DRAFT: Direct Radiance Fields Editing with Composable Operations

Zhihan Cai *1
cai-zh21@mails.tsinghua.edu.cn
Kailu Wu *1
wkl22@mails.tsinghua.edu.cn
Dapeng Cao2
dapengcao@stu.xjtu.edu.cn
Feng Chen1
chenf20@mails.tsinghua.edu.cn
Kaisheng Ma †1
kaisheng@mails.tsinghua.edu.cn

[1] Tsinghua University
[2] Xi'an Jiaotong University

Abstract

Neural implicit representations, especially neural radiance fields (NeRF), have shown promising results in novel view synthesis of 3D scenes. However, radiance fields often implicitly model the 3D geometry, making direct manipulation challenging for users. Current editing methods often rely on customized NeRFs or employ ray bending, which lacks universality. To address these issues, we propose DRAFT, a novel editing system that translates user edits directly into implicit representations, bypassing the need for ray bending or customized labeling. DRAFT focuses on voxel-based radiance fields, mapping user edits to the latent space defined on the voxel representation. By modeling user edits as two steps - *Selection* and *Modification* - we implement various composable editing operations, including standard operations such as copy-and-paste, rotation, stretching, and removal. Additionally, for more realistic scene editing purposes, we introduce a stamp operation to address the artifacts caused by editing and seam carving on radiance fields to achieve content-aware resizing. Extensive experiments demonstrate that DRAFT produces realistic editing results for both synthetic and real-world scenes.

1 Introduction

Neural implicit scene representations, particularly neural radiance fields (NeRF) [21], have garnered increasing attention for their high-quality representations of 3D scenes. NeRF encodes a static 3D scene as a radiance field using a neural network to model volume density and view-dependent radiance from point samples along cast rays and synthesizes views at arbitrary viewpoints through volume rendering. It has demonstrated promising results in photorealistic novel view synthesis and 3D reconstruction of real-world scenes, inspiring

* Equal Contribution. † Corresponding author.

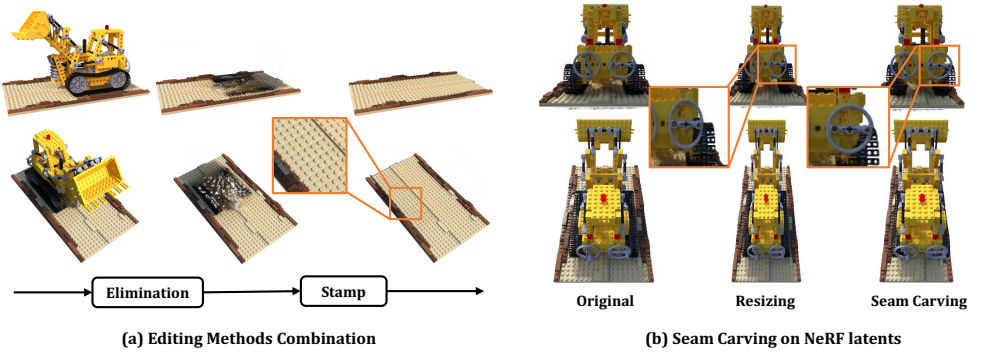


Figure 1: (a) DRAFT enables users to perform a series of modifications. For example, we implement the removal of the Lego Truck followed by a stamp operation to resolve the artifacts. (b) DRAFT supports content-aware resizing based on seam carving.

numerous subsequent works [10, 13, 20, 25, 26, 33, 36] that extended upon it to achieve better visual results and support diverse scenarios.

While neural implicit representation shows great potential in 3D content creation, how to edit a 3D scene based on this representation has become a new exploration direction. However, since radiance fields often implicitly model 3D geometry during training, it is difficult for users to manipulate the 3D content directly. EditNeRF [19] pioneered the concept of editing in radiance fields, but its application is limited to toy examples and falls short in realistic scenes. Object-NeRF [32] advances by separating objects and background, but its accessible operations are limited to object duplication, rotation, and removal. While editing implicit representations poses challenges, some research focuses on editing explicit representations derived from implicit representations, such as NeRF-Editing [34] targets mesh editing and Deforming-NeRF [31] focusing on cages. However, these explicit editing methods primarily adjust the volume rendering process by altering camera rays without directly engaging with neural radiance fields themselves. This limitation hinders integrating these techniques with other editing methods designed for implicit representations. Furthermore, there are challenges in carrying out certain editing tasks with ray-bending, such as seam carving.

In this paper, we study on how to facilitate user-friendly manipulation of realistic 3D content. We propose DRAFT, a system that directly applies user edits to implicit representations, bypassing the need for ray bending. DRAFT supports editing on most voxel-based NeRF methods, such as Plenoxels [33] and DVGO [25]. Our method translates user edits into manipulations on corresponding volume density and features without altering the rendering procedure. We leverage the inherent compatibility and benefits of voxel grids, ensuring gap-free and smooth post-editing of voxel grid features by interpolation. Significantly, DRAFT facilitates a direct correlation between user edits and representation edits, which preserves the representation form, leaves the rendering procedure unmodified, and ensures high-quality rendering results. This consistency allows for the composition of multiple editing operations, such as removal and stamp operation, as depicted in Figure 1(a), enhancing the system's capability to handle complex editing tasks.

Specifically, our editing system, DRAFT, is structured around two primary steps for each operation: *Selection* of the target area and feature *Modification*. We provide two selection

Method	Rigid Deformation	Non-rigid Deformation	Ray Bending	Re-training Requirement	Form Consistency	Applicable NeRFs
NeRF-Editing [34]	✓	✓	✓			Any
Deforming-NeRF [31]	✓	✓	✓			Any
Object-NeRF [31]	✓			✓		Object-NeRF
NeuMesh [9]	✓	✓			✓	NeuMesh
Control-NeRF [18]	✓			✓	✓	Control-NeRF
NSVF [18]	✓				✓	Sparse Voxel-based NeRF
Ours	✓	✓			✓	Voxel-based NeRF

Table 1: Comparison with other editing methods.

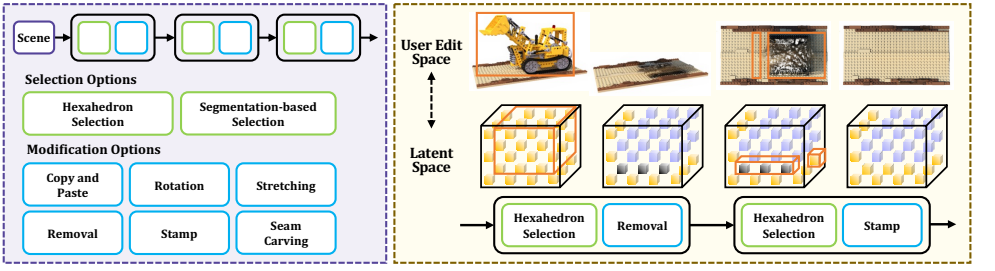
techniques: *Hexahedron Selection* and *Segmentation-based Selection*. The modification step includes various standard operations, such as copy-paste and partial rotation. Additionally, we introduce stamp operation to effectively resolve the artifacts due to the shape part removal and 3D seam carving on voxels with local features to achieve high-quality content-aware resizing of 3D scenes, as illustrated in Figure1(b). We believe that our proposed editing system is practical and can be widely adopted, especially with the recent emergence of NeRF methods based on voxel representations. In summary, this paper has the following contributions:

- We propose a new system DRAFT for editing voxel-based radiance fields with composable operations. Importantly, representations before and after editing remain consistent, which allows further refinement of edited scenes by other works.
- We are the first to explore the application of seam carving to radiance fields, enabling high-quality content-aware resizing of 3D scenes.
- We implement different kinds of editing operations and demonstrate the effectiveness on various synthetic and real-world datasets.

2 Related work

2.1 Editing methods for Neural Radiance Fields

In NeRF editing, the majority of approaches design customized radiance fields to enable various editing capabilities, including but not limited to modifying facial expressions [1, 9], adjusting human poses [22, 30], applying stylization effects [11], decomposing objects [15], editing textures [3], performing text-driven editing [27], and manipulating toy objects [19]. Additionally, certain methods [23, 31, 34] modify the ray path to effect editing changes, without necessitating a particular type of NeRF. However, these methods are limited to visualizing edited scenes and cannot export the edited scenes. Most relevant to us, NeRFShop [12] introduced an interactive framework for manipulating NeRFs, but it only incorporated some existing editing methods. NSVF [18] proposes a method for editing sparse voxel fields by copying or moving voxels. However, this technique is limited to performing rigid deformations and cannot rotate objects. In contrast, our method enables both rigid and non-rigid deformations on most voxel-based radiance fields. We make a detailed comparison with other articles in Tab. 1. Rigid deformation here involves voxel translation without affecting the radiance through rotation. Conversely, non-rigid deformation encompasses operations like rotation and cage deformation, which cannot be well solved by voxel displacement.



(a) Editing Pipeline (b) Editing Example: Object Removal and Stamp

Figure 2: (a) Our editing process is structured into composable operation groups, each comprising a selection step (Section 3.2) and a modification step (Section 3.3). (b) With each operation group, users initially identify the target object for editing and then apply editing operations. The spatial positions within the scene correspond directly to the feature grid, translating object-based editing actions into feature grid modifications.

2.2 3D Model Deformation and Editing

A series of editing techniques have been developed in recent years. Traditional mesh deformation methods mainly analyze the geometry of the shape and define the constraints accordingly. In addition to these mesh-based deformation methods, there are some methods [17, 29] that explore the use of voxel to perform deformation, which derives the form of voxel grids from RGB-D data and allows users to edit the voxels. Furthermore, there are some methods applying seam carving [2] to voxel [24] or voxel-based particle system [4, 5] to achieve content-aware resizing. Unlike volumetric seam carving methods [5, 24], they solve the optimization problem with an approximation algorithm, while we solve it precisely with a parallel algorithm. As far as we know, we are the first to explore the application of seam carving to radiance fields.

3 Editing Methods

3.1 Applicable Radiance Fields

Our editing system is applicable to most voxel-based hybrid representations that consist of an explicit voxel grid and a spatially independent decoder. Such voxel-based NeRF methods map a query position $\mathbf{x}$ and a viewing direction $\mathbf{d}$ to the corresponding volume density $\sigma \in \mathbb{R}$ and view-dependent color emission $\mathbf{c} \in \mathbb{R}^3$, which can be written as:

$$\sigma = F_\sigma(\mathrm{INTR}(\mathbf{x}, \mathbf{V}^{Dens})), \tag{1}$$

$$\mathbf{c} = F_{rgb}(\mathrm{INTR}(\mathbf{x}, \mathbf{V}^{CLR}), \mathbf{d}), \tag{2}$$

where F_σ and F_{rgb} denote two mapping functions depending on the specific NeRF method, INTR represents an interpolation function, $\mathbf{V}^{CLR} \in \mathbb{R}^{C \times N_x \times N_y \times N_z}$ and $\mathbf{V}^{Dens} \in \mathbb{R}^{N_x \times N_y \times N_z}$ denote the dense voxel grids with resolution $N_x \times N_y \times N_z$.

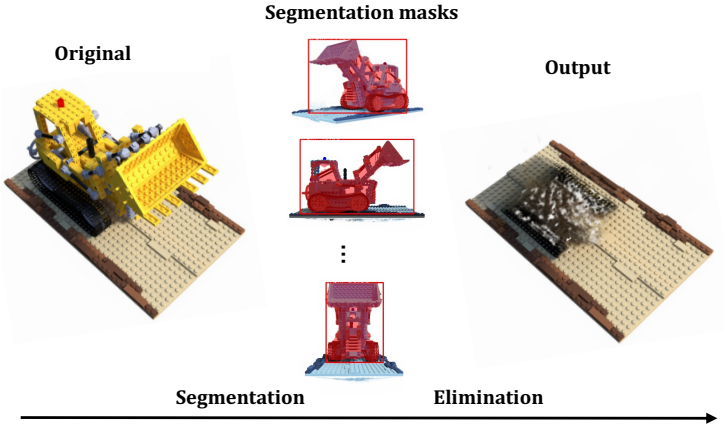


Figure 3: Illustration of segmentation-based selection. To select an object with a segmentation network, we first synthesize multiple views of the scene, select those high-quality segmentation views, and finally compute the convex hull for the selected object.

3.2 Selection Operations

Figure 2 illustrates the editing pipeline. First, we need to locate the space corresponding to the desired object for editing. Unlike image editing, where the user can intuitively locate where each pixel is, locating a precise point in stereoscopic space is cumbersome. In practice, however, the points used for editing are typically distributed over the surface of the object, as these key points hold crucial significance for the editing process.

In our approach, we offer two kinds of selection operations: Hexahedron Selection and Segmentation-based Selection. The choice of which selection method to use is entirely left to the user's discretion. Hexahedron selection is suitable for regular-shaped objects while segmentation-based selection is an alternative for more accurate extraction.

Hexahedron Selection. Most objects can be selected using a hexahedron. Specifically, we use the point selection method to select four co-planar spatial points $\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}$ as a base, and calculate the normal $\mathbf{n}$ of face $\mathbf{abc}$. Next, the hexahedron $\mathbf{S}$ formed by $\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}$ and $\mathbf{a}+t\mathbf{n},\mathbf{b}+t\mathbf{n},\mathbf{c}+t\mathbf{n},\mathbf{d}+t\mathbf{n}$ is used as the bounding box of the object, where the height t is determined by the user.

Segmentation-based Selection. In complex scenes where Hexahedron Selection struggles to extract the object accurately, we use a pre-trained 2D instance segmentation network [14] to precisely locate points on the object surface, as shown in Figure 3. First, we select several camera views and render corresponding images. These images are fed into the pre-trained image segmentation network to obtain the segmentation marks. Subsequently, we compute the collection of 3D coordinate points on the object surface corresponding to the masks, followed by calculating the convex hull to define the selection subspace.

3.3 Modification Operations

Once the object selection is completed, we proceed to object editing. We introduce stamp and seam carving along with some standard editing operations.

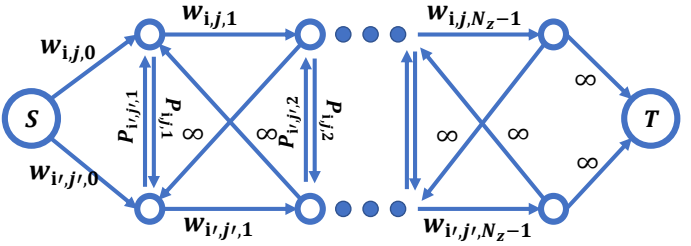


Figure 4: Illustration of maximum flow graph. S represents the source, and T represents the sink of the maximum flow problem. The value on the edge represents the capacity of this edge. The graph cut of this graph corresponds to one time of seam carving operation.

The edited scene can be represented as a combination of the pre-editing scene. We use the following form to represent an editing operation:

$$\mathbf{x}_{pre}, R_{rot} = F_{edit}(\mathbf{x}), \tag{3}$$

where $\mathbf{x} \in \mathbb{U}$ is the spatial location after editing, $\mathbf{x}_{pre} \in \mathbb{U}$ is the corresponding location before editing, $R_{rot} \in SO(3)$ is the rotation from $\mathbf{x}$ to $\mathbf{x}_{pre}$, and F_{edit} is the edit function for specific edit operation. We use F_{edit} to represent the location and rotation mapping from the edited scene to the pre-editing scene, which helps to determine the value at each location according to the previous value. Notably, the function F_{edit} varies with each editing action, depending on the specific operation performed.

After the edit operation has been applied, we reconstruct the voxel grids of the edited scene. Specifically, we solve the feature $\mathbf{v} \in \mathbf{V}^{CLR}$ for each grid point $\mathbf{x}$ by:

$$\arg\min_{\mathbf{v}} \ \mathbb{E}_{\mathbf{d}} \| F_{rgb}(\mathbf{v}, \mathbf{d}) - F_{rgb}(\mathrm{INTR}(\mathbf{x}_{pre}, \mathbf{V}^{CLR}), \mathbf{R}_{rot}\mathbf{d}) \|. \tag{4}$$

We solve $\mathbf{v}$ by stochastic gradient descent (SGD). In particular, when we delete object parts, it is done by mapping the corresponding position to an empty point in the original scene.

Standard Operation. We implement a wide range of frequently used operations:

- Copy-and-Paste. The copy operation involves selecting an object and then specifying the translation for it.
- Rotation. Rotation is achieved through a rotation mapping technique that defines both the axis of rotation and the degree of clockwise rotation, applicable to an entire object or a specific segment thereof.
- Stretching. We can simply select a part of the object and specify the stretch direction and the stretch ratio to stretch the part of the object.
- Removal. Removal is done by replacing specific parts of objects with air.

Stamp Operation. Most of the time, removing an object that is originally placed on a plane will leave an artifact. To repair such an artifact, we use a stamp operation similar to the one used for 2D images, i.e., we select a small patch from the vicinity and repeat the patch to fill the target defective area, as we perform in the scene Lego truck shown in Figure 1(a). This repair operation usually works well when the radiance field is well converged. We acknowledge that the stamp operation is primarily suitable for cases involving flat surfaces (e.g., a table). It may not yield optimal results for non-flat surfaces, such as a sphere.

Seam Carving. We apply the 3D seam carving algorithm on voxel grids to achieve content-aware resizing. Compared to the 3D seam carving algorithm applied on video and voxel, the main issue here is how to find the proper distance between two different features. In the following, we present the 3D seam carving algorithm based on the maximum flow algorithm [6] and possible ways of calculating the feature distance.

First, we introduce a seam carving weighting and its corresponding solution, *forward energy*. In forward energy-based seam carving, the plane with the lowest energy represents the one that introduces the least amount of defects into the retargeted result. For convenience, we use $D(i,j)$ to denote the distance between the grid point i and grid point j in the voxel grids of shape $N_x \times N_y \times N_z$. We need to find a plane $g : [N_x] \times [N_y] \to [N_z]$ satisfying:

$$\begin{aligned} \arg\min_{g} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left(W_{i,j,g(i,j)} + \sum_{(i',j') \in neighbor(i,j)} [g(i,j) = g(i',j') + 1] \times P_{i,j,g(i,j)} \right) \\ \left| g(i,j) - g(i',j') \right| \leq 1, \forall (i',j') \in neighbor(i,j) \end{aligned} \tag{5}$$

where the $W_{i,j,k} = D\big((i,j,k+1),(i,j,k-1)\big)$ and $P_{i,j,k} = D\big((i,j,k-1),(i,j,k)\big)$. Solving Equ. 5 is equivalent to solving a graph cut in Figure 4.

Implicit representations lack an explicit definition of distance, so we aim to find metrics to measure the visual differences in this paper. The two main issues we take into account here are: (1) there is no difference between two invisible points; (2) implicit feature distance depends strongly on the chosen representations. Considering that the human eye is more sensitive to changes in grayscale, we propose the following metric to calculate the distance:

$$\begin{aligned} F_{gray}(\mathbf{v},\mathbf{d}) &= \text{RGB-to-GRAY}(F_{rgb}(\mathbf{v},\mathbf{d})) \\ gray(\mathbf{v}_i,\mathbf{v}_j) &= \mathbb{E}_{\mathbf{d}} \left[\left\| F_{gray}(\mathbf{v}_i,\mathbf{d}) - F_{gray}(\mathbf{v}_j,\mathbf{d}) \right\| \right] \\ D_{gray}(i,j) &= gray(\mathbf{v}_i,\mathbf{v}_j) \times (\sigma_i + \sigma_j). \end{aligned} \tag{6}$$

Here RGB-to-GRAY denotes the function that converts RGB to grayscale, F_{rgb} is the function in Equ. 2, $\mathbf{d}$ is the view direction, $\mathbb{E}_{\mathbf{d}}$ presents the expectation over the view direction, σ is the density in Equ. 1 corresponding to the grid point. In experiments, we will show that D_{gray} is more effective than directly using seam carving.

4 Experiments

4.1 Qualitative Results and Comparisons

We apply our proposed method to a suite of scenes and show that all the edited scenes maintain high-quality novel view synthesis and spatial consistency.

Bounded Scenes Editing. In Figure 5, we present a comparison between our Standard operation 180°*Rotation* on Chair with NSVF. Upon the editing result of rotating the entire chair by 180 degrees, it's evident that NSVF, lacking consideration for view dependence during rotation, leads to the swapping of the chair's front and back colors. In contrast, our method effectively preserves the chair's original color scheme. This example highlights the robustness of our approach in handling rotational transformations without compromising visual fidelity. Standard operation *Stretching* is shown in Figure 7, where We stretch the trunk part of the Ficus to get different heights of the plant. Accessibility of non-rigid deformation is demonstrated with *Cage Deformation* in Figure 6. Specifically, we use cage

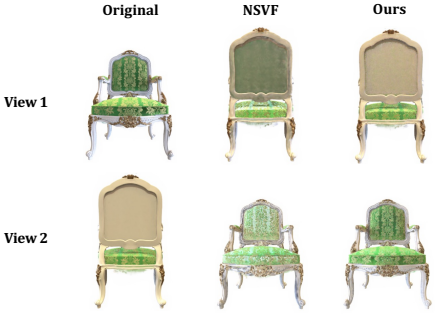


Figure 5: Comparison for 180°*Rotation* on Chair against NSVF.

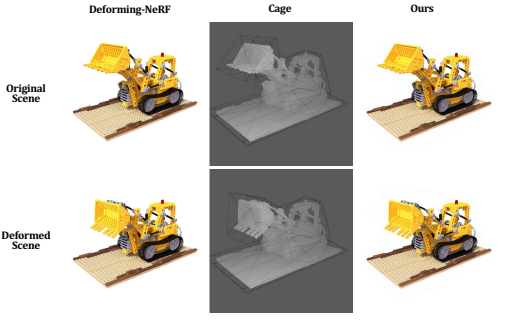


Figure 6: Visual comparison for *Cage Deformation* with Deforming-NeRF on Lego.

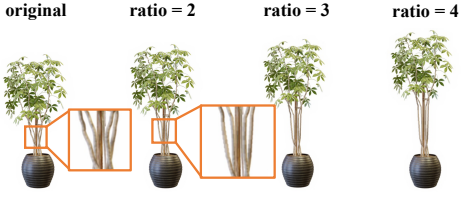


Figure 7: Results of *Streching* to heighten the trunk of Ficus from NeRF dataset.

Figure 8: The same edit operation on scenes trained with different representations.

generation followed by cage-based deformation to guide voxel modification, aligning with our stepwise approach of selection followed by modification. We present visual comparison results on the Lego scene from the NeRF synthetic dataset between our approach and Deforming-NeRF [31]. It can be seen that our results demonstrate comparable quality to Deforming-NeRF.

We next assess the performance of the *Seam Carving* in different bounded scenes. We compare (i) seam carving with our proposed metric D_{gray} with (ii) simple resizing, (iii) Instruct-NeRF2NeRF [7], and (iv) naive seam carving with distance metric $D_{direct}(i,j) = \|\mathbf{v}_i - \mathbf{v}_j\| + \|\sigma_i - \sigma_j\|$ on multiple synthetic scenes from NSVF dataset. For each scene, we change the aspect ratio along one of the axes. For Instruct-NeRF2NeRF, we use prompts to make the change. As shown in Figure 9, seam carving exhibits strong performance across all types of scenes, outperforming simple resizing by a large margin. Seam carving will not alter the important part and achieve content-aware resizing.

Unbounded Scenes Editing. As *Object Removal* demonstrated in Figure 10, we pick the triangular toy and remove the toy from the desktop of the ToyDesk. When comparing object removal results with Object-NeRF [32], We can observe that the removed toy leaves a rectangular artifact in Object-NeRF, while our results are more visually satisfactory. More results of seam carving in unbounded scenes can be seen in the supplementary material.

Forward-facing Scenes Editing. The *Seam Carving* results in forward-facing scenes are shown in Figure 11. We can see that both Fern and Flower scenes with seam carving preserve their contents better than with direct resizing.

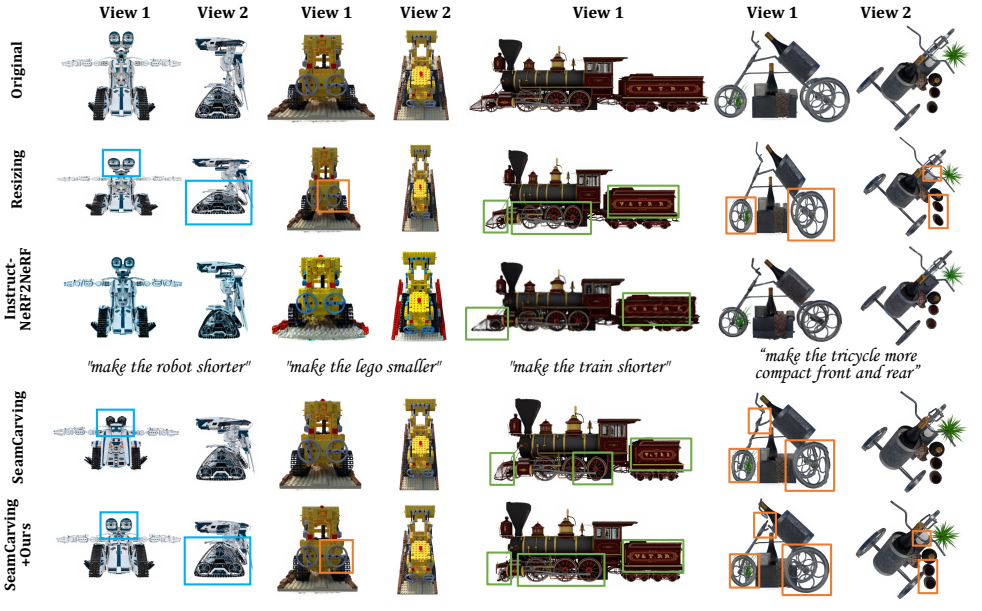


Figure 9: Seam Carving results on NSVF dataset: The first row depicts the original scenes trained by DVGO. The 2^{th} row shows simple resizing of local features via interpolation. The 3^{th} row is Instruct-NeRF2NeRF results. The 4^{th} row presents seam carving with naive distance metric. The last row demonstrates seam carving with our proposed metric D_{gray}.

4.2 Quantitative Evaluation

Evaluation Metrics. Acquiring ground truths for user-edited novel view synthesis is challenging due to the absence of physically existing scenes. Similar to NeRF-Editing [34], we employ SSIM [28], LPIPS [35], and PSNR as evaluation metrics with ground truth provided, and FID score [8] when ground truth is not provided.

Result Analysis. Our approach, which covers a wider range of editing methods, is compared quantitatively with existing techniques within their applicable domains, as detailed in Table 2. In summary, we compare rotation operation with NSVF [18], object removal with Object-NeRF [32], and seam carving with direct resizing via interpolation. Table 2 showcases the versatility of our framework across both generalized and specialized methods. Firstly, comparable performance with NSVF in rotation is attributed to the shared principles of simple edits across different representations. These results primarily underscore the generality of our framework for fundamental operations. We compare object removal with Object-NeRF, with FID scores calculated before and after the edit. We see that our method introduces fewer artifacts, leading to lower FID increases. For seam carving, we quantitatively evaluate the average FID for rendered images between the original scene and the modified scene. We compare our seam carving with direct resizing in multiple datasets. Given that seam carving is not defined for unbounded scenes, we carve only the foreground part, setting the background to white. The results demonstrate the superior performance of our proposed seam carving method which significantly leads to better FID scores.

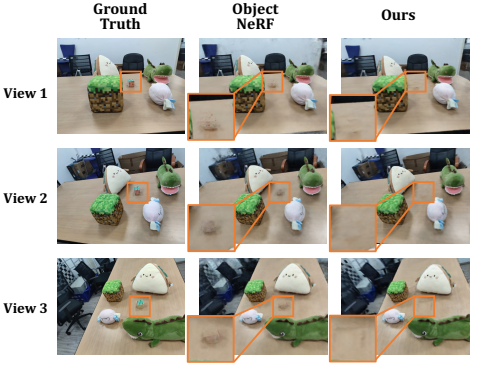


Figure 10: Editing results of *Object Removal* on the unbounded scene from Toydesk.

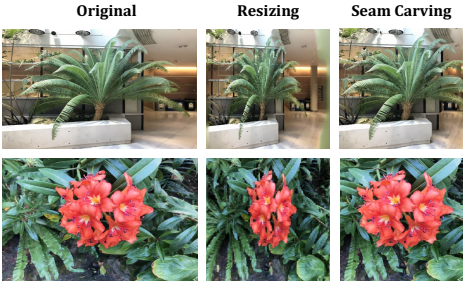


Figure 11: Improved version of *Seam Carving* on Fern and Flower from LLFF dataset to validate that our system still works on the forward-facing scenes.

Rotation	60°Rotation			180°Rotation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NSVF	20.677	**0.851**	0.084	18.626	0.832	0.125
Ours	**20.707**	0.850	0.084	**18.897**	0.832	**0.123**
Object Removal	FID↓ (before)		FID↓ (after)		increased ↓	
Object-NeRF	**1.519**		**1.594**		+4.9%	
Ours	2.814		2.901		**+3.1%**	
Resizing	FID↓ (NeRF Synthetic)		FID↓ (LLFF)		FID↓ (Mip-360 Foreground)	
Direct Resizing	3.066		7.097		1.404	
Ours	**1.288**		**1.330**		**1.176**	

Table 2: Quantitative comparison with other edit methods.

4.3 Ablation Study

We show the effectiveness of our editing method across various representations in Figure 8. We perform seam carving on representation based on DVGO or Plenoxels, yielding comparable outcomes. Due to the inconsistency in the choice of representation and the quality of scene reconstruction, the edited scenes naturally have slight differences, but this does not affect the universality of our proposed editing system. More ablation studies focusing on the size of Voxel Grids re detailed in the supplementary material.

5 Conclusions

In this paper, we proposed a novel editing system DRAFT for voxel-based radiance fields, which consists of composable operations and yields high-quality editing results. We implement several editing methods commonly utilized in radiance field editing. And we introduce seam curving to achieve content-aware resizing of radiance fields. Moreover, our system maintains the same format of representations before and after editing, enabling further refinement of edited scenes by other works. Experimental results demonstrate the effectiveness of our proposed system on real-world and synthetic datasets, as well as on both neural-based and non-neural-based representations.

References

[1] ShahRukh Athar, Zexiang Xu, Kalyan Sunkavalli, Eli Shechtman, and Zhixin Shu. Rignerf: Fully controllable neural 3d portraits. *CoRR*, abs/2206.06481, 2022. doi: 10.48550/arXiv.2206.06481. URL https://doi.org/10.48550/arXiv.2206.06481.

[2] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. *ACM Trans. Graph.*, 26(3):10, 2007. doi: 10.1145/1276377.1276390. URL https://doi.org/10.1145/1276377.1276390.

[3] Bao and Yang, Zeng Junyi, Bao Hujun, Zhang Yinda, Cui Zhaopeng, and Zhang Guofeng. Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In *European Conference on Computer Vision (ECCV)*, 2022.

[4] Sean Flynn, Parris K. Egbert, Seth Holladay, and Bryan S. Morse. Fluid carving: intelligent resizing for fluid simulation data. *ACM Trans. Graph.*, 38(6):238:1–238:14, 2019. doi: 10.1145/3355089.3356572. URL https://doi.org/10.1145/3355089.3356572.

[5] Sean Flynn, David Hart, Bryan S. Morse, Seth Holladay, and Parris K. Egbert. Generalized fluid carving with fast lattice-guided seam computation. *ACM Trans. Graph.*, 40(6):255:1–255:15, 2021. doi: 10.1145/3478513.3480544. URL https://doi.org/10.1145/3478513.3480544.

[6] Lester Randolph Ford and Delbert R Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8:399–404, 1956.

[7] Ayaan Haque, Matthew Tancik, Alexei A. Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. *CoRR*, abs/2303.12789, 2023. doi: 10.48550/arXiv.2303.12789. URL https://doi.org/10.48550/arXiv.2303.12789.

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6626–6637, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.

[9] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. *CoRR*, abs/2112.05637, 2021. URL https://arxiv.org/abs/2112.05637.

[10] Xin Huang, Qi Zhang, Feng Ying, Hongdong Li, Xuan Wang, and Qing Wang. Hdr-nerf: High dynamic range neural radiance fields. *CoRR*, abs/2111.14451, 2021. URL https://arxiv.org/abs/2111.14451.

[11] Yihua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. Stylizednerf: Consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. *CoRR*, abs/2205.12183, 2022. doi: 10.48550/arXiv.2205.12183. URL https://doi.org/10.48550/arXiv.2205.12183.

[12] Clément Jambon, Bernhard Kerbl, Georgios Kopanas, Stavros Diolatzis, Thomas Leimkühler, and George Drettakis. Nerfshop: Interactive editing of neural radiance fields. *Proc. ACM Comput. Graph. Interact. Tech.*, 6(1):1:1–1:21, 2023. doi: 10.1145/3585499. URL https://doi.org/10.1145/3585499.

[13] Kacper Kania, Kwang Moo Yi, Marek Kowalski, Tomasz Trzcinski, and Andrea Tagliasacchi. Conerf: Controllable neural radiance fields. *CoRR*, abs/2112.01983, 2021. URL https://arxiv.org/abs/2112.01983.

[14] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross B. Girshick. Pointrend: Image segmentation as rendering. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9796–9805. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00982. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Kirillov_PointRend_Image_Segmentation_As_Rendering_CVPR_2020_paper.html.

[15] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *CoRR*, abs/2205.15585, 2022. doi: 10.48550/arXiv.2205.15585. URL https://doi.org/10.48550/arXiv.2205.15585.

[16] Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, and Gerard Pons-Moll. Control-nerf: Editable feature volumes for scene rendering and manipulation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4340–4350, January 2023.

[17] Jerry Liu, Fisher Yu, and Thomas A. Funkhouser. Interactive 3d modeling with a generative adversarial network. In *2017 International Conference on 3D Vision, 3DV 2017, Qingdao, China, October 10-12, 2017*, pages 126–134. IEEE Computer Society, 2017. doi: 10.1109/3DV.2017.00024. URL https://doi.org/10.1109/3DV.2017.00024.

[18] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b4b758962f17808746e9bb832a6fa4b8-Abstract.html.

[19] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 5753–5763. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00572. URL https://doi.org/10.1109/ICCV48922.2021.00572.

[20] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 7210–7219. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00713. URL https://openaccess.thecvf.com/content/CVPR2021/html/Martin-Brualla_NeRF_in_the_Wild_Neural_Radiance_Fields_for_Unconstrained_Photo_CVPR_2021_paper.html.

[21] 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. *Commun. ACM*, 65(1):99–106, 2022. doi: 10.1145/3503250. URL https://doi.org/10.1145/3503250.

[22] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 9054–9063. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00894. URL https://openaccess.thecvf.com/content/CVPR2021/html/Peng_Neural_Body_Implicit_Neural_Representations_With_Structured_Latent_Codes_for_CVPR_2021_paper.html.

[23] Yicong Peng, Yichao Yan, Shenqi Liu, Yuhao Cheng, Shanyan Guan, Bowen Pan, Guangtao Zhai, and Xiaokang Yang. Cagenerf: Cage-based neural radiance fields for genrenlized 3d deformation and animation. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022.

[24] Michael Rubinstein, Ariel Shamir, and Shai Avidan. Improved seam carving for video retargeting. *ACM Trans. Graph.*, 27(3):16, 2008. doi: 10.1145/1360612.1360615. URL https://doi.org/10.1145/1360612.1360615.

[25] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *CoRR*, abs/2111.11215, 2021. URL https://arxiv.org/abs/2111.11215.

[26] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. *CoRR*, abs/2202.05263, 2022. URL https://arxiv.org/abs/2202.05263.

[27] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 3825–3834. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00381. URL https://doi.org/10.1109/CVPR52688.2022.00381.

[28] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*,

13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861. URL https://doi.org/10.1109/TIP.2003.819861.

[29] Ole Wegen, Jürgen Döllner, and Matthias Trapp. Interactive editing of voxel-based signed distance fields. *J. WSCG*, 30(1-2):72–81, 2022. doi: 10.24132/JWSCG.2022.9. URL https://doi.org/10.24132/JWSCG.2022.9.

[30] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. Humannerf: Free-viewpoint rendering of moving people from monocular video. *CoRR*, abs/2201.04127, 2022. URL https://arxiv.org/abs/2201.04127.

[31] Tianhan Xu and Tatsuya Harada. Deforming radiance fields with cages. *CoRR*, abs/2207.12298, 2022. doi: 10.48550/arXiv.2207.12298. URL https://doi.org/10.48550/arXiv.2207.12298.

[32] Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 13759–13768. IEEE, 2021. doi: 10.1109/ICCV48922.2021.01352. URL https://doi.org/10.1109/ICCV48922.2021.01352.

[33] Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *CoRR*, abs/2112.05131, 2021. URL https://arxiv.org/abs/2112.05131.

[34] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. Nerf-editing: Geometry editing of neural radiance fields. *CoRR*, abs/2205.04978, 2022. doi: 10.48550/arXiv.2205.04978. URL https://doi.org/10.48550/arXiv.2205.04978.

[35] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 586–595. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00068. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

[36] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul E. Debevec, William T. Freeman, and Jonathan T. Barron. Nerfactor: neural factorization of shape and reflectance under an unknown illumination. *ACM Trans. Graph.*, 40(6):237:1–237:18, 2021. doi: 10.1145/3478513.3480496. URL https://doi.org/10.1145/3478513.3480496.