4D-Editor: Interactive Object-level Editing in Dynamic Neural Radiance Fields via Semantic Distillation

Dadong Jiang Zhihui Ke Xiaobo Zhou† Tie Qiu Xidong Shi
Hao Yan
College of Intelligence and Computing, Tianjin University
{patrickdd, kezhihui, xiaobo.zhou, qiutie, suif}@tju.edu.cn

Abstract

This paper targets interactive object-level editing (e.g., deletion, recoloring, transformation, composition) in dynamic scenes. Recently, some methods aiming for flexible editing static scenes represented by neural radiance field (NeRF) have shown impressive synthesis quality, while similar capabilities in time-variant dynamic scenes remain limited. To solve this problem, we propose 4D-Editor, an interactive semantic-driven editing framework, allowing editing multiple objects in a dynamic NeRF with user strokes on a single frame. Specifically, we extend the original dynamic NeRF by incorporating Hybrid Semantic Feature Distillation to maintain spatial-temporal consistency after editing. In addition, a Recursive Selection Refinement module is presented to significantly boost object segmentation accuracy within a dynamic NeRF to aid the editing process. Moreover, we develop Multi-view Reprojection Inpainting to fill holes caused by incomplete scene capture after editing. Extensive quantitative and qualitative experiments on real application scenarios demonstrate that 4D-Editor achieves photo-realistic editing on dynamic NeRFs. Project page: https://patrickddj.github.io/4D-Editor

1. Introduction

Neural Radiance Field (NeRF [35]) and its following works [6, 10, 14, 18, 31, 39, 43, 49, 52] enable free-viewpoint photographic rendering on real-world scenes, and attract wide range interest from the community. With the development of neural reconstruction and rendering, it is highly required to edit the implicit neural representations for downstream applications, *e.g.*, VR/AR, computer animation, and education. Semantic-NeRF [69] utilizes existing semantic labels for scene understanding and facilitates object-level editing, while it requires manual labels. Therefore, some recent studies [15, 20, 54] adopt semantic distillation to extract 3D semantic features in a self-supervised way from large pre-trained models like DINO [5] or LSeg [23], which can generate open-vocabulary scene semantic labels as prior information. Nevertheless, those approaches [15, 20, 54, 69] are limited to 3D static NeRFs, and fail to edit time-variant dynamic scenes.

Our goal is to interactively edit multiple target objects within dynamic NeRFs. MonoNeRF [51] and NeuPhysics [45] allows partially editing dynamic scenes while it simply decomposes a scene into the dynamic foreground and the static background, meaning that the foreground and background are treated as distinct entities, allowing only the complete editing of either the foreground or background. Thus, object-level editing within a scene containing multiple objects is not supported.(*e.g.*, recolor a street sign in a static background or remove one specific car when existing multiple moving cars). Moreover, NeuPhysics relies on a mesh proxy for editing and does not provide any editing interface, which is not user-friendly. Alternatively, NSG [40] and Total-Recon [48] model each foreground object within a scene as a individual NeRF to support object-level editing. However, they model the whole background as a single NeRF model, which means there is no way to edit objects in the background. The differences between our work and existing works are listed in Table 1.

In this paper, we propose an interactive object-level editing framework for dynamic NeRFs, named 4D-Editor. 4D-Editor enables users to select and edit different objects by strokes on one reference frame, and then the modification effect will propagate to all novel views. However, it is difficult to apply editing on one single frame to the entire dynamic scene directly. Therefore, we employs Hybrid Semantic Feature Distillation to extract and distill 2D semantic information from a pre-trained DINO model into hybrid semantic radiance fields. These fields store 3D and 4D semantic features separately to aid object segmentation and editing process and maintain multi-view and time-variant consistency after editing operations. It is also challenging to achieve precise selection on target objects according to

† Corresponding Author

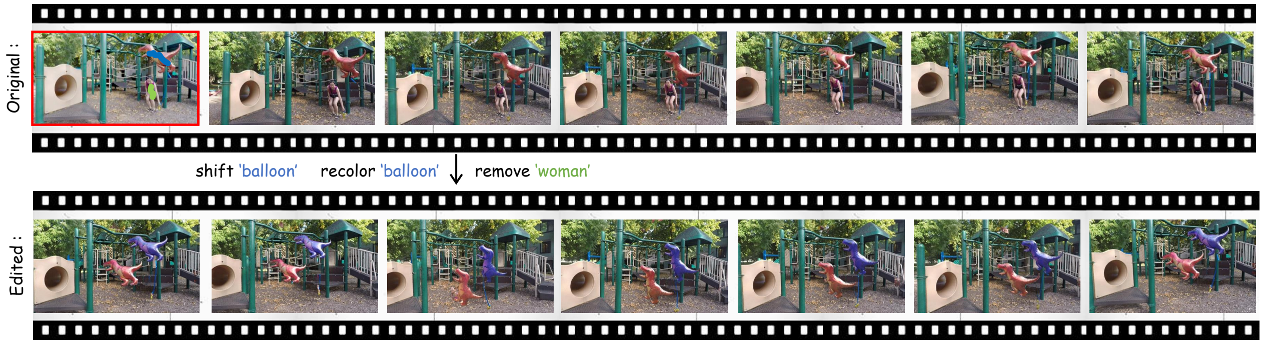


Figure 1. 4D-Editor can interactively edit objects in a dynamic NeRF. For example, with strokes drawn on a reference frame by users, 4D-Editor can remove the human, recolor and shift the balloon within the dynamic NeRF. After the novel view synthesis, the women disappears, the balloon is changed into purple and shifted in both spatial and temporal dimensions.

sparse user brushes, due to ambiguous semantic information distilled from low-resolution DINO outputs. To solve this mismatch, we propose Recursive Selection Refinement method to ensure accurate object segmentation. Additionally, deletion without any refinement can lead to *holes* in background(Fig. 6a) when observation views are limited or foreground objects are too large. However, both the existing static NeRF inpainting method [14, 51] and direct inpainting method [37] result in obvious artifacts in such cases. For refinement on background after deletion, we propose Multi-view Reprojection Inpainting strategy, which completes visible parts through static NeRF and generating invisible parts using an inpainting model.

We evaluate 4D-Editor's ability to edit a dynamic NeRF on Dynamic View Synthesis [66] and DAVIS [44] datasets. The experimental results demonstrate that 4D-Editor outperforms N3F [53] baselines and is capable of editing on complex and static objects, which is difficult for Total-Recon [48]. Moreover, 4D-Editor also achieves best object segmentation compared with extended ISRF [15], SAFF [26] and D^{2}NeRF [60] on dynamic scenes.

We summarize our contributions as follows:

- We propose 4D-Editor, which supports interactive and object-level editing on multiple objects within a dynamic NeRF. 4D-Editor enables diverse editing operations *e.g.*, deletion, recoloring, transformation, composition.
- We incorporate Hybrid Semantic Feature Distillation to extract semantic information in 4D space from pre-trained DINO models for accurate object segmentation in 4D space and maintaining spatial-temporal consistency after editing operations.
- We propose Recursive Selection Refinement, an effective searching method for rapid and precise selection of target objects in the dynamic NeRF. Our experimental results confirm its accuracy and efficiency.
- Multi-view Reprojection Inpainting is developed to fill holes caused by incomplete scene capture from sparse views, particularly after deletion.

Method	Interactive	Object-Level	Dynamic	Operations
Ours	✓	✓	✓	①②③④
N3F [53]	✓	✓	✓	①
ISRF [15]	✓	✓	✗	①②③④
NeuPhysics [45]	✗	✗	✓	①②③④
MonoNeRF [51]	✗	✗	✓	①④
NSG [40]	✗	✓	✓	①③
Total-Recon [48]	✗	✓	✓	①

Table 1. Comparison with NeRF Editing methods. ① deletion, ② recoloring, ③ transformation, ④ composition

2. Related work

Dynamic Scene Representations. In the past few years, dynamic scene representations based on NeRF [4, 11, 13, 18, 22, 24, 29, 31, 52, 57, 58, 64, 66] or Gaussian [9, 32, 59, 62] have experienced great development both in terms of reconstruction quality and training speed. DyNeRF [24] is an early contribution to this field, introducing expressive time-variant latent codes into implicit volume representations for reconstructing dynamic scenes. However, the training process requires 7 days and is extremely slow. Thus, explicit volume representation methods like K-Planes[11] and D-TensoRF [18] use multiple low-dimension planes or vectors to represent the 4D dynamic scene, greatly accelerating training and rendering speed. Different from directly learning dynamic scene representation, DynamicNeRF [66], RobustNeRF [31], and MixVoxels [57] divide the scene into static and dynamic parts and utilize hybrid radiance field representation to model them separately.

Scene Decomposition. Some works mainly focus on decomposition and understanding of NeRF-based scenes, which enables downstream tasks such as segmentation or editing. For dynamic scenes, NSFF [25] and D^{2}NeRF [60] address foreground segmentation by training decoupled static NeRF and dynamic NeRF in a self-supervised way. SAFF [26] employs additional semantic and attention networks and uses saliency-aware clustering to decompose the scene, resulting in a better performance than NSFF and D^{2}NeRF. However, SAFF performs object segmentation by calculating the similarity to stored salient clusters, which may result in erroneous selection. Other methods introduce semantic features into NeRF for spatial semantic segmentation [7, 8, 12, 21, 27, 33, 41, 55, 69]. Semantic-NeRF [69] encodes scene semantic information with appearance and geometry, achieving interactive segmentation using semantic label propagation. PNF [21] optimizes an object-aware neural scene representation that decomposes a scene into a set of objects and background using pseudo-supervision from predicted semantic segmentation. However, these methods require manual annotated labels. In contrast, N3F [54], ISRF [15] and DFF [20] treat pre-trained semantic models(*e.g.*, DINO[5]) as a teacher and distill 2D semantic features into feature fields in a self-supervised manner, enabling object segmentation and editing in 3D space. We follow these methods and incorporate semantic distillation into dynamic NeRFs.

Scene Editing. One direct approach to editing NeRF is to convert it into textured mesh [30, 61, 63, 67], enabling the use of existing 3D editing tools (*e.g.*, Blender [3]). However, this method cannot handle large scenes with high quality and is limited to static scenes. Researchers also utilize language-guided models like CLIP model [46], diffusion model [2, 17, 34, 38, 42, 47, 68], or combine an editing field [1, 16] to perform creative editing using text/image patches. Nevertheless, all above methods are limited to editing static NeRFs. Regarding editing on dynamic NeRF, NeuPhysics [45] utilizes time-invariant signed distance function (SDF) with a deformation field to reconstruct dynamic scenes. MonoNeRF [51] enables deletion or composition in dynamic scenes. Both of them can only edit the entire foreground or background, and cannot edit individual objects separately. NSG [40] and Total-Recon [48] utilize multiple NeRFs to model distinct objects within a scene. This prevents editing on targets that are not pre-modeled during reconstruction, and the lack of interactivity is user-friendless. In contrast, our proposed 4D-Editor directly interacts with arbitrary targets using 2D-4D feature matching and firstly supports applying different editing operations on multiple objects, with spatial-temporal consistency. There are several methods [19, 28, 36, 37, 56, 65] concentrate on completing the background after object removal. SPIn-NeRF[37] utilizes an inpainting model [50] to fill empty background, after which these inpainted images are incorporated as updated training sets to retrain NeRF. However, these methods ignore multi-view consistency, leading to artifacts and inconsistencies in the 3D reconstruction. MonoNeRF [51] and DynNeRF [14] use filtered static parts to construct static fields. Despite ensuring multi-view consistency, these methods still leave holes (Fig. 6a) due to the omission of invisible areas that were not captured during the reconstruction process. We design a simple, effective strategy, combining the advantages of both methods.

3. Method

In this section, we first briefly introduce the background of hybrid radiance field representation of dynamic scenes (Sec. 3.1). Subsequently, we propose our Hybrid Semantic Features Distillation method which serves as a guidance for editing one dynamic NeRF (Sec. 3.2). We then detail the interactive editing pipeline including K-means clustering, 2D-4D feature matching, recursive selection refinement (Sec. 3.3) and editing module (Sec. 3.4). Finally, we demonstrate the Multi-view Reprojection Inpainting method for hole-filling after editing (Sec. 3.5). The proposed framework is shown in Fig. 2.

3.1. Preliminary: Hybrid Radiance Field Reconstruction of Dynamic Scenes

Hybrid radiance field representations of dynamic scenes, such as DynNeRF [14] and RoDyn-NeRF [31], usually consist of a static radiance field F^s and a dynamic radiance field F^d. Given a 3D position $\mathbf{x} \in \mathbb{R}^3$ with its normalized viewing direction $\mathbf{d} \in \mathbb{R}^3$ and time $t \in \mathbb{R}$, F^s maps radiance values as time-invariant density σ^s and RGB color $\mathbf{c}^s$ for static background. While F^d maps radiance values as time-variant density σ^d and RGB color $\mathbf{c}^d$ for dynamic foreground. Furthermore, F^d also predicts blending weight b for blending the output of F^s and F^d.

$$(\sigma^s, \mathbf{c}^s) = F^s(\mathbf{x}, \mathbf{d}), (\sigma^d, \mathbf{c}^d, b) = F^d(\mathbf{x}, \mathbf{d}, t) \tag{1}$$

While F^d maps radiance values as time-variant density σ^d and RGB color $\mathbf{c}^d$ for dynamic foreground. Furthermore, F^d also predicts blending weight b for blending the output of F^s and F^d:

The calculated density and color are then used in volume rendering alone the ray $\mathbf{r}$ emitted from the camera to obtain corresponding pixel color:

$$\begin{aligned}\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T(u_i) \cdot (\alpha(\sigma^s(u_i)\delta_i) \cdot \mathbf{c}^s(u_i) \cdot b \\ +\alpha(\sigma^d(u_i)\delta_i) \cdot \mathbf{c}^d(u_i) \cdot (1-b))\end{aligned} \tag{2}$$

$$T(u_i) = exp\left(\sum_{j=1}^{i-1} \sigma^s(u_j)\delta_j b + \sigma^d(u_j)\delta_j(1-b)\right) \tag{3}$$

where $\alpha(z) = 1 - exp(-z)$, $\delta_i = u_{i+1} - u_i$ is the distance between two neighbor sampled points along the ray, the N points $\{u_i\}_{i=1}^{N}$ are uniformly sampled between near plane and far plane [35], and $T(u_i)$ indicates the accumulated transmittance.

3.2. Hybrid Semantic Features Distillation

In order to maintain the spatial-temporal consistency during object-level editing, we extend original NeRFs and use two

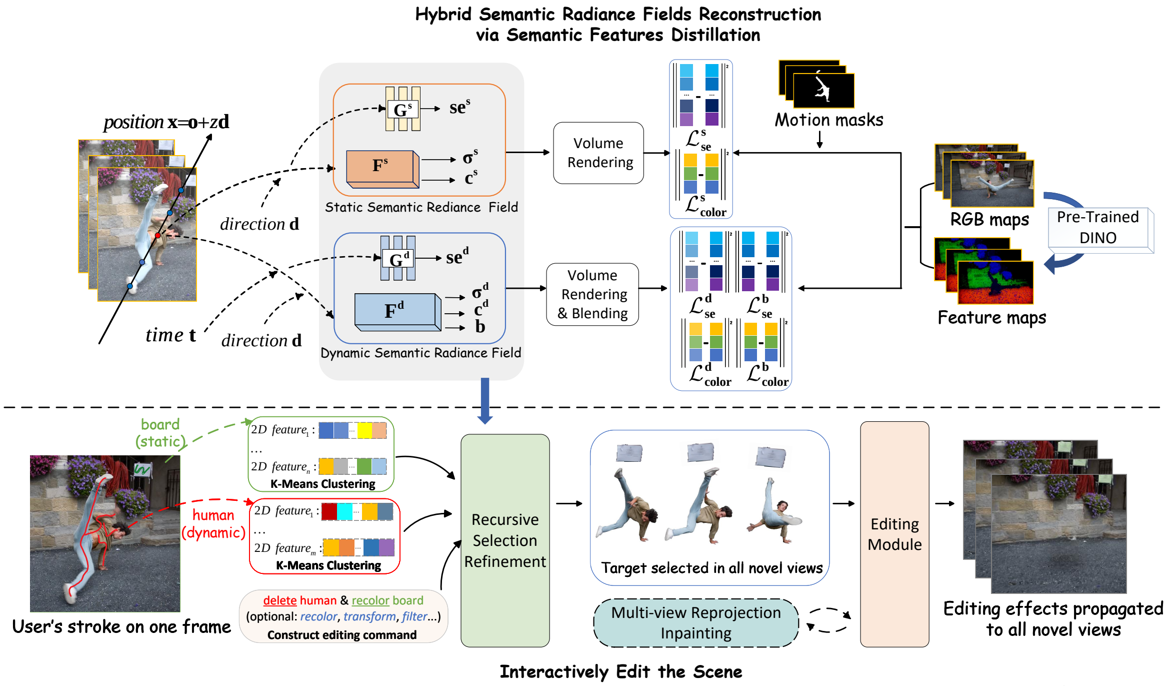

Hybrid Semantic Radiance Fields Reconstruction
via Semantic Features Distillation
position x=o+zd
direction d
time t
direction d
G^s
se^s
F^s
σ^s
c^s
Static Semantic Rediance Field
G^d
se^d
F^d
σ^d
c^d
b
Dynamic Semantic Radiance Field
Volume Rendering
Volume Rendering & Blending
$\mathcal{L}^s_{se}$
$\mathcal{L}^s_{color}$
$\mathcal{L}^d_{se}$
$\mathcal{L}^b_{se}$
$\mathcal{L}^d_{color}$
$\mathcal{L}^b_{color}$
Motion masks
RGB maps
Pre-Trained DINO
Feature maps
board (static)
human (dynamic)
2D feature₁ :
2D featureₙ :
K-Means Clustering
2D feature₁ :
2D featureₘ :
K-Means Clustering
delete human & recolor board
(optional: recolor, transform, filter...)
Construct editing command
User's stroke on one frame
Recursive Selection Refinement
Target selected in all novel views
Multi-view Reprojection Inpainting
Editing Module
Editing effects propagated to all novel views
Interactively Edit the Scene

Figure 2. **4D-Editor framework overview.** The dynamic scene is represented by hybrid radiance fields F^s, F^d and corresponding semantic fields G^s and G^d which are distilled from DINO teacher model. A user can choose a reference view, mark on desired edited objects and assign desirable operations. For example, **delete** dynamic human(red[~] stroke) and **recolor** static board(green[~] stroke). Then, two groups of 2D semantic features are collected and clustered by K-Means to recursively match the corresponding 4D features in dynamic NeRF to achieve precise object segmentation. Finally, editing operations on these objects are applied and spread to the whole NeRF(the human disappeared and the board turned green).

additional fields to store semantic features of static and dynamic parts of a scene, respectively, which are denoted as G^s and G^d. We employ a large pre-trained teacher model (*e.g.*, DINO[5]) to distill semantic features into two semantic fields, serving as guidance for editing. Therefore, the static semantic radiance field is represented by F^s and G^s that stores time-invariant radiance and semantic features. Similarity, the dynamic semantic radiance field is represented by F^d and G^d that stores time-variant features, as shown in Fig. 2.

We obtain the time-invariant semantic feature $\mathbf{se}^s \in \mathbb{R}^C$ and time-variant semantic feature $\mathbf{se}^d \in \mathbb{R}^C$ of a sampled point according to follows:

$$\mathbf{se}^s = G^s(\mathbf{x}), \quad \mathbf{se}^d = G^d(\mathbf{x}, t) \tag{4}$$

Note that we disregard view direction $\mathbf{d}$ due to the direction-agnostic nature of scene semantics. Specifically, given a set of N consecutive frames $\mathcal{I} : \{I_i\}_{i=1}^N \in \mathbb{R}^{H\times W\times 3}$, we utilize the DINO ViT-b8 model to generate corresponding semantic feature maps $\in \mathbb{R}^{H/8\times W/8\times C}$. Then, we upsample these low-resolution feature maps through an upsampling layer to output the final feature maps $\{Se_i\}_{i=1}^N \in \mathbb{R}^{H\times W\times C}$. Next, we employ volume rendering to obtain the pixel-aligned semantic features $\hat{Se}^s(\mathbf{r})$ and $\hat{Se}^d(\mathbf{r})$, which represent the accumulated semantic feature along a ray $\mathbf{r}$:

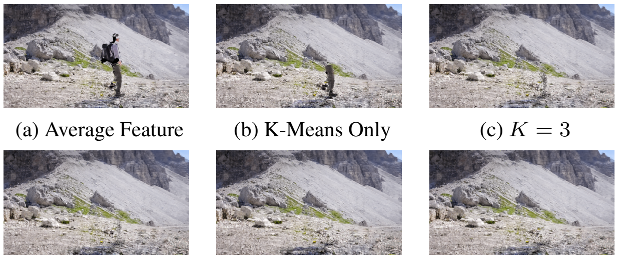
$$\begin{aligned}\hat{Se}^s(\mathbf{r}) &= \sum_{i=1}^{N} T^s(u_i)\alpha(\sigma^s(u_i)\delta_i)\mathbf{se}^s(u_i),\\ \hat{Se}^d(\mathbf{r}) &= \sum_{i=1}^{N} T^d(u_i)\alpha(\sigma^d(u_i)\delta_i)\mathbf{se}^d(u_i)\end{aligned} \tag{5}$$

where $T^s(u_i) = exp(\sum_{j=1}^{i-1}\sigma^s(u_j)\delta_j)$ and $T^d(u_i) = exp(\sum_{j=1}^{i-1}\sigma^d(u_j)\delta_j)$.

Finally, we calculate the final semantic feature by blending semantic features of G^s and G^d outputs, similar to Equation 2:

$$\begin{aligned}\hat{Se}^b(\mathbf{r}) = \sum_{i=1}^{K} T^b(u_i)(\alpha(\sigma^s(u_i)\delta_i)\mathbf{se}^s(u_i)b\\ +\alpha(\sigma^d(u_i)\delta_i)\mathbf{se}^d(u_i)(1-b))\end{aligned} \tag{6}$$

We add three new losses to train G^s and G^d: $\mathcal{L}^s_{se}$ for pixels belonging to the static part, $\mathcal{L}^d_{se}$ and $\mathcal{L}^b_{se}$ for all pixels. We treat the output of the teacher model as the ground truth. By minimizing the difference between predicted features and the ground truth, these two semantic fields can



(a) Average Feature (b) K-Means Only (c) $K = 3$

(d) $K = 5$ (e) $K = 10$ (f) $K = 30$

Figure 3. **Refinement with different recursive number.** Compared to **(a)** and **(b)**, our method **(c)-(f)** achieve great improvement in object removal. With larger recursion number, artifacts are eliminated, while maintaining spatial-temporal consistency in other areas. Similar to the convergence of ray-tracing, our method also achieve high quality when $K = 10$ ($\alpha = 0.6, \beta = 0.1$). The benefits generated by a larger K are very small.

learn scene semantics.

$$\mathcal{L}_{se}^s = \sum_{\mathbf{r}\in\mathcal{R}^s} \left\|Se(\mathbf{r}) - \hat{Se}^s(\mathbf{r})\right\|_2^2,$$
$$\mathcal{L}_{se}^d = \sum_{\mathbf{r}\in\mathcal{R}^s+\mathcal{R}^d} \left\|Se(\mathbf{r}) - \hat{Se}^d(\mathbf{r})\right\|_2^2, \tag{7}$$
$$\mathcal{L}_{se}^b = \sum_{\mathbf{r}\in\mathcal{R}^s+\mathcal{R}^d} \left\|Se(\mathbf{r}) - \hat{Se}^b(\mathbf{r})\right\|_2^2$$

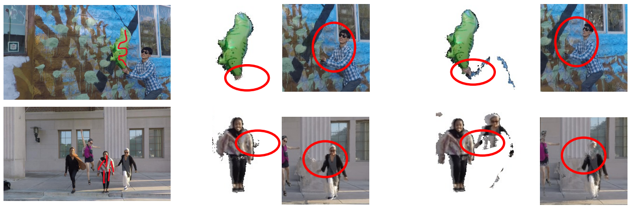
where $\mathcal{R}^s$, $\mathcal{R}^d$ are sampled rays from static and dynamic part of a scene, respectively. The total semantic loss is

$$\mathcal{L}_{se} = \mathcal{L}_{se}^s + \lambda_1\mathcal{L}_{se}^d + \lambda_2\mathcal{L}_{se}^b \tag{8}$$

3.3. Recursive Selection Refinement

K-Means Query for Multi-Objects Selection. In 4D-Editor, users can mark on multiple objects on a reference frame. Based on user's strokes, 4D-Editor extracts target 2D semantic features from corresponding feature maps generated by DINO. These semantic features are utilized to construct different queries for matching multiple objects in semantic fields G^s and G^d. However, the user-provided strokes are sparse, resulting in naturally insufficient and inexpressive semantic features, where a simple query such as averaging features (Fig. 3a) may cause incorrect 2D-4D feature matches. Therefore, inspired by ISRF[15], we utilize K-Means to group the most significant features for each individual object, aiming to improve the accuracy of feature matching.

Recursive Refinement. Despite employing K-Means for effective 2D-4D feature matching, achieving accurate object segmentation still remains challenging (Fig. 3b): Owing to 8x up-sampled feature maps, unsupervised semantic feature distillation inherently leads to imprecise 4D semantic information, especially for the semantic confusion between object edges and background.



(a) Reference Frame (b) RSR (Ours) (c) Bilateral Search (ISRF)

Figure 4. **Comparison with Bilateral Search in ISRF [15].**

To solve this problem, we propose recursive selection refinement(RSR) algorithm which recursively refines selection instead of relying on a fixed threshold. Thus, we not only avoid heavy manual threshold setting but also achieve precise object selection, especially on object edges. Specifically, we add exploration range β and divide all sampled points into three sets according to the feature distance $\boldsymbol{d}$: (1) Valid point set $\mathcal{V}(u_i \in \mathcal{V}, d(\gamma, u_i) \leq \alpha)$ (2) Possible point set $\mathcal{P}(u_i \in \mathcal{P}, \alpha < d(\gamma, u_i) \leq \alpha + \beta)$. (3) Impossible point set $\mathcal{Q}(u_i \in \mathcal{Q}\ , d(\gamma, u_i) > \alpha + \beta)$. Note that our proposed RSR algorithm is not sensitive to parameter β (supplementary material Sec.5).

Then, we distinguish these sampled points truly belonging to the target object in possible point set $\mathcal{P}$. We record their original indexes and apply random offsets to points in $\mathcal{P}$ to nudge them into valid point set and impossible point set, thereby forming new valid point set $\mathcal{V}'$, possible point set $\mathcal{P}'$, and impossible point set $\mathcal{Q}'$. We repeat this recursive refinement process until reaching the maximum recursion number K or the possible point set is empty. Finally, we can obtain the approximate unbiased estimation on the object segmentation (or object selection) as demonstrated in Fig. 3. The visualization of RSR algorithm and the reason for final convergence are in supplementary material Sec.4.

We compare RSR with Bilateral Search proposed in ISRF [15], which firstly identifies high-confidence regions and expands them based on feature disparities and spatial proximities. Explicitly, the features of the newly selected areas are used for the subsequent query. However, if there are connected objects, the iteration becomes hard to terminate and results in redundant selections due to semantic ambiguity of such areas, as illustrated in Fig. 4c. In contrast, we ensure precise selection by maintaining globally consistent feature queries in RSR and confirming the validity of points based on the expected probability in regions with semantic confusion, as in Fig. 4b. Additionally, RSR can be utilized with a variety of NeRF structures, *e.g.*, pure implicit NeRF and NeRF with voxels. However, Bilateral Search is limited to voxel grid.

3.4. Editing Module

After the object segmentation, for each target object, we can obtain two sampling point sets: $\mathcal{T}$ inside the object and $\mathcal{S}$

outside. We can edit the specific object as follows:

Remove. Removing one object in hybrid semantic radiance fields, actually means that we need to treat the object as transparent in order to expose the background behind the object. With reconstruction of hybrid semantic features fields ahead, we can use static or dynamic semantic fields according to the dynamics of the target object to achieve more accurate object segmentation. For sampling point $t_i \in \mathcal{T}$ of a static object, we set the density $\sigma^s(t_i) = 0$, so that the point will be ignored during volume rendering. As for the dynamic object, we not only set density $\sigma^d(t_i) = 0$ but also set blending weight $b(t_i) = 1$. This is because we expect the dynamic object removal operation not to affect the static field.

Filter. To filter an object, we set $\sigma(s_i) = 0$ where $s_i \in \mathcal{S}$, which means making all points outside the object invisible.

Composite. We can composite objects filtered from other scenes($\mathcal{Z}$) into the current scene by setting $\sigma(s_i) = \sigma(z_i)$, $c(s_i) = c(z_i)$ where $z_i \in \mathcal{Z}$(supplementary material Sec.6).

Recolor. We find editing color $\mathbf{c}(s_i)$ in 4D space is the similar to that on 2D images: we exchange RGB channels to change hue parameter, change the corresponding RGB channel to change RGB saturation parameter, and add all RGB channels to change lightness parameter. The recoloring results are showed in supplementary material Sec.6.

Transform. We allow users to apply various self-defined transforming operations to the object, such as translating, scaling, mirroring or duplicating. Users only need to define a transforming function $mapping(\mathbf{x}) : \mathbf{x} \rightarrow \mathbf{x}'$ to set the spatial or temporal mapping relation of the target object (*e.g.*, $mirror(x, y, z) : (-x, -y, z)$, $reverse(x, y, z, t) : (x, y, z, -t)$). Then, we set $\sigma(mapping(s_i)) = \sigma(s_i)$, $\mathbf{c}(mapping(s_i)) = \mathbf{c}(s_i)$ for volume rendering. The detailed experimental results can be seen in Fig. 9 .

3.5. Multi-view Reprojection Inpainting

In cases where observations views are limited or foreground objects are large, the removal operation without refinement may cause "holes" in the novel views. This limitation is observed in static field inpainting methods like DynNeRF [14] and MonoNeRF [51] in Fig. 6a, as they overlook invisible background regions during reconstruction. However, direct inpainting on motion masks and retraining NeRF, as in SPIn-NeRF [37], can introduce multi-view inconsistency, leading to blurry artifacts and distortion of original contents of the scene in Fig. 6b.

Therefore, we propose multi-view reprojection inpainting method, combining the advantages of both methods. As Fig. 5 shows, for one image under a certain pose, we divide its empty areas(foreground motion masks) into two parts: visible background $\mathcal{J}_{vis}$, invisible holes $\mathcal{J}_{invis}$: area $\mathcal{J}_{vis}$ is invisible in the current timestamp while visible in other timestamps, while area $\mathcal{J}_{invis}$ remains invisible in all timestamps. We fill $\mathcal{J}_{vis}$ with static NeRF to preserve the original scene content as much as possible, and minimize the area requiring generation. This belief stems from the notion that, given a specific perspective, if a occluded 3D point is present in some other views, its geometry and appearance information are inherently captured during NeRF's reconstruction, thereby enabling inherent filling by NeRF. Conversely, if the occluded points lack observation in other views, then it is necessary to apply generative inpainting.

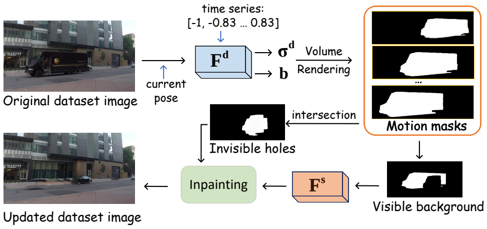
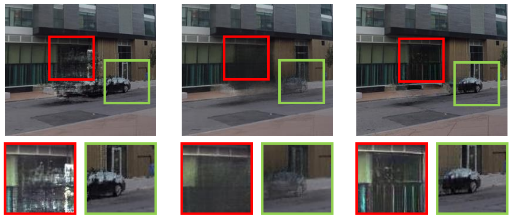


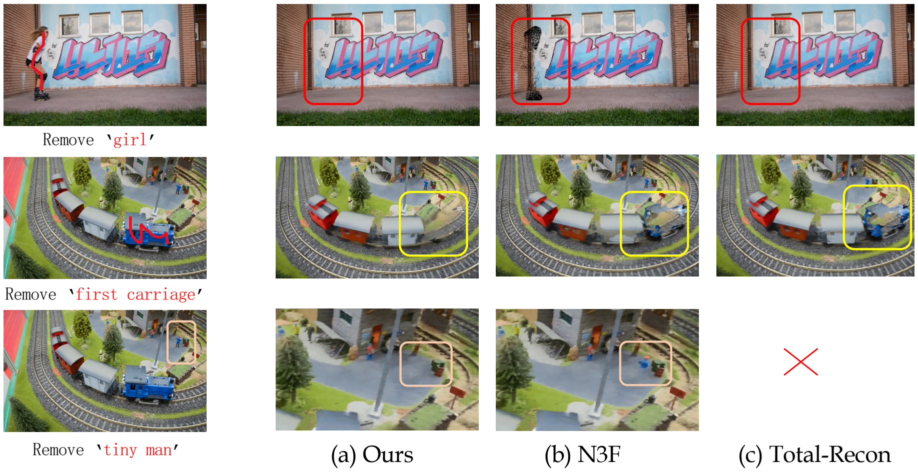
Figure 5. **Updating via Multi-view Reprojection Inpainting.**



(a) w/o inpainting (b) Inpainting w/o reprojection (c) Ours

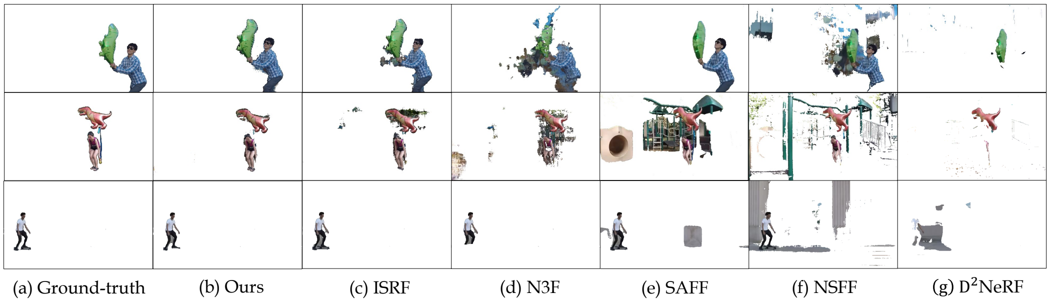
Figure 6. **Comparison of different inpainting methods.**

We distinguish visible background $\mathcal{J}_{vis}$ and invisible holes $\mathcal{J}_{invis}$ from time-variant motion masks, for each pose in the original dataset. These masks are generated by performing volume rendering on the blending weight b across the entire time series under the current camera pose. The overlapping area of these masks represents $\mathcal{J}_{invis}$, while the remaining regions constitute $\mathcal{J}_{vis}$. For inpainting $\mathcal{J}_{vis}$, we eschew the traditional pixel reprojection method due to its drawbacks such as potential loss of fine details and sensitivity to geometric inconsistencies or occlusions between views. Instead, we leverage NeRF's inherent multi-view information to accomplish the inpainting of $\mathcal{J}_{vis}$ in all training images. In this way, these areas are filled with rendering results from F^s (Fig. 5). Subsequently, lama model [50] is utilized for inpainting on the remaining invisible parts, $\mathcal{J}_{invis}$. By pre-filling the background, we narrow down areas that need to be generated, which strengthens the reliability of inpainting results. Static radiance field F^s is then retrained based on these inpainted images, after which the hybrid semantic radiance fields are also retrained. As Fig. 6 shows, our proposed inpainting method demonstrates enhanced inpainting results in comparison with static NeRF(Fig. 6a) or direct inpainting (Fig. 6b) only.



(a) Ours (b) N3F (c) Total-Recon

Figure 7. **Comparison on Deletion.** With user-provided strokes on a reference frame, we search for semantically similar regions. Subsequently, the deletion operation affects all novel views accordingly. Compared to N3F and Total-Recon, 4D-Editor achieves cleaner deletion without artifacts and supports object-level editing on the background.



(a) Ground-truth (b) Ours (c) ISRF (d) N3F (e) SAFF (f) NSFF (g) D^{2}NeRF

Figure 8. **Qualitative comparison of 4D-Editor's foreground segmentation in 4D space.**

	Balloon1	Balloon2	Playground	Skating	Jumping	Umbrella	Average
SAFF [26]	93.01	91.54	51.22	53.20	82.11	92.10	77.19
NSFF [25]	75.23	51.84	75.63	35.64	26.52	67.23	55.34
D^{2}NeRF [60]	47.41	39.86	52.35	32.37	62.39	59.52	48.98
N3F [53]	78.96	61.13	56.52	64.11	65.53	67.92	65.86
ISRF [15]	83.42	85.97	82.38	87.51	76.54	81.19	82.83
Ours	90.62	93.19	90.26	89.68	85.31	89.77	89.80

Table 2. **The IoU performance of foreground segmentation on Dynamic View Synthesis Dataset [66]**

4. Experiments

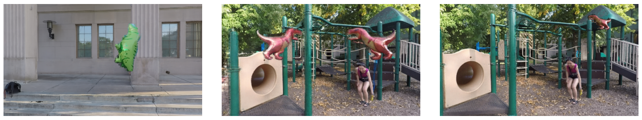
4.1. Experimental setup

Datasets. We experiment on two datasets: Dynamic View Synthesis [66], DAVIS [44].

Implements Details. We implement 4D-Editor with Pytorch, using Adam optimizer to update learnable parameters on one NVIDIA A6000 GPU. We train original hybrid NeRFs and semantic parts separately, and time for hybrid semantic features distillation is only 10-15 minutes. Editing one frame can take 1 seconds based on RobustNeRF and 7 seconds based on DynNeRF.

4.2. Interactive Object-level Editing

Our method enables users to perform interactive object-level editing with strokes. As Fig. 7 shows, the user simply annotates the target object on a reference frame in original videos(Column 1). After constructing the editing command (*e.g.*, removal), this operation can be propagated to the whole scenes. Compared with N3F [54], our method can achieve cleaner removal results, whereas N3F leaves obvious artifacts. Since NeuPhysics[45] or MonoNeRF [51] can only edit the whole foreground and not support object-level editing, we make no comparison with them. Total-Recon [48] uses a *background field* to represent all static objects, it cannot edit on these objects, which are not premodeled ahead(Fig. 7c line 3: the tiny man in static background). Moreover, Total-Recon leaves obvious artifacts af-



(a) composition (b) mirror (c) scale

Figure 9. **Composition & Transformation.** 4D-Editor supports composition across different scenes and flexible transformations(More results in supplementary material Sec.6).

ter deleting *first carriage* (Fig. 7c line 2). This is because Total-Recon must use 4 object fields for each carriage to support object-level editing, and such method hard to handle connected objects that occlude each other.

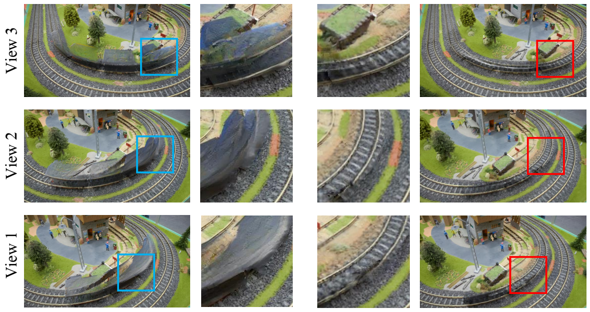
Besides, 4D-Editor also supports editing multiple objects, such as deleting the women, recoloring, and shifting the balloon simultaneously, as demonstrated in Fig. 1. 4D-Editor offers flexibility by allowing users to define their own transformation functions. Fig. 9 gives some examples of object-level composition across multiple scenes and a variety of transformations. The green balloon is actually filtered from another scene and inserted into the playground in Fig. 9a. As for different transformation operations, we can still keep the correct spatial information.

4.3. Foreground Segmentation

Since scene decomposition methods like SAFF [26], NSFF [25] and D^{2}NeRF [60] cannot produce object-level masks, we evaluate on foreground segmentation as shown in Fig.8. SAFF simply clusters the semantic features of spatial points, but the semantic information distilled from 2D features does not entirely align with actual spatial distribution. Consequently, background elements are included in foreground segmentation(Fig. 8e). NSFF and D^2 NeRF utilize blending weight v for reconstructing dynamic scenes. However, this parameter does not directly indicate whether the current spatial point is dynamic or not. Therefore, this causes ambiguity when distinguishing foreground and background(Fig. 8f, Fig. 8g). As demonstrated in Fig. 4, Bilateral Search from ISRF struggles to handle connected objects with semantic ambiguity. Additionally, Table. 2 demonstrates that 4D-Editor achieves higher IoU performance on foreground segmentation, compared with N3F and ISRF. Fig. 4 shows that our methods achieve higher segmentation quality for individual objects. Further examples are provided in supplementary material Sec. 5.

4.4. Ablation Studies

Recursive Selection Refinement. In Fig. 3, we evaluate three different methods for 2D-4D feature matching: (1) Average feature, (2) K-Means clustered features [15], and (3) Recursive selection refinement (our method). Fig. 3a illustrates ineffective feature matching resulting from the limited capability of average features to extract meaningful features. K-Means clustered features only remove part of object as shown in Fig. 3b, leading to artifacts(*e.g.*, remaining legs). However, RSR improves the precise of feature matching and achieves nearly perfect removal, as demonstrated in Fig. 3c-3f. We achieve 93% Acc and 79% IoU, in supplementary material Sec. 5.



(a) Inpainting w/o reprojection (b) Ours

Figure 10. **Visulaizations of multi-view consistency.** The left column demonstrates that direct inpainting causes view-inconsistent with obvious artifacts, while our method (the right column) exhibits better multi-view consistency.

Multi-view Reprojection Inpainting. Fig. 6 shows that Multi-view Reprojection Inpainting not only fills holes(Fig. 6a), but also maintains original contents in the scene as much as possible(compared with SPIn-NeRF in Fig. 6b). Moreover, Fig. 10 proves that we can achieve better multi-view with the reprojection module. The inpainting performance can be further improved by employing more powerful inpainting models.

5. Conclusion

We propose a novel interactive editing framework for dynamic scenes that enables object-level editing operations through user-provided strokes on a single reference frame, and delivers spatial-temporal consistency across the entire time series. We present several excellent results from multiple challenging scenes. However, our method has limitations in removing shadows of moving objects. Additionally, while we can handle invisible background completion, in some times, the scene inpainting may still have spatial-temporal inconsistencies. We will investigate ways to address the problem in future works.

6. Acknowledgement

This work is supported in part by the National Natural Science Foundation of China (No. 62272339) and in part by Tianjin Science and Technology Plan Project(No. 24JCZXJC00050).

References

[1] Chong Bao, Yinda Zhang, Bangbang Yang, Tianxing Fan, Zesong Yang, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Sine: Semantic-driven image-based nerf editing with prior-guided editing field. In *The IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2023. 3

[2] Edward Bartrum, Thu Nguyen-Phuoc, Chris Xie, Zhengqin Li, Numair Khan, Armen Avetisyan, Douglas Lanman, and Lei Xiao. Replaceanything3d:text-guided 3d scene editing with compositional neural radiance fields, 2024. 3

[3] Blender Foundation. Blender. https://www.blender.org/, 2023. Version 3.6. 3

[4] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023. 2

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 1, 3, 4

[6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields, 2022. 1

[7] Jiahao Chen, Yipeng Qin, Lingjie Liu, Jiangbo Lu, and Guanbin Li. Nerf-hugs: Improved neural radiance fields in non-static scenes using heuristics-guided segmentation. *CVPR*, 2024. 3

[8] Yuedong Chen, Qianyi Wu, Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Sem2nerf: Converting single-view semantic masks to neural radiance fields. In *European Conference on Computer Vision*, pages 730–748. Springer, 2022. 3

[9] Yuanxing Duan, Fangyin Wei, Qiyu Dai, Yuhang He, Wenzheng Chen, and Baoquan Chen. 4d gaussian splatting: Towards efficient novel view synthesis for dynamic scenes. *arXiv preprint arXiv:2402.03307*, 2024. 2

[10] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 1

[11] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023. 2

[12] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In *2022 International Conference on 3D Vision (3DV)*, pages 1–11. IEEE, 2022. 3

[13] Wanshui Gan, Hongbin Xu, Yi Huang, Shifeng Chen, and Naoto Yokoya. V4d: Voxel for 4d novel view synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 2

[14] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021. 1, 2, 3, 6

[15] Rahul Goel, Dhawal Sirikonda, Saurabh Saini, and PJ Narayanan. Interactive segmentation of radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4201–4211, 2023. 1, 2, 3, 5, 7, 8

[16] Ori Gordon, Omri Avrahami, and Dani Lischinski. Blended-nerf: Zero-shot object generation and blending in existing neural radiance fields. *arXiv preprint arXiv:2306.12760*, 2023. 3

[17] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3

[18] Hankyu Jang and Daeyoung Kim. D-tensorf: Tensorial radiance fields for dynamic scenes, 2022. 1, 2

[19] Han Jiang, Haosen Sun, Ruoxuan Li, Chi-Keung Tang, and Yu-Wing Tai. Inpaint4dnerf: Promptable spatio-temporal nerf inpainting with generative diffusion models. *arXiv preprint arXiv:2401.00208*, 2023. 3

[20] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *Advances in Neural Information Processing Systems*, 35:23311–23330, 2022. 1, 3

[21] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022. 3

[22] Sacha Lewin, Maxime Vandegar, Thomas Hoyoux, Olivier Barnich, and Gilles Louppe. Dynamic nerfs for soccer scenes. *arXiv preprint arXiv:2309.06802*, 2023. 2

[23] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 1

[24] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, and Zhaoyang Lv. Neural 3d video synthesis from multi-view video, 2022. 2

[25] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 7, 8

[26] Yiqing Liang, Eliot Laidlaw, Alexander Meyerowitz, Srinath Sridhar, and James Tompkin. Semantic attention flow fields for monocular dynamic scene decomposition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 7, 8

[27] Dezhi Liu, Weibing Wan, Zhijun Fang, and Xiuyuan Zheng. Gsnerf: Fast novel view synthesis of dynamic radiance fields. *Computers & Graphics*, 116:491–499, 2023. 3

[28] Hao-Kang Liu, I-Chao Shen, and Bing-Yu Chen. Nerf-in: Free-form nerf inpainting with rgb-d priors, 2022. 3

[29] Jia-Wei Liu, Yan-Pei Cao, Weijia Mao, Wenqiao Zhang, David Junhao Zhang, Jussi Keppo, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Devrf: Fast deformable voxel radiance fields for dynamic scenes. *Advances in Neural Information Processing Systems*, 35:36762–36775, 2022. 2

[30] Ruiyang Liu, Jinxu Xiang, Bowen Zhao, Ran Zhang, Jingyi Yu, and Changxi Zheng. Neural impostor: Editing neural radiance fields with explicit shape manipulation, 2023. 3

[31] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13–23, 2023. 1, 2, 3

[32] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. 2

[33] Xiaoyang Lyu, Chirui Chang, Peng Dai, Yang-Tian Sun, and Xiaojuan Qi. Total-decom: Decomposed 3d scene reconstruction with minimal interaction. *arXiv preprint arXiv:2403.19314*, 2024. 3

[34] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12663–12673, 2023. 3

[35] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 3

[36] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Marcus A. Brubaker, Jonathan Kelly, Alex Levinshtein, Konstantinos G. Derpanis, and Igor Gilitschenski. Reference-guided controllable inpainting of neural radiance fields, 2023. 3

[37] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G Derpanis, Jonathan Kelly, Marcus A Brubaker, Igor Gilitschenski, and Alex Levinshtein. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20669–20679, 2023. 2, 3, 6

[38] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulo, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4328–4338, 2023. 3

[39] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 1

[40] Ost. Neural scene graphs for dynamic scenes. In *CVPR*, pages 2856–2865, 2021. 1, 2, 3

[41] Takashi Otonari, Satoshi Ikehata, and Kiyoharu Aizawa. Entity-nerf: Detecting and removing moving entities in urban scenes. *CVPR*, 2024. 3

[42] Jangho Park, Gihyun Kwon, and Jong Chul Ye. Ed-nerf: Efficient text-guided editing of 3d scene using latent space nerf, 2023. 3

[43] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), 2021. 1

[44] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 7

[45] Yi-Ling Qiao, Alexander Gao, and Ming Lin. Neuphysics: Editable neural geometry and physics from monocular videos. *Advances in Neural Information Processing Systems*, 35:12841–12854, 2022. 1, 2, 3, 7

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[47] Hoigi Seo, Hayeon Kim, Gwanghyun Kim, and Se Young Chun. Ditto-nerf: Diffusion-based iterative text to omni-directional 3d model, 2023. 3

[48] Song. Total-recon: Deformable scene reconstruction for embodied view synthesis. In *ICCV*, 2023. 1, 2, 3, 7

[49] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023. 1

[50] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 3, 6

[51] Tian. MonoNeRF: Learning a generalizable dynamic radiance field from monocular videos. In *ICCV*, 2023. 1, 2, 3, 6, 7

[52] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021. 1, 2

[53] Vadim Tschernezki. Neural feature fusion fields: 3D distillation of self-supervised 2D image representations. In *3DV*, 2022. 2, 7

[54] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In *2022 Inter-*

national Conference on 3D Vision (3DV), pages 443–453. IEEE, 2022. 1, 3, 7

[55] Suhani Vora, Noha Radwan, Klaus Greff, Henning Meyer, Kyle Genova, Mehdi SM Sajjadi, Etienne Pot, Andrea Tagliasacchi, and Daniel Duckworth. Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes. *arXiv preprint arXiv:2111.13260*, 2021. 3

[56] Dongqing Wang, Tong Zhang, Alaa Abboud, and Sabine Süsstrunk. Inpaintnerf360: Text-guided 3d inpainting on unbounded neural radiance fields, 2023. 3

[57] Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, Yafei Song, and Huaping Liu. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19706–19716, 2023. 2

[58] Liao Wang, Qiang Hu, Qihan He, Ziyu Wang, Jingyi Yu, Tinne Tuytelaars, Lan Xu, and Minye Wu. Neural residual radiance fields for streamably free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 76–87, 2023. 2

[59] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Wang Xinggang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint arXiv:2310.08528*, 2023. 2

[60] TH Wu, FC Zhong, A Tagliasacchi, F Cole, and C Oztireli. D2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. arxiv preprint [2022-07-07], 2022. 2, 7, 8

[61] Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In *European Conference on Computer Vision*, pages 597–614. Springer, 2022. 3

[62] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023. 2

[63] Zheyuan Yang, Yibo Liu, Guile Wu, Tongtong Cao, Yuan Ren, Yang Liu, and Bingbing Liu. Learning effective nerfs and sdfs representations with 3d generative adversarial networks for 3d object generation: Technical report for iccv 2023 omniobject3d challenge, 2023. 3

[64] Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023. 2

[65] Youtan Yin, Zhoujie Fu, Fan Yang, and Guosheng Lin. Ornerf: Object removing from 3d scenes guided by multiview segmentation with neural radiance fields, 2023. 3

[66] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5336–5345, 2020. 2, 7

[67] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364, 2022. 3

[68] Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao. Text2nerf: Text-driven 3d scene generation with neural radiance fields, 2023. 3

[69] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. 1, 3