

Dyn-E: Local Appearance Editing of Dynamic Neural Radiance Fields

SHANGZHAN ZHANG, State Key Laboratory of CAD&CG, Zhejiang University, China

SIDA PENG, Zhejiang University, China

YINJI SHENTU, Zhejiang University, China

QING SHUAI, State Key Laboratory of CAD&CG, Zhejiang University, China

TIANRUN CHEN, Zhejiang University, China

KAICHENG YU, Alibaba Group, China

HUJUN BAO, State Key Laboratory of CAD&CG, Zhejiang University, China

XIAOWEI ZHOU*, State Key Laboratory of CAD&CG, Zhejiang University, China

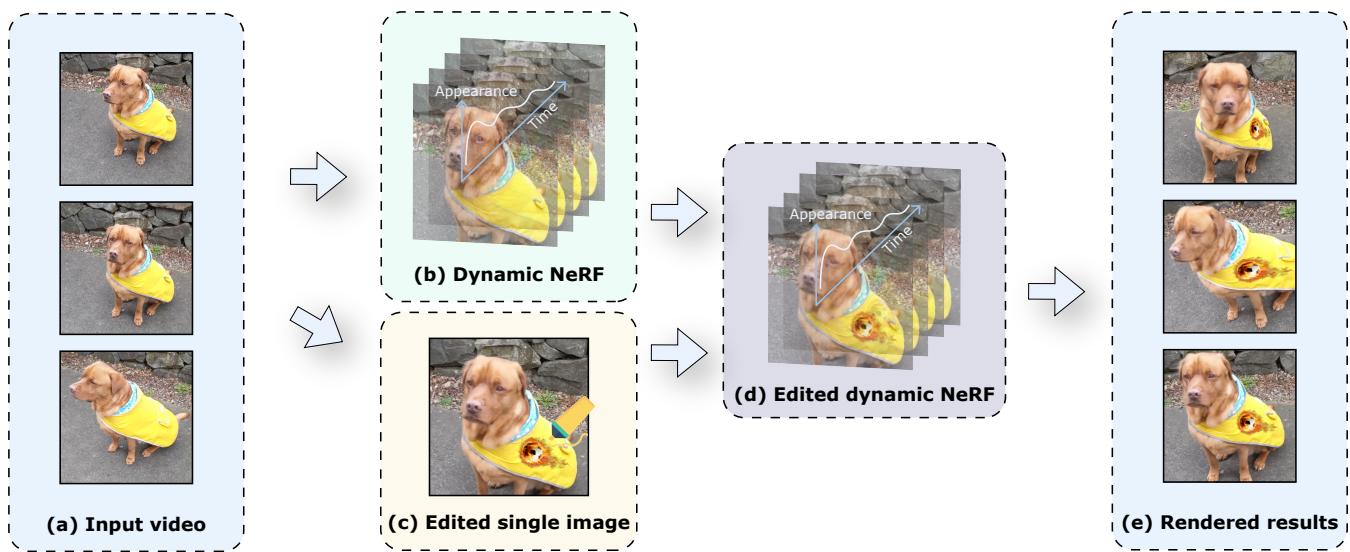


Fig. 1. Our proposed approach allows users to locally edit the appearance of a dynamic 3D scene in a user-friendly manner. Given training videos (a) and the reconstructed dynamic NeRF (b) as input, users can edit the appearance of the dynamic NeRF (d) by manipulating pixels in a single image (c). Experiment results demonstrate that our method can produce spatially and temporally consistent renderings (e).

Recently, the editing of neural radiance fields (NeRFs) has gained considerable attention, but most prior works focus on static scenes while research on the appearance editing of dynamic scenes is relatively lacking. In this paper, we propose a novel framework to edit the local appearance of dynamic NeRFs by manipulating pixels in a single frame of training video. Specifically, to locally edit the appearance of dynamic NeRFs while preserving unedited regions, we introduce a local surface representation of the edited region, which can be inserted into and rendered along with the original NeRF and warped to arbitrary other frames through a learned invertible motion representation network. By employing our method, users without

professional expertise can easily add desired content to the appearance of a dynamic scene. We extensively evaluate our approach on various scenes and show that our approach achieves spatially and temporally consistent editing results. Notably, our approach is versatile and applicable to different variants of dynamic NeRF representations.

CCS Concepts: • Computing methodologies → Computer vision representations.

Additional Key Words and Phrases: Dynamic view synthesis, neural radiance fields, appearance editing.

ACM Reference Format:

Shangzhan Zhang, Sida Peng, Yinji ShenTu, Qing Shuai, Tianrun Chen, Kaicheng Yu, Hujun Bao, and Xiaowei Zhou. 2023. Dyn-E: Local Appearance Editing of Dynamic Neural Radiance Fields. 1, 1 (July 2023), 11 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

3D content editing is a fast-growing research area with significant potential to shape the future of digital media. Recently, NeRF [Mildenhall et al. 2020] and its extended methods [Barron et al. 2021, 2022;

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Association for Computing Machinery.

XXXX-XXXX/2023/7-ART \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Li et al. 2021; Park et al. 2021a; Peng et al. 2021b] have made it easy to reconstruct static and even dynamic 3D scenes. Therefore, research on 3D scene editing based on NeRFs has also emerged. Many methods [Huang et al. 2022; Liu et al. 2021b; Yang et al. 2022] were proposed for editing static 3D scenes, which provide friendly editing tools for non-professional users. However, there is a noticeable lack of research related to editing the appearance of dynamic 3D scenes, despite a great demand for users to add their desired content to volumetric videos.

This paper specifically focuses on fine-grained local appearance editing for 3D dynamic scenes. Given a dynamic NeRF (probably in various representations) and its original training videos, we aim to edit the appearance of the dynamic NeRF by modifying a single 2D frame in the training videos, as illustrated in Fig. 1. This problem is challenging for three reasons. First, it is non-trivial to propagate the edited 2D pixels to the implicit neural scene representation, especially for dynamic scenes. Moreover, how to propagate the single-frame edited content to other frames in a temporally consistent manner is not well explored. Finally, designing a generally applicable editing tool that can be plugged into different variants of dynamic NeRF representations [Gao et al. 2021; Li et al. 2022, 2021; Park et al. 2021a; Peng et al. 2021b] presents a significant challenge. A naive baseline is to directly fine-tune a dynamic NeRF on the edited image. However, training the dynamic NeRF directly using a single image tends to cause the network to overfit to a single view, leading to the degradation of the rendering quality on other views and the failure of propagating the edited result to other frames, despite the tedious computation.

To tackle these challenges, we propose a novel framework called Dyn-E, which enables users to locally edit the appearance of dynamic NeRFs by modifying a single image. Our approach first lifts the edited local region to 3D space to form a local surface and then uses an invertible network to represent the motion of the local surface, allowing us to propagate the edited results across video frames. Specifically, our local surface is a textured mesh lifted from the edited region of the single image through the rendered depth map of the given dynamic NeRF. We convert the textured mesh into a local density and color field, which can be rendered together with the given dynamic NeRF. To propagate the edits to other frames, an invertible motion representation is learned from the input videos, enabling efficient warping of the local surface to different time steps.

Our proposed local surface representation is an independent layer that does not make assumptions about the underlying scene structure, thus it is versatile and can be inserted into most existing dynamic NeRF representations. Thanks to the local surface representation, we are able to maintain the rendering performance of the dynamic NeRFs outside the edited region. The surface-based representation also allows us to leverage smoothness and photometric constraints on the surface deformation to regularize the learning of the invertible motion network, which makes the spatial positions of the local surface in all frames more accurate. We conduct extensive experiments on three commonly used dynamic scene datasets to verify the effectiveness of our algorithm. Additionally, we demonstrate that our method is suitable for most dynamic NeRFs by editing HyperNeRF [Park et al. 2021b], DynamicNeRF [Gao et al. 2021], and Neural Body [Peng et al. 2021b].

Our main contributions are summarized as follows: (1) We propose a novel approach to the task of image-based local appearance editing for dynamic NeRFs. (2) We design a trackable local surface representation that facilitates spatio-temporally consistent dynamic scene appearance editing. (3) We extensively verify the effectiveness and versatility of our approach with various dynamic NeRF representations and multiple complex datasets.

2 RELATED WORKS

Dynamic NeRFs. Recently, Neural Radiance Fields (NeRFs) [Mildenhall et al. 2020] demonstrate remarkable performance in the domain of novel view synthesis. Given a set of multi-view images of static scenes, NeRF has the ability to reconstruct scenes and generate free-viewpoint videos. Some studies [Lin et al. 2022; Park et al. 2021a,b; Peng et al. 2023; Pumarola et al. 2020; Tretschk et al. 2021] extend NeRFs to dynamic scenes, showcasing promising results. D-NeRF [Pumarola et al. 2020] and Nerfies [Park et al. 2021a] employ canonical NeRFs and a series of deformation fields to capture dynamic scenes. NSFF [Li et al. 2021] uses MLPs to model the scene flow fields, ensuring temporal consistency. DyNeRF [Li et al. 2022] directly utilizes time-conditioned neural radiance fields for representing dynamic scenes. Several works [Liu et al. 2021a; Peng et al. 2021a,b; Zheng et al. 2022] utilize prior knowledge of the human body to handle dynamic human bodies with a wide range of motion. Despite achieving high-quality rendering results, these works do not allow users to freely edit their appearance.

Neural scene editing. Scene editing based on neural fields is gaining increasing attention. Some works [Bao et al. 2023; Liu et al. 2021b; Xu and Harada 2022; Yuan et al. 2022] are able to modify the geometry of scenes, producing impressive results, while others focus on editing the appearance of scenes, similar to our work. A series of works [Huang et al. 2022; Nguyen-Phuoc et al. 2022; Zhang et al. 2022a] utilize pretrained 2D neural networks to modify the style of neural radiance fields. Some works [Zhang et al. 2021a,b, 2022b] enable physics-based relighting and material editing on NeRFs by recovering the material properties of objects and the environmental lighting conditions. Another line of work [Das et al. 2022; Xiang et al. 2021; Yang et al. 2022] focuses on fine-grained appearance editing of NeRFs. NeuMesh [Yang et al. 2022] empowers users to modify the appearance of objects by editing a 2D image captured from a specific viewpoint and achieves consistent rendering of the edited appearance from multiple viewpoints. However, these methods only enable editing the appearance of static scenes and do not explore editing dynamic scenes. Recently, some works [Chen et al. 2023; Ho et al. 2023; Jafarian et al. 2023] attempt to edit dynamic human bodies, but they need to utilize prior knowledge of the human body which is not required in our work.

Video editing. Achieving consistent video editing [Liu et al. 2023; Molad et al. 2023; Qi et al. 2023; Yu et al. 2023] has always been a long-standing challenge in the field of video editing. To achieve temporal consistency, some studies [Ruder et al. 2016; Xu et al. 2022] utilize optical flow as a constraint. Other studies [Jamriška et al. 2019; Texler et al. 2020] utilize keyframe propagation methods to propagate modifications made in keyframes to other frames. Additionally, there

are approaches [Bar-Tal et al. 2022; Kasten et al. 2021; Ye et al. 2022] that represent videos as 2D atlases, enabling appearance editing of the videos by modifying the atlases. However, these methods lack 3D awareness, making it non-trivial to extend them for performing novel view synthesis. Some traditional methods [Deng et al. 2022; Habermann et al. 2019; Xu et al. 2014] attempt to edit the appearance of dynamic videos by reconstructing the explicit 3D representation of dynamic objects. However, they only show the editing results for simple scenes or in the presence of RGBD training data.

3 METHOD

Given a dynamic NeRF and its training video data, our task is to edit the local appearance of the dynamic NeRF by modifying a single 2D image in the training video. To this end, we propose a framework that can edit the appearance of dynamic NeRF by lifting the 2D edited content to a sequence of temporally consistent 3D edited content, as illustrated in Fig. 2. We first briefly introduce dynamic NeRFs for modeling dynamic scenes and our problem setting in Section 3.1. Then, we describe a local surface representation in Section 3.2. Meanwhile, we propose a motion representation that allows for the tracking of the local surface in Section 3.3. Next, Section 3.4 describes our training losses for the motion representation and the constraints that are applied to the local surface. Finally, Section 3.5 illustrates how to render edited results after training.

3.1 Preliminary

Dynamic NeRFs. In this paper, we use the term “dynamic NeRF” to refer to the neural radiance fields that can represent dynamic scenes. Existing dynamic NeRFs [Gao et al. 2021; Li et al. 2022, 2021; Park et al. 2021a,b; Peng et al. 2021b] represent dynamic scenes as a time-varying continuous neural radiance field, i.e., a mapping function f_θ from a spatial position \mathbf{x} , time t , and viewing direction \mathbf{d} to color and density. It can be represented as a general equation:

$$f_\theta : (\mathbf{x} \in \mathbb{R}^3, t \in \mathbb{R}, \mathbf{d} \in \mathbb{S}^2) \mapsto (\sigma \in \mathbb{R}^+, \mathbf{c} \in \mathbb{R}^3). \quad (1)$$

Different dynamic NeRFs have different ways of modelling time-varying scene content. Some works [Gao et al. 2021; Li et al. 2021] adopt MLPs taking an additional “time coordinate” as input to represent the dynamic components of the scene. Nerfies [Park et al. 2021a] uses a series of time-varying deformation fields to represent the change of scene over time. Neural Body [Peng et al. 2021b] represents the dynamic content of the scene by a time-varying human parametric model. Our method is designed to be compatible with most dynamic NeRFs.

Problem setting. Our goal is to edit the appearance of the given dynamic NeRF f_θ by modifying a single 2D image in the training video $\{\mathbf{I}_i\}_{i=1}^N$, and then render a free-viewpoint video that is temporally consistent and high-quality. For convenience, we refer to the image that the user wants to edit as the reference image. Our method can be trained on both monocular and multi-view data. N_v denotes the number of cameras.

We assume that the region to be edited can be observed without occlusion in most cases. Additionally, we assume that the user only edits part of the image, rather than the entire image.

3.2 Local Surface

It is difficult to edit the appearance of dynamic NeRFs by finetuning them on a single image. This is because optimizing dynamic NeRFs on a single image tends to cause overfitting problems. Most dynamic NeRFs use an MLP to represent the entire space, thus finetuning their local regions will also affect other regions. For local appearance editing tasks, we observe that most 3D regions do not need to be edited. Hence, we only need to design a local layer to be inserted into the dynamic NeRF. By doing so, we can solely modify the parameters of this local layer, ensuring that other regions remain unaffected. Motivated by this, we propose a plug-and-play local surface representation to edit the appearance of dynamic NeRFs. Our local surface representation adopts a mesh-based surface representation that can be easily constructed. Moreover, we design our surface representation to be compatible with the volume rendering equation to handle occlusion. To generate the local surface, we first render the depth map for the reference image using the given dynamic NeRF. Then, we unproject the user-edited region of the reference image back to 3D space to form the mesh, similar to Shih et al. [Shih et al. 2020]. Specifically, we unproject user edited pixels $\{\mathbf{p}_i\}_{i=1}^K$ back to 3D space in world coordinates to form the vertices $\{\mathbf{v}_r^i\}_{i=1}^K$ of the mesh, as shown in Eq. 2:

$$\{\mathbf{v}_r^i\}_{i=1}^K = \mathbf{M}_{c2w}\Pi^{-1}(\{\mathbf{p}_i\}_{i=1}^K), \quad (2)$$

where Π^{-1} represents the inverse perspective projection operation. \mathbf{M}_{c2w} is the camera-to-world transformation matrix, which is used to transform the mesh vertices from camera coordinates to world coordinates. K is the number of mesh vertices. The vertices of neighboring pixels are connected to form the faces of the mesh.

We define the color of each vertex of the mesh as the color of the corresponding pixel in the reference image. When rendering the mesh, we manually define the color for each point along the ray. For any point on the rays intersecting the mesh, its color \mathbf{c}^s is calculated by interpolating the vertex colors of the intersecting face of the 3D mesh using barycentric coordinates as weights. For any point on the rays that does not intersect the mesh, its color \mathbf{c}^s is set to zero.

Although the local surface representation is easy to construct, it is difficult to handle occlusion relationships between the local surface and the original dynamic NeRF, as shown in Fig. 4. To address this problem, we design our surface representation to be seamlessly integrated with the given dynamic NeRF for volume rendering. We convert the mesh into a distance field according to the nearest distance from a point to the mesh. Then, the distance field is converted into a density field. Inspired by VolSDF [Yariv et al. 2021], we convert the distance field into the density field using Eq. 3:

$$\sigma^d(\mathbf{x}) = \alpha\Psi_\beta(-d(\mathbf{x})), \quad (3)$$

where $\alpha = \beta^{-1}$, β is a manually defined parameter, $d(\mathbf{x})$ is the distance from the mesh to the point \mathbf{x} , and Ψ_β represents the cumulative distribution function of the Laplace distribution with zero mean and a scale of β . To ensure that the local surface replace the original dynamic NeRF in the edited region and does not affect other regions of the dynamic NeRF, we define a mask field to indicate which regions belong to the local surface and which regions belong

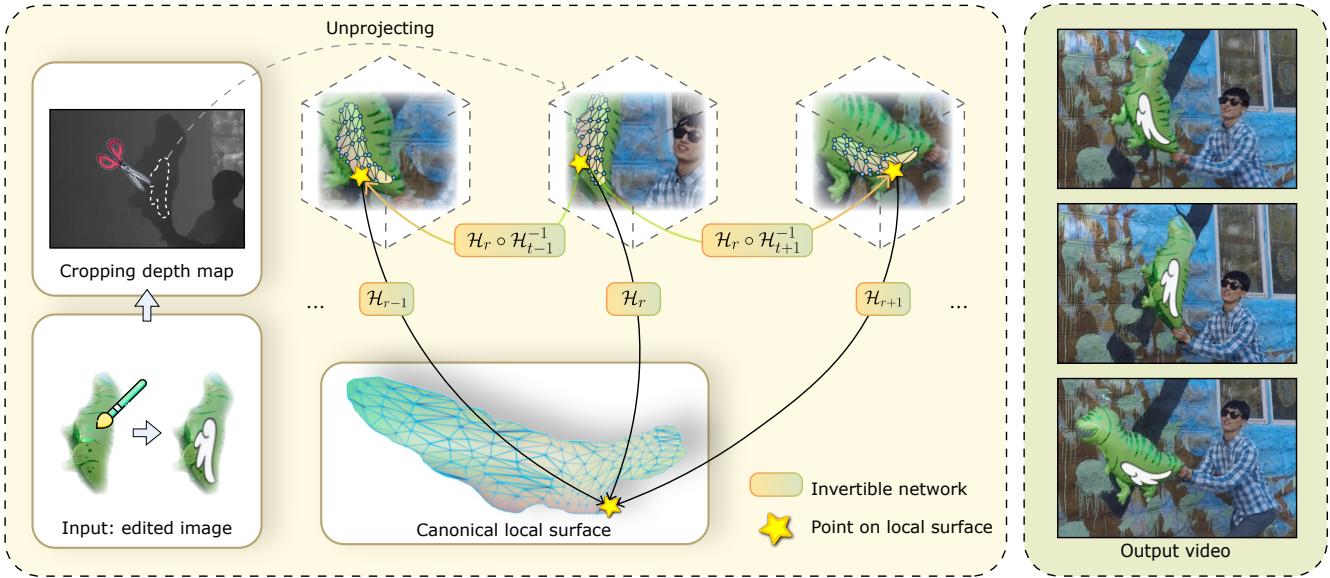


Fig. 2. **Illustration of our pipeline.** Given a single edited image and a dynamic NeRF, we first lift the edited region to the 3D space through rendered depth maps to form a textured mesh. Then, we train an invertible network to propagate the textured mesh to other frames. Finally, we combine the textured mesh with the original dynamic NeRF and render them to obtain the final results.

to the dynamic NeRF. For points within a distance of γ from the surface, the mask value is set to 1. For points beyond a distance of γ from the surface, the mask value is set to 0. If a ray $\mathbf{r}(t)$ does not intersect with the mesh, we set the mask value to 0 for all points on the ray. The mask field is defined as:

$$M(\mathbf{x}) = \begin{cases} 1, & \text{if } d(\mathbf{x}) < \gamma \text{ and } \mathbf{r}(t) \in \mathcal{R}_h \\ 0, & \text{otherwise} \end{cases}, \quad (4)$$

where \mathcal{R}_h is the set of all rays that intersect with the mesh. Our ultimate volume rendering equation is:

$$\mathbf{C}^{full}(\mathbf{r}) = \sum_{i=1}^N T_i^{full} \left(\alpha \left(\sigma_i^d \delta_i \right) (1 - M_i) \mathbf{c}_i^d + \alpha \left(\sigma_i^s \delta_i \right) M_i \mathbf{c}_i^s \right), \quad (5)$$

where \mathbf{C}^{full} is the rendered color of the ray \mathbf{r} and $\alpha(x) = 1 - \exp(-x)$. σ^d and σ^s are the densities of the dynamic NeRF and the local surface, δ is the distance between adjacent sample points. M_i is the mask value at the i -th sample point. \mathbf{c}^d and \mathbf{c}^s are the colors of the dynamic NeRF and the local surface. $T_i^{full} = \exp(-\sum_{j=1}^{i-1} (\sigma_j^d \delta_j (1 - M_j) + \sigma_j^s \delta_j M_j))$ is the accumulated transmittance of the ray \mathbf{r} up to the i -th sample point.

3.3 Motion Representations

Through the above method, we only generate the local surface of a certain frame. The positions of the local surface of other frames are still unknown. Therefore, we propose our motion representation to propagate the local surface to other frames. Inspired by CaDeX [Lei and Daniilidis 2022], we use invertible networks to model our scene motion. Invertible networks are strictly bijective maps, which fits the natural properties of non-rigid motion better than the traditional method [Tewari et al. 2022] of using two MLPs to represent

the deformation and the inverse deformation. By leveraging the invertible network, we can know the point-to-point correspondence between arbitrary two frames, as shown in Eq. 6:

$$\mathbf{x}_{t \rightarrow t'} = \mathcal{H}_{t'}^{-1} \circ \mathcal{H}_t(\mathbf{x}_t), \quad (6)$$

where \mathcal{H}_t represents the operation of mapping the point \mathbf{x}_t from the observation space at frame t to the canonical space by the invertible network, and $\mathcal{H}_{t'}^{-1}$ is the inverse operation of mapping the point $\mathbf{x}_{t'}$ from the canonical space to the observation space at frame t' .

Theoretically, we can train the invertible network to learn the correspondence between adjacent frames in the same way as NSFF [Li et al. 2021]. Because the invertible network is a strictly bijective map, after learning the correspondence between each pair of adjacent frames, the correspondence between any two frames is naturally known. However, this training strategy needs to densely sample the points in 3D space to transform the scene flows into optical flows on the 2D image plane for 2D supervision. Due to the large GPU memory consumption of the invertible network, we cannot use the same strategy as above to train the invertible network. To resolve this problem, we employ an MLP f_ϕ to model the scene flow field and then use f_ϕ to distill the motion information to the invertible network. The MLP f_ϕ is defined as:

$$(\mathbf{f}_{t \rightarrow t+1}, \mathbf{f}_{t \rightarrow t-1}) = f_\phi(\mathbf{x}, \gamma(t)), \quad (7)$$

where $\mathbf{f}_{t \rightarrow t+1}$ and $\mathbf{f}_{t \rightarrow t-1}$ are the scene flow fields from frame t to frame $t+1$ and frame t to frame $t-1$, and $\gamma(t)$ is the positional encoding of t . We adopt the loss function described in Eq. 8 to train the invertible network and only sample training points near the

local surface, which can greatly reduce memory consumption.

$$\mathcal{L}_{distill} = \sum_{\mathbf{x} \in \mathcal{X}_{surf}} (||\mathbf{f}_{t \rightarrow t+1}^{\text{inv}} - \mathbf{f}_{t \rightarrow t+1}|| + ||\mathbf{f}_{t \rightarrow t-1}^{\text{inv}} - \mathbf{f}_{t \rightarrow t-1}||), \quad (8)$$

where $\mathbf{f}_{t \rightarrow t+1}^{\text{inv}} = \mathbf{x}_{t \rightarrow t+1} - \mathbf{x}_t$ and $\mathbf{f}_{t \rightarrow t-1}^{\text{inv}} = \mathbf{x}_{t \rightarrow t-1} - \mathbf{x}_t$ are the scene flows predicted by the invertible networks. \mathcal{X}_{surf} is the set of points near the local surface.

3.4 Training

We utilize the motion matching loss to supervise the scene flow field f_ϕ :

$$\mathcal{L}_{motion} = \sum_{\mathbf{r} \in \mathcal{R}} \sum_{j \in \{i \pm 1\}} \|\hat{\mathbf{p}}_{i \rightarrow j}(\mathbf{r}_i) - \mathbf{p}_{i \rightarrow j}(\mathbf{r}_i)\|, \quad (9)$$

where \mathcal{R} is the set of rays, $\hat{\mathbf{p}}_{i \rightarrow j}$ is the ground truth optical flow from frame i to frame j predicted by RAFT [Teed and Deng 2020], and $\mathbf{p}_{i \rightarrow j}$ is the optical flow induced by the scene flow $\mathbf{f}_{i \rightarrow j}$.

The deformation sequence of our local surface can be recovered easily through the invertible networks, which allows us to easily add regularization terms on the local surface to eliminate accumulated errors caused by the scene flow. We introduce the Laplacian smooth term and the feature-based photometric term as follows.

The Laplacian smooth term is a common regularization term used to make the mesh more smooth, as shown in Eq. 10:

$$\mathcal{L}_{lap} = \sum_{\mathbf{v} \in \mathcal{V}} \|\mathbf{v}_t^i - \bar{\mathbf{v}}_t^i\|^2, \text{ where } \bar{\mathbf{v}}_t^i = \sum_{j \in \mathcal{N}(i)} \mathbf{v}_t^j, \quad (10)$$

where $\mathcal{N}(i)$ is the set of neighboring vertices of the i -th vertex, and \mathbf{v}_t^i is the i -th vertex of the local surface at frame t . \mathcal{V} is the set of all the vertices of the local surface. To align the surface with the image in each frame, the feature-based photometric term is utilized to measure the alignment between the surface and the image, as shown in Eq. 11:

$$\begin{aligned} \mathcal{L}_{pho} &= \sum_{\mathbf{v}^i \in \mathcal{V}} \sum_{j=0}^{N_v} s_{\cos}(F_r(\Pi(\mathbf{v}_r^i)), F_t^j(\Pi(\mathbf{v}_t^i))), \\ s_{\cos}(x, y) &= \max(1 - \frac{\langle x, y \rangle}{\|x\| \cdot \|y\|}, \epsilon), \end{aligned} \quad (11)$$

where F_r and F_t are the features extracted from the reference image and the image at frame t , Π is the projection function and \mathbf{v}_r and \mathbf{v}_t are the vertices of the local surface at the reference frame and frame t . N_v is the number of cameras. ϵ is a threshold to prevent inaccurate supervision signals caused by occlusion. We adopt S2DNet [Germain et al. 2020] as the feature extractor, for it can extract robust local features. Note that only \mathcal{L}_{pho} may be supervised by multi-view images, while the other loss functions solely require supervision from single-view videos. The complete loss functions are shown in the supplementary material.

3.5 Rendering Edited Results

After training, we can warp the local surface to any frame we want by the learned invertible networks. Then, we combine the local surface and the given dynamic NeRF, rendering them together according to Eq. 5.

Table 1. User study results. Our method is preferred by users in terms of temporal consistency and photo-realism.

	Temporal consistency↑	Photo-realism↑
Ours	3.75	3.65
Nerfies	1.50	1.55
HyperNeRF	1.20	1.30
SF+DynNeRF	2.15	2.30
OF+DynNeRF	3.20	2.30

4 EXPERIMENTS

4.1 Experimental Settings

Comparison methods. We compare our proposed method with the following baseline methods using the Nvidia Dynamic Scenes Dataset [Yoon et al. 2020]: (1) Gao *et al.* [Gao et al. 2021] is a widely used method for modeling dynamic scenes. To support editing, we make the following modifications: we project 3D scene flows onto 2D space to render optical flows, and use optical flows to warp the edited content to other frames in the training video. After obtaining the edited training video, we use it to supervise the time-varying dynamic NeRF. This baseline is abbreviated as “SF+DynNeRF”. (2) “OF+DynNeRF” is another strong baseline. We employ optical flows predicted by the RAFT [Teed and Deng 2020] to warp the edited content in the reference image to other frames in the training video and use the edited video to supervise the time-varying dynamic NeRF. We choose Gao *et al.* [Gao et al. 2021] as the dynamic NeRF for this baseline. This baseline is abbreviated as “OF+DynNeRF”. (3) Nerfies [Park et al. 2021a] is a commonly used algorithm for modeling non-rigidly deforming scenes for monocular videos. It models the non-rigidly deforming scene as a canonical space and a set of deformation fields. To edit Nerfies, we finetune it using the single edited image. To prevent overfitting, we freeze the parameters of the deformation fields and only train the MLP representing the canonical space. (4) HyperNeRF [Park et al. 2021b] extends Nerfies and adopts a family of higher-dimensional spaces to handle topological variations. HyperNeRF is also finetuned on the single edited image for editing. We only train the MLP that represents the canonical space while freezing the parameters of other components.

Datasets. We validate our algorithm on three datasets to demonstrate its wide applicability to various types of dynamic scenes. (1) The Nvidia Dynamic Scenes Dataset [Yoon et al. 2020] is a dataset containing 9 videos, which is widely used to measure the performance of dynamic view synthesis. We use the data processed by Gao *et al.* [Gao et al. 2021] for comparison experiments. To reconstruct the dynamic scenes in this dataset for editing, we choose the method proposed by Gao *et al.* [Gao et al. 2021]. Twenty users were asked to edit these videos and then provide ratings for our algorithm and baseline. We consider two aspects for user evaluations: (1) temporal consistency and (2) photo-realism. The scores are integers ranging from 1 to 5, where 1 represents lower quality and 5 represents the highest quality. (2) The Nerfies-HyperNeRF-CoNeRF dataset is utilized by Nerfies [Park et al. 2021a], HyperNeRF [Park et al. 2021b], and CoNeRF [Kania et al. 2022] for their respective research. Since their capture protocols are similar, we treat them

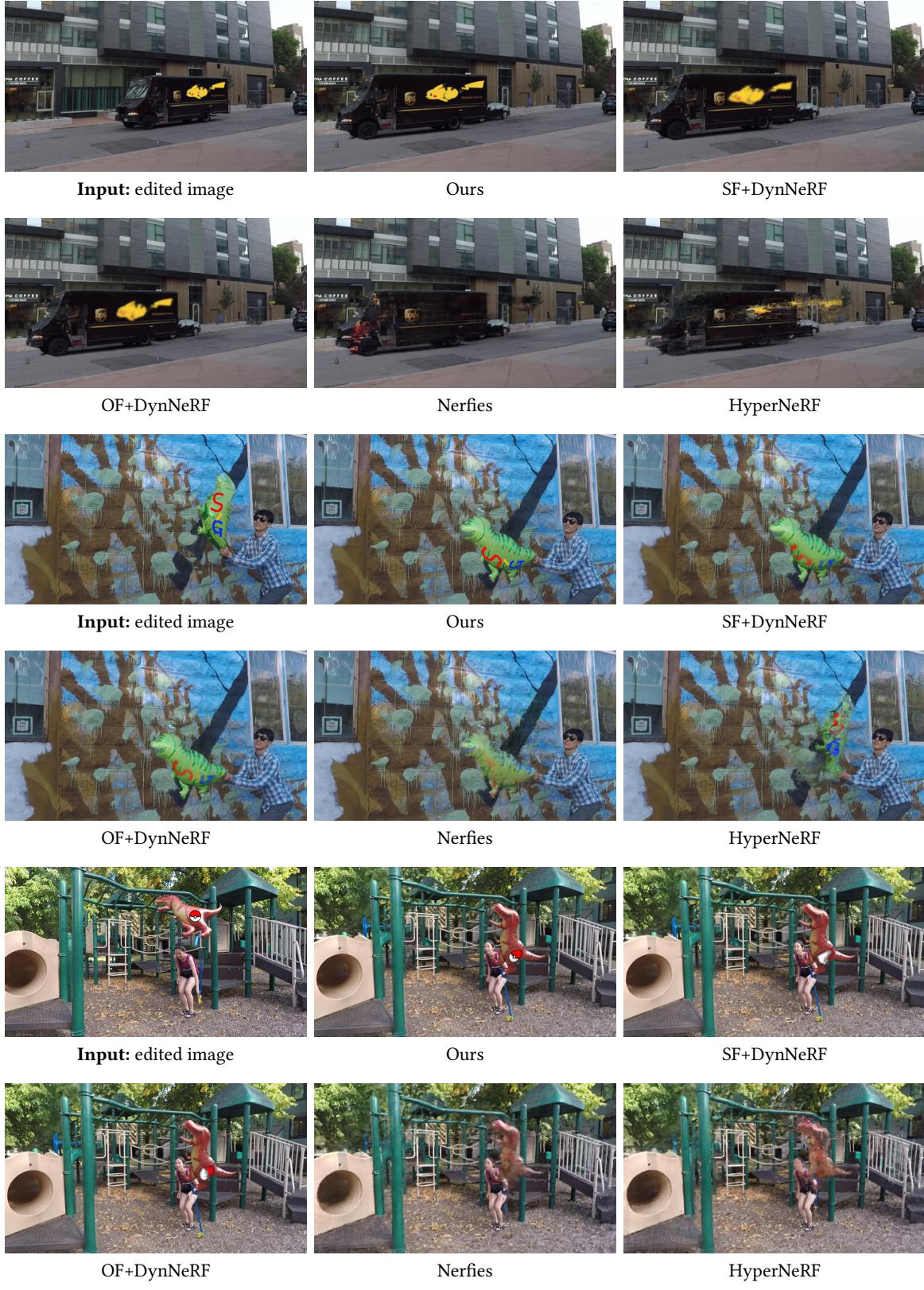


Fig. 3. **Qualitative comparisons.** We generate more realistic results than the baseline methods.

Table 2. Quantitative results of the ablation studies. We compare our method with baselines in terms of PCK-T and EPE.

	EPE \downarrow	PCK-1 \uparrow	PCK-2 \uparrow
Ours	0.72	0.83	0.99
Ours w/o Lap	1.33	0.35	0.81
Ours w/o FP	1.29	0.46	0.80
Ours w/o SFF	1.55	0.33	0.74
Ours w/o Inv	1.57	0.37	0.65
SF warping	1.34	0.34	0.88

as a single dataset. We reconstruct the dynamic scenes using HyperNeRFs on this dataset and apply our method for editing. (3) The ZJU-MoCap [Peng et al. 2021b; Shuai et al. 2022] is a dataset widely used to test the novel view synthesis for human bodies. Following previous work [Shuai et al. 2022], we use eight synchronized cameras as training views. We adopt the Neural Body [Peng et al. 2021b] to reconstruct the dynamic human body and utilize our method to edit it. We present our qualitative results and user study results with baselines on the Nvidia Dynamic Scenes Dataset. On the Nerfies-HyperNeRF-CoNeRF and ZJU-MoCap datasets, we show qualitative results of our method.

Implementation details. We use the Adam optimizer to train our network with a learning rate of 5e-4. The specific network architecture is in the supplementary material. Our network needs to be trained for 12 hours on an NVIDIA RTX 3090 for the Nvidia Dynamic Scenes Dataset, and two NVIDIA RTX 3090 GPUs for the Nerfies-HyperNeRF-CoNeRF and ZJU-MoCap datasets. The detailed parameter selection of β and γ is in the supplementary material.

4.2 Experiment Results

Comparisons. We present some comparison results for our qualitative evaluation in Fig. 3. Our algorithm performs much better than other baselines. Using optical flows and scene flows to warp the edited content is prone to accumulate errors, resulting in poor temporal consistency. Additionally, because “SF+DynNeRF” lacks a well-defined and smooth surface, the rendered results of “SF+DynNeRF” and “OF+DynNeRF” at novel views suffer from blur and ghosting artifacts. The results of Nerfies and HyperNeRF also exhibit serious temporal inconsistency due to inaccurate correspondences. Additionally, HyperNeRF tends to overfit on a single edited frame, which seriously affects the results of other frames. The user study results also show that our method is preferred by users, as shown in Table 1.

More qualitative results. In Fig. 5, we show the qualitative results on the Nerfies-HyperNeRF-CoNeRF dataset. In Fig. 6, we show the qualitative results of the ZJU-MoCap dataset.

4.3 Ablation Studies

Temporal consistency. We analyze how the proposed components in motion representation affect temporal consistency. We measure temporal consistency by evaluating pixel-wise correspondence accuracy between the reference image and other frames. We adopt PCK-T [Truong et al. 2021] and EPE [Ilg et al. 2017] to measure the pixel-wise correspondence accuracy. In the context of PCK-T, the letter “T” represents a given pixel threshold, which we set to 1 and

2 in our experiments. The “Truck” scene in the Nvidia Dynamic Scenes Dataset is selected for this experiment because it contains large areas of textureless planar regions, which is easy to obtain the ground truth of dense correspondence and is more challenging. By annotating the four corners of the truck plane in each frame, the homography matrix is able to be calculated, and the ground truth of dense correspondence can be obtained.

We remove two important regularizations to verify their effectiveness, including the feature-based photo-metric term and the Laplacian smooth term. The ablation experiments of removing the feature-based photo-metric term and Laplacian smooth term are abbreviated as “Ours w/o FP” and “Ours w/o Lap”, respectively. To demonstrate that we need scene flow fields to distill the motion information to invertible networks, we design a baseline named “Ours w/o SFF”. This baseline removes the scene flow fields. To supervise the invertible networks, we directly project the local surface to a 2D plane and then utilize the optical flow estimated by RAFT to constrain the position of the local surface between adjacent frames. In order to show the effectiveness of invertible networks, we propose a baseline abbreviated as “Ours w/o Inv”. In this baseline, the invertible networks are replaced by forward-warping MLP and backward-warping MLP, like Disentangled3d [Tewari et al. 2022]. We design a baseline named “SF warping” which directly warps the edited region with scene flows to show that there will be cumulative errors.

As shown in Table 2, “Ours w/o FP” and “Ours w/o Lap” have lower PCK and higher EPE than our method, which indicates that the regularization added on the local surface can improve the temporal consistency. The performance of “Ours w/o SFF” demonstrates that it is necessary to utilize scene flow fields to distill the motion information to invertible networks. This is because using scene flow fields to distill the motion information is a more direct 3D supervision signal than 2D optical flow constraints. Additionally, following NSFF [Li et al. 2021] and Gao *et al.* [Gao et al. 2021], temporal photometric consistency and some regularization can be added to help eliminate noise in scene flow fields during training. Because our method strictly satisfies cycle consistency, it is less likely to accumulate errors and performs better than “Ours w/o Inv”. Our method is better than “SF warping”, for we add the regularization on the local surface to eliminate the accumulated errors.

Importance of handling occlusion relationship. Our method combines the local surface and the given dynamic NeRF and uses the volume rendering equation to render them to handle the occlusion relationship. To demonstrate the significance of this design, we directly render the local surface with the mesh renderer and combine it with the rendered results of the dynamic NeRF. We abbreviate this baseline as “Ours w/o Occ”. As shown in Fig. 4, if the occlusion relationship is not handled, the rendered results would be incorrect.

5 CONCLUSIONS

Editing the appearance of dynamic 3D scenes is an important and challenging task. In this paper, we propose a method for editing the local appearance of dynamic 3D scenes by modifying individual frames in the video. Through our proposed trackable local surface, we are able to precisely edit the local appearance of the dynamic

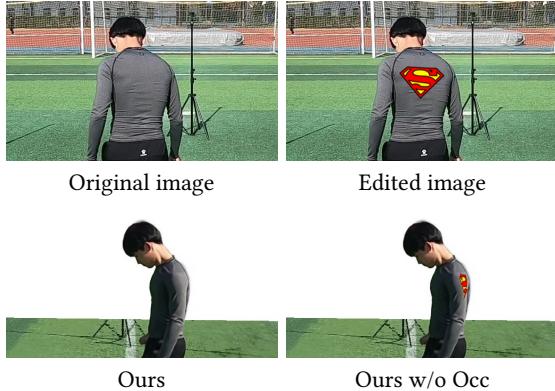


Fig. 4. Importance of handling occlusion relationship. We show the results of our method with and without handling the occlusion relationship. The baseline “Ours w/o Occ” fails to correctly handle the occlusion, resulting in the edited content behind the human body being visible.

scene in a temporally coherent manner. Experiments have demonstrated that our method can achieve high-quality results on a variety of dynamic scenes.

REFERENCES

- Chong Bao, Yinda Zhang, Bangbang Yang, Tianxing Fan, Zesong Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. 2023. SINE: Semantic-driven Image-based NeRF Editing with Prior-guided Editing Field. In *CVPR*.
- Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. 2022. Text2live: Text-driven layered image and video editing. In *ECCV*.
- Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*.
- Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*.
- Yue Chen, Xuan Wang, Qi Zhang, Xiaoyu Li, Xingyu Chen, Yu Guo, Jue Wang, and Fei Wang. 2023. UV Volumes for Real-time Rendering of Editable Free-view Human Performance. In *CVPR*.
- Sagnik Das, Ke Ma, Zhixin Shu, and Dimitris Samaras. 2022. Learning an Isometric Surface Parameterization for Texture Unwrapping. In *ECCV*.
- Bailin Deng, Yuxin Yao, Roberto M Dyke, and Juyong Zhang. 2022. A Survey of Non-Rigid 3D Registration. In *Computer Graphics Forum*.
- Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. 2021. Dynamic View Synthesis from Dynamic Monocular Video. In *ICCV*.
- Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. 2020. S2DNet: Learning Image Features for Accurate Sparse-to-Dense Matching. In *ECCV*.
- Marc Habermann, Weipeng Xu, Helge Rhodin, Michael Zollhöfer, Gerard Pons-Moll, and Christian Theobalt. 2019. Nrst: Non-rigid surface tracking from monocular video. In *GCPR*.
- Hsuan-I Ho, Lixin Xue, Jie Song, and Otmar Hilliges. 2023. Learning Locally Editable Virtual Humans. In *CVPR*.
- Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. 2022. Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In *CVPR*.
- Eddy Ilg, Nikolaus Mayer, Tommoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*.
- Yasamin Jafarian, Tuanfeng Wang, Duygu Ceylan, Jimei Yang, Nathan Carr, Yi Zhou, and Hyun Soo Park. 2023. Normal-guided Garment UV Prediction for Human Re-texturing. In *CVPR*.
- Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. 2019. Styling video by example. *TOG* (2019).
- Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzcinski, and Andrea Tagliasacchi. 2022. CoNeRF: Controllable Neural Radiance Fields. In *CVPR*.
- Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. 2021. Layered neural atlases for consistent video editing. *TOG* (2021).
- Jiahui Lei and Kostas Daniilidis. 2022. CaDeX: Learning Canonical Deformation Coordinate Space for Dynamic Surface Representation via Neural Homeomorphism. In *CVPR*.
- Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. 2022. Neural 3d video synthesis from multi-view video. In *CVPR*.
- Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. 2021. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. In *CVPR*.
- Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2022. Efficient Neural Radiance Fields with Learned Depth-Guided Sampling. In *SIGGRAPH Asia Conference Proceedings*.
- Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. 2021a. Neural actor: Neural free-view synthesis of human actors with pose control. *TOG* (2021).
- Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. 2021b. Editing conditional radiance fields. In *ICCV*.
- Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. 2023. Video-p2p: Video editing with cross-attention control. *arXiv preprint arXiv:2303.04761* (2023).
- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.
- Eyal Molad, Eliah Horwitz, Danialevski, Alex Rav Acha, Yossi Matias, Yael Pritch, Yaniv Leviathan, and Yedid Hoshen. 2023. Dreamix: Video Diffusion Models are General Video Editors. *arXiv* (2023).
- Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. 2022. Snerf: stylized neural implicit representations for 3d scenes. *arXiv* (2022).
- Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. 2021a. Nerfies: Deformable Neural Radiance Fields. In *ICCV*.
- Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. 2021b. HyperNeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields. *TOG* (2021).
- Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. 2021a. Animatable neural radiance fields for modeling dynamic human bodies. In *ICCV*.
- Sida Peng, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2023. Representing Volumetric Videos as Dynamic MLP Maps. In *CVPR*.
- Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2021b. Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans. In *CVPR*.
- Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2020. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *CVPR*.
- Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. 2023. Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv* (2023).
- Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. 2016. Artistic style transfer for videos. In *GCPR*.
- Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 2020. 3D Photography using Context-aware Layered Depth Inpainting. In *CVPR*.
- Qing Shuai, Chen Geng, Qi Fang, Sida Peng, Wenhao Shen, Xiaowei Zhou, and Hujun Bao. 2022. Novel View Synthesis of Human Interactions from Sparse Multi-view Videos. In *SIGGRAPH Conference Proceedings*.
- Zachary Teed and Jia Deng. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*.
- Ayush Tewari, Xingang Pan, Ohad Fried, Maneesh Agrawala, Christian Theobalt, et al. 2022. Disentangled3d: Learning a 3d generative model with disentangled geometry and appearance from monocular images. In *CVPR*.
- Ondřej Texler, David Futschik, Michal Kučera, Ondřej Jamriška, Šárka Sochorová, Menclej Chai, Sergey Tulyakov, and Daniel Sýkora. 2020. Interactive video stylization using few-shot patch-based training. *TOG* (2020).
- Edgar Treitschek, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. 2021. Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Dynamic Scene From Monocular Video. In *ICCV*.
- Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. 2021. Learning accurate dense correspondences and when to trust them. In *CVPR*.
- Fanbo Xiang, Zexiang Xu, Milos Hasan, Yannick Hold-Geoffroy, Kalyan Sunkavalli, and Hao Su. 2021. Neutex: Neural texture mapping for volumetric neural rendering. In *CVPR*.
- Tianhan Xu and Tatsuya Harada. 2022. Deforming radiance fields with cages. In *ECCV*.
- Weipeng Xu, Mathieu Salzmann, Yongtian Wang, and Yue Liu. 2014. Nonrigid surface registration and completion from RGBD images. In *ECCV*.
- Yiran Xu, Badour Albaran, and Jia-Bin Huang. 2022. Temporally consistent semantic video editing. In *ECCV*.
- Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. 2022. Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In *ECCV*.
- Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. 2021. Volume rendering of neural implicit surfaces. In *NeurIPS*.

- Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely. 2022. Deformable sprites for unsupervised video decomposition. In *CVPR*.
- Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. 2020. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *CVPR*.
- Emilie Yu, Kevin Blackburn-Matzen, Cuong Nguyen, Oliver Wang, Rubaiat Habib Kazi, and Adrien Bousseau. 2023. VideoDoodles: Hand-Drawn Animations on Videos with Scene-Aware Canvases. *ACM Transactions on Graphics* (2023).
- Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. 2022. NeRF-editing: geometry editing of neural radiance fields. In *CVPR*.
- Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. 2022a. Arf: Artistic radiance fields. In *ECCV*.
- Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. 2021a. Physig: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *CVPR*.
- Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. 2021b. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *TOG* (2021).
- Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. 2022b. Modeling indirect illumination for inverse rendering. In *CVPR*.
- Zerong Zheng, Han Huang, Tao Yu, Hongwen Zhang, Yandong Guo, and Yebin Liu. 2022. Structured local radiance fields for human avatar modeling. In *CVPR*.

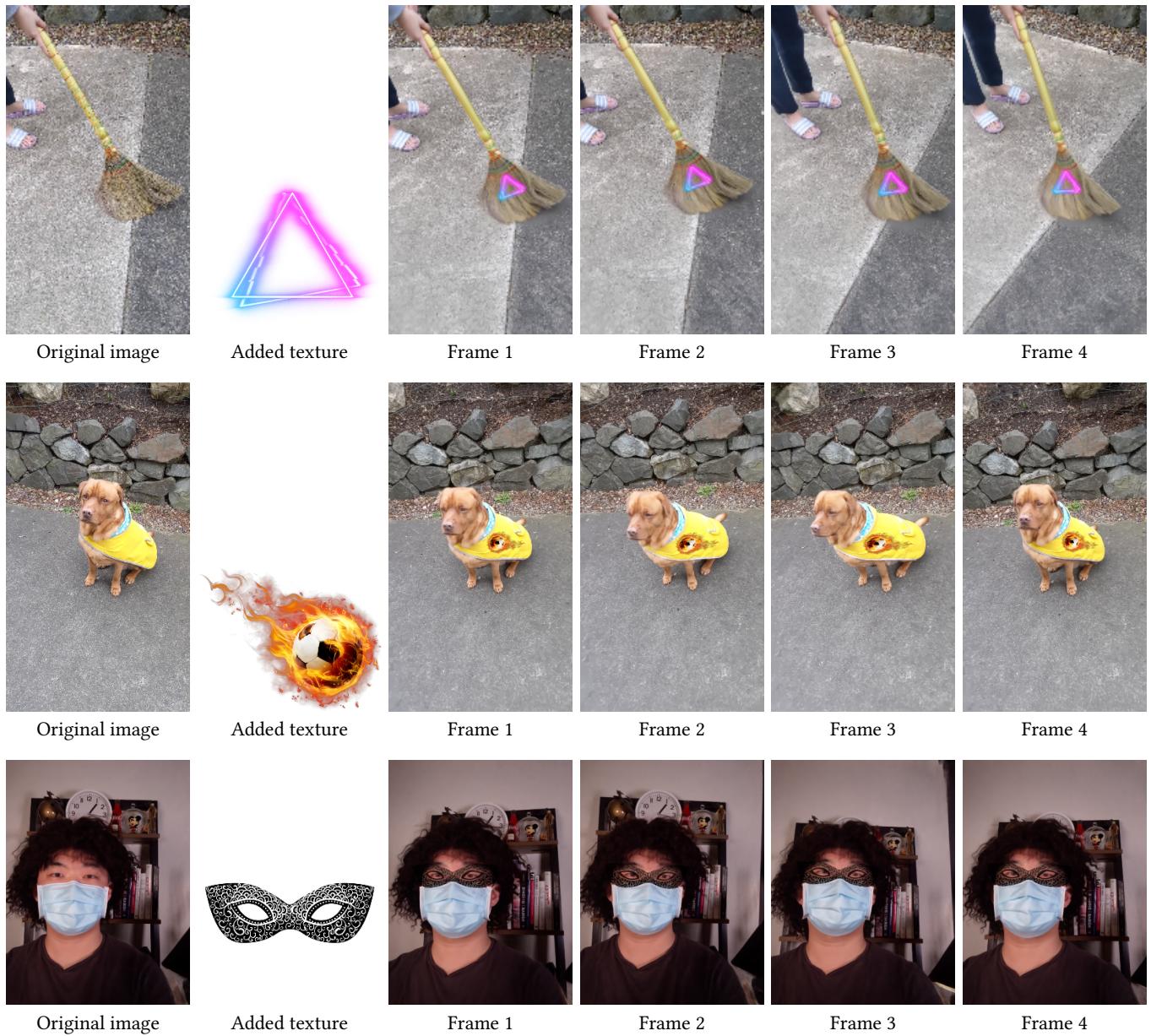


Fig. 5. **Qualitative results on the Nerfies-HyperNeRF-CoNeRF dataset.** More results are shown in the supplementary video.

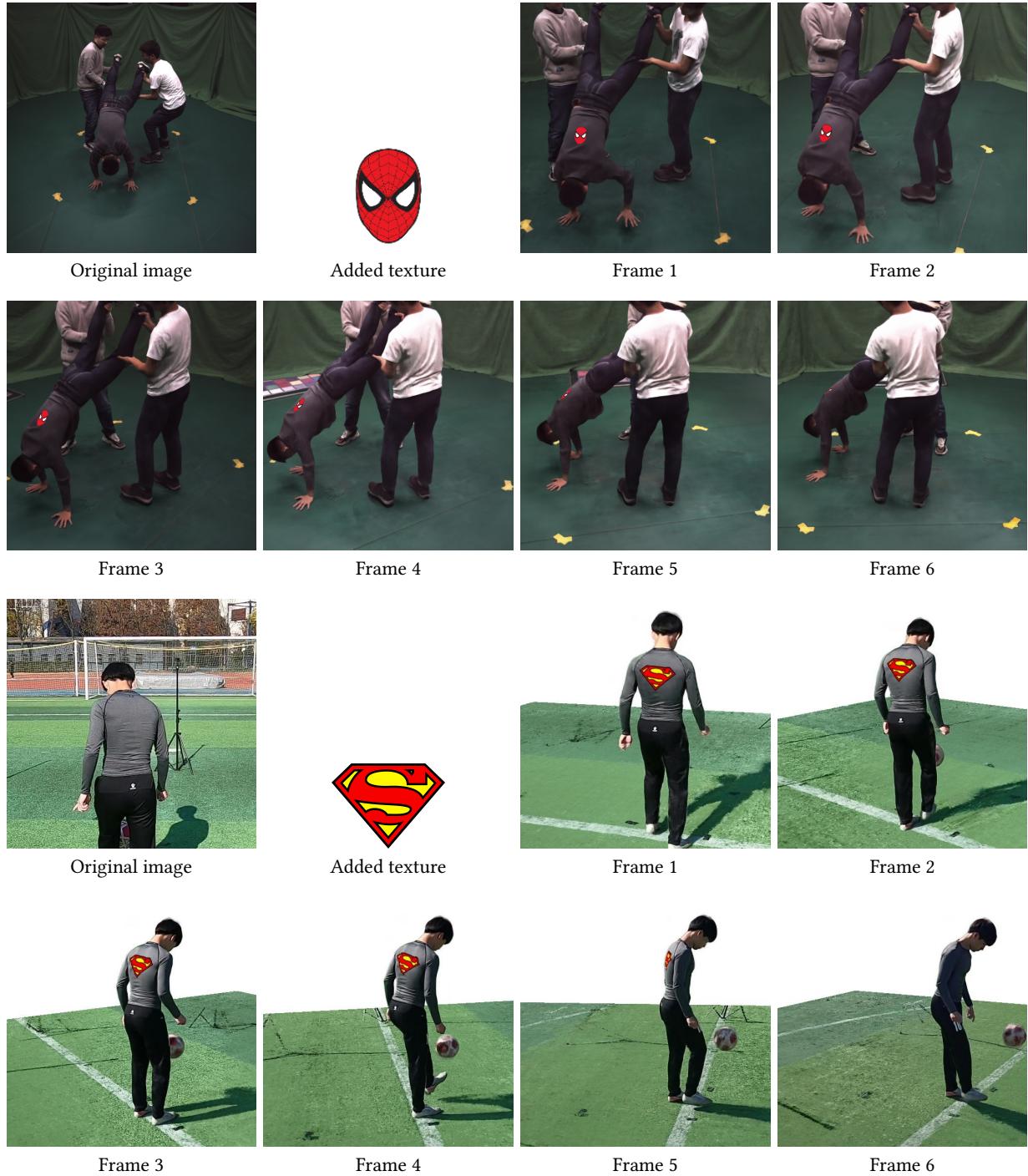


Fig. 6. Qualitative results on the ZJU-MoCap dataset. More results are shown in the supplementary video.