# Volumetric Disentanglement for 3D Scene Manipulation

Sagie Benaim[1] Frederik Warburg[2,*] Peter Ebert Christensen[1,*] Serge Belongie[1]
[1]University of Copenhagen [2]Technical University of Denmark

## Abstract

*Recently, advances in differential volumetric rendering enabled significant breakthroughs in the photo-realistic and fine-detailed reconstruction of complex 3D scenes, which is key for many virtual reality applications. However, in the context of augmented reality, one may also wish to effect semantic manipulations or augmentations of objects within a scene. To this end, we propose a volumetric framework for (i) disentangling or separating, the volumetric representation of a given foreground object from the background, and (ii) semantically manipulating the foreground object, as well as the background. Our framework takes as input a set of 2D masks specifying the desired foreground object for training views, together with the associated 2D views and poses, and produces a foreground-background disentanglement that respects the surrounding illumination, reflections, and partial occlusions, which can be applied to both training and novel views. Our method enables the separate control of pixel color and depth as well as 3D similarity transformations of both the foreground and background objects. We subsequently demonstrate our framework's applicability on several downstream manipulation tasks, going beyond the placement and movement of foreground objects. These tasks include object camouflage, non-negative 3D object inpainting, 3D object translation, 3D object inpainting, and 3D text-based object manipulation. The project webpage is provided in* https://sagiebenaim.github.io/volumetric-disentanglement/.

## 1. Introduction

The ability to interact with a 3D environment is of fundamental importance for many augmented reality (AR) application domains such as interactive visualization, entertainment, games, and robotics [28]. Such interactions are often semantic in nature, capturing specified entities in a 3D scene and manipulating them accordingly. To this end, we propose a novel framework for the disentanglement and manipulation of objects in a 3D scene. Given a small set of 2D masks delineating the desired foreground object together with the associated 2D views and poses, and no other 3D information, our method produces a volumetric representation of both the foreground object and the background. Our volumetric representation enables separate control of pixel color and depth, as well as scale, rotation, and translation of the foreground object and the background. Using this disentangled representation, we demonstrate a suite of downstream manipulation tasks involving both the foreground and background volumes, going beyond previous work, and including 3D camouflage and 3D semantic text-based manipulation. Fig. 1 illustrates our proposed volumetric disentanglement and a sampling of the downstream volumetric manipulations that this disentanglement enables. We note that while the *foreground/background* terminology is useful for painting a mental picture, we wish to emphasize that the disentanglement is not limited to foreground objects, and works equally well for objects positioned further back (and partially occluded).

Neural Radiance Fields (NeRF) [30] delivered a significant breakthrough in reconstructing complex 3D scenes with high fidelity and detail. However, NeRF has no control over individual semantic objects within a scene. To this end, ObjectNeRF [56] proposed to represent foreground objects by rendering rays with masked regions. While ObjectNeRF learns foreground object representation independently from the background, our method instead disentangles the foreground from the background using a volumetric composition. In particular, the foreground object is extracted using a volumetric "subtraction" of the background from the full scene. In doing so, our method correctly captures objects occluded by the background, as well as objects with noisy and inaccurate masks which may include occluding objects, which ObjectNeRF does not handle well. Further, our method is able to reduce the level of required supervision. While we require a set of 2D mask annotation for training views, ObjectNeRF also requires additional 3D information in the form of 3D bounding boxes to render the background and edit objects at test time and relies on an accurate estimation of depth for training.

Given a set of 2D training views and poses of a scene, as well as masks, specifying the foreground object, our method

*Contributed equally.

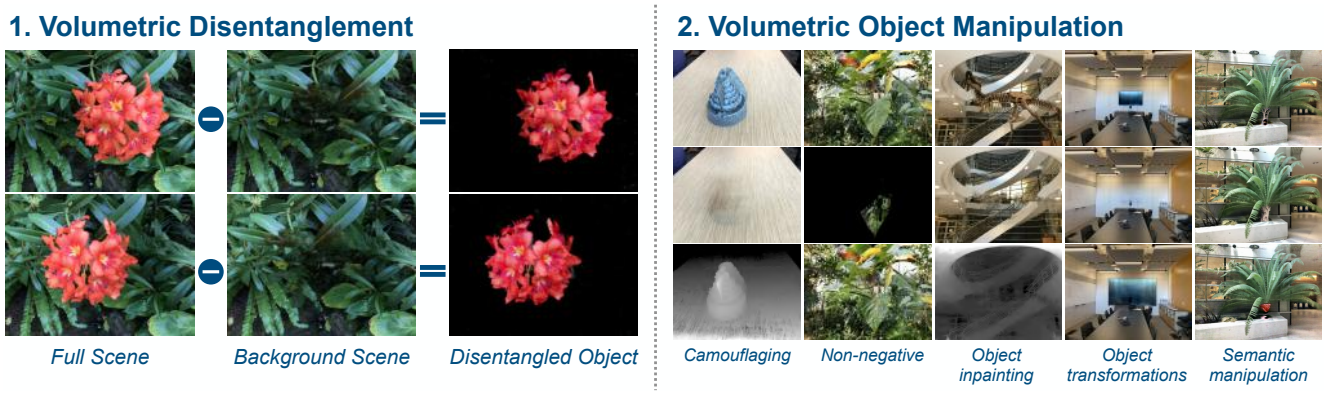



Figure 1. **Volumetric disentanglement framework.** Our framework disentangles the foreground objects and the background from a full scene which can then be rendered from novel views (1). Our volumetric disentanglement can then be used for many downstream tasks of interest to designers and artists in AR applications (2), including 3D object camouflage, non-negative 3D inpainting, 3D object inpainting, 3D object transformation, and 3D text-based semantic manipulation.

first trains a neural radiance field to reconstruct the background and its associated effects, following a similar procedure to NeRF [30]. Due to the prior induced through volumetric rendering, the resulting neural field captures the background volume that also includes objects appearing behind or occluding the foreground object. By training a neural radiance field to reconstruct the volume of the entire 3D scene and the volume of the background separately, the representation of the foreground can be computed in a compositional manner from the two volumes [9] as illustrated in Fig. 1, without using any other 3D information. We note that the background and foreground can be rendered from both training and novel views.

Having disentangled the foreground object from the rest of the 3D scene, we can now perform a range of downstream tasks, going beyond the placement and movement of objects, as shown in [56]. For example, optical-see-through devices can only add light to the scene, meaning that the generation must be non-negative with respect to the input scene [27]. In other cases, one may wish to keep the depth of the original scene intact [16, 41], and only modify the textures or colors. Our framework enables properties such as color, depth, and affine transformations of both the foreground object and background to be manipulated separately, and therefore can handle such manipulation tasks.

Lastly, we consider the ability to semantic manipulate the foreground. To this end, we consider the recently proposed multi-modal embedding of CLIP [43]. Using CLIP, we manipulate the foreground object semantically using text. Recent work such as [29, 46, 49] considered the ability to manipulate 3D scenes semantically using text. We demonstrate a similar capability, but one which transcends to individual objects in our 3D scene, while adhering to the semantics of the background. We also note that while 2D counterparts may exist for each of the proposed manipulations, our disentangled volumetric manipulation offers 3D-consistent semantic manipulation of foreground objects.

## 2. Related Work

**3D Disentanglement** We focus on the disentanglement of semantic and geometric properties in 3D scenes. For a more comprehensive overview, see [2]. CLIP-NeRF [49] disentangle the shape and appearance of NeRF [30] and, subsequently, uses CLIP [43] to manipulate these properties. Other works disentangle pose [51, 58], illumination [4, 48], texture and shape [8, 20, 26, 39]. These works are limited to an entire volumetric scene or object but not to objects within a scene. Further, they are limited to specific categories on constrained domains (e.g human parts).

Another line of work considers the disentanglement of objects in a full 3D scene. [36, 38] consider the generation of scenes in a compositional manner. In contrast, we disentangle an *existing* scene into the foreground and background volumes, while they generate such volumes from scratch. A subsequent line of works considers the disentanglement of objects in an existing scene. Several representations can be used to learn 3D scenes such as point clouds [1, 18, 47, 57], meshes [13, 17, 42, 50], or voxels [6, 44, 53, 55]. However, work using these representations for disentanglement [5, 23] are typically restricted in topology or resolution or make strong assumptions about scenes.

Recently, a number of methods proposed to use neural fields (NeRFs) to represent individual objects in the scene. [15] use an object library and learn a per object scattering field which can then be composed together to represent a scene where the object's movement, lighting, and reflection can be controlled. Our method instead decomposes an existing scene into foreground and background objects, capturing their relations, and subsequently allowing for object-specific edits. [40] use a scene graph representation to decompose dynamic objects, but rely on a dynamic scene as

input, and are restricted to one class of objects with similar shapes. [12, 22] consider specific types of semantic categories, for instance specialized domains s.a traffic scenes. Unlike these works, our work is not limited to the type of editable objects in the scene and enables a wider variety of manipulations including 3D object camouflage and 3D semantic manipulation of individual objects in a scene.

ObjectNeRF [56] is a recently proposed method that uses an object branch to render rays with masked regions for foreground objects. Our method differs from ObjectNeRF in multiple ways: (i). Our method requires input 2D segmentation masks for input training views. ObjectNeRF also requires 3D bounding boxes for editing foreground objects in addition to the 2D annotation. Similarly, our method does not require 3D structure in the form of a voxel grid during training. (ii). Unlike ObjectNeRF, our method correctly captures objects with noisy and inaccurate masks and objects occluded by the background. (iii). Our method relies on ground truth RGB images for existing views for our loss objectives, and does not require an occlusion loss which requires an accurate estimate of the scene's depth of existing and novel views. (iv). Lastly, our method goes beyond the editing of objects' movement and placement and enables zero-shot manipulations (does not require any 3D or 2D training data) such as 3D object camouflage, and 3D text-based semantic manipulation of individual objects.

Recently, [21] proposed a disentanglement framework for neural fields using text or image patches. While it enables the disentanglement of coarse concepts based on text or image patch, it does not allow for the fine-grained control which a mask can provide in selecting the object to be disentangled. Further, our approach offers a lighter indirect object representation in the form of masks whereas [21] uses CNN-based features per pixel instead.

Recent work has also considered the disentanglement of 3D objects from a neural radiance field using 2D masks [25, 32, 33, 52, 54]. Our method enables a richer set of applications including 3D object camouflage, 3D non-negative inpainting, and 3D semantic manipulation.

**3D Manipulation** Our framework enables the manipulation of localized regions in a scene. While 2D counterparts, such as 2D inpainting approaches exist [10, 11, 14, 59], they cannot generate 3D consistent manipulations. One set of approaches consider editing the entire scene. [7] considers texture and shape manipulation of 3D meshes. CLIP-Forge [45] generates objects matching a text prompt using CLIP embeddings. Text2Mesh [29] manipulate the texture or style of an object. DreamFields [19] learn a neural radiance field representing 3D objects. Unlike these works, our work is concerned with manipulating a local region in an existing scene. [20] and [26] modify the shape and color code of objects using coarse 2D user scribbles, but require a curated dataset of objects under different colors and views, and are limited to synthetic objects.

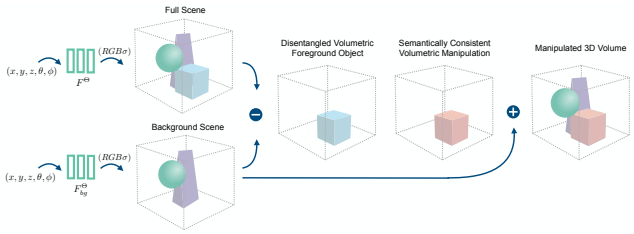



Figure 2. **Overview of our disentanglement framework.** First, we learn a volumetric representation of the background and full scene (Sec. 3.1). Second, by subtracting the full and the background volumes, we obtain a disentangled foreground volume. Third, we perform a wide range of manipulations on this volume, which adhere to the background volume. This is illustrated here by changing the color of the cube from blue to red. Finally, we can place the foreground object back into the original scene by adding it volumetrically to the background scene.

## 3. Method

Given a 3D scene, we wish to disentangle semantic objects from the rest of the scene. First, we describe the 3D volumetric representation used to disentangle objects and control objects separately (Sec. 3.1). The disentanglement of foreground and background volumes opens a wide range of downstream applications. We provide a framework that explores some of these applications by manipulating objects in a semantic manner (Sec. 3.2). An illustration of our framework is provided in Fig. 2. Additional training and implementation details are provided in the supplementary.

### 3.1. Disentangled Object Representation

The ability to disentangle the foreground object volumetrically from the background requires a volumetric representation that correctly handles multiple challenges: (i). Foreground occluding objects, which may be covered by a foreground mask, should not be included in the foreground volume, (ii). Regions occluded by the foreground object should be visible in the background volume, (iii). Illumination and reflectance effects, affecting the foreground object in the full scene volume, should affect the now unoccluded regions of the background in a natural way. To this end, we build upon the representation of neural radiance fields [30].

**Neural Radiance Fields.** A neural radiance field [30] is a continuous function $f$ whose input is a 3D position $p = (x, y, z) \in \mathbb{R}^3$ along with a viewing direction $d = (\theta, \phi) \in \mathbb{S}^2$, indicating a position along a camera ray. The output of $f$ is an RGB color $c \in \mathbb{R}^3$ and volume density $\alpha \in \mathbb{R}^+$. We first apply a frequency-based encoding $\gamma$ to correctly capture high-frequency details using $\gamma(p) = [\cos(2\pi \mathbf{B}p), \sin(2\pi \mathbf{B}p)]^{\mathrm{T}}$, where $\mathbf{B} \in \mathbb{R}^{n \times 3}$ is a randomly drawn Gaussian matrix whose entries are drawn

from $\mathcal{N}\left(0, \sigma^2\right)$, where $\sigma$ is a hyperparameter. $f$ is then parameterized as an MLP $f_\theta$ whose input is $(\gamma(p), \gamma(d))$ and output is $c$ and $\sigma$.

**Object Representation.** Given camera extrinsics $\xi$, we assume a set $\{(c_r^i, \sigma_r^i)\}_{i=1}^N$ of color and volume density values predicted by $f_\theta$ for $N$ randomly chosen points along camera ray $r$. A rendering operator then maps these values to an RGB color $c_r$ as $\sum_{i=1}^N w_r^i \cdot c_r^i$ where:

$$w_r^i = \prod_{j=1}^{i} (1 - \sigma_j^r) \cdot T_r^i \qquad T_r^i = \quad 1 - \exp(\sigma_r^i \cdot \delta_i^r) \tag{1}$$

where $\sigma_r^i$ and $T_r^i$ are the alpha and transmittance values for point $i$ along ray $r$ and $\delta_i^r = t^{i+1} - t^i$ is the distance between adjacent samples. For training, we assume a set of posed views $\{x^i\}_{i=1}^M$ together with their associated foreground object masks $\{m_i\}_{i=1}^M$. We set $\{\hat{x^i}\}_{i=1}^M$ to be the corresponding colors to $\{x^i\}_{i=1}^M$ as predicted by Eq. (1). To train the background (resp. full) volume, we minimize the masked (resp. unmasked) reconstruction loss between real and estimated views:

$$\mathcal{L}_{bg} = \sum_{i=1}^{M} ||(1 - m_i) \odot (x^i - \hat{x}^i)||_2^2 \tag{2}$$

$$\mathcal{L}_{full} = \sum_{i=1}^{M} ||x^i - \hat{x}^i||_2^2 \tag{3}$$

We note that $L_{bg}$ is sufficient to obtain a background representation. However, inspired by SPIn-NeRF [33], one can obtain a higher-quality background representation. In particular, we first inpaint each of the training views and their associated depths (obtained by training NeRF) in the region provided by the input masks, using an off-the-shelf 2D inpainting method, which may generate inconsistencies. We then train our background NeRF representation using (1). $L_{bg}$, which is modified to use a perceptual loss (LPIPS) instead of MSE loss. (2). A standard reconstruction loss (following volumetric rendering) using both the inpainted RGB and depth views.

Let $w_{bg}^{i_r}$ and $c_{bg}^{i_r}$ be the value of $w_r^i$ and $c_r^i$ in Eq. (1) predicted for the background volume and similarly let $w_{full}^{i_r}$ and $c_{full}^{i_r}$ be the value of $w_r^i$ and $c_r^i$ in Eq. (1) predicted for the full volume. A natural representation of the foreground can then be found using the volume mixing principle [9]:

$$c_{fg}^r = \sum_{i=1}^{N} w_{fg}^{i_r} \cdot c_{fg}^{i_r}, \text{where} \tag{4}$$
$$w_{fg}^{i_r} = w_{full}^{i_r} - w_{bg}^{i_r} \qquad c_{fg}^{i_r} = c_{full}^{i_r} - c_{bg}^{i_r}$$

$c_{fg}^r$ is the foreground volume color at the pixel corresponding to ray $r$. This can be used to render the color of the foreground object for all pixels across different views.

**Object Controls.** We note that camera parameters, as well as chosen poses, rays, and sampled points along the rays, are chosen to be identical for both the full volume and the background volume, and hence also identical to the foreground volume. Given this canonical setting, the corresponding points along the rays for both the foreground and background can be easily found.

Due to the above-mentioned correspondence, one can independently modify ${w_{fg}}^{i_r}$ and ${c_{fg}}^{i_r}$ to get ${w'_{fg}}^{i_r}$ and ${c'_{fg}}^{i_r}$ for the foreground volume as well as ${w_{bg}}^{i_r}$ and ${c_{bg}}^{i_r}$ to get ${w'_{bg}}^{i_r}$ and ${c'_{bg}}^{i_r}$ for the background volume. In order to recombine the modified background with the modified foreground, we note that every 3D point along the ray should only be colored, either according to the background volume or according to the foreground volumes, but not by both, as they are disentangled. We can then recombine the modified foreground and background:

$$c^r = \sum_{i=1}^{N} {w'_{bg}}^{i_r} \cdot {c'_{bg}}^{i_r} + {w'_{fg}}^{i_r} \cdot {c'_{fg}}^{i_r} \tag{5}$$

$c^r$ is the recombined color of the pixel corresponding to ray $r$. In our experiments, we only modify the foreground and so ${w'_{bg}}^{i_r} = {w_{bg}}^{i_r}$, ${c'_{bg}}^{i_r} = {c_{bg}}^{i_r}$.

### 3.2. Object Manipulation

Given the ability to control the foreground and background volumes separately, we now propose a set of downstream manipulation tasks that emerge from our disentangled representation. As noted in Sec. 3.1, we can now control the weights, colors as well as translation parameters separately for the foreground and background volumes and so introduce a set of manipulation tasks that use the controls. We note that the task of *Object Removal* is equivalent to displaying the background.

**Object Transformation.** Due to the alignment of camera parameters and chosen poses, rays, and sampled points along the rays, one can apply a transformation on the background and foreground volumes separately, before recombining the volumes together. For either the foreground or the background, and for a transformation $T$, we evaluate the color and weight of point $p$ using $f_\theta$ at position $T^{-1}(p)$ and then recombine the volumes together using Eq. (5).

**Object Camouflage.** Here we wish to change the texture of the foreground 3D object such that it is difficult to detect from its background [16, 41]. Such an ability can be useful in the context of diminished reality [34]. To do so, we fix the depth of the foreground object while manipulating its texture. As the depth of the foreground is derived from ${w_{fg}}^{i_r}$, we fix ${w'_{fg}}^{i_r} = {w_{fg}}^{i_r}$ and only optimize ${c'_{fg}}^{i_r}$. We follow Eq. (5), in compositing the foreground and background volumes. Let the resulting output for each view $i$ be $\hat{x}_c^i$, and let $\hat{x}_{bg}^i$ be the corresponding

output for the background volume. We optimize a neural radiance field for foreground colors ${c'_{fg}}^{i_r}$ to minimize $\mathcal{L}_{camouflage} = \sum_{i=1}^{M} ||\hat{x}_c^i - \hat{x}_{bg}^i||_2^2$. As depth is fixed, only the foreground object colors are changed to match the background volume as closely.

**Non-negative 3D Inpainting.** Next, we consider the setting of non-negative image generation [27]. We are interested in performing non-negative changes to views of the full scene so as to most closely resemble the background. This constraint is imposed in optical-see-through devices that can only add light onto an image. In this case, we learn a residual volume to render views $\hat{x}_{residual}^i$ as in Eq. (1) to minimize $\mathcal{L}_{non-negative} = \sum_{i=1}^{M} ||\hat{x}_{full}^i + \hat{x}_{residual}^i - \hat{x}_{bg}^i||_2^2$. where $\hat{x}_{full}^i$ are rendered views of the full scene as in Eq. (3). That is, we learn a residual volume whose views are $\hat{x}_{residual}^i$, such that when added to the full volume views, most closely resemble the background.

**Semantic Manipulation.** Next, we consider a mechanism for the semantic manipulation of the foreground. We consider the recently proposed model of CLIP [43], which can be used to embed an image $I$ and text prompt $t$ (or image $I_2$), and to subsequently compare the cosine similarity of the embeddings. Let this operation be $sim(I, t)$ (resp. $sim(I, I_2)$), where a value of 1 indicates perceptually similar text (resp. image) and image. Let $\hat{x}_c^i$ be the result of applying Eq. (5), while fixing the background colors and weights as well as the foreground weights. That is, we only optimize the foreground colors ${c'_{fg}}^{i_r}$. For a user-specified target text $t$, we consider the objective:

$$\sum_{i=1}^{M} 1 - sim\left(\hat{x}_c^i \odot m_i + \hat{x}_{bg}^i \odot (1 - m_i), t\right) \tag{6}$$

$$+ 1 - sim\left(\hat{x}_c^i \odot m_i + \hat{x}_{bg}^i \odot (1 - m_i), \hat{x}_{bg}^i \odot (1 - m_i)\right) \tag{7}$$

$$+ ||\hat{x}_c^i \odot (1 - m_i) - \hat{x}_{bg}^i \odot (1 - m_i)||_2^2 \tag{8}$$

While only the colors of the foreground volume can be manipulated, we enforce that such changes only occur within the localized masked region of the foreground, and so take the background from the fixed background volume. To do so, instead of applying clip similarity directly with $\hat{x}_c^i$, we apply it with $\hat{x}_c^i \odot m_i + \hat{x}_{bg}^i \odot (1 - m_i)$. Therefore, CLIP's similarity can only be improved by making local changes that occur within the masked region of the foreground object, but can 'see' the background and foreground for context. We enforce the generated volume views are similar to both the target text (Eq. (6)) and the background (Eq. (7)). To further enforce that no changes are made to the background, we constrain the background of the combined volume views to match those of the background using Eq. (8).

# 4. Experiments

We divide the experiments into two parts. First, we consider the ability to disentangle the foreground and background volumes from the rest of the scene. Second, we demonstrate some of the many manipulation tasks this disentanglement enables, as described in Sec. 3.2. Corresponding and additional 3D scenes from novel views are provided on the project webpage. All comparisons to baselines are made with the same set of input views and masks.

## 4.1. Object Disentanglement

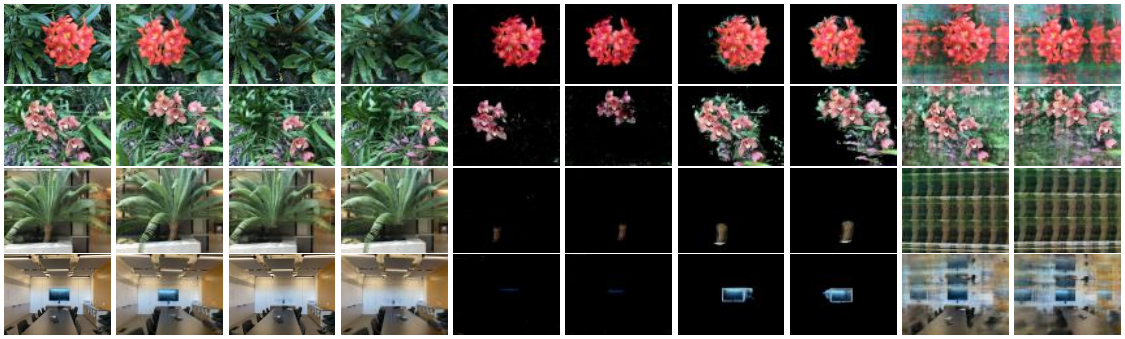
Fig. 3 shows novel views from different scenes of the LLFF dataset [31], where we separate the full scene, background, and foreground in a volumetrically and semantically consistent manner. As these are novel views, no foreground object mask is used. In the supplementary, we provide examples of training views and associated masks for the provided scenes. We compare our method to ObjectNeRF [56]. Note that ObjectNeRF requires 3D bounding boxes to extract the background volume which we do not use. Hence, we consider only the extracted foreground by ObjectNeRF. As can be seen, ObjectNeRF's extracted foreground object captures much of the background. This is visible for the tree trunk example. We note that extracted object representation captures the pixels commonly masked by all training masks, and hence for the orchids (second row), only some of the petals are shown. As a further comparison we consider a neural field trained to reconstruct only the masked region. Due to noisy masks, shown in the supplementary, this results in a noisy result which captures much of the background.

Fig. 4 depicts the consistency of the removal of a leaf, a T-rex, and a whiteboard for two different novel views. The background neural radiance field makes plausible predictions of the background scene via multi-view geometry and the inductive bias introduced by the positional encoding and the neural radiance field. *E.g.* the background behind the leaf or the legs of the T-rex might be occluded by the 2D mask from one view, but visible from another. However, the background behind the whiteboard is occluded from every angle. Nevertheless, the background neural radiance field makes a plausible prediction of the background. Further, our model can handle the disentanglement of non-planar objects, such as the T-rex, well.

In the 2D domain, as far as we can ascertain, the closest 2D task to object disentanglement is that of object inpainting. We consider two prominent baselines of DeepFill-v2 [59] and EdgeConnect [35] for this task and compare our method on the scenes of leaves and whiteboard removal as in Fig. 4. We train the baseline on the same training images and their associated masks. In order to compare our method on the same novel views, we train a NeRF [30] on the resulting outputs, resulting in a scene with the same novel views

| | Object Removal | | | Object Extraction | | Semantic Object Manipulation | | |
|---|---|---|---|---|---|---|---|---|
| | Ours | DeepFill-v2 [59] | EdgeConnect [35] | Ours | ObjectNeRF [56] | Ours | GLIDE [37] | Blended [3] |
| Q1 | **3.86** | 2.44 | 2.37 | **3.87** | 2.85 | **3.85** | 1.10 | 1.26 |
| Q2 | **3.84** | 1.52 | 1.86 | **3.91** | 2.62 | **3.78** | 1.20 | 1.26 |

Table 1. **User study**. We consider Object Removal, Extraction, and 3D Semantic Manipulation and use a MOS score (1-5).



Full Background Foreground [56] Foreground Training

Figure 3. **Two rendered *novel* views of the full scene, background, and foreground.** Note that as the TV screen is black, the foreground object appears black as well. We also show the result of directly training a neural field on the masked foreground object itself (foreground training). As some masks are noisy and may include much of the background, this results in a very noisy result which also include much of the background. While no 2D mask annotation is given for novel views, corresponding masks for training views are provided in the supplementary.

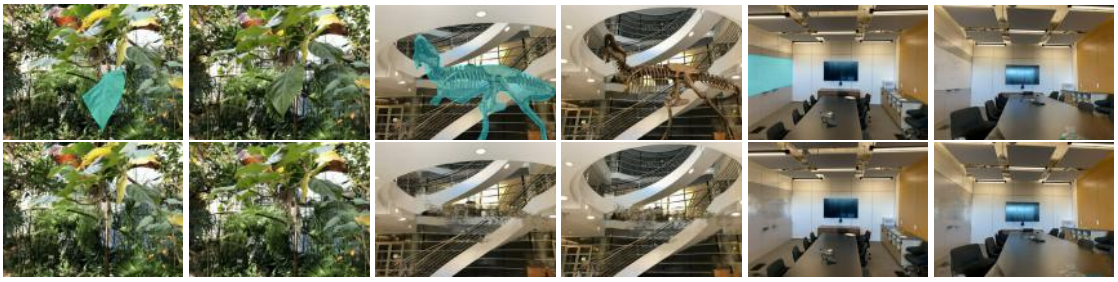
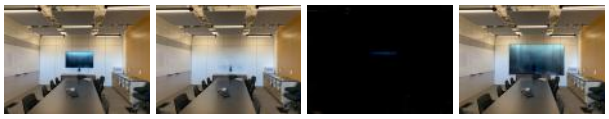


Figure 4. **Two uniformly sampled novel views of the full and the background volumes.** The removed object is visually enhanced by a 2D mask for illustration purposes (2D mask annotation is not given for novel views).



Full Background Foreground New Scene

Figure 5. **Foreground object transformation**. Our method makes plausible predictions in occluded regions (behind the TV). Scaling the foreground and placing it back into the scene results in photo-realistic and view consistent scene.

Figure 6. **Foreground rotation and translation.** 30° rotations (top) or translations of 0.4 (bottom) along the x (left), y (middle), and z (right) axes.

as ours. Unlike our method, the results have 3D inconsistencies, artifacts, and flickering between views. The visual comparison is provided on the project webpage.

To assess our method numerically, we first conduct a user study and ask users to rate from a scale of $1-5$: (Q1) "How well was the object removed/extracted?" and (Q2) "How realistic is the resulting object/scene?" We consider 25 users and mean opinion scores are shown in Tab. 1. For object extraction we consider ObjectNeRF [56] and consider the scenes in Fig. 3, For object removal, we consider 2D baselines of DeepFill-v2 [59], EdgeConnect [35], as de-

(a) (b) (c) (d) (e)

Figure 7. **Object camouflage for two different novel views of a fortress scene.** (a) original scene, (b) background scene, (c) disparity map of the background scene, (d) camouflaged scene, (e) disparity map of the camouflaged scene.

(a) (b) (c) (d)

Figure 8. **Non-negative object inpainting for two novel views for a scene of leaves.** Given the full scene (a), a residual scene is added (b) resulting in scene (c), with the aim of being close to the background without the leaf (d).

tailed above, and consider the leaves and whiteboard scenes as in Fig. 4. As no 3D bounding box is provided, we did not consider ObjectNeRF [56] for object removal. Second, we consider the dataset introduced by [33] which uses human-annotated object masks. Our method is comparable to SPIn-NeRF and superior to other 3D-based inpainting methods of NeRF-In [25] and Object-NeRF: We get an LPIPS/FID of **0.4623**/158.24 vs SPIn-NeRF's 0.4654/**156.64**, NeRF-In's 0.4884/183.23 and Object-NeRF's 0.6829/271.80.

### 4.2. Object Manipulation

**Foreground Transformation.** We consider the ability to scale the foreground object and place the rescaled object back into the scene by changing the focal length used to generate the rays of the foreground object, and then volumetrically adding it back into our background volume. Fig. 5 shows an example novel view where the disentangled TV is twice as large. Fig. 5 highlights that the network is able to "hallucinate" how a plausible background looks in regions occluded across all views (*e.g.* behind the TV). In Fig. 6, we consider various rotations and translations for the flower object. While our method can handle some transformations, very large transformations are not handled well.

**Object Camouflage.** Another manipulation is of camouflaging an object [16, 41], *i.e.* only changing the texture of the object and not its shape. Fig. 7 illustrates an example novel view of camouflaging with fixed depth, but free texture changes. While the depth of the camouflaged object and that of the foreground object match, the appearance is that of the background.

**Non-Negative Inpainting.** In optical see-through AR, one might also wish to camouflage objects [27] or inpaint them. However, in see-through AR one can only add light. Fig. 8 shows how adding light can make the appearance of camouflage in a 3D consistent manner for novel views.

**3D Object Manipulation.** Fig. 9 shows an example of a manipulated fern scene. We disentangled both the window mullion in the upper left corner and the tree trunk from the rest of the scene. Although the window mullion is occluded in the first view, and thus our 2D mask is masking the occluding leaf in front of the window mullion, this occluding object is not part of the disentangled window mullion object, demonstrating the advantage of our volumetric "subtraction" approach. This is because the extracted object representation captures the pixels commonly masked by all training masks. The 3D manipulations are shown in (c)-(e) in Fig. 9 for a novel view. See the project webpage for additional views. For the strawberry manipulation in (e), note how part of the tree trunk was camouflaged to more closely resemble the shape of a strawberry. We compare to 2D text-based inpainting methods of GLIDE [37] and Blended Diffusion [3], where we follow the same procedure as in Sec. 4.1. We consider a similar user study as detailed in Sec. 4.1, where Q1 is modified to: "How well was the object semantically manipulated according to the target text prompt?" and consider the fern scene of Fig. 9, for the text prompts of "strawberry" and "old tree". We compare to CLIP-NeRF and [21] on the scenes depicted in Fig. 9 (and project webpage). We measure CLIP similarity of test views to the input text: Our method achieved a clip-similarity of **0.126** vs CLIP-NeRF's 0.34 and [21]'s 0.31.

### 4.3. Mask Supervision and Limitations

**Noisy masks.** Our work can handle noisy masks, which may also include occluding objects and may cover regions outside of the foreground object, as shown in the supplementary. For instance, for the leaves scene, masks were extracted using an off-the-shelf segmentation algorithm and contain significant noise from the background. The number of noisy masks used in training (see supplementary for examples) is: leaf: 11 of 26 (42%), orchids: 20 of 25 (42%), TV: 10 of 41 (24%), T-rex: 35 of 55 (64%).

**Automatic Annotation.** While masks allow for fine-grained control, training masks can be generated automatically using either (1). *off-the-shelf background-foreground separation tools*, as is done for the leaf, orchids, TV, and T-rex scenes (Fig. 3, row 2 and 4, Fig. 4 LHS, Figs. 5-8 and project webpage), or (2). *text-based segmentation*

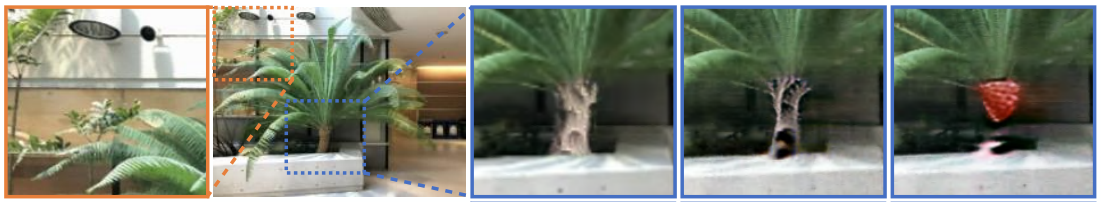



Figure 9. **3D semantic object manipulation.** Insets of the disentangled (a) window mullion and manipulated (c)-(e) tree trunk in the original scene (b). The window mullion is removed without removing the leaf that occludes it. Masks for training views also mask the occluding tree leaf. The text to manipulate the trunks is (c) *Old tree*, (d) *Aspen tree*, and (e) *Strawberry*.

*tools*, such as CLIP-LSeg [24]. To illustrate this, for the TV and leaf scenes, we run CLIP-LSeg using the text "tv" and "red flower" respectively, producing training masks on which our method is applied. Fig. 10 depicts an example novel view of the background, in comparison to standard mask annotation, and an example training mask generated.

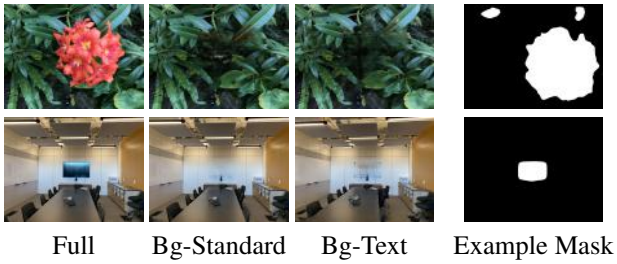



Figure 10. **Text-based object removal.** Removal of the flower and TV screen using masks generated automatically using CLIP-Lseg (Bg-Text) in comparison to standard masks (Bg-Standard). On the RHS, an example training mask is shown.

**Limitations.** When light from the background affects the foreground object, we correctly disentangle the illuminations on the object. However, when the object is a light source, we cannot completely disentangle the object as seen in the supplementary. Another limitation is with respect to semantic manipulation. We found that manipulating too large objects results in an under-constrained optimization because the signal provided by CLIP is not sufficient.

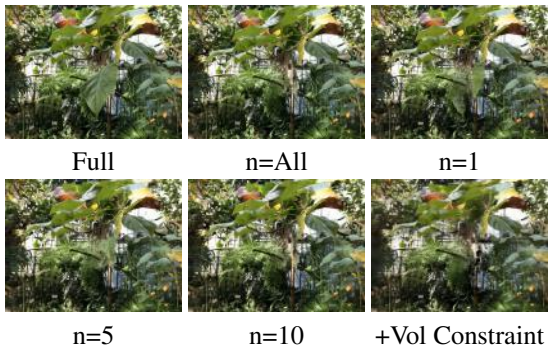



Figure 11. **Ablation.** Varying the number of masks (n) used (1, 5, 10, or all) for the removal of the leaf. Bottom RHS: Adding the volumetric constraint as described in the Sec. 4.3.

## 5. Ablation Study

In the supplementary, we consider, for the task of foreground object translation (Fig. 5), alternatives to the recombining method of Eq. (5). As a further ablation, we consider whether direct volume constraints improve our background scene reconstruction. More specifically, we reconstructed the full leaf scene and marked visible but empty regions of space, according to the predicted density of sampled points. For the background, we then added an additional constraint penalizing adding density to these regions. We find that such a constraint does not improve the result (Fig. 11). We hypothesize that our 2D background reconstruction loss already penalizes these regions.

**Number of masks.** Manual annotation can be used when exact and fine-grained control is desired. Here, the number of views to be manually annotated is minimal. To illustrate this, for the leaf scene, we varied the number of masks ($n$) used (1, 5, 10, or all). We randomly selected $n$ training masks on which we trained a NeRF to provide masks for other training views, which were then used in our method. As seen in Fig. 11, only little or no noise is introduced.

## 6. Conclusion

We presented a framework for the volumetric disentanglement of foreground objects from a background scene. The disentangled foreground object is obtained by volumetrically subtracting a learned volume representation of the background with one from the entire scene. We established that our disentanglement facilitates separate control of color, depth, and transformations for both the foreground and background objects. This enables a wide range of applications going beyond object movement and placement, of which we have demonstrated those of object camouflage, non-negative generation, and object manipulation.

## References


[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J Guibas. Learning representations and generative models for 3d point clouds. *arXiv preprint arXiv:1707.02392*, 2017. 2

[2] Eman Ahmed, Alexandre Saint, Abd El Rahman Shabayek, Kseniya Cherenkova, Rig Das, Gleb Gusev, Djamila Aouada, and Björn E. Ottersten. A survey on deep learning advances on different 3d data representations. *arXiv: Computer Vision and Pattern Recognition*, 2018. 2

[3] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. *arXiv preprint arXiv:2111.14818*, 2021. 6, 7

[4] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[5] Adrian Broadhurst, Tom W Drummond, and Roberto Cipolla. A probabilistic framework for space carving. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, volume 1, pages 388–393. IEEE, 2001. 2

[6] Andrew Brock, Theodore Lim, James Millar Ritchie, and Nicholas J. Weston. Generative and discriminative voxel modeling with convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2016. 2

[7] Zehranaz Canfes, M Furkan Atasoy, Alara Dirik, and Pinar Yanardag. Text and image guided 3d avatar generation and manipulation. *arXiv preprint arXiv:2202.06079*, 2022. 3

[8] Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa neural articulated shape approximation. In *European Conference on Computer Vision*, pages 612–628. Springer, 2020. 2

[9] Robert A Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *ACM Siggraph Computer Graphics*, 22(4):65–74, 1988. 2, 4

[10] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001. 3

[11] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1033–1038. IEEE, 1999. 3

[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. *arXiv preprint arXiv:2203.15224*, 2022. 3

[13] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[14] Christine Guillemot and Olivier Le Meur. Image inpainting: Overview and recent advances. *IEEE signal processing magazine*, 31(1):127–144, 2013. 3

[15] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503*, 2020. 2

[16] Rui Guo, Jasmine Collins, Oscar de Lima, and Andrew Owens. Ganmouflage: 3d object nondetection with texture fields. *arXiv preprint arXiv:2201.07202*, 2022. 2, 4, 7

[17] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: A network with an edge. *ACM Trans. Graph.*, 38(4), jul 2019. 2

[18] Le Hui, Rui Xu, Jin Xie, Jianjun Qian, and Jian Yang. Progressive point cloud deconvolution generation network. In *ECCV*, 2020. 2

[19] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. *arXiv preprint arXiv:2112.01455*, 2021. 3

[20] Won Jun Jang and Lourdes de Agapito. Codenerf: Disentangled neural radiance fields for object categories. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12929–12938, 2021. 2, 3

[21] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *arXiv preprint arXiv:2205.15585*, 2022. 3, 7

[22] 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

[23] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38(3):199–218, 2000. 2

[24] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022. 8

[25] Hao-Kang Liu, I Shen, Bing-Yu Chen, et al. Nerf-in: Free-form nerf inpainting with rgb-d priors. *arXiv preprint arXiv:2206.04901*, 2022. 3, 7

[26] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2, 3

[27] Katie Luo, Guandao Yang, Wenqi Xian, Harald Haraldsson, Bharath Hariharan, and Serge Belongie. Stay positive: Non-negative image synthesis for augmented reality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10050–10060, 2021. 2, 5, 7

[28] Mehdi Mekni and Andre Lemieux. Augmented reality: Applications, challenges and future trends. *Applied computational science*, 20:205–214, 2014. 1

[29] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. *arXiv preprint arXiv:2112.03221*, 2021. 2, 3

[30] B Mildenhall, P Srinivasan, M Tancik, JT Barron, and RRR Ng. Representing scenes as neural radiance fields for view synthesis. In *Proc. of European Conference on Computer Vision, Virtual*, 2020. 1, 2, 3, 5

[31] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019. 5

[32] 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. 3

[33] 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 *CVPR*, 2023. 3, 4, 7

[34] Shohei Mori, Sei Ikeda, and Hideo Saito. A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects. *IPSJ Transactions on Computer Vision and Applications*, 9(1):1–14, 2017. 4

[35] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019. 5, 6

[36] Thu H Nguyen-Phuoc, Christian Richardt, Long Mai, Yongliang Yang, and Niloy Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. *Advances in Neural Information Processing Systems*, 33:6767–6778, 2020. 2

[37] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 6, 7

[38] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 2

[39] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5762–5772, 2021. 2

[40] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2856–2865, 2021. 2

[41] Andrew Owens, Connelly Barnes, Alex Flint, Hanumant Singh, and William Freeman. Camouflaging an object from many viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2782–2789, 2014. 2, 4, 7

[42] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9964–9973, 2019. 2

[43] 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. 2, 5

[44] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[45] Aditya Sanghi, Hang Chu, J. Lambourne, Ye Wang, Chin-Yi Cheng, and Marco Fumero. Clip-forge: Towards zero-shot text-to-shape generation. *ArXiv*, abs/2110.02624, 2021. 3

[46] Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, and Marco Fumero. Clip-forge: Towards zero-shot text-to-shape generation. *arXiv preprint arXiv:2110.02624*, 2021. 2

[47] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3858–3867, 2019. 2

[48] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *CVPR*, 2021. 2

[49] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. *arXiv preprint arXiv:2112.05139*, 2021. 2

[50] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018. 2

[51] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf-: Neural radiance fields without known camera parameters. *ArXiv*, abs/2102.07064, 2021. 2

[52] Silvan Weder, Guillermo Garcia-Hernando, Aron Monszpart, Marc Pollefeys, Gabriel J Brostow, Michael Firman, and Sara Vicente. Removing objects from neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16528–16538, 2023. 3

[53] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016. 2

[54] Qianyi Wu, Xian Liu, Yuedong Chen, Kejie Li, Chuanxia Zheng, Jianfei Cai, and Jianmin Zheng. Object-compositional neural implicit surfaces. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 197–213. Springer, 2022. 3

[55] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *ICCV*, 2019. 2

[56] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13779–13788, 2021. 1, 2, 3, 5, 6, 7

[57] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4541–4550, 2019. 2

[58] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. iNeRF: Inverting neural radiance fields for pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021. 2

[59] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019. 3, 5, 6