene flow field, and
- image-prior-based 2D feature addition and semantic constraints, achieving local interactivity and global consistency for each pixel in the scene, leading to
- generation of results superior to previous state-of-the-art dynamic space-time view synthesis methods.

2 Related work

2.1 Implicit neural representations

Continuous and differentiable functions parameterized by fully-connected networks have been successfully applied as compact implicit representations for modeling 3D scenes [10–13], object appearances [14, 15] and 3D shapes [16–22]. These methods train MLPs to regress input coordinates, *e.g.* points in 3D space, to the desired quantities, for example, volume densities [10], colors [10, 11, 15, 23], signed distances [19, 24, 25], or occupancy values [12, 23, 26]. Recently several works have shown training MLPs with 2D images under multi-view without directly using 3D supervision [10, 14, 27], leveraging differentiable rendering [28, 29].

Most existing methods deal with static scenes. Due to motion entanglement and the complexity of 3D shapes, directly extending MLPs to encode additional temporal dimensions is ineffective. The method in [30] extends NeRF [10] to process diverse photographs containing lighting changes and transparent objects. The most relevant work for our method is [7], which learns continuous motion fields over space and time. Our method follows it, but the focus of our task is to resolve the previously ambiguous fast motion for dynamic space-time novel view synthesis. Unlike [7], we additionally extract features from 2D images as supplementary information to guide the model for dynamic scene learning.

2.2 Novel view synthesis for static scenes

Synthesizing novel views for static scenes is a long standing vision and graphics problem that aims to synthesize new

images from arbitrary viewpoints of a scene captured by multiple cameras. Different methods represent the underlying geometry using different representations. Mesh-based approaches [31–36] represent scenes using compact and easily renderable surfaces, while optimizing a mesh for complex scenes remains challenging. Volume-based works, *e.g.* multi-plane images (MPIs) [17, 37–41] and voxel-meshes [42–46] are more suitable for modeling those complex and translucent scenes which are smooth and fluid. In particular, the realistic rendering quality of NeRF [10] has led to an explosion of developments in the field. Progress has been made in training speed [16, 47–50], improving the rendering quality [51, 52], accelerating rendering, and adapting to more general scenes [30, 53–55], *etc*.

2.3 Novel view synthesis for dynamic scenes

Synthesizing novel views for dynamic scenes is a more practical and challenging problem. However, existing methods do not perform well with dynamic scenes. Discrepancies between the actual capture process and the existing experimental protocols for monocular videos have been shown in [56].

Most methods are limited to certain scenarios, *e.g.* constrained motions or human models. For complex scenes in the real world, reconstruction from synchronized multi-view videos is more promising due to the intensive supervision of each viewpoint and point in time. Earlier works [57, 58] explore this issue and show the possibility of rendering novel videos from a set of input views. The Neural Volumes approach [44] uses volumetric representations. It employs an encoder-decoder network to convert the input images into 3D volumes and decode the latent representation by differentiable ray marching operations. [59] proposes a data-driven strategy for 4D space-time visualization of dynamic scenes. They split the static and dynamic components and convert the intermediate representations into images using spatial U-Net structures. More recently, Li *et al.* [60] used a time-aware neural radiance field to address the problem, and proposed several new sampling strategies to train the model efficiently. They presented a more complex real-world dataset and validate the improvements of their method compared to the previous methods. To accelerate the reconstruction of dynamic scenes, the Fourier Plenoctree approach [61] proposes to model dynamic components in the frequency domain, and generates a Plenoctree by multi-view blending to accelerate the rendering. The authors focus on the foreground moving components extracted through chroma key segmentation, which requires that the background should be a solid color.

With advances in rendering, view synthesis methods have shown state-of-the-art results from a monocular video depicting a dynamic scene. These methods can be divided into explicit modeling [62–65] and implicit modeling of deformations [7, 66–69]. Despite the improvements achieved, it is still difficult to reconstruct complex dynamic scenes only using monocular videos. In particular, DynamicNeRF [67] decomposes a dynamic scene into static, deforming components and jointly trains a time-invariant static NeRF and a time-variant dynamic NeRF, and learns how to blend the results in an unsupervised manner. However, this approach is unsuitable for fast motions and often leads to incorrect flows. NSFF [7] proposes a new representation that models a dynamic scene as a time-variant continuous function of appearance, geometry, and 3D scene motion. However, this method only works well for short (1–2 s) videos without fast or drastic motions.

Differing from these works, we introduce fusion regularization fields to eliminate the influence of the inaccurate 3D flow fields, which can further enhance the correlation between adjacent video frames. We also propose a 2D image feature extractor to achieve local interactivity and global consistency for each pixel in the dynamic scenes.

3 Method

3.1 Background: static scene rendering

NeRFs represent a continuous static scene as a function with an input of 5D vectors, including the 3D coordinate position $o = (x, y, z)$ of a space point, and the viewpoint direction $d = (\theta, \Phi)$. In NeRF, F_Θ represents an MLP network that models the volume density σ and color $c = (r, g, b)$ corresponding to each position and view direction in the space, forming an implicit representation of the 3D scene:

$$F_\Theta : (\boldsymbol{o}, \boldsymbol{d}) \rightarrow (\boldsymbol{c}, \sigma), \tag{1}$$

The RGB value of each pixel in the view of a single novel viewpoint requires the (r, g, b, σ) values of all the sampled points on the ray to be determined. To render the color of an image pixel, NeRF approximates a volume rendering integral. Light is emitted and sampled from the camera position to a pixel in the scene and the expected color $\hat{\mathbf{C}}$ of that pixel is then given by:

$$\hat{\mathbf{C}}(\mathbf{r}) = \int_{\hat{t}_{\text{near}}}^{\hat{t}_{\text{far}}} T(\hat{t})\sigma(\mathbf{r}(\hat{t}))\mathbf{c}(\mathbf{r}(\hat{t}), \mathbf{d})d\hat{t},$$

$$T(\hat{t}) = \exp\left(-\int_{\hat{t}_n}^{\hat{t}} \sigma(\mathbf{r}(t))dt\right). \tag{2}$$

where the function $T(\hat{t})$ represents the accumulated transmittance of the ray from $\hat{t}_{\text{near}}$ to $\hat{t}$, and $\hat{t}_{\text{near}}$ and $\hat{t}_{\text{far}}$ correspond to the samples at the near and far planes.

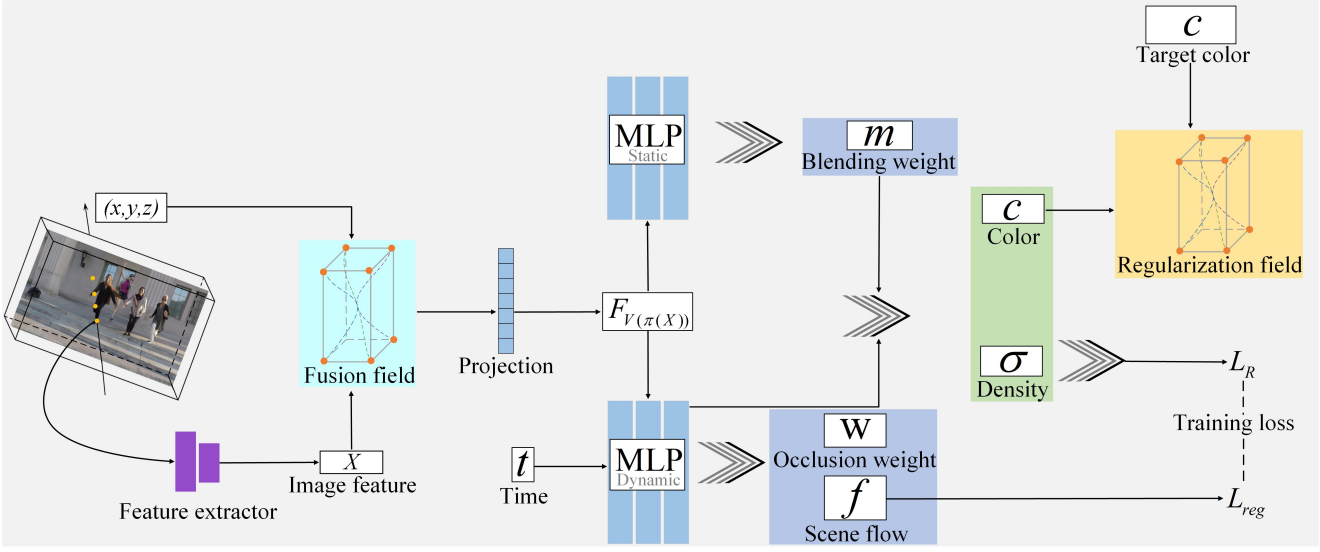

Fig. 2 Framework. Our dynamic NeRF with fusion regularization fields can take 2D-3D knowledge transfer as input to predict the flow fields from frame t to frame $t-1$ and frame $t+1$. For static components, we train a following NeRF model, but exclude all pixels marked as dynamic from model training. This allows us to reconstruct the background structure and appearance without conflicting with moving or deforming objects.

x

The goal of our work is to synthesize novel viewpoints at any desired time within the video. Figure 2 shows the framework of our method. The inputs are a monocular video frame sequence $(t_1, \ldots, t_n)$ of a dynamic scene and the known camera parameters $(k_1, \ldots, k_n)$. The most significant differences from existing work are that we predict the fusion regularization fields from frame t to frame $t-1$ and to frame $t+1$ for bi-directional consistency of adjacent video frames, and globally align the rendered pixels with the corresponding input video frame pixels. In addition, we propose a feature enhancement strategy for hierarchical volume sampling. Our method consists of three steps: (i) extraction of 2D semantic features from scene images using a pre-trained feature extractor and attachment of 2D feature information to the original hierarchical sampling points (see Sec. 3.2), (ii) knowledge transfer for 2D-3D fusion (see Sec. 3.3.1), and (iii) alignment regularization of the 3D scene flow field (see Sec. 3.3.2). To synthesize seamless and sharp dynamic scenes, we have designed a hybrid dynamic-static neural rendering network (see Sec. 3.4). Utilizing both 2D images and 3D scene data, we address the pixel misalignment in consecutive novel views.

3.2 Feature enhancement for hierarchical volume sampling

NeRF uses a combination of implicit neural fields and volume rendering techniques [70] to render 3D scenes by hierarchical volume sampling. However, the hierarchical volume sampling mechanism cannot provide sufficient pixel information for dynamic radiance fields where the input is a monocular video frame sequence. In order to increase the available input information for the model, we introduce additional 2D image information to the original hierarchical sampling points:

$$\left(\sigma_t, c_t, f_t^{3D}, f_t^{2D}, w_t\right) = \mathcal{F}_d\left(x_t, d, t, F_{V(\pi(x_t))}\right), \quad (3)$$

where x_t is the spatial position of 3D point x at frame t, d is the view direction. σ_t and $c_t = (r, g, b)$ are the volume density and color at frame t for the 3D point, respectively. π denotes projecting 3D point x_t onto the image. $\mathcal{F}_d$ is the dynamic representation model. $F_{V(\pi(x_t))}$ represents image feature extraction by bilinear interpolation.

In addition, the model predicts the forward and backward 3D scene flow fields $f_t^{3D} = (f_{t\to t+1}^{3D}, f_{t\to t-1}^{3D})$, and the forward and backward 2D feature fields $f_t^{2D} = (f_{t\to t+1}^{2D}, f_{t\to t-1}^{2D})$, which represent the 3D offset vectors and 2D feature offsets corresponding to x_t and its projection points at frames $t+1$ and $t-1$, respectively. To handle motion occlusion in 3D space, the model also predicts the occlusion weights $w_t = (w_{t\to t+1}, w_{t\to t-1})$ for previous frame $t-1$ and next frame $t+1$.

As Figure 3 shows, given an input image I of the scene, we extract the features $F_V = E(I)$, where E denotes the feature extraction network. Specifically, we choose the masked autoencoder (MAE) feature extractor [71]. The core idea of the masked autoencoder is to allow the model to learn a

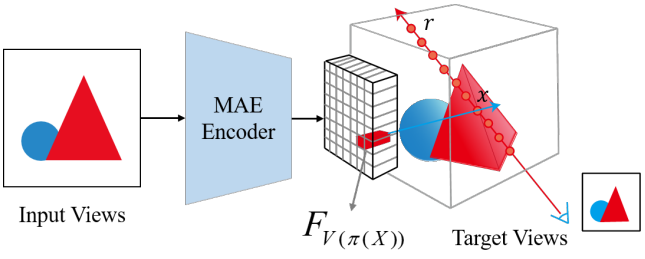

Fig. 3 Image feature extraction. Given the input image, $F_{V(\pi(X))}$ is extracted with the feature extraction network.

generalized intermediate representation, which increases the NeRF's ability to reason about under-observed pixels. After obtaining the 2D image features of the current frame through the feature extraction network, the features of the current frame and the 2D feature field can be used to calculate the pixel features of the previous frame and the next frame. In this paper, we retain the convolutional embedding operations in the first two stages of the feature encoder module. This enables the network to collect local image regions during each iteration. Thus, utilizing 2D local neighborhood information when rendering each pixel can be formulated as follows:

$$F_{\text{encoder 1}} = \text{StrideConv}\left(\text{MAE}_1, 2\right), \tag{4}$$

$$F_{\text{encoder 2}} = \text{StrideConv}\left(\text{MAE}_2, 2\right), \tag{5}$$

$$F_{2D} = \left[F_{\text{encoder1}}, F_{\text{encoder2}}\right], \tag{6}$$

where StrideConv$(\cdot, 2)$ represents the mask convolution operation with a stride of 2, MAE_1 and MAE_2 represent the two stages of the masked autoencoder feature extractor respectively, $F_{\text{encoder 1}}$ and $F_{\text{encoder 2}}$ are the two scales of features extracted by the masked autoencoder feature extractor, and F_{2D} is the final 2D fusion feature. Then, for each sampling point x on the ray r, we project x to the corresponding coordinate $\pi(x)$ on the image plane using the known camera intrinsics and then retrieve the corresponding image features by $\pi(x)$ and use bilinear interpolation to extract the feature vector $F_{V(\pi(x))}$.

3.3 Fusion regularization field

3.3.1 Knowledge transfer for 2D-3D fusion

Previous methods [7, 9] predict the forward and backward 3D scene flow of a dynamic scene, representing the offset of pixels moving at a uniform speed for frames $t-1$ and $t+1$. However, the 3D scene flow contains inaccurate depth information. In this paper, we use the same datasets as NSFF [7], in which the depth is not the ground truth, but it is generated by the depth estimation method. As a result, significant artifacts can be produced when we model dynamic objects in the scene using a scene flow that includes inaccurate depth information. The 2D feature field estimates the offset of pixels in adjacent frames, which does not use inaccurate depth. As shown in Figure 2, we introduce 2D image features in the fusion field to alleviate the artifacts of objects in adjacent frames and making the NeRF effectively handle rigid and non-rigid deformations of moving objects. At the same time, the 2D feature field with image priors can further enhance the accuracy and extend the NeRF's expressive capabilities for dynamic space-time view synthesis.

For each pixel, the training process starts with three frames of the scene to train the model: the current frame t, the previous frame $t-1$, and the next frame $t+1$. After the number of training iterations reaches 50,000 steps, the model can be trained using five adjacent frames of the scene, *i.e.* frames $t-2$ and $t+2$ are added.

Taking frame t as the reference frame, the volume density σ and color c of the 3D point x in frames $t-1$ and $t+1$ can be calculated according to the 3D scene flow field and 2D feature field, expressed as:

$$(\sigma_{t\to t-1}, c_{t\to t-1}) = \mathcal{F}_d\left(x_t + f^{3D}_{t\to t-1}, d, t-1, F_{V\left(\pi(x_t)+f^{2D}_{t\to t-1}\right)}\right), \tag{7}$$

$$(\sigma_{t\to t+1}, c_{t\to t+1}) = \mathcal{F}_d\left(x_t + f^{3D}_{t\to t+1}, d, t+1, F_{V\left(\pi(x_t)+f^{2D}_{t\to t+1}\right)}\right), \tag{8}$$

When rendering the image, we warp the 3D point x_t of the current frame t to frames $t-1$ and $t+1$ using the predicted 3D scene flow field f^{3D}_t and the 2D feature field f^{2D}_t. The volume density σ and color c of this 3D point x_t at adjacent frames $t-1$ and $t+1$ are rendered along the ray r_t. It can be seen that the volume density σ and color c of pixels in adjacent frames should be consistent.

3.3.2 Alignment regularization of 3D scene flow field

Adjacent monocular video frames in the same scene share the same semantic features; capturing and realigning the semantic features of the scene is beneficial to improve the consistency and fidelity of the scene flow. Since the contents and styles of adjacent frames are similar, the deep learning network can learn invariant representations. Therefore, we alleviate the artifacts due to inaccurate depth information by re-aligning the original 3D flow field at the semantic level through a vision transformer (ViT) [1]. In the ViT module, a monocular video frame is flattened into n non-overlapping patch sequences. Then each patch is linearly embedded into a c-dimensional vector, and the learned position embedding (using a value to characterize the absolute position of each patch) is added. ViT extracts abstract high-level semantic representations from non-overlapping patches using global

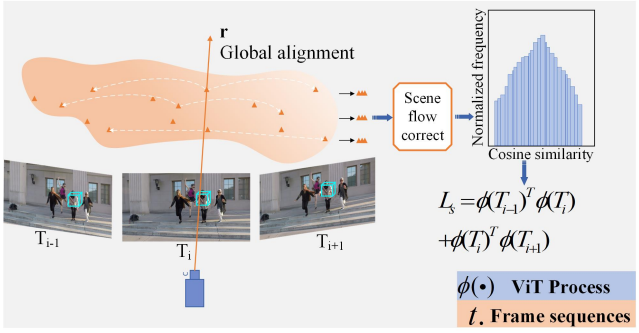

Fig. 4 3D scene flow regularization. High-level semantic information of adjacent video frame pixels is extracted by ViT, and then optimized using cosine similarity loss.

self-attention and generates a single global embedding vector. As Figure 4 shows, we use this mechanism to correct the high-level semantic features of adjacent frames.

The cost of matching the corresponding pixel points of adjacent video frames is determined based on the high-level semantic feature representation learned by ViT. Scene flow features after ViT processing are of the form $X_{t+l}, X_t \in \mathbb{R}^{n\times p}$, where p is the dimension of the features. The losses are calculated as follows:

$$\mathcal{L}_s = \frac{\Phi(X_{t+l}^i)\cdot\Phi(X_t^j)^\top}{\|\Phi(X_{t+l}^i)\|_2\,\|\Phi(X_i^j)\|_2}, \quad l\in\{-1,1\}, \tag{9}$$

where X_{t+l}^i, $X_t^j \in \mathbb{R}^{n\times p}$ are the i-th and j-th patches of X_{t+l} and X_t, respectively, and $\Phi(\cdot)$ denotes the normalized embedding of the image.

3.4 Hybrid rendering

In this paper, we use the same hybrid (dynamic and static) neural fields as NSFF [7] to render the scene. The static representation model $\mathcal{F}_s$ takes 3D coordinates x, view direction d, and feature vector $V(\pi(x))$ as input and aims to generate the volume density σ, and color $c=(r,g,b)$ of this 3D point:

$$(\sigma, c, m) = \mathcal{F}_s(x, d, F_{V(\pi(X))}), \tag{10}$$

where m is an unsupervised 3D mixture weight for linearly fusing σ and c from the static and dynamic representation models; m is generated by training the static representation network. The feature vector $F_{V(\pi(X))}$ provides a priori information to generate more accurate color details. Intuitively, m should assign a low weight to the dynamic representation in static regions with sufficient observations, as these can be rendered at higher fidelity by the static representation, while assigning a lower weight to the static representation in regions that are moving, as these can be better modeled by the dynamic representation:

$$\hat{C}_t\left(r_{\hat{t}}\right) = \int_{\hat{t}_{\text{near}}}^{\hat{t}_{\text{far}}} T_t(\hat{t})\sigma_t\left(r_t(\hat{t})\right) c_t\left(r_t(\hat{t})\right) d\hat{t}, \tag{11}$$

where $\sigma_t(r_t(\hat{t}))$ and $c_t(r_t(\hat{t}))$ are linear combinations of static and dynamic scene representations weighted by $m(\hat{t})$, given by:

$$\begin{aligned}\sigma_t\left(r_t(\hat{t})\right) c_t\left(r_t(\hat{t})\right) &= m(\hat{t})c\left(r_t(\hat{t})\right)\sigma\left(r_t(\hat{t})\right)\\ &\quad + (1-m(\hat{t}))c_t\left(r_t(\hat{t})\right)\sigma_t\left(r_t(\hat{t})\right),\end{aligned} \tag{12}$$

The final blended rendering loss $\mathcal{L}_R$ calculates the mean squared error between the blended rendered pixel value $\hat{C}_t\left(r_{\hat{t}}\right)$ and its corresponding true pixel value $C_t(r_{\hat{t}})$ along the ray $r_{\hat{t}}$:

$$\mathcal{L}_R = \sum_{r_{\hat{t}}} \left\| \hat{C}_t\left(r_{\hat{t}}\right) - C_t\left(r_{\hat{t}}\right) \right\|_2^2, \tag{13}$$

Based on blended rendering loss, we align the rendered pixels globally to enhance the overall spatial consistency of the rendered scene images (dynamic and static) with the original monocular video frames. First, we flatten the raw input video frames and the rendered scene images and then put them into ViT [1] to obtain their high-level semantic features. The pre-trained ViT has high robustness to unaligned pixels in the scene flow. Following [72], we compute the $\mathcal{L}_2$ distance between the high-level semantic features extracted from the rendered scene images and the semantic features of the original video frames to construct the global loss function:

$$\mathcal{L}_{\text{global}} = \left\| E\left(t_d\right) - E\left(t_f\right) \right\|^2, \tag{14}$$

where $E(\cdot)$ denotes the advanced semantic feature extractor, t_d denotes the scene images generated by NeRF rendering, and t_f denotes the original monocular video frames. Therefore, the improved combined loss for the space-time viewpoints synthesis task is as follows:

$$\mathcal{L} = \lambda_r\mathcal{L}_R + \lambda_s\mathcal{L}_S + \lambda_g\mathcal{L}_{\text{global}}, \tag{15}$$

where λ_r, λ_s, and λ_g are balancing weights for the corresponding loss terms, which are set to $1, 0.1, 0.1$ in this paper, respectively.

4 Experiments

4.1 Setup

4.1.1 Implementation details

Our framework is implemented in PyTorch. The hyperparameters λ_r, λ_t, λ_c, λ_{eg}, λ_g, and λ_z are set to 1.0, 1.0, 1.0, 0.1, 0.2, and 0.4 during training. We use COLMAP [73] to estimate the camera intrinsics and extrinsics, and since COLMAP can only estimate camera parameters for static scenes, we use instance segmentation [74] to hide the features from the regions that are associated with the common dynamic objects. During training and testing, we sample 64 points along each camera ray. In addition, we use the Adam

Table 1 Quantitative evaluation of space-time novel view synthesis for dynamic scenes on the *Nvidia Dynamic Scenes* Dataset.

Scene	Method	Dynamic Only		Full Image	
		SSIM ↑	LPIPS ↓	SSIM ↑	LPIPS ↓
Jumping	DynamicNeRF	0.203	0.392	0.709	0.205
	NSFF	0.685	0.176	0.918	0.072
	Ours	**0.705**	**0.131**	**0.925**	**0.055**
Umbrella	DynamicNeRF	0.225	0.649	0.510	0.433
	NSFF	**0.549**	0.171	0.842	**0.097**
	Ours	**0.549**	**0.162**	**0.844**	0.098
Playground	DynamicNeRF	0.196	0.366	0.490	0.325
	NSFF	0.716	0.143	0.876	0.081
	Ours	**0.725**	**0.127**	**0.877**	**0.075**
Skating	DynamicNeRF	0.663	0.159	0.812	0.054
	NSFF	0.788	0.106	0.971	0.035
	Ours	**0.789**	**0.098**	**0.977**	**0.023**
Truck	DynamicNeRF	0.218	0.149	0.492	0.134
	NSFF	0.839	0.056	0.691	0.026
	Ours	**0.913**	**0.046**	**0.963**	**0.024**
Average	DynamicNeRF	0.301	0.343	0.603	0.230
	NSFF	0.715	0.130	0.860	0.062
	Ours	**0.736**	**0.113**	**0.917**	**0.055**

Table 2 Comparison of our approach to state-of-the-art novel view synthesis methods NeRFPlayer, K-planes, and DynIBaR on the *Nvidia Dynamic Scenes* Dataset.

Scene	Methods	Dynamic Only		Full Image	
		SSIM ↑	LPIPS ↓	SSIM ↑	LPIPS ↓
Jumping	NeRFPlayer	0.532	0.289	0.813	0.102
	K-planes	0.639	0.366	0.835	0.078
	DynIBaR	0.701	0.133	0.922	0.057
	Ours	**0.705**	**0.131**	**0.925**	**0.055**
Playground	NeRFPlayer	0.598	0.245	0.812	0.163
	K-planes	0.654	0.198	0.822	0.147
	DynIBaR	0.721	0.129	0.875	0.077
	Ours	**0.725**	**0.127**	**0.877**	**0.075**
Average	NeRFPlayer	0.565	0.267	0.813	0.133
	K-planes	0.647	0.282	0.829	0.113
	DynIBaR	0.711	0.131	0.899	0.067
	Ours	**0.715**	**0.129**	**0.901**	**0.065**

optimizer [75] to train a separate model for each scene, with learning rate 5×10^{-4}. Training a full model takes about seven days per scene using two NVIDIA 2080ti GPUs and rendering takes roughly 6 seconds for each 512×288 frame.

4.1.2 Datasets

We evaluate our method on the *Nvidia Dynamic Scene* Dataset [76]; it consists of 8 scenes with human motions and inanimate objects and background. These sequences were captured with 12 cameras using a static camera setup. All cameras synchronously captured images at 24 different time steps $\{t_0, \ldots, t_{23}\}$. The input is 24 frames of monocular videos $\{i_0, \ldots, i_{23}\}$ obtained by sampling the image taken by the k-th camera at time t_j. Note that each frame of the video uses a different camera to simulate camera motion in order to obtain information about the perspective transformation. Frame i contains a background that does not change over time, and dynamic objects which change over time. We use positional encoding to transform the inputs and parameterize the scenes using standardized coordinates. We assume that all cameras share the same intrinsic parameters. Following [7], we simulate the moving monocular camera by extracting images sampled from each camera viewpoint at different time instances and evaluate the results of view synthesis with respect to known held-out viewpoints and frames. For each scene, we use 24 frames from the original video for training and use the remaining 11 held-out images from each time instance for evaluation.

4.1.3 Metrics

We evaluate the results of novel view synthesis using the following metrics: (i) structural similarity index measure (SSIM) [77], which measures the similarity between images from three aspects: brightness, contrast, and structural similarity, and (ii) learned perceptual image patch similarity (LPIPS) [78], which computes the distance between the generated image and the ground-truth image in the perceptual domain. LPIPS is generally considered to be closer to human perception, when assessing reconstruction errors. Furthermore, we calculate errors both over the entire scene (Full

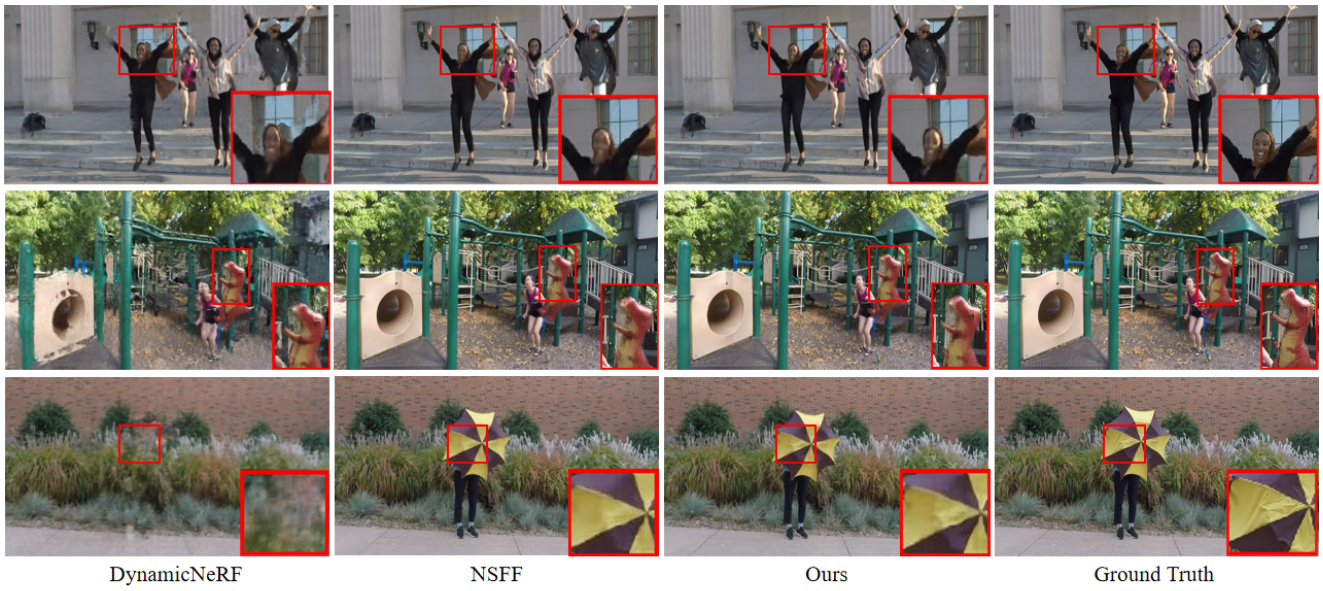

Fig. 5 Qualitative comparisons to state-of-the-art space-time novel view synthesis methods. Left to right: results from DynamicNeRF, NSFF, our method and ground truth. The images generated by our model more closely match the ground-truth, and include fewer artifacts, especially in the highlighted regions.

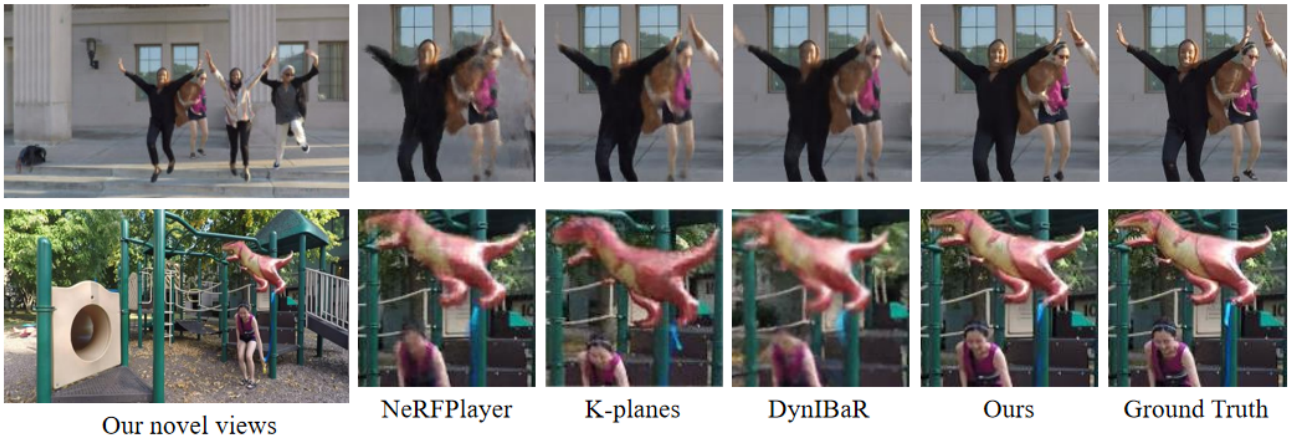

Fig. 6 Qualitative comparisons to other state-of-the-art novel view synthesis methods. Left to right: our results, close ups of results from NeRFPlayer K-planes, DynIBaR, our method, and ground truth.

Image) and restricted to dynamic regions only (Dynamic Only). The dynamic component is obtained using the binary masks in the initial inputs.

4.2 Quantitative results

We first compare our approach to state-of-the-art monocular dynamic view synthesis methods: DynamicNeRF [67] and NSFF [7], which specifically generate novel space-time view images with monocular videos. Note that DynamicNeRF [67] was originally trained with 12 input images. We re-trained DynamicNeRF [67] and NSFF [7] with 24 frames of the monocular videos for fair comparisons, and generated qualitative and quantitative results using the same test set.

Table 1 reports our results and compares them to those of DynamicNeRF and NSFF, which are specialized to space-time novel view synthesis for dynamic scenes. Our method outperforms them for all metrics. In particular, our method achieves much better results on the dynamic components, demonstrating that our model is better adapted to handle non-rigid motions and blur in the entire scenes. In addition, calculating the mean values on all datasets, our method gets the best scores: our model outperforms existing methods.

In order to demonstrate the superiority and robustness of our method for novel view synthesis, we also conducted comparative experiments with other state-of-the-art methods: NeRFPlayer [79], K-planes [80], and DynIBaR [69]. NeRF-

Table 3 Quantitative comparison with four alternative designs.

Scene	Methods	Dynamic Only		Full Image	
		SSIM ↑	LPIPS ↓	SSIM ↑	LPIPS ↓
Jumping	w/o Feature Enhancement	0.694	0.133	0.926	**0.052**
	w/o 2D-3D Fusion	0.698	0.136	0.922	0.057
	w/o 3D Regularization	0.699	0.136	**0.927**	0.054
	w/o $\mathcal{L}_{\text{Constraint}}$	0.696	0.136	0.923	0.056
	Full Model	**0.705**	**0.131**	0.925	0.055
Skating	w/o Feature Enhancement	0.769	0.118	**0.982**	**0.019**
	w/o 2D-3D Fusion	**0.814**	**0.087**	0.971	0.027
	w/o 3D Regularization	0.813	0.098	0.974	0.027
	w/o $\mathcal{L}_{\text{Constraint}}$	0.800	0.104	0.976	0.024
	Full Model	0.805	0.098	0.977	0.023
Truck	w/o Feature Enhancement	0.892	0.054	0.965	**0.023**
	w/o 2D-3D Fusion	0.907	0.053	0.961	0.026
	w/o 3D Regularization	0.908	0.054	0.961	0.027
	w/o $\mathcal{L}_{\text{Constraint}}$	0.901	0.053	**0.967**	**0.023**
	Full Model	**0.913**	**0.046**	0.963	0.024
Average	w/o Feature Enhancement	0.785	0.102	**0.958**	**0.031**
	w/o 2D-3D Fusion	0.806	**0.092**	0.951	0.037
	w/o 3D Regularization	0.807	0.096	0.954	0.036
	w/o $\mathcal{L}_{\text{Constraint}}$	0.799	0.098	0.955	0.034
	Full Model	**0.808**	**0.092**	0.955	0.034

Fig. 7 Comparison to NSFF, showing that our proposed fusion regularization field is the key to better visual results.

Player and K-planes both support novel view synthesis with multiple cameras; NeRFPlayer focuses on reducing training and rendering times, while K-planes focuses on low memory usage and DynIBaR focuses on synthesizing novel views from long videos. Table 2 compares our method to NeRFPayer, K-planes and DynIBaR. We randomly selected two scenarios from the *Nvidia Dynamic Scene* dataset, *i.e.* Playground and Jumping for evaluation.

4.3 Qualitative results

We provide qualitative comparisons between our approach and two state-of-the-art monocular and dynamic-scene-based view synthesis methods in Figure 5. DynamicNeRF [67] produces many artifacts in both static and dynamic regions of the scenes. NSFF [7] reconstructs most static regions correctly since it treats all the moving objects as view-dependent effects leading to loss of certain details, *e.g.* the head of the woman, the eyes of the toy dinosaur, and the folds of the umbrella when the dynamic objects move rapidly. In contrast, our model is able to model the fast motion of dynamic objects and still retains the complete structure even if the dynamic regions are widely separated between two adjacent frames. We also make a comparison to other state-of-the-art novel view synthesis methods: K-planes [80], NeRFPlayer [79] and DynIBaR [69], in Figure 6. Our approach outperforms them both in terms of overall quality and scene details.

Figure 7 shows a comparison to NSFF [7] on sequences with large motion, *e.g.* a moving truck and a man skating. Unlike NSFF [7], which uses a static NeRF to predict the blending weights, we propose a fusion regularization field to fuse the 2D feature field features to enhance the quality of the foreground. The benefits of this 2D feature field include extracting features of dynamic regions and generating the images with less blur.

Figure 8 shows some novel viewpoints synthesized by our method at any desired time within the video, demonstrating that our method can achieve space-time novel view synthesis and generate realistic images, specifically generating accurate results for dynamic regions of the entire scene.

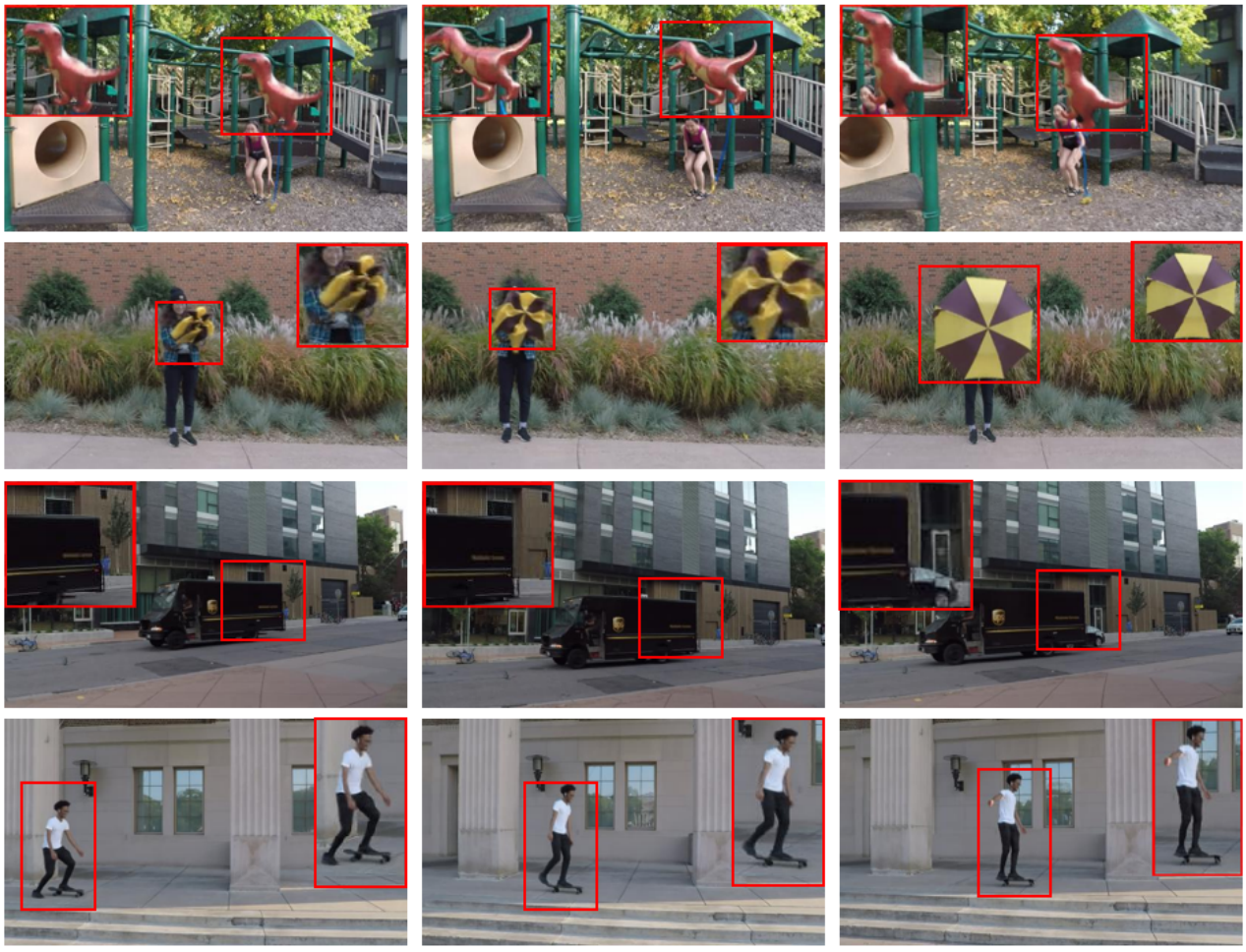

Fig. 8 Further visual results of space-time novel view synthesis.

4.4 Ablation study

We conducted ablation experiments on the *Nvidia Dynamic Scene* Dataset [76] for each of the four proposed components and the full model. We analyzed the effectiveness of each proposed component in novel view synthesis by removing (i) feature enhancement (w/o Feature Enhancement), (ii) 2D-3D fusion (w/o 2D-3D Fusion), (iii) 3D regularization (w/o 3D regularization), and (iv) the constraint loss (w/o $\mathcal{L}_{\text{Constraint}}$); our full method is denoted (Full Model).

Quantitative results are shown in Table 3, which demonstrate the relative importance of each component, with the full model performing the best. As shown in Table 3, the results of the dynamic components in *Jumping* and *Truck* scenes of the Full Model outperform the other four variants. The Full Model does not get the best numerical performance for the dynamic regions in the *Skating* scene. This is because the movement of the skating man is very smooth which leads the model to learn more background information. Nevertheless, the average results for all cases on the dynamic regions, indicate that the full model performs best.

As for the full image, since the dynamic regions in the foreground take up a small proportion of the whole image, and the background information is dominant, this leads the full model to achieve a performance comparable to the best scores.

Visual results are shown in Figure 9. Compared to the four variants, the full model achieves better results especially for non-rigid motions and dynamic regions. More results can be found in the demo video at https://cic.tju.edu.cn/faculty/likun/projects/FRNerf/demo.mp4.

5 Conclusions and limitations

In this paper, we propose a novel framework for novel view synthesis from a monocular video. Our core contribution lies in the fusion regularization fields and the addition of sampled features to enhance consistency between video frames and to mitigate the ambiguity in synthesizing dynamic scenes due to inaccurate depth information and under-sampled features. We

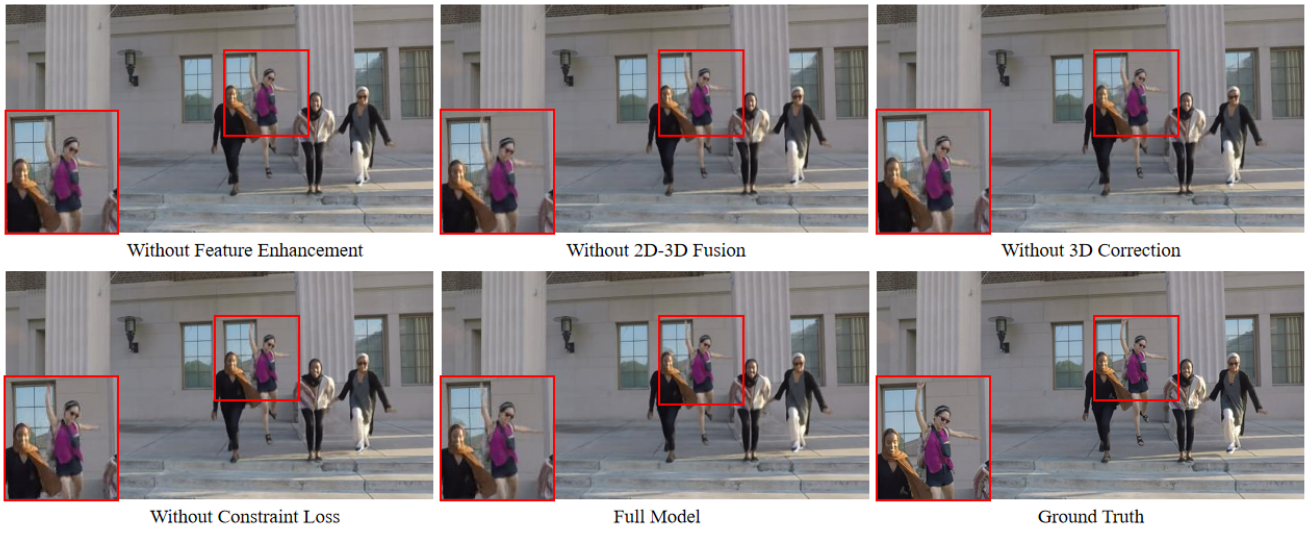

Fig. 9 Qualitative results of ablation study.

demonstrate qualitatively and quantitatively, using multiple dynamic datasets, that our approach can synthesize photo-realistic novel-view images from a monocular video, and can achieve significant improvements over state-of-the-art methods on the dynamic scene benchmarks.

Space-time view synthesis on dynamic scenes is a highly challenging task, and our proposed method addresses the problems of temporal consistency and under-sampled features. However, our method still has a limitation in common with most existing methods [7, 67]; it has many learnable parameters, leading to a long training time. In future, we hope to improve training efficiency using a Jittor model [81, 82], which is 2.26 times faster than the equivalent PyTorch model on average.

Declarations

- The authors have no competing interests to declare that are relevant to the content of this article.
- Compliance with ethical standards.
- This study does not contain any studies with human or animal subjects performed by any of the authors.
- Availability of data and materials: 'Code is available at https://cic.tju.edu.cn/faculty/likun/projects/FRNerf.
- Funding: This work was supported in part by National Key R&D Program of China (2023YFC3082100), National Natural Science Foundation of China (62122058 and 62171317), and Science Fund for Distinguished Young Scholars of Tianjin (No. 22JCJQJC00040).
- Acknowledgements: We are grateful to the Associate Editor and anonymous reviewers for their help in improving this paper.
- Authors' contributions:
  Xinyi Jing: theoretical development, experiment implementation, paper writing, approving the final version of the article publication, including references,
  Tao Yu: theoretical development, experiment implementation, paper writing, approving the final version of the article publication, including references,
  Renyuan He: theoretical development, experiment implementation, paper writing, approving the final version of the article publication, including references,
  Yu-Kun Lai: guidance, theoretical development, experimental design, paper revision, approving the final version of the article for publication, including references,
  Kun Li: guidance, theoretical development, experimental design, paper writing, approving the final version of the article for publication, including references.

References

[1] Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, Krueger G, Sutskever I. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, 2021, 8748–8763.

[2] Yoon JS, Kim K, Gallo O, Park HS, Kautz J. 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, 5336–5345.

[3] Broxton M, Flynn J, Overbeck R, Erickson D, Hedman P, Duvall M, Dourgarian J, Busch J, Whalen M, Debevec P. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics*, 2020, 39(4).

[4] Collet A, Chuang M, Sweeney P, Gillett D, Evseev D, Calabrese D, Hoppe H, Kirk A, Sullivan S. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics*, 2015, 34(4).

[5] Zheng Z, Yu T, Wei Y, Dai Q, Liu Y. Deephuman: 3D human reconstruction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 7739–7749.

[6] Guler RA, Kokkinos I. Holopose: holistic 3D human reconstruction in-the-wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 10884–10894.

[7] Li Z, Niklaus S, Snavely N, Wang O. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 6498–6508.

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

[9] Gao C, Saraf A, Kopf J, Huang JB. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 5712–5721.

[10] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R. NeRF: representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the European Conference on Computer Vision*, 2020, 405–421.

[11] Sitzmann V, Zollhöfer M, Wetzstein G. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 2019, 32.

[12] Peng S, Niemeyer M, Mescheder L, Pollefeys M, Geiger A. Convolutional occupancy networks. In *Proceedings of the European Conference on Computer Vision*, 2020, 523–540.

[13] Müller T, Evans A, Schied C, Keller A. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 2022, 41(4): 1–15.

[14] Niemeyer M, Mescheder L, Oechsle M, Geiger A. Differentiable volumetric rendering: learning implicit 3D representations without 3D supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 3504–3515.

[15] Oechsle M, Mescheder L, Niemeyer M, Strauss T, Geiger A. Texture fields: learning texture representations in function space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 4531–4540.

[16] Chen Z, Zhang H. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 5939–5948.

[17] Genova K, Cole F, Sud A, Sarna A, Funkhouser T. Local deep implicit functions for 3D shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 4857–4866.

[18] Genova K, Cole F, Vlasic D, Sarna A, Freeman WT, Funkhouser T. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 7154–7164.

[19] Park JJ, Florence P, Straub J, Newcombe R, Lovegrove S. DeepSDF: learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 165–174.

[20] Xu Q, Wang W, Ceylan D, Mech R, Neumann U. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *Advances in neural information processing systems*, 2019, 32.

[21] Shao R, Zheng Z, Tu H, Liu B, Zhang H, Liu Y. Tensor4D: efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 16632–16642.

[22] Chen A, Xu Z, Geiger A, Yu J, Su H. TensoRF: tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision*, 2022, 333–350.

[23] Saito S, Huang Z, Natsume R, Morishima S, Kanazawa A, Li H. Pifu: pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 2304–2314.

[24] Atzmon M, Lipman Y. Sal: sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 2565–2574.

[25] Michalkiewicz M, Pontes JK, Jack D, Baktashmotlagh M, Eriksson A. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 4743–4752.

[26] Mescheder L, Oechsle M, Niemeyer M, Nowozin S, Geiger A. Occupancy networks: learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 4460–4470.

[27] Yariv L, Atzmon M, Lipman Y. Universal differentiable renderer for implicit neural representations. *arXiv preprint arXiv:2003.09852*, 2020, 2.

[28] Tewari A, Fried O, Thies J, Sitzmann V, Lombardi S, Sunkavalli K, Martin-Brualla R, Simon T, Saragih J, Nießner M, Pandey R, Fanello S, Wetzstein G, Zhu JY, Theobalt C, Agrawala M, Shechtman E, Goldman DB, Zollhöfer M. State of the art on neural rendering. *Computer Graphics Forum*, 2020, 39(2): 701–727.

[29] Kato H, Beker D, Morariu M, Ando T, Matsuoka T, Kehl W, Gaidon A. Differentiable rendering: a survey. *arXiv preprint arXiv:2006.12057*, 2020.

[30] Martin-Brualla R, Radwan N, Sajjadi MS, Barron JT, Dosovitskiy A, Duckworth D. Nerf in the wild: neural radiance fields for unconstrained photo collections. In *Proceedings of*

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, 7210–7219.

[31] Buehler C, Bosse M, McMillan L, Gortler S, Cohen M. Unstructured lumigraph rendering. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 2023, 497–504.

[32] Debevec PE, Taylor CJ, Malik J. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 2023, 465–474.

[33] Riegler G, Koltun V. Free view synthesis. In *Proceedings of the European Conference on Computer Vision*, 2020, 623–640.

[34] Waechter M, Moehrle N, Goesele M. Let there be color! large-scale texturing of 3D reconstructions. In *Proceedings of the European Conference on Computer Vision*, 2014, 836–850.

[35] Thies J, Zollhöfer M, Nießner M. Deferred neural rendering: image synthesis using neural textures. *ACM Transactions on Graphics*, 2019, 38(4): 1–12.

[36] Wood DN, Azuma DI, Aldinger K, Curless B, Duchamp T, Salesin DH, Stuetzle W. Surface light fields for 3D photography. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 2023, 487–496.

[37] Flynn J, Broxton M, Debevec P, DuVall M, Fyffe G, Overbeck R, Snavely N, Tucker R. Deepview: view synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 2367–2376.

[38] Srinivasan PP, Tucker R, Barron JT, Ramamoorthi R, Ng R, Snavely N. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 175–184.

[39] Srinivasan PP, Mildenhall B, Tancik M, Barron JT, Tucker R, Snavely N. Lighthouse: predicting lighting volumes for spatially-coherent illumination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 8080–8089.

[40] Zhou T, Tucker R, Flynn J, Fyffe G, Snavely N. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)*, 2018, 37(4): 1–12.

[41] Tucker R, Snavely N. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 551–560.

[42] Penner E, Zhang L. Soft 3D reconstruction for view synthesis. *ACM Transactions on Graphics*, 2017, 36(6): 1–11.

[43] Kutulakos KN, Seitz SM. A theory of shape by space carving. *International Journal of Computer Vision*, 2000, 38(3): 199–218.

[44] Lombardi S, Simon T, Saragih J, Schwartz G, Lehrmann A, Sheikh Y. Neural volumes: learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)*, 2019, 38(4): 1–14.

[45] Seitz SM, Dyer CR. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 1999, 35(2): 151–173.

[46] Sitzmann V, Thies J, Heide F, Nießner M, Wetzstein G, Zollhofer M. Deepvoxels: learning persistent 3D feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 2437–2446.

[47] Liu L, Gu J, Zaw Lin K, Chua TS, Theobalt C. Neural sparse voxel fields. In *Advances in Neural Information Processing Systems*, 2020, 15651–15663.

[48] Sun C, Sun M, Chen HT. Direct voxel grid optimization: super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 5459–5469.

[49] Yu A, Li R, Tancik M, Li H, Ng R, Kanazawa A. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 5752–5761.

[50] Fridovich-Keil S, Yu A, Tancik M, Chen Q, Recht B, Kanazawa A. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 5501–5510.

[51] Tancik M, Srinivasan P, Mildenhall B, Fridovich-Keil S, Raghavan N, Singhal U, Ramamoorthi R, Barron J, Ng R. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, 2020, 7537–7547.

[52] Barron JT, Mildenhall B, Tancik M, Hedman P, Martin-Brualla R, Srinivasan PP. Mip-nerf: a multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 5855–5864.

[53] Zhang K, Riegler G, Snavely N, Koltun V. Nerf++: analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

[54] Tancik M, Casser V, Yan X, Pradhan S, Mildenhall B, Srinivasan PP, Barron JT, Kretzschmar H. Block-nerf: scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 8248–8258.

[55] Barron JT, Mildenhall B, Verbin D, Srinivasan PP, Hedman P. Mip-nerf 360: unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 5470–5479.

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

[57] Zitnick CL, Kang SB, Uyttendaele M, Winder S, Szeliski R. High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics*, 2004, 23(3): 600–608.

[58] Kanade T, Rander P, Narayanan P. Virtualized reality: constructing virtual worlds from real scenes. *IEEE Multimedia*, 1997, 4(1): 34–47.

[59] Bansal A, Vo M, Sheikh Y, Ramanan D, Narasimhan S. 4D visualization of dynamic events from unconstrained multi-view videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 5366–5375.

[60] Li T, Slavcheva M, Zollhoefer M, Green S, Lassner C, Kim C, Schmidt T, Lovegrove S, Goesele M, Newcombe R, Lv Z. Neural 3D video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 5521–5531.

[61] Wang L, Zhang J, Liu X, Zhao F, Zhang Y, Zhang Y, Wu M, Yu J, Xu L. Fourier plenOctrees for dynamic radiance field rendering in real-time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 13524–13534.

[62] Park K, Sinha U, Barron JT, Bouaziz S, Goldman DB, Seitz SM, Martin-Brualla R. Nerfies: deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 5865–5874.

[63] Park K, Sinha U, Hedman P, Barron JT, Bouaziz S, Goldman DB, Martin-Brualla R, Seitz SM. HyperNeRF: a higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics (TOG)*, 2021, 40(6): 1–12.

[64] Pumarola A, Corona E, Pons-Moll G, Moreno-Noguer F. D-nerf: neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 10318–10327.

[65] Tretschk E, Tewari A, Golyanik V, Zollhöfer M, Lassner C, Theobalt C. 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, 12959–12970.

[66] Du Y, Zhang Y, Yu HX, Tenenbaum JB, Wu J. Neural radiance flow for 4D view synthesis and video processing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 14304–14314.

[67] Gao C, Saraf A, Kopf J, Huang JB. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 5712–5721.

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

[69] Li Z, Wang Q, Cole F, Tucker R, Snavely N. DynIBaR: neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 4273–4284.

[70] Kajiya JT, Von Herzen BP. Ray tracing volume densities. *ACM SIGGRAPH Computer Graphics*, 1984, 18(3): 165–174.

[71] Gao P, Ma T, Li H, Lin Z, Dai J, Qiao Y. MCMAE: masked convolution meets masked autoencoders. *Advances in Neural Information Processing Systems*, 2022, 35: 35632–35644.

[72] Xu D, Jiang Y, Wang P, Fan Z, Shi H, Wang Z. SinNeRF: training neural radiance fields on complex scenes from a single image. In *Proceedings of the European Conference on Computer Vision*, 2022, 736–753.

[73] Schonberger JL, Frahm JM. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, 4104–4113.

[74] Girshick R. Fast R-CNN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2015, 1440–1448.

[75] Kingma DP, Ba J. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[76] Yoon JS, Kim K, Gallo O, Park HS, Kautz J. 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, 5336–5345.

[77] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004, 13(4): 600–612.

[78] Zhang R, Isola P, Efros AA, Shechtman E, Wang O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, 586–595.

[79] Song L, Chen A, Li Z, Chen Z, Chen L, Yuan J, Xu Y, Geiger A. NeRFPlayer: a streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 2023: 2732–2742.

[80] Fridovich-Keil S, Meanti G, Warburg FR, Recht B, Kanazawa A. K-planes: explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 12479–12488.

[81] Hu SM, Liang D, Yang GY, Yang GW, Zhou WY. Jittor: A novel deep learning framework with meta-operators and unified graph execution. *Science China Information Sciences*, 2020, 63(12): 1–21.

[82] Zhou WY, Yang GW, Hu SM. Jittor-GAN: A fast-training generative adversarial network model zoo based on Jittor. *Computational Visual Media*, 2021, 7(1): 153–157.

Xinyi Jing received a B.E. degree from the School of Computer Science, Shaanxi Normal University, Xi'an, China, in 2020. She is currently pursuing an M.E. degree in the College of Intelligence and Computing, Tianjin University, China. Her research interests are in computer vision and computer graphics.

Tao Yu is currently pursuing a Ph.D. degree in the College of Intelligence and Computing, Tianjin University, China. His research interests are in computer vision and computer graphics.

Renyuan He received a B.E. degree from the School of Cyberscience and Engineering, Zhengzhou University, China, in 2021. He is currently pursuing an M.E. degree in the College of Intelligence and Computing, Tianjin University, Tianjin, China. His research interests are in computer vision and computer graphics.

Yu-Kun Lai received his bachelor's and Ph.D. degrees in computer science from Tsinghua University in 2003 and 2008, respectively. He is currently a Professor in the School of Computer Science & Informatics, Cardiff University, UK. His research interests include computer graphics, geometry processing, image processing and computer vision. He is on the editorial boards of *IEEE Transactions on Visualization and Computer Graphics* and *The Visual Computer*.

Kun Li received a B.E. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2006, and master's and Ph.D. degrees from Tsinghua University, Beijing, in 2011. She is currently a Professor in the College of Intelligence and Computing, Tianjin University. She served as an Area Chair in ACM Multimedia 2021 and a Local Chair in VALSE 2022. Her research interests include 3D reconstruction and generative AI.