Neural Panoramic Representation for Spatially and Temporally Consistent 360° Video Editing

Simin Kou
Victoria University of Wellington

Fang-Lue Zhang*
Victoria University of Wellington

Yu-Kun Lai
Cardiff University

Neil A. Dodgson
Victoria University of Wellington

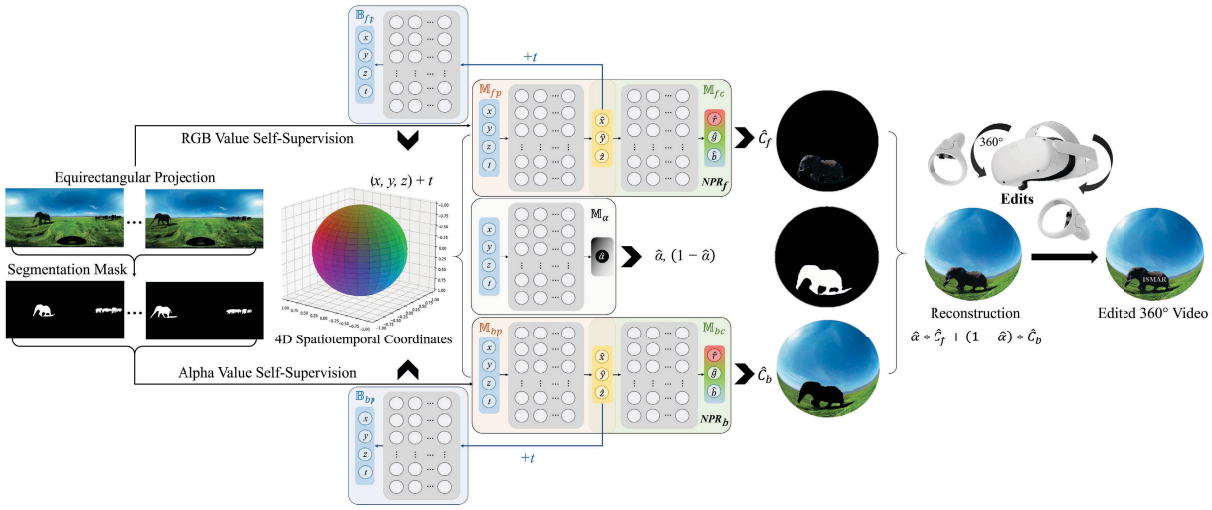


Figure 1: The framework of our proposed **N**eural **P**anoramic **R**epresentation (**NPR**). Our model represents 360° videos using MLPs, allowing for easy video editing in the true spherical space. Given the captured 360° video, its segmentation masks, and the designed 4D spatiotemporal coordinates as inputs, our model predicts implicit spherical positions for generating spherical content layers, providing each layer's appearance for reconstruction. We incorporate bi-directional mapping by introducing an additional pair of backward mapping MLPs to model the global motion of individual dynamic scenes, facilitating flexible 360° video editing.

Abstract

Content-based 360° video editing allows users to manipulate panoramic content for interaction in a dynamic visual world. However, the current related methods (2D neural representation and optical flow) show limitations in producing high-quality panoramic content from 360° videos due to their lack of capacity to model the inherent spatiotemporal relationships among pixels in the true panoramic space. To address this issue, we propose a Neural Panoramic Representation (NPR) method to model the global inter-pixel relationships, facilitating immersive video editing. Specifically, our method utilizes MLP-based networks to learn spherical implicit content layers, by encoding the spherical spatiotemporal positions and appearance details within the panoramic video, and bi-directional mapping between the original video frames and the learned content layers, to capture the interpretable and global omnidirectional visual characteristics of individual dynamic scenes. Additionally, we introduce innovative loss functions (spherical neighborhood consistency and unit spherical regularization) to ensure the creation of appropriate implicit spherical content layers. We further provide an interactive layer neural panoramic editing approach based on the proposed NPR, in the head-mounted display device. We evaluate this framework on diverse real-world 360° videos, showing superior performance on both reconstruction and consistent editing compared to existing state-of-the-art (SOTA) neural representation techniques.

Index Terms: Virtual reality, immersive video editing, 360° video manipulation, spherical content modeling.

*email: fanglue.zhang@vuw.ac.nz (Corresponding author)

1 Introduction

Recent advances in mixed reality (MR) and virtual reality (VR) have revolutionized how individuals can experience different parts of the physical world virtually. This seamless immersion offers vast potential across entertainment, education, and enriched experiences, overcoming previous barriers of safety and cost [57]. The growth of the metaverse has further fueled research in this field. The advent of 360° videos offers a cost-effective way to capture and present dynamic real-world environments for MR and VR applications. Recent research has established a solid foundation in capturing [30, 40], analyzing [22, 47, 51], stabilizing [19, 44], and presenting [37, 46] 360° videos, enabling high-quality, immersive content delivery. However, current MR and VR applications based on 360° videos offer only limited interaction capabilities, such as the ability to change view angles and navigate within a confined area [7, 13]. This limits the extent to which users can feel fully immersed in the experience.

This research focuses on content-based 360° video understanding and editing, heightening the immersion of MR and VR applications, which enables users to interactively manipulate panoramic video content. The main challenge is the absence of an effective approach to learning the inherent spatiotemporal relationships within the 360° video content. Although optical-flow-based approaches show promise for editing tasks like video completion [25, 52], they struggle with pixel-level operations like drawing strokes due to issues like occlusions and drifting. Additionally, the absence of a real-world panoramic optical flow dataset means that current 360° optical flow technologies [22, 41] fall short of meeting the requirements for practical 360° video editing. Recent advances in deep learning technologies enable self-supervised representation learning to support temporally consistent video editing [17, 54]. However, these methods are designed for 2D planar videos and do not consider the unique spatial properties of 360° data. 360° videos repre-

sent signals across a spherical surface, which is non-developable, as widely known from cartography [16]. This mathematical fact implies that common sphere-to-plane projections (e.g., cubemap, equirectangular projection, etc.) cannot produce entirely conformal 2D maps. These projections inherently suffer from distortion and discontinuities. Consequently, existing 2D planar neural video representations [17, 54] are inadequate for producing high-quality atlases or consistent editing results in 360° videos. Recently, EgoNeRF [9] introduced a neural spherical representation focusing on static scenes for novel view synthesis. However, it still struggles with discontinuity issues due to its design based on two separate hemispheres and lacks editability. Thus, there is a need for a new editable spherical representation for dynamic environments in 360° videos.

Motivated by this, we propose a neural panoramic representation (NPR) method to model the inherent spatiotemporal relationships among the spherical pixels in a 360° video, facilitating immersive video editing in MR and VR applications. Unlike existing 2D neural representation learning methods [17, 54], our approach eschews the learning of planar layers, developing a network that directly acquires the ability to model spherical surface layers for video representation.

Our method employs Multi-Layer Perceptron (MLP)-based networks to learn spherical implicit content layers, effectively capturing the omnidirectional visual characteristics of the dynamic scene in the 360° video. The trained MLPs encode motion information for each spherical content layer, enabling high-fidelity reconstruction of dynamic 360° video frames. The motion information encompasses the movements of the foreground objects and any background changes induced by camera motion. To ensure accurate encoding and interpretation of individual motion information for different objects, we begin by encoding appearance details into spherical content layers. These layers are used to restore both foreground and background elements within the panoramic view. Subsequently, the restored elements in different layers are combined by alpha-blending to generate a seamless, full-frame panoramic image. Compared to an alternative approach that segments a panoramic video into multiple perspective videos and applies 2D learning techniques, our method excels in generating globally consistent motion and appearance data while circumventing the extra computational expenses associated with integrating information from various perspectives. Additionally, we introduce innovative loss functions, such as spherical neighborhood consistency and unit spherical regularization, to ensure the creation of appropriate implicit spherical content layers. To achieve intuitive editing, we introduce a bi-directional mapping scheme allowing global tracking to identify user-specifying regions in any frame to apply customized edits. By training our NPR using this network architecture, we empower the capability for spatially and temporally consistent 360° video editing, overcoming fundamental limitations and surpassing the performance of existing neural representation techniques. Our contributions are summarized as follows:

- We propose a novel **N**eural **P**anoramic **R**epresentation (NPR) that captures the inherent spherical spatiotemporal relationships among 360° video elements, tackling the challenges in existing video representations for handling 360° videos.
- We develop a learning framework for NPR, featuring innovative loss functions and a bi-directional mapping scheme that bridges a neural panoramic space with its original video space.
- We design a novel 360° video editing approach capable of generating spatially and temporally consistent results.

2 Related Work

360° Video Understanding. The progress of next-generation MR and VR applications centered around 360° media hinges on advancing content understanding. Some recent efforts focused on proposing novel deep architectures and learning schemes to encode the spatial features of spherical pixels. Among these, temporal analysis is crucial in 360° videos for capturing omnidirectional motion, distinguishing them from static images. Despite the effectiveness of 2D optical flow estimation techniques [43, 45], extending their applicability to wide field-of-view videos poses a persistent challenge. Bhandari et al. [5] employed 2D flow projection onto equirectangular images as pseudo-ground truth for training, while Yuan et al. [55] employed tangent images but faced boundary discontinuities. To tackle discontinuities, Li et al. [22] introduced a framework combining equirectangular, cube-padding, and cylindrical projections to integrate motion information. However, the lack of a real-world 360° optical flow dataset and the limitation to adjacent frame motion restricts flow-based editing. Efforts have been made in semantic understanding and 3D information extraction from 360° data, including depth estimation [1, 23, 47], normal estimation [26], semantic segmentation [31, 53], and object detection [50]. However, compact 360° video representation learning remains underexplored.

Neural Video Representation. Using layered neural mapping designs that bridge abstract or parametric proxies with video frames, Layered Neural Atlas [17] and Deformable Sprites [54] show promise in 2D planar video editing. This field has expanded to applications such as text-driven video stylization [3, 28], deflickering [20], and face video manipulation [27]. By using MLPs to translate coordinates into color and volume density, Neural Radiance Fields (NeRF) [32] advanced neural rendering and novel view synthesis, inspiring numerous studies to address its limitations from various aspects [4, 34, 36, 39]. Most recently, the 3D Gaussian Splatting technique [18, 29] has been proposed to enable real-time view synthesis. To provide such neural representations with editability, various techniques have been introduced to enable modifications to learned volumetric representations [6, 56, 60]. Existing NeRF-based editing methods are primarily designed for static scenes learned from 2D planar imagery. They fall short in representing 360° inputs due to their lack of consideration for spherical properties. To tackle this issue, Choi et al. [9] introduced EgoNeRF, a specialized neural representation for static scenes in 360° videos. Deng et al. [11] presented an egocentric sampling approach to construct radiance fields, enhancing foveated rendering quality in VR applications. 360-GS [2] and OmniGS [21], have expanded real-time omnidirectional rendering applications. However, they face challenges with dynamic objects and lack editibility in 360° videos. OmniLocalRF [10] focuses on reconstructing the inherent static scene in 360° videos by removing dynamic objects. In contrast, we aim to fully capture the dynamic environment in 360° videos, representing both static and dynamic elements, thereby facilitating 360° video editing.

360° Content Manipulation. Zhu et al. [61] pioneered an inpainting method for 360° images, using structure-rectifying warping to address distortions and applying 2D completion technique [15] to fill gaps. Xu et al. [49] advanced this by considering spherical geometry and iteratively restoring missing pixels and motion information. Video stabilization techniques [19, 44] have enhanced visual comfort by reducing distortion in equirectangular representations. Zhang et al. [58, 59] introduced edit propagation methods for global color changes in 360° panoramas. Li et al. [24] integrated bullet comments into 360° videos, and Wong et al. [48] devised a view-adaptive asymmetric detail enhancement solution. To employ the benefits of 360° data, neural illumination representations [12, 38] provide solutions based on implicit neural models, reconstructing light information by encoding the directions of spherical pixels to enhance inverse rendering. Unlike these neural illumination models, our method focuses on learning interpretable spherical content layers by encoding spatiotemporal positions, thereby expressing the detailed appearance of dynamic 360° videos. While these existing 360° content manipulating approaches have demonstrated promise in specific tasks, they fall short of providing a representation that maintains temporal coherence across all elements of a dynamic scene, which is essential for supporting consistent 360° video editing tasks.

3 METHODOLOGY

3.1 Overview

Our objective is to develop a specialized, editable representation for 360° videos, which is capable of simplifying 360° video manipulation problems, such as tracking specific areas or objects, video editing, and video completion. Unlike traditional 2D planar videos, 360° videos, captured via panoramic cameras or stitched from multiple planar captures, are distributed over a spherical surface and intended for VR head-mounted displays. Editing these spherical videos is more challenging due to the inability to view the entire scene simultaneously in its native format.

Various projection methods enable mapping 360° videos onto one or multiple planes. Equirectangular projection stands out as it unwraps the entire scene into a rectangle, enabling the application of 2D planar video editing techniques. However, this method introduces significant distortion, particularly near the two poles, due to the non-developable nature of spherical data, leading to inaccurate deformations compared to the original spherical format. Hence, exploring learning-based representations on the unit spherical surface shows promise for 360° videos. Such spherical data, characterized by non-grid structure and non-uniform sampling, favor a coordinate-based representation over grid-like approximations [54]. MLPs, adept at handling coordinate-based inputs, are flexible in modeling non-rigid motion in videos, as evidenced by recent studies [17, 32].

Motivated by these, we propose a spherical neural representation NPR, using MLPs to create a continuous, global representation of 360° video frames, encoding both temporal information (timestamp t) and spatial information (spherical positions and their color mapping). This model results in a compact and efficient embedding from the original 4D space (time and 3D spherical positions) to an editable canonical 3D space, preserving the spherical properties of the captured content while enabling direct visualization for user-centric editing. This addresses the crucial challenge of maintaining content correlation over time. Our framework is summarized in fig. 1.

3.2 Neural Panoramic Representation

Given an input 360° video, we represent the spatiotemporal position of each pixel as a 4D point on a unit spherical surface. We obtain the equirectangular coordinates (θ, ϕ) of one video frame and convert them to their 3D Cartesian coordinates (x, y, z) as follows:

$$x = \cos\phi\cos\theta,\ y = \cos\phi\sin\theta,\ z = \sin\phi, \tag{1}$$

where x, y, and z are each bounded within the interval $[-1, 1]$, due to their distribution on a unit sphere. To accommodate coordinate-based computations, it is essential to discretize the continuous video signal into an $H \times W$ pixel grid for each frame. Accordingly, we set θ within $[-\pi, \pi)$ and ϕ within $[-\frac{\pi}{2}, \frac{\pi}{2})$. This arrangement strategically excludes the boundary values $\theta = \pi$ and $\phi = \frac{\pi}{2}$, thereby preserving the central points $\theta = 0$ and $\phi = 0$ that are crucial for accurate representation in the equirectangular format. A timestamp t is added as the fourth dimension, resulting in 4D spatiotemporal coordinates (x, y, z, t), where t ranges from 1 to N, with N representing the total frame count. These coordinates are structured into a matrix with dimensions $[4, H \times W \times N]$, serving as the input for further training, where H and W denote the height and width of each frame.

Our neural panoramic representation (NPR) employs several MLPs to map these 4D spatiotemporal coordinates to a canonical 3D space with multiple layers for different scene elements, like the background or specific foreground objects. Through this method, any point (x, y, z, t) in the given 360° video will be mapped into its corresponding canonical point $(\hat{x}_l, \hat{y}_l, \hat{z}_l)(l \in \mathbb{Z},\ l \geq 0)$, with layer number l determined by pre-segmentation. We also generate the opacity $\hat{M}^l(x, y, z, t)$ for each pixel in each layer to represent the visible content of the background and foreground objects.

Given the distinct motion patterns between the background and foregrounds, the NPR adopts a layered implicit representation, implemented via the neural network F formulated as:

$$\hat{\mathbf{c}} = F_\Theta(\mathbf{x}, \{M^l_\mathbf{x}\}_{l=0,1,2,\ldots}) \tag{2}$$

$\mathbf{x}$ represents the 4D spatiotemporal coordinates (x, y, z, t). $M^l_\mathbf{x}$ is the initial mask for $\mathbf{x}$ at layer l, set by pre-segmentation to reflect its initial layer affiliation. This value can be refined during training to compensate for any initial segmentation imperfections. $\hat{\mathbf{c}}$ denotes the output color, and Θ is the set of parameters in F. To facilitate the description, we assume a two-layer setup: the 0-th layer for the background (b), and another for the foregrounds (f). This setup efficiently addresses most real-world scenarios, accommodating scenes with one foreground object or multiple objects exhibiting moderate motion variations. In cases with multiple dynamic objects, we combine them into a single foreground layer for streamlined learning, balancing computational efficiency with performance. When specific editing operations for distinct objects are needed, more foreground layers can be added with the same architecture.

Adhering to the assumed two-layer setup in our NPR framework, network F employs two groups of MLP-based modules for reconstructing the foreground and background of 360° video frames. Within this dual-layer structure, the first MLP in each layer—$\mathbb{M}_{fp}$ for the foreground and $\mathbb{M}_{bp}$ for the background—maps the 4D spatiotemporal coordinates to a 3D position $\hat{\mathbf{x}} = (\hat{x}, \hat{y}, \hat{z})$ in the neural spherical space. The second MLP in each group, $\mathbb{M}_{fc}$ for the foreground and $\mathbb{M}_{bc}$ for the background, then recovers the respective color information of $\hat{\mathbf{x}}$. The color reconstruction is formulated as:

$$\hat{\mathbf{c}}_f = \mathbb{M}_{fc}(\hat{\mathbf{x}}),\quad \hat{\mathbf{c}}_b = \mathbb{M}_{bc}(\hat{\mathbf{x}}) \tag{3}$$

Our model uses an opacity prediction module $\mathbb{M}_\alpha$ for seamlessly merging foreground and background. It determines optimal opacity value at each 4D point $\mathbf{x}$, enabling the final color reconstruction:

$$\hat{\mathbf{c}} = (1 - \hat{\alpha})\hat{\mathbf{c}}_b + \hat{\alpha}\hat{\mathbf{c}}_f, \tag{4}$$

where $\hat{\mathbf{c}}$ denotes the reconstructed pixel color, with $\hat{\mathbf{c}}_b$ and $\hat{\mathbf{c}}_f$ being the predicted background and foreground colors. Pixel-wise opacity value $\hat{\alpha}$, learned in the foreground layer $\hat{M}^f(x, y, z, t)$, is refined based on each pixel's opacity value α on the pre-computed frame-wise semantic masks using a SOTA video object segmentation method [8]. Our learned NPRs establish dependable relationships among spherical pixels, enhancing the accuracy of 360° video representation and yielding more precise object motion estimations than 2D layered planar neural representations [17, 54].

3.3 Bi-directional Position Mapping for Modeling Global Panoramic Motion

The above modules learn a one-directional mapping from 360° video to neural panoramic space, namely $f(x, y, z, t) \rightarrow (\hat{x}, \hat{y}, \hat{z})$, providing neural panoramic positions at a specific time but cannot predict their position movements in other frames. For example, a point P in a real scene is located at (x_1, y_1, z_1) at time t and (x_2, y_2, z_2) at time $t+1$. To obtain the corresponding neural panoramic coordinates $(\hat{x}, \hat{y}, \hat{z})$, both t and $t+1$ must be provided. This process does not allow direct determination of positions in the different 360° video frames from their implicit space positions. To address this and ease editing tasks, we introduce bi-directional position mapping by adding an additional pair of MLPs $\mathbb{B}_{fp}$ and $\mathbb{B}_{bp}$ for foreground and background, to get the real video position of point P for any time t. It implements the backward mapping $g(\hat{x}, \hat{y}, \hat{z}, t) \rightarrow (x, y, z, t)$, from the learned neural panoramic space back to the original video. We refer to $\mathbb{M}_{fp}$ and $\mathbb{M}_{bp}$ as the forward position mapping module, and $\mathbb{B}_{fp}$ and $\mathbb{B}_{bp}$ as the backward position mapping module. The backward mappings are learned by solving the position-recovery problem for both the foreground and background layers:

$$\mathbf{x}' = \mathbb{B}_{fp}(\mathbb{M}_{fp}(\mathbf{x}), t),\ \text{or}\ \mathbf{x}' = \mathbb{B}_{bp}(\mathbb{M}_{bp}(\mathbf{x}), t) \tag{5}$$

This design enables global tracking for each pixel in any given 360° video, ensuring per-pixel correspondence across all frames. As a result, users can perform edits directly on a specific frame

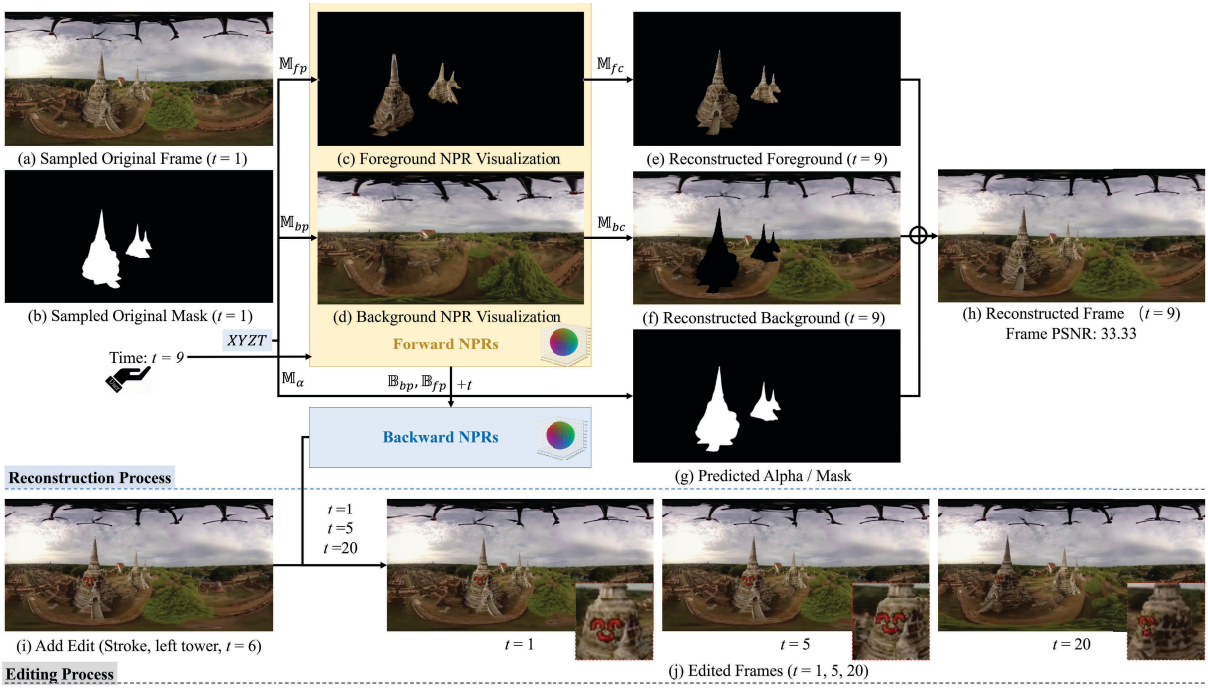


Figure 2: Illustration of our framework for the reconstruction and editing process. The user provides any time t, for example, $t = 9$ as shown in the figure, the learned forward NPRs can reconstruct the corresponding video frame. In editing, by using the learned backward NPRs, we generate the edited frames at any time t the user likes respecting their edits, such as adding a hand-drawn stroke (smile emoji) in this example.

rather than on an implicit content layer derived from a learned neural panoramic representation. This module can automatically identify the positions affected by edits on the implicit content layer, and then map these changes to the corresponding areas in all other frames.

3.4 Training for Neural Panoramic Representation

Our training is self-supervised, involving forward and backward mapping in two stages. In the forward mapping training, the primary constraint is the minimization of reconstruction loss, defined as:

$$\mathfrak{L}_{rec} = \sum_{s \in S} ||\mathbf{c}_s - \hat{\mathbf{c}}_s||_2^2, \tag{6}$$

$s \in S$ is a point randomly sampled from the video, where S is the whole set of sampled points per iteration. c_s and $\hat{c}_s$ are the original colors in the input video and reconstructed colors, respectively. Sole reliance on reconstruction loss is inadequate for effective inter-pixel spatiotemporal relationships in spherical space, as training can be dominated by target colors. For example, if two distinct points A and B in one frame share a color, they might map to the same point in the neural panoramic space. This does not hinder reconstruction but limits editability, as any edit to A will cause the same edit to B. To avoid this, we design the following two constraints: spherical neighborhood consistency loss and unit spherical regularization.

Spherical Neighborhood Consistency (SNC) loss. For each point in the scene described by the given 360° video, this loss aims to constrain the effective one-to-one mapping between the original space and neural panoramic space by preserving the relative position with its eight-connected neighbors. In a spherical space, it is harder to constrain the distance between any two points than on a plane, but the direction vector from one point to another is easy to get by subtracting their spatial coordinates. Thus, we use the spatial direction vectors between each sampled point and its eight-connected neighbors as the position self-supervision to constrain the relative position in the target space. We define the SNC loss as:

$$\mathfrak{L}_{snc} = \sum_{i \in I} ||(P - P_i) - (\hat{P} - \hat{P}_i)||_2^2, \tag{7}$$

where $I = \{0, 1, 2, ..., 7\}$ is the index set for neighbors, $P \in \mathbb{R}^{|S| \times 4}$ is the sampled S positions in the original input video and each position is a 4D spatiotemporal coordinate, P_i is one of the neighbors of P in the original input video, $\hat{P} \in \mathbb{R}^{|S| \times 3}$ and $\hat{P}_i$ are the predicted positions in the neural panoramic space.

Unit Spherical Regularization (USR). The predicted 3D coordinates of any one implicit content layer are expected to distribute on a unit spherical surface, rather than deviate from this surface by any significant amount. As mentioned in section 3.2, the input 4D coordinates of each layer are on a unit spherical surface and in the range $[-1, 1]$. We could force the forward mapping outputs to be in $[-1, 1]$ by setting the activation function of the output layer. However, points with the 3D Cartesian format in $[-1, 1]^3$ do not distribute on a unit spherical surface, some lie inside the sphere body and some outside. To this end, we introduce a unit spherical regularization to encourage the learned implicit 3D coordinates from MLPs to lie on a unit spherical surface, which is defined by:

$$\mathfrak{L}_{usr} = \sum_{s \in S} \left\| \sqrt{(\hat{P}_{s1})^2 + (\hat{P}_{s2})^2 + (\hat{P}_{s3})^2} - 1 \right\|_2^2, \tag{8}$$

where $\hat{P}_{s1}$, $\hat{P}_{s2}$, and $\hat{P}_{s3}$ are the neural panoramic space axes, akin to the standard 3D Cartesian system's x-, y-, and z-axes.

The predicted opacity value of the foreground is crucial for video reconstruction. To ensure consistency with input masks over the frames, we introduce alpha loss for self-supervision, formulated as:

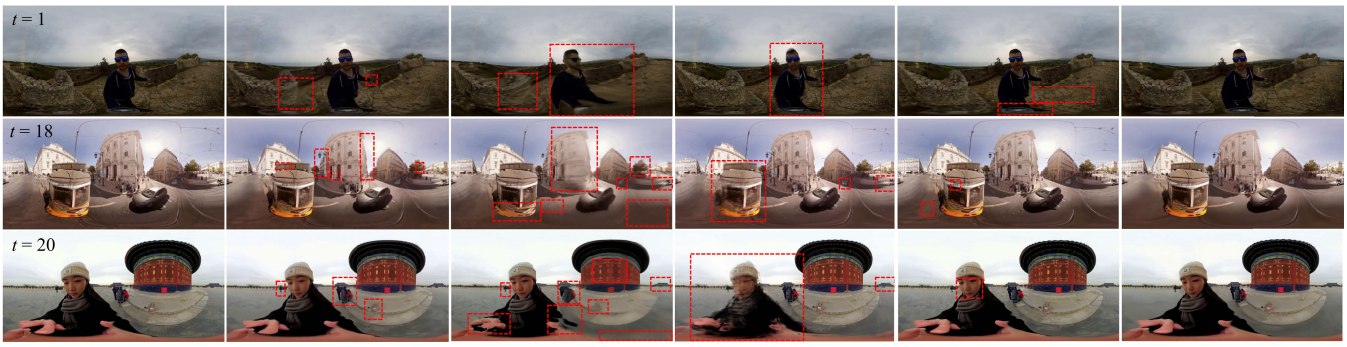
$$\mathfrak{L}_{alpha} = -\sum_{s \in S} \alpha_s \cdot \log(\hat{\alpha}_s) + (1 - \alpha_s) \cdot \log(1 - \hat{\alpha}_s), \tag{9}$$

Initialization. We implement a warm-start to initialize $\mathbb{M}_{fp}$ and $\mathbb{M}_{bp}$ with the spherical positions of each video frame within the 4D spatiotemporal coordinates, as mentioned in section 3.2. This approach strategically positions each NPR on a unit spherical surface, providing a standardized starting point for training. Additionally, we initialize $\mathbb{M}_{\alpha}$ close to the pre-computed alpha values, facilitating the effective learning of NPRs for distinct content layers.

Detail Enhancement via Positional Encoding. Considering that deep networks are typically biased towards learning lower-frequency

Methods	LNA			Deformable Sprites			EgoNeRF			EgoNeRF+t			Ours (NPR)		
Metrics	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Walking Boy	36.21	0.923	0.141	28.75	0.819	0.213	31.72	0.905	0.267	35.05	0.923	0.231	**38.88**	**0.955**	**0.091**
Walking Girl	32.35	0.922	0.138	23.45	0.778	0.234	27.52	0.897	0.268	34.09	0.932	0.181	**35.38**	**0.934**	**0.112**
Classroom	37.32	0.970	0.091	31.68	0.946	0.089	*	*	*	*	*	*	**40.01**	**0.974**	**0.068**
Farm	36.73	0.938	0.109	31.00	0.912	0.099	*	*	*	*	*	*	**39.04**	**0.968**	**0.054**
Towers	30.55	0.886	0.174	25.49	0.772	0.276	27.70	0.901	0.247	32.79	0.888	0.214	**33.73**	**0.924**	**0.114**
Bus Stop	36.04	0.964	0.071	29.49	0.945	0.072	*	*	*	*	*	*	**39.34**	**0.973**	**0.048**
Tram	31.18	0.903	0.177	22.90	0.742	0.317	26.53	0.865	0.312	31.41	0.916	0.221	**32.93**	**0.917**	**0.149**
Motorcycle	29.59	0.850	0.234	23.03	0.656	0.396	27.55	0.844	0.335	**31.05**	**0.902**	0.255	30.13	0.852	**0.224**
Elephant	35.41	0.957	0.099	30.17	0.884	0.112	*	*	*	*	*	*	**37.58**	**0.969**	**0.076**
Average	33.93	0.924	0.137	27.33	0.828	0.201	28.20	0.882	0.286	32.88	0.912	0.220	**36.34**	**0.941**	**0.104**

Table 1: Average PSNR, SSIM, and LPIPS for each video are presented, with the last row indicating the overall averages for the entire dataset. For some videos, EgoNeRF cannot reconstruct the scene due to the static camera position, hindering camera pose extraction by OpenMVG.



(a) Original Frame (b) LNA (c) Deformable Sprites (d) EgoNeRF (e) EgoNeRF+t (f) Ours

Figure 3: Qualitative comparisons of reconstruction on several real-world 360° videos. Refer to our supplementary video for the full results.

functions [35], we employ positional encoding (PE) techniques to map input positions to a high-dimensional space, ensuring that even closely situated positions retain unique characteristics. This strategy enables the model to differentiate positions while recognizing their proximity or similarity, thereby aiding in the capture of high-frequency details. Given the unique 4D coordinates of each input spherical pixel, PE is not necessary for the position mapping MLPs, $\mathbb{M}_{fp}$ and $\mathbb{M}_{bp}$. However, since distinct spherical positions might display similar color or opacity values, PE is crucial for the color mapping MLPs, $\mathbb{M}_{fc}$ and $\mathbb{M}_{bc}$, and the alpha prediction MLP, $\mathbb{M}_{\alpha}$. This application of PE enhances the capture of high-frequency appearance details influenced by subtle differences among implicit spherical positions, preventing an overly smooth appearance. To balance efficiency and performance, we opt for a fixed sinusoidal positional encoding function defined as $PE(\mathbf{x}) = \sin(2^d\pi\mathbf{x}), \cos(2^d\pi\mathbf{x}), d = 0, ..., D$, which avoids the need for additional learnable parameters and mitigates the higher computational demands and parameter sensitivity associated with learnable functions or wavelet transformations.

Bi-directional Training Scheme. Our model learns two mappings: forward and backward. The forward mapping encodes 4D spatiotemporal positions into 3D implicit positions within a canonical content space through self-supervision and regularization, unaffected by the backward mapping. The backward mapping converts these 3D positions back to the original video space. To avoid interference and unnecessary complications, we train these mappings sequentially—first forward, then backward—instead of simultaneously.

3.5 Layered Neural Panoramic Editing

Our framework of the reconstruction and editing process is shown in fig. 2. Post-training, we obtain multiple learned MLPs forming a dynamic representation for 360° video reconstruction and editing at any time. Unlike previous neural video representations [17, 54] that rely on abstract texture maps, making meaningful edits challenging due to the lack of actual appearance, our model enables direct editing on original frames using the learned NPRs, offering a more practical and user-friendly approach for 360° video editing. Leveraging bi-directional position mapping, as mentioned in section 3.3, users can choose any frame at time t (e.g., $t = 6$ in the figure) for precise pixel-level editing. This involves mapping edits to implicit positions via forward NPRs ($\mathbb{M}_{fp}$ and $\mathbb{M}_{bp}$) and then adjusting corresponding pixels across all frames using backward NPRs ($\mathbb{B}_{fp}$ and $\mathbb{B}_{bp}$). After specifying whether the edits should be applied to the foreground or background, users can perform a variety of edits, such as adding strokes (e.g., a hand-drawn smile emoji in the figure), applying graphic overlays, and removing elements. The learned NPRs will identify the required editing locations, allowing seamless integration of these changes throughout the video sequence.

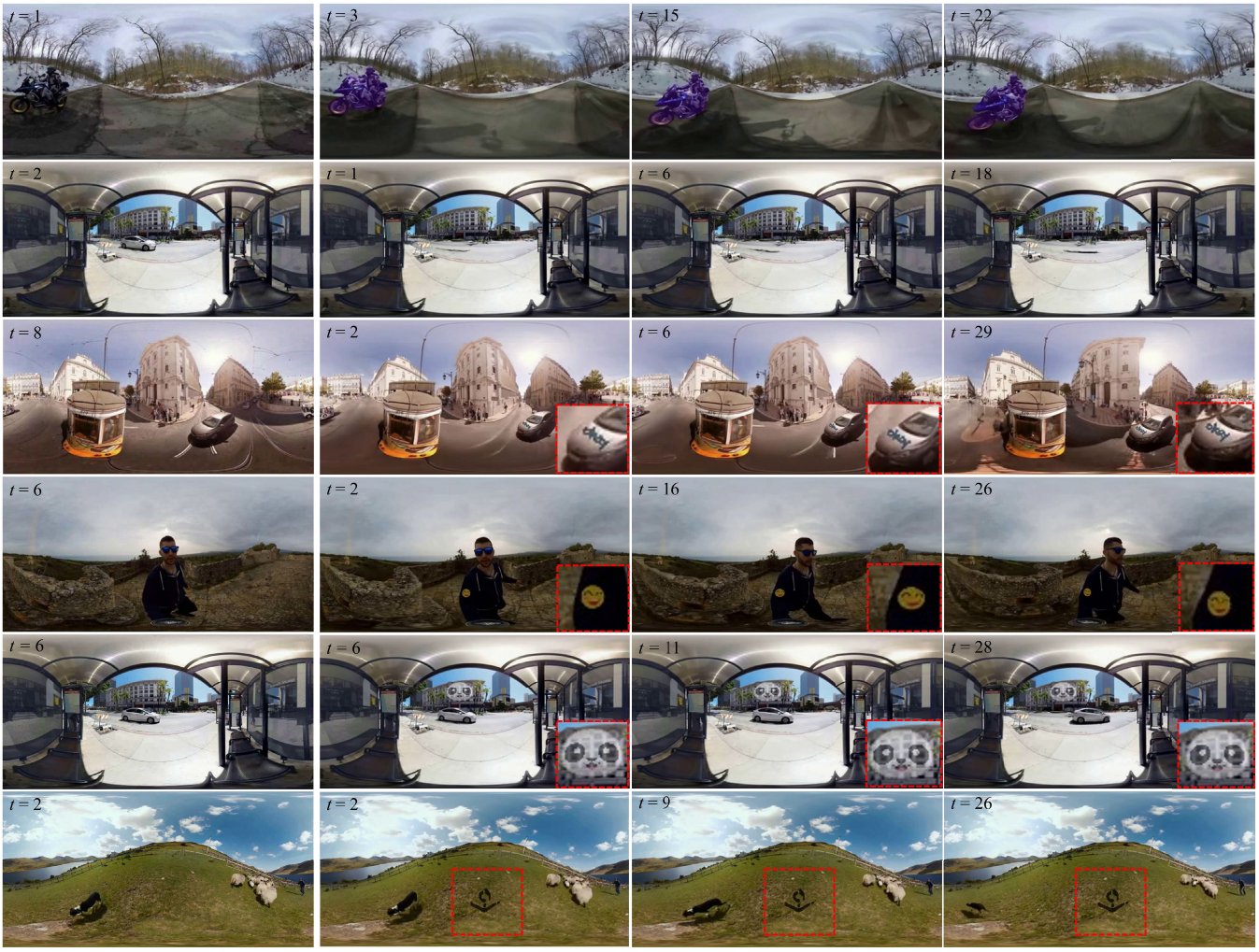
4 Experiments

4.1 Implementation Details

Configurations. Our framework, depicted in fig. 1, includes several MLPs, each comprising 8 linear layers. Each hidden layer is configured with 256 channels. We assign the dimensions D of positional encoding as follows: 10 for $\mathbb{M}_{fc}$ and $\mathbb{M}_{bc}$, and 5 for $\mathbb{M}_{\alpha}$. The empirical trade-off coefficients of our losses are: $w_{recon} = 1.0$, $w_{snc} = 0.1$, $w_{usr} = 10.0$, and $w_{alpha} = 10.0$. All experiments described in this paper were conducted on an NVIDIA GeForce RTX 3080 GPU.

Dataset. Due to the lack of publicly accessible real-world 360° video datasets featuring significant object and camera motion, we developed a new dataset to evaluate our approach. This dataset consists of nine real-world 360° videos captured by a fixed or moving spherical camera, with each video containing 30 frames.

Baseline Methods. We evaluated the effectiveness of our method in reconstructing 360° videos by comparing it with SOTA neural representations: Layered Neural Atlases (LNA) [17], Deformable Sprites [54], and EgoNeRF [9]. LNA learns an implicit representation to map planar video frames onto a 2D plane in neural space, while Deformable Sprites uses learnable B-Splines to model non-



(a) Original Frame (b) Edited Video Frames

Figure 4: Results of 360° video editing: Each row shows an original frame on the left and three sampled frames from edited videos on the right.

rigid transformations in planar videos. EgoNeRF, designed with yin-yang feature grids in spherical coordinates, represents static scenes in 360° videos. We also compared our method with EgoNeRF+t, a straightforward extension of EgoNeRF for dynamic scenes, by directly incorporating a time dimension into yin-yang feature grids.

Evaluation Metrics. We selected well-recognized visual quality metrics, including PSNR, SSIM, and LPIPS, for the quantitative evaluation of our reconstruction performance against baseline methods. For qualitative assessment, we displayed reconstructed frames. In the context of 360° video editing, where no established metrics exist, we demonstrated our consistent editing performance with various edited videos alongside their original frames.

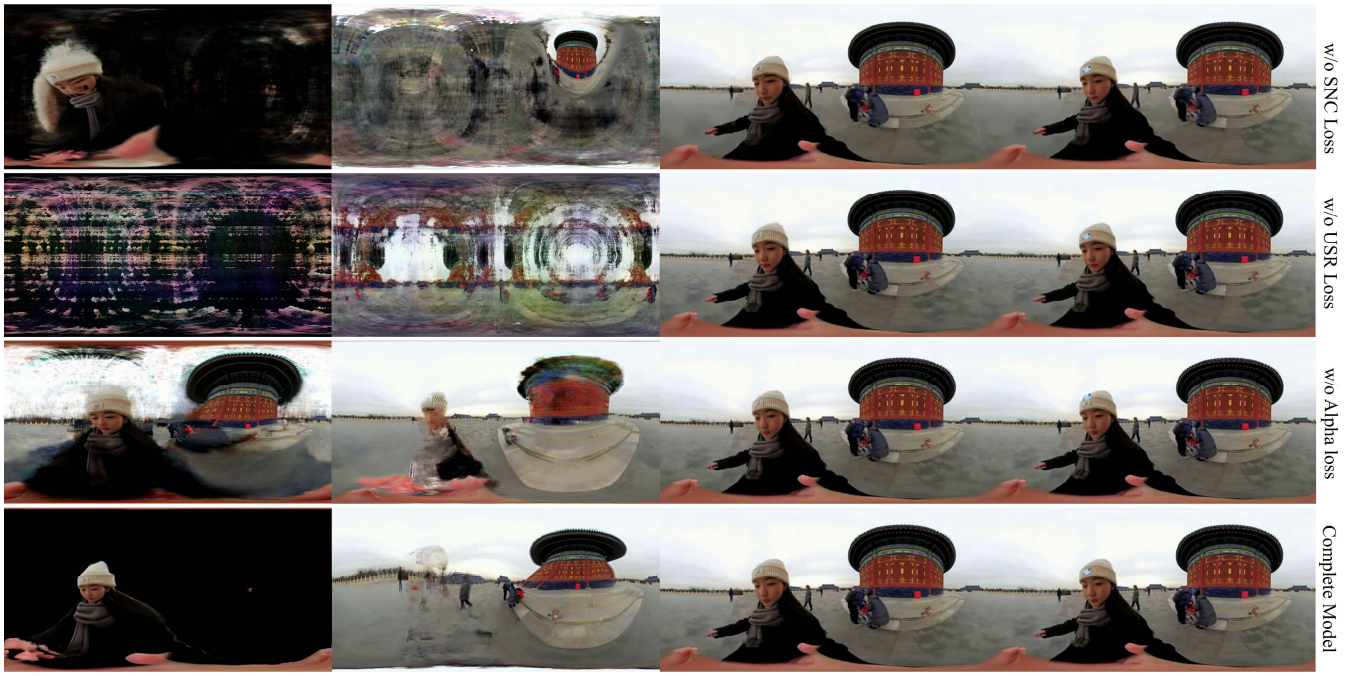
4.2 Comparison with Baseline Methods

Since two of the baseline methods, LNA [17] and Deformable Sprites [54], are designed for 2D planar videos, we adapted these models to handle 360° video inputs using equirectangular projection (ERP). This adjustment allows us to treat ERP videos as 2D planar formats through the use of image coordinates. To ensure a fair comparison, we aligned our model's training parameters with those of LNA, involving 10,000 sample points per iteration for a total of 300,000 iterations. Another baseline method, EgoNeRF, tailored specifically for 360° videos, concentrates on 3D scene representation. However, it cannot separate foregrounds from backgrounds using layered representations due to its dependency on precise camera pose information. This limitation is exacerbated by the inadequate textural features of separated foregrounds, which impede effective pose extraction. For EgoNeRF's evaluation, we adhered to its native setup and utilized OpenMVG [33] to extract spherical camera poses from our dataset, as detailed in its foundational paper. For EgoNeRF+t, a dynamic extension of EgoNeRF, we implemented it by building upon EgoNeRF with an additional time dimension.

The quantitative and qualitative comparison results for the 360° video reconstruction task are presented in table 1 and fig. 3. Our method consistently achieves superior average metrics across the reconstructed videos, enhancing visual details such as sharper wall textures, defined building edges, intact street lamp structures, and clearer distant objects. Notably, our approach excels in modeling subtle motions in both foreground and background elements, effectively capturing reflections, shadows, and other real-world effects in dynamic scenes. In the first example shown in fig. 3, our proposed model accurately captures and globally represents the motion of a walking boy, successfully reconstructing appearance details in each frame. While EgoNeRF performs well in reconstructing static backgrounds, it fails to accurately capture moving objects in all examples, as it treats scenes as static and lacks effective mechanisms for handling foreground motion. EgoNeR+t mitigates the blurriness in dynamic regions compared to EgoNeRF, but it exhibits grid-like blurring on less textured dynamic parts, such as the human face, as demonstrated in the third example in fig. 3.

(a) Edit on an Original Frame (b) Intermediate Result (c) Edited Video Frames ($t = 1, 23$)

Figure 5: An intermediate result of adding a red stroke to the foreground (a car): (a) A specific frame showing the stroke edit applied to the moving car. (b) The intermediate result on the learned canonical content layer, foreground NPR. (c) Edited frames based on the user's edit.



(a) Foreground NPR (b) Background NPR (c) Reconstructed Video Frame (d) Edited Video Frame

Figure 6: Qualitative results of an ablation study on a real-world 360° video (Walking Girl): Effects of ablating each designed loss term.

4.3 Results of 360° Video Editing

Our proposed neural panoramic representation (NPR) method models the spatiotemporal relationships among pixels in 360° videos using neural spherical layers, thereby facilitating consistent and flexible editing operations. Fig. 2 and fig. 4 show several video editing results of our method, with various effects, including adding strokes, applying graphic overlays, changing the appearance of certain interest areas, and removing the foreground (video completion).

The editing process illustrated in fig. 2 demonstrates the seamless integration of a hand-drawn cartoon smile emoji onto a dynamic foreground object—the left tower—ensuring an exact fit. Our method excels in editing both dynamic foreground objects, which pose challenges due to their variable dynamics, and static background regions. This versatility is showcased in fig. 4, where each row highlights a distinct editing achievement: (1) The first row shows a uniform enhancement by applying a purple highlight to both a rider and his motorcycle, maintaining consistency throughout the video. (2) The second row demonstrates our ability to make a car disappear from the road while realistically reconstructing the occluded background. (3) The third row features the consistent addition of the text stroke "okay" onto a moving car, aligned with the car throughout the video. (4) The fourth row presents the effective integration of graffiti onto a walking boy, demonstrating dynamic scene adaptation. (5) The fifth and sixth rows highlight our precise content mapping in adding graffiti to the background, resulting in believable and stable edits. These examples underline our method's success in delivering meaningful, spatiotemporally consistent edits in 360° video editing.

To demonstrate the effectiveness of our proposed bi-directional mapping module, we present an intermediate result in one of the learned implicit content layers (foreground NPR on the video "Bus Stop") in fig. 5. This figure visualizes the points on the foreground NPR affected by the user's edits in the sixth frame. It clearly shows that an added red stroke is accurately mapped across the corresponding affected pixels in other frames.

4.4 Ablation Study

We evaluated the impact of our designed loss terms within the proposed framework by observing their effects on reconstruction and editing performance during an ablation study (fig. 6 and table 2). The omission of the Spherical Neighborhood Consistency (SNC) loss, detailed in the first row, resulted in noticeable dislocation issues in the background NPR, especially in large areas like the ground. This occurred because the implicit neural panoramic space allocated fewer points to these regions, resulting in a loss of inter-pixel spatial relationships. In the second row, we noted that removing the Unit Spherical Regularization (USR) loss caused dynamic objects in the reconstructions to appear as if they were behind transparent glass or showed disturbing depth of field effects. This observation suggests that some points in the implicit space deviated from the unit spherical surface, leading to inaccurately layered spherical representations. The third row demonstrates the consequences of excluding the alpha loss, which resulted in an incorrect foreground mask. Pixels that should have been identified as background were incorrectly included

(a) Original Frame (b) w/o PE (c) Wavelet-based PE (d) Learnable Sine PE (e) Fixed Sinusoidal PE

Figure 7: Qualitative results of an ablation study on a real-world 360° video (Walking Boy): Effects of absence and different PE functions.

as part of the transparent foreground, which could confuse users during video editing. Although eliminating the alpha loss led to higher metrics in table 2, retaining this loss is essential. It plays a crucial role in accurately separating spherical content layers, enabling clearer and more reasonable edits in 360° videos. Furthermore, it is crucial to note the significant risk of overfitting associated with the removal of any of these loss terms, especially the alpha loss. While the impact on reconstruction quality may be minimal in scenes primarily composed of sky, grass, or water, the overall consistency in 360° video editing performance could be adversely affected.

Ablating Loss	PSNR ↑	SSIM ↑	LPIPS ↓
w/o SNC Loss	31.86	0.908	0.151
w/o USR Loss	34.68	0.933	0.113
w/o Alpha Loss	37.03	0.941	0.105
Complete Model	35.38	0.934	0.112

Table 2: Quantitative results of an ablation study on designed loss terms using all nine real-world 360° videos in our dataset.

In another ablation study, we assessed the impact of different positional encoding (PE) techniques, including the absence of PE, to clarify our choice. We explored three types: a fixed sinusoidal PE function as outlined in section 3.4, a wavelet-based PE applying a 1D discrete wavelet transform to each dimension of our 4D inputs, and a learnable periodic activation function, $\sin(\omega_i \times x)$ (cited in [42]). The fixed sinusoidal and wavelet-based PEs were applied before training, while the learnable PE was adjusted during training. Results in table 3 show that the fixed sinusoidal PE not only outperformed other methods across all metrics with only a modest increase in training time but also provided the most detailed appearances in fig. 7, capturing high-frequency details more effectively than other approaches. While the learnable sine PE offered acceptable results with similar training time, it was the second-best option. In contrast, the wavelet-based PE resulted in the poorest visual quality, and the absence of any PE led to overly smooth regions, emphasizing the need for PE to preserve details in complex images. Consequently, the fixed sinusoidal PE emerges as the superior choice for enhancing the model's capability to handle detailed appearances.

PE Functions	PSNR ↑	SSIM ↑	LPIPS ↓	Train Time ↓
w/o PE	29.88	0.848	0.239	**20 hours**
Wavelet-based	26.30	0.755	0.313	1 week
Learnable Sine	35.28	0.936	0.108	22 hours
Fixed Sinusoidal	**36.34**	**0.941**	**0.104**	24 hours

Table 3: Quantitative results of an ablation study on PE functions.

5 Discussion

Model Scalability for Higher Resolution Videos. The results in this paper are derived from models trained on 480×240 360° videos. The alpha prediction MLP $\mathbb{M}_\alpha$ comprises 405,505 parameters. The position and color mapping MLPs for each content layer have 397,839 and 426,679 parameters, respectively. Our method demonstrates notable scalability to higher resolutions; for instance, training on a 960×480 video requires quadrupling the time, scaling approximately linearly with the increase in pixel count. The main advantage of our layer representation lies in facilitating intuitive user edits by specifying the content to be edited at different layers. Even with imperfect initial foreground segmentation, our approach consistently achieves high-quality reconstruction results. Tests with alternative segmentation methods [14], indicate similar performance, suggesting a reduced dependency on precise foreground segmentation. In our preliminary experiments, we observed that the MLPs effectively represent and reconstruct 360° videos without needing to segment the video into layers, thanks to their capability to fit continuous functions and model pixel-wise motion. However, excessive motion components within a single layer necessitate additional network layers, significantly elevating computational costs. Separating the video into layers not only enhances the manageability of motion components but also reduces overall training time.

Achieving Global Motion: Advantages Over Optical Flow. We discuss the differences between our method and optical flow approaches in terms of modeling the motion in the dynamic scene, to highlight our merits. Optical flow describes the motion between two adjacent frames, which is a kind of local correspondence in the temporal domain, and thus it could not support the holistic understanding of the scene of the given video. Compared to that, we leverage MLPs to record the global correspondence that represents a holistic spatiotemporal relationship of the dynamic scene in the given 360° video, describing motions over time globally. The baseline methods we compared, LNA [17] and Deformable Sprites [54], already utilize optical flow as guidance, showing inferior results to ours due to the accumulated error in frame-to-frame optical flow. Additionally, state-of-the-art optical flow models [22,41,45] struggle to provide accurate results on various real-world 360° videos.

6 Conclusion

We propose the first neural representation for 360° videos that supports spatiotemporally consistent, immersive pixel-level editing. Distinct from existing methods tailored for 2D planar videos, our approach empowers neural networks to effectively capture spherical appearances. The innovative bi-directional mapping facilitates global content tracking in dynamic panoramic scenes. This advancement enables more convenient 360° video editing by providing an interpretable content representation, allowing users to intuitively specify regions or objects for customized modification. It shows promise for a wide range of users, regardless of their VR or video editing experience. Experimental results on real-world 360° videos validate our model's effectiveness, showcasing superior reconstruction and achieving spatiotemporally consistent 360° video editing.

Future Work. Our method specializes in pixel-wise appearance edits for panoramic videos with dynamic objects, however, it currently lacks the capability to model the temporal information of object motion. In the future, we plan to integrate a temporal modeling component to more effectively encode motion information within our video representation. This enhancement will expand the applicability of our method, enabling functionalities such as video prediction and timeline editing.

Acknowledgments

This work was supported by Marsden Fund Council managed by the Royal Society of New Zealand under Grant MFP-20-VUW-180 and the Royal Society (UK) under Grant No. IES\R1\180126.

REFERENCES

[1] H. Ai, Z. Cao, Y.-P. Cao, Y. Shan, and L. Wang. HRDFuse: Monocular 360° depth estimation by collaboratively learning holistic-with-regional depth distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13273–13282, 2023.

[2] J. Bai, L. Huang, J. Guo, W. Gong, Y. Li, and Y. Guo. 360-GS: Layout-guided panoramic Gaussian splatting for indoor roaming. *arXiv preprint arXiv:2402.00763*, 2024.

[3] O. Bar-Tal, D. Ofri-Amar, R. Fridman, Y. Kasten, and T. Dekel. Text2LIVE: Text-driven layered image and video editing. In *European conference on computer vision*, pp. 707–723. Springer, 2022.

[4] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5470–5479, 2022.

[5] K. Bhandari, Z. Zong, and Y. Yan. Revisiting optical flow estimation in 360 videos. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 8196–8203. IEEE, 2021.

[6] J.-K. Chen, J. Lyu, and Y.-X. Wang. NeuralEditor: Editing neural radiance fields via manipulating point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12439–12448, 2023.

[7] R. Chen, F.-L. Zhang, S. Finnie, A. Chalmers, and T. Rhee. Casual 6-DoF: free-viewpoint panorama using a handheld 360° camera. *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[8] Y. Cheng, L. Li, Y. Xu, X. Li, Z. Yang, W. Wang, and Y. Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023.

[9] C. Choi, S. M. Kim, and Y. M. Kim. Balanced spherical grid for egocentric view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16590–16599, June 2023.

[10] D. Choi, H. Jang, and M. H. Kim. OmniLocalRF: Omnidirectional local radiance fields from dynamic videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6871–6880, 2024.

[11] N. Deng, Z. He, J. Ye, B. Duinkharjav, P. Chakravarthula, X. Yang, and Q. Sun. Fov-NeRF: Foveated neural radiance fields for virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3854–3864, 2022.

[12] J. Gardner, B. Egger, and W. Smith. Rotation-equivariant conditional spherical neural fields for learning a natural illumination prior. *Advances in Neural Information Processing Systems*, 35:26309–26323, 2022.

[13] K. Gu, T. Maugey, S. Knorr, and C. Guillemot. Omni-NeRF: neural radiance field from 360° image captures. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2022.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[15] K. He and J. Sun. Image completion approaches using the statistics of similar patches. *IEEE transactions on pattern analysis and machine intelligence*, 36(12):2423–2435, 2014.

[16] D. Hilbert and S. Cohn-Vossen. *Geometry and the Imagination*, vol. 87. American Mathematical Soc., 2021.

[17] Y. Kasten, D. Ofri, O. Wang, and T. Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021.

[18] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

[19] J. Kopf. 360 video stabilization. *ACM Transactions on Graphics (TOG)*, 35(6):1–9, 2016.

[20] C. Lei, X. Ren, Z. Zhang, and Q. Chen. Blind video deflickering by neural filtering with a flawed atlas. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10439–10448, June 2023.

[21] L. Li, H. Huang, S.-K. Yeung, and H. Cheng. OmniGS: Omnidirectional Gaussian splatting for fast radiance field reconstruction using omnidirectional images. *arXiv preprint arXiv:2404.03202*, 2024.

[22] Y. Li, C. Barnes, K. Huang, and F.-L. Zhang. Deep 360° optical flow estimation based on multi-projection fusion. In *European Conference on Computer Vision*, pp. 336–352. Springer, 2022.

[23] Y. Li, Y. Guo, Z. Yan, X. Huang, Y. Duan, and L. Ren. OmniFusion: 360 monocular depth estimation via geometry-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2801–2810, 2022.

[24] Y.-J. Li, J. Shi, F.-L. Zhang, and M. Wang. Bullet comments for 360° video. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 1–10. IEEE, 2022.

[25] Z. Li, C.-Z. Lu, J. Qin, C.-L. Guo, and M.-M. Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17562–17571, 2022.

[26] S. Liao, E. Gavves, and C. G. Snoek. Spherical regression: Learning viewpoints, surface normals and 3d rotations on n-spheres. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9759–9767, 2019.

[27] F.-L. Liu, S.-Y. Chen, Y. Lai, C. Li, Y.-R. Jiang, H. Fu, and L. Gao. DeepFaceVideoEditing: sketch-based deep editing of face videos. *ACM Transactions on Graphics*, 41(4):167, 2022.

[28] S. Loeschcke, S. Belongie, and S. Benaim. Text-driven stylization of video objects. In *European Conference on Computer Vision*, pp. 594–609. Springer, 2022.

[29] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan. Dynamic 3D Gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.

[30] K. Matzen, M. F. Cohen, B. Evans, J. Kopf, and R. Szeliski. Low-cost 360 stereo photography and video capture. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017.

[31] J. Mei, A. Z. Zhu, X. Yan, H. Yan, S. Qiao, L.-C. Chen, and H. Kretzschmar. Waymo open dataset: Panoramic video panoptic segmentation. In *European Conference on Computer Vision*, pp. 53–72. Springer, 2022.

[32] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[33] P. Moulon, P. Monasse, R. Perrot, and R. Marlet. OpenMVG: Open multiple view geometry. In *Reproducible Research in Pattern Recognition: First International Workshop, RRPR 2016, Cancún, Mexico, December 4, 2016, Revised Selected Papers 1*, pp. 60–74. Springer, 2017.

[34] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, and S. M. Seitz. Hypernerf: a higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021.

[35] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pp. 5301–5310. PMLR, 2019.

[36] C. Reiser, S. Peng, Y. Liao, and A. Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14335–14345, 2021.

[37] T. Rhee, L. Petikam, B. Allen, and A. Chalmers. MR360: Mixed reality rendering for 360° panoramic videos. *IEEE transactions on visualization and computer graphics*, 23(4):1379–1388, 2017.

[38] C. Rodriguez-Pardo, J. Fabre, E. Garces, and J. Lopez-Moreno. NEnv: Neural environment maps for global illumination. In *Computer Graphics Forum*, vol. 42, p. e14883. Wiley Online Library, 2023.

[39] K. Sato, S. Yamaguchi, T. Takeda, and S. Morishima. Deformable neural radiance fields for object motion blur removal. In *ACM SIGGRAPH 2023 Posters*, pp. 1–2. Association for Computing Machinery, 2023.

[40] C. Schroers, J.-C. Bazin, and A. Sorkine-Hornung. An omnistereoscopic video pipeline for capture and display of real-world VR. *ACM Transactions on Graphics (TOG)*, 37(3):1–13, 2018.

[41] H. Shi, Y. Zhou, K. Yang, X. Yin, Z. Wang, Y. Ye, Z. Yin, S. Meng, P. Li, and K. Wang. PanoFlow: Learning 360° optical flow for surrounding temporal understanding. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[42] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

[43] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2827. IEEE, 2018. doi: 10.1109/CVPR.2018.00297

[44] C. Tang, O. Wang, F. Liu, and P. Tan. Joint stabilization and direction of 360° videos. *ACM Transactions on Graphics (TOG)*, 38(2):1–13, 2019.

[45] Z. Teed and J. Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 402–419. Springer, 2020.

[46] O. T. Tursun, E. Arabadzhiyska-Koleva, M. Wernikowski, R. Mantiuk, H.-P. Seidel, K. Myszkowski, and P. Didyk. Luminance-contrast-aware foveated rendering. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019.

[47] F.-E. Wang, Y.-H. Yeh, Y.-H. Tsai, W.-C. Chiu, and M. Sun. BiFuse++: Self-supervised and efficient bi-projection fusion for 360° depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5448–5460, 2022.

[48] K.-M. Wong. View-adaptive asymmetric image detail enhancement for 360-degree stereoscopic VR content. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 23–26. IEEE, 2022.

[49] B. Xu, S. Pathak, H. Fujii, A. Yamashita, and H. Asama. Spatio-temporal video completion in spherical image sequences. *IEEE Robotics and Automation Letters*, 2(4):2032–2039, 2017.

[50] H. Xu, X. Liu, Q. Zhao, Y. Ma, C. Yan, and F. Dai. Gaussian label distribution learning for spherical image object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1033–1042, 2023.

[51] M. Xu, C. Li, S. Zhang, and P. Le Callet. State-of-the-art in 360° video/image processing: Perception, assessment and compression. *IEEE Journal of Selected Topics in Signal Processing*, 14(1):5–26, 2020.

[52] R. Xu, X. Li, B. Zhou, and C. C. Loy. Deep flow-guided video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3723–3732, 2019.

[53] K. Yang, J. Zhang, S. Reiß, X. Hu, and R. Stiefelhagen. Capturing omni-range context for omnidirectional segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1376–1386, 2021.

[54] V. Ye, Z. Li, R. Tucker, A. Kanazawa, and N. Snavely. Deformable sprites for unsupervised video decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2657–2666, 2022.

[55] M. Yuan and R. Christian. 360° optical flow using tangent images. In *Proceedings of the 32nd British Machine Vision Conference (BMVC)*, 2021.

[56] Y.-J. Yuan, Y.-T. Sun, Y.-K. Lai, Y. Ma, R. Jia, and L. Gao. NeRF-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18353–18364, 2022.

[57] F. Zhang, J. Zhao, Y. Zhang, and S. Zollmann. A survey on 360° images and videos in mixed reality: Algorithms and applications. *Journal of Computer Science and Technology*, 38(3):473–491, 2023.

[58] Y. Zhang, F.-L. Zhang, Y.-K. Lai, and Z. Zhu. Efficient propagation of sparse edits on 360° panoramas. *Computers & Graphics*, 96:61–70, 2021.

[59] Y. Zhang, F.-L. Zhang, Z. Zhu, L. Wang, and Y. Jin. Fast edit propagation for 360 degree panoramas using function interpolation. *IEEE Access*, 10:43882–43894, 2022.

[60] C. Zheng, W. Lin, and F. Xu. EditableNeRF: Editing topologically varying neural radiance fields by key points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8317–8327, 2023.

[61] Z. Zhu, R. R. Martin, and S.-M. Hu. Panorama completion for street views. *Computational Visual Media*, 1(1):49–57, 2015.