Decomposing NeRF for Editing via Feature Field Distillation

Sosuke Kobayashi
Preferred Networks, Inc.
sosk@preferred.jp

Eiichi Matsumoto
Preferred Networks, Inc.
matsumoto@preferred.jp

Vincent Sitzmann
Massachusetts Institute of Technology
sitzmann@mit.edu

pfnet-research.github.io/distilled-feature-fields/

Abstract

Emerging neural radiance fields (NeRF) are a promising scene representation for computer graphics, enabling high-quality 3D reconstruction and novel view synthesis from image observations. However, editing a scene represented by a NeRF is challenging, as the underlying connectionist representations such as MLPs or voxel grids are not object-centric or compositional. In particular, it has been difficult to selectively edit specific regions or objects. In this work, we tackle the problem of semantic scene decomposition of NeRFs to enable query-based local editing of the represented 3D scenes. We propose to distill the knowledge of off-the-shelf, supervised and self-supervised 2D image feature extractors such as CLIP-LSeg or DINO into a 3D feature field optimized in parallel to the radiance field. Given a user-specified query of various modalities such as text, an image patch, or a point-and-click selection, 3D feature fields semantically decompose 3D space without the need for re-training and enable us to semantically select and edit regions in the radiance field. Our experiments validate that the distilled feature fields can transfer recent progress in 2D vision and language foundation models to 3D scene representations, enabling convincing 3D segmentation and selective editing of emerging neural graphics representations.

1 Introduction

Emerging neural implicit representations or neural fields have been shown to be a promising approach for representing a variety of signals [82, 53, 65, 106, 56]. In particular, they play an important role in 3D scene reconstruction and novel view synthesis from a limited number of context images. Neural radiance fields (NeRF) [56] enabled the recovery of a continuous volume density and radiance field from a limited number of observations, producing high-quality images from arbitrary views via volume rendering with promising applications in computer graphics. However, editing a scene reconstructed by NeRF is non-obvious because the scene is not object-centric and is implicitly encoded in the weights of a connectionist representation such as an MLP [56] or a voxelgrid [23]. Although we can transform the scene in input or output space or via optimization-based editing [37, 97], this does not enable selective *object-centric* or *semantic*, local edits, such as moving a single object. Prior work has addressed this challenge via coordinate-level, semantic decompositions which allow to selectively move, deform, paint, or optimize parts of a NeRF, but relies on costly annotation of instance segmentations and training of instance-specific networks [104]. While this can be alleviated with pre-trained segmentation models [25, 41], such models require pre-defined closed label sets and

Preprint. Under review.

domains (e.g., traffic scenes), limiting decomposition and editing. Local editing of NeRFs ideally requires an efficient, open-set method for coordinate-level decomposition.

In this work, we present *distilled feature fields* (DFFs), a novel approach to query-based scene decomposition for local, interactive editing of NeRFs. We focus on 3D neural *feature* fields, which map every 3D coordinate to a semantic feature descriptor of that coordinate. Conditioned on a user query such as a text or image patch, this 3D feature field can compute a decomposition of a scene without re-training. We train a scene-specific DFF via teacher-student distillation [34], using supervision from feature encoders pre-trained on the image domain. Unlike the domain of 3D scenes, the image domain boasts massive high-quality datasets and abundant prior work on self-supervised and supervised training of effective feature extraction models. Notably, recently proposed transformer-based models [96, 22] have demonstrated impressive capabilities across various vision- and text-based tasks (e.g., CLIP [69], LSeg [44], DINO [12]). Such feature spaces capture the semantic properties of regions and make it possible to correspond and segment them well by text, image queries, or clustering. We employ these models as teacher networks and distill them into 3D feature fields via volume rendering. The trained feature field enables us to semantically select and edit specific regions in 3D NeRF scenes and render multi-view consistent images from the locally edited scenes.

In extensive experiments, we investigate the applications of neural feature fields with two different pre-trained teacher networks, (1) LSeg [44], a CLIP-inspired language-driven semantic segmentation network, and (2) DINO [12, 3], a self-supervised network aware of various object boundaries and correspondences. LSeg and DINO features allow us to select 3D regions by a simple text query or an image patch, respectively. We first quantitatively demonstrate that LSeg-based DFFs with label queries can have high 3D segmentation performance compared with an existing point-cloud based 3D segmentation baseline trained on ScanNet [20], a supervised point-cloud dataset. We then demonstrate a variety of 3D appearance and geometry edits across real-world NeRF scenes with no annotations of segmentation; and show that we may edit regions with a single query of text, image, pixel, or cluster choice.

2 Related Work

Neural Implicit Representations. Neural implicit representations or neural fields have recently advanced neural processing for 3D data and multi-view 2D images [82, 53, 65, 106, 56]. For a review of this emerging space we point the reader to the reports by Kato et al. [39], Tewari et al. [90], and Xie et al. [102]. In particular, a neural radiance field (NeRF) can be fitted to a set of posed 2D images and maps a 3D point coordinate and a view direction to RGB color and density. When observations are limited, NeRF often overfits and fails to synthesize novel views with correct geometry and appearance. Pre-trained vision models have been used for regularizing NeRF via flows [62], multi-view consistency [35], perceptual loss [110], or depth estimation [100, 77]. Some pre-trained models operate not only in the visual world but also in other modalities such as language. The recently proposed CLIP model [69] has demonstrated impressive performance in image-and-text alignment, with strong generalization to various textual and visual concepts. Wang et al. [97], Jain et al. [36], and Poole et al. [68] use CLIP or Imagen [79] to edit or generate a single-object NeRF with a text prompt query by optimizing the NeRF parameters to generate images matched with the text. While such methods are promising, they do not enable accurate selective editing of specific scene regions. For example, the prompt "yellow flowers" may affect unintended scene regions, such as the leaves of a plant. Our proposed decomposition method leverages pre-trained foundation models to enable selective editing of real-world NeRF scenes. Neural descriptor fields [80] use intermediate features that emerge in a 3D occupancy field network [53] for efficiently teaching robots object grasping. Instead of a pre-trained object-centric 3D model, we use 2D vision models as teacher networks via distillation, exploiting recent progress in pre-trained foundation models [7].

Geometric Decomposition of Neural Scene Representations Kohli et al. [40] and Zhi et al. [112] show that neural implicit representations can be combined with the supervision of semantic labels. Yang et al. [104] demonstrate that given view-consistent ground-truth instance segmentation masks during training, NeRF can be trained to represent each object as different volumes, although such an annotation is expensive in practice. Concurrently, Benaim et al. [6] also experiment with the different parametarization. Conditional [49, 37, 21, 63] and generative models [60, 61, 31] enable a degree

of category-specific decomposition (e.g., human bodyparts) and editing on constrained domains with large datasets. Regular structures such as voxelgrids or octrees [13, 48, 14, 88, 94, 89, 43, 81, 107, 59, 60] or unsupervised decomposition [73, 85, 109, 83] enable editability via manipulation of localized parameters. However, the decomposition is limited due to the inflexibly structured boundaries or strong assumptions about scenes; self-supervised object-centric learning is a difficult task. Other studies also explored reconstruction with more structured hybrid representations via pipelines specialized to a domain (e.g., traffic scene) [64, 25, 41] or situation (e.g., each object data is independently accessible) [28, 27, 105]. Note that this line of work defines and constrains domains or the types of segmentation during or before training and thus limits the degrees of freedom for editable scenes and objects. In contrast, our method can decompose scene-specific NeRFs into arbitrary semantic units via text and image queries, enabling versatile scene edits without re-training. A concurrent paper by Tschernezki et al. [93] also explores the same training framework and, in particular, investigates how fused features are improved from 2D teacher networks. It also complementarily shows the results with other teacher models (MoCo-v3 [17] and DeiT [91]), dimension reduction via PCA, and NeuralDiff [92]-based neural fields. Other concurrent studies explore decomposition through training scene-specific segmentation field [113] or 3DCNN [76] supervised by click or scribble annotations. Lastly, in a different but related task, video editing, Kasten et al. [38] use foreground-background decomposition and atlas representation for time-consistent, local editing; Loeschcke et al. [51] and Bar-Tal et al. [5] further use CLIP for editing.

Zero-shot Semantic Segmentation. Zero-shot semantic segmentation is a challenging task [24, 2, 10] where a model has to predict semantic labels of pixels in images without a-priori information of the categories. A typical solution is to use vision-and-language cross-modal encoders. They are trained to encode images (pixels) and text labels into the same semantic space and perform zero-shot prediction based on the similarity or alignments of the two inputs. Recent development of image encoder architectures [96, 22, 71] and large-scale training [69, 12] have improved the ability and generalization of vision models, including zero-shot models [44, 52, 99, 103, 114, 72]. On the other hand, ongoing studies on zero-shot perception in 3D still suffer from the lack of effective, efficient, and high-resolution architectures and large-scale annotated datasets [54, 33, 29, 101, 78, 30]. Our method is a new approach to perform zero-shot semantic segmentation on scene-specific 3D fields by exploiting progress in the image domain without semantic 3D supervision. We note that the goal of this paper is not to achieve state-of-the-art performance on 3D semantic segmentation tasks. Instead, our goal is the decomposition of neural scene representations for editing, which requires smooth segmentation results on continuous 3D space rather than segmentation of discrete point clouds or voxelgrids.

3 Preliminaries

3.1 Neural Radiance Fields (NeRF)

NeRF [56] uses MLPs to output density σ and color $\mathbf{c}$ given a point coordinate $\mathbf{x} = (x, y, z)$ in a 3D scene. This simple scene representation can be rendered and optimized via volume rendering. Given a pixel's camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, depth t with bounds $[t_{\text{near}}, t_{\text{far}}]$, camera position $\mathbf{o}$, and its view direction $\mathbf{d}$, NeRF calculates the color of a ray using quadrature of K sampled points $\{\mathbf{x}_k\}_{k=1}^K$ with depths $\{t_k\}_{k=1}^K$ as

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{k=1}^{K} \hat{T}(t_k)\, \alpha\left(\sigma(\mathbf{x}_k)\delta_k\right) \mathbf{c}(\mathbf{x}_k, \mathbf{d}), \quad \hat{T}(t_k) = \exp\left(-\sum_{k'=1}^{k-1} \sigma(\mathbf{x}_{k'})\delta_{k'}\right), \tag{1}$$

where $\alpha(x) = 1 - \exp(-x)$, and $\delta_k = t_{k+1} - t_k$ is the distance between adjacent point samples. NeRFs are optimized solely on a dataset of images and their camera poses by minimizing a re-rendering loss.

3.2 Pre-trained Models and Zero-shot Segmentation of Images

Most semantic segmentation models pre-define a closed set of labels, and cannot flexibly change the segmentation categories or boundaries without supervised training. In contrast, zero-shot semantic segmentation predicts target regions given open-set queries. Li et al. [44] proposes LSeg, a model to

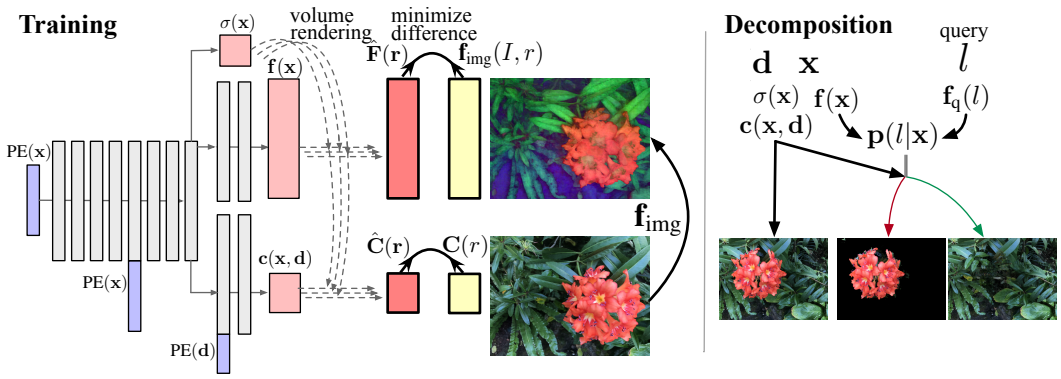

Figure 1: Left: A Distilled Feature Field (DFF) maps a coordinate $\mathbf{x}$ and a viewing direction $\mathbf{d}$ to density σ, color $\mathbf{c}$, and feature $\mathbf{f}$. It is trained by minimizing the difference between rendered features and features as predicted by a pre-trained image feature encoder, as well as the rendered color and ground-truth pixel color. Right: At test time, we may decompose and edit 3D space via selecting and manipulating different 3D regions with a variety of queries.

perform zero-shot semantic segmentation by aligning pixel-level features and a text query feature. LSeg employs an image feature encoder with the DPT architecture [71] and a CLIP-based text label feature encoder [69], trained via large-scale language-image contrastive learning. The probability of a text label l given a pixel r in an image I, $\mathbf{p}(l|I, r)$, is then calculated via dot product of pixel-level image feature $\mathbf{f}_{\text{img}}(I, r)$ and queried text feature $\mathbf{f}_{\text{q}}(l)$ followed by a softmax:

$$\mathbf{p}(l|I, r) = \frac{\exp(\mathbf{f}_{\text{img}}(I, r)\mathbf{f}_{\text{q}}(l)^{\text{T}})}{\sum_{l' \in \mathcal{L}} \exp(\mathbf{f}_{\text{img}}(I, r)\mathbf{f}_{\text{q}}(l')^{\text{T}})} \quad , \tag{2}$$

where $\mathcal{L}$ is a set of possible labels. If negative labels are not available, we may use other scores like thresholded cosine similarity to directly compute the probability of a label. During training, LSeg optimizes only the image encoder $\mathbf{f}_{\text{img}}(I, r)$ by minimizing cross-entropy on supervised semantic segmentation datasets. The text encoder $\mathbf{f}_{\text{q}}(l)$ is obtained from a pre-trained CLIP model [69]. Recently, pre-trained CLIP has been leveraged as the backbone for a variety of tasks and has been extended with additional modules sharing the same latent space. For example, Reimers and Gurevych [74, 75] trains a multi-lingual (more than 50+ languages) text encoder, which enables CLIP and CLIP-inspired variants to use non-English queries like Japanese. We similarly use the latent space of a pre-trained CLIP for LSeg via distillation, enabling the decomposition of NeRFs with both English and non-English queries. Segmentation can further be performed with other modalities such as image, patch or pixel query features $\mathbf{f}_{\text{q}}$ using a similar dot-product similarity formulation as in Eq. 2. Notably, DINO [12], a self-supervised vision model, solves video instance segmentation and tracking by calculating similarity among features in adjacent frames. Amir et al. [3] also demonstrate that DINO features work well on co-segmentation and point correspondence by similarity and clustering. In our experiments, we use these two publicly available models, LSeg and DINO, to obtain features of images and texts for 3D decomposition.

4 Distilled Feature Fields

4.1 Distilling Foundation Modules into 3D Feature Fields via Volume Rendering

NeRF learns a neural field to compute the density and view-dependent color, $\sigma(\mathbf{x})$ and $\mathbf{c}(\mathbf{x}, \mathbf{d})$. We may extend NeRF by adding decoders for other quantities of interest. For example, SemanticNeRF [112] adds a branch outputting a probability distribution of closed-set semantic labels, trained with supervision via images with ground-truth semantic labels. This enables the prediction of pairs of RGB and semantic segmentation masks from novel views, useful for data augmentation. However, because ground-truth annotation is costly, the method is inefficient as a means of scene editing [104]. For specific domains like traffic scenes [25, 41], we may instead train a closed-set segmentation model and use its prediction for training object-aware neural fields. However, this approach is possible only if the types of objects are limited and the domain-specific supervised dataset is available; limiting the application of scene editing in terms of domain and flexibility of decomposition.

We build on top of these ideas and perform 3D zero-shot segmentation of NeRFs using open-set text labels or other feature queries. Instead of a branch performing closed-set classification, we propose to add a feature branch outputting a feature vector itself. This branch models a 3D feature field describing the semantics of each spatial point. We supervise the feature field by a pretrained pixel-level image encoder $\mathbf{f}_{\text{img}}$ as a teacher network. Given a 3D coordinate $\mathbf{x}$, the feature field outputs a feature vector $\mathbf{f}(\mathbf{x})$ in addition to density $\sigma(\mathbf{x})$ and color $\mathbf{c}(\mathbf{x}, \mathbf{d})$, as shown in Fig. 1. Volume rendering of the feature field is similarly performed via

$$\hat{\mathbf{F}}(\mathbf{r}) = \sum_{k=1}^{K} \hat{T}(t_k)\, \alpha(\sigma(\mathbf{x}_k)\delta_k)\, \mathbf{f}(\mathbf{x}_k) \ . \tag{3}$$

We can optimize $\mathbf{f}$ by minimizing the difference between rendered features $\hat{\mathbf{F}}(\mathbf{r})$ and the teacher's features $\mathbf{f}_{\text{img}}(I, r)$. Effectively, we are distilling [34] the 2D teacher network into our 3D student network via differentiable rendering, and thus dub this model a *distilled feature field* (DFF). We add a feature objective $\mathcal{L}_f$ penalizing the difference between rendered features $\hat{\mathbf{F}}(\mathbf{r})$ and the teacher's outputs $\mathbf{f}_{\text{img}}(I, r)$ to the photometric loss of the original NeRF. We use two networks for volume rendering with coarse-and-fine hierarchical sampling. We thus minimize the sum of photometric loss L_p and feature loss L_f, in total, L:

$$L = L_p + \lambda L_f, \quad L_p = \sum_{\mathbf{r}\in\mathcal{R}} \left\| \hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(r) \right\|_2^2, \quad L_f = \sum_{\mathbf{r}\in\mathcal{R}} \left\| \hat{\mathbf{F}}(\mathbf{r}) - \mathbf{f}_{\text{img}}(I, r) \right\|_1, \tag{4}$$

where $\mathcal{R}$ are sampled rays, $\mathbf{C}(r)$ is the ground truth pixel color of ray r, λ is the weight of the feature loss and is set to 0.04 to balance the losses [112]. We apply stop-gradient to density in rendering of features $\hat{\mathbf{F}}(\mathbf{r})$ in Equation 3 as the teacher's features $\mathbf{f}_{\text{img}}(I, r)$ are not fully multi-view consistent, which could harm the quality of reconstructed geometry.

4.2 Query-based Decomposition and Editing

A trained DFF model can perform 3D zero-shot segmentation by its feature field $\mathbf{f}$ and a query encoder $\mathbf{f}_{\text{q}}$. Probability of a label l of a point $\mathbf{x}$ in the 3D space, $\mathbf{p}(l|\mathbf{x})$, is calculated by dot product of the 3D feature $\mathbf{f}(\mathbf{x})$ and text label feature $\mathbf{f}_{\text{q}}(l)$ followed by a softmax:

$$\mathbf{p}(l|\mathbf{x}) = \frac{\exp(\mathbf{f}(\mathbf{x})\mathbf{f}_{\text{q}}(l)^{\text{T}})}{\sum_{l'\in\mathcal{L}} \exp(\mathbf{f}(\mathbf{x})\mathbf{f}_{\text{q}}(l')^{\text{T}})} \ . \tag{5}$$

This query-based segmentation field is at the core of the proposed method. It can be calculated at any 3D point without limiting resolution, naturally used in tandem with a radiance field and volume rendering. Note that the segmentation depends on only the 3D coordinate and the query[1]. As the original NeRF, it is thus multi-view consistent. In addition and important for interactive editing, we can change the segmentation via queries without re-training, which cannot be realized by closed-set methods using semantic [112] or instance segmentation annotation [104]. We may now use this query-conditional segmentation to identify a specific 3D region for editing. Various edits can be generalized to the merging of two NeRF scenes $\sigma_1(\mathbf{x}), \mathbf{c}_1(\mathbf{x}, \mathbf{d})$ and $\sigma_2(\mathbf{x}), \mathbf{c}_2(\mathbf{x}, \mathbf{d})$, where we use the segmentation field $\mathbf{p}$ for blending. In the experiments section, we simply modify Eq. 1 as a blend of two scenes based on the ratio of α:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{k=1}^{K} \hat{T}(t_k) \left(\alpha(\sigma_1(\mathbf{x}_k)\delta_k)\, \mathbf{c}_1(\mathbf{x}_k, \mathbf{d})\rho_k + \alpha(\sigma_2(\mathbf{x}_k)\delta_k)\, \mathbf{c}_2(\mathbf{x}_k, \mathbf{d})(1-\rho_k) \right) , \tag{6}$$

$$\text{where} \quad \rho_k = \frac{\alpha(\sigma_1(\mathbf{x}_k)\delta_k)}{\alpha(\sigma_1(\mathbf{x}_k)\delta_k) + \alpha(\sigma_2(\mathbf{x}_k)\delta_k)} \ , \ \hat{T}(t_k) = \prod_{k'=1}^{k-1} \alpha(\sigma_1(\mathbf{x}_{k'})\delta_{k'}) + \alpha(\sigma_2(\mathbf{x}_{k'})\delta_{k'}) \ . \tag{7}$$

For example, if we want to apply a geometric transformation $\mathbf{g}$ to a region of a query l in a NeRF scene $(\sigma, \mathbf{c})$, we can render the transformed scene via Eqs. 6 and 7 by setting $\alpha(\sigma_1(\mathbf{x}_k)\delta_k) = (1 - \mathbf{p}(l|\mathbf{x}_k))\alpha(\sigma(\mathbf{x}_k)\delta_k)$, $\alpha(\sigma_2(\mathbf{x}_k)\delta_k) = \mathbf{p}(l|\mathbf{g}^{-1}(\mathbf{x}_k))\alpha(\sigma(\mathbf{g}^{-1}(\mathbf{x}_k))\delta_k)$, $\mathbf{c}_1(\mathbf{x}_k, \mathbf{d}) = (1 - \mathbf{p}(l|\mathbf{x}_k))\mathbf{c}(\mathbf{x}_k, \mathbf{d})$, and $\mathbf{c}_2(\mathbf{x}_k, \mathbf{d}) = \mathbf{p}(l|\mathbf{g}^{-1}(\mathbf{x}_k))\mathbf{c}(\mathbf{g}^{-1}(\mathbf{x}_k), \mathbf{g}^{-1}(\mathbf{d}))$. More details of editing for

[1] It is an interesting extension to introduce a user's viewpoint to the function for recognizing view-dependent queries like referring expressions (e.g., "the chair left to the table") [16, 1, 47, 4]. We leave this to future work.

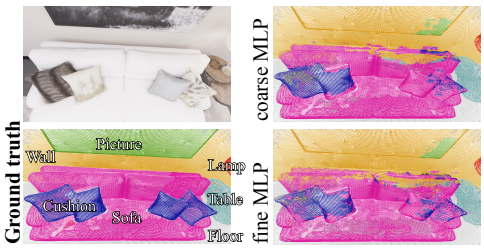

Figure 2: Comparison of predictions by coarse and fine MLPs.

Table 1: Performance of 3D semantic segmentation on Replica dataset. DFF outperforms a *supervised* point-cloud segmentation model MinkowskiNet42.

	mIoU	accuracy
Supervised 3DCNN	0.475	0.758
DFF (Coarse)	0.589	0.855
DFF (Fine)	0.583	0.855

colorization, translation, and deletion are shown in Appendix B. We can combine this with more complex edits, including optimization-based methods like CLIPNeRF [97]. While CLIPNeRF itself cannot selectively edit specific regions in multi-object scenes, our decomposition method enables it to update only desired objects without breaking unintended areas.

5 Experiments

We first conduct a quantitative evaluation of the decomposition achieved by DFF. We demonstrate that DFF enables 3D semantic segmentation in a benchmark dataset using scanned point clouds with human-annotated semantic segmentation labels. We then investigate the capabilities of DFF for editing and subsequent novel-view synthesis on real-world datasets. We use two teacher networks, LSeg [44] and DINO [12], which are pre-trained and publicly available. Each training image is encoded by the image encoders of the networks and used as target feature maps, $\mathbf{f}_{\text{img}}(I, r)$, defined in Equation 4. Because the feature maps are of reduced sizes due to the limitation of the networks, we first resize them to the original image size. The implementation and settings of NeRF, unless otherwise stated, follow Zhi et al. [112]. During the training of 200K iterations, the loss L in Equation 4 is minimized by Adam with a linearly decaying learning rate (5e-4 to 8e-5). During training, Gaussian noise for density is also applied. The number of coarse and fine samplings is 64 and 128, respectively. The MLP of the neural radiance field consists of eight ReLU layers with 256 dimensions, followed by a linear layer for density, three layers for color, and three layers for feature, as shown in Fig. 1. Positional encoding of length 10 is used for the input coordinate and its skip connection, and that of length 4 is for viewing direction. If an independent MLP is prepared for the feature field, it consists of four layers (with a skip connection at the third layer if the positional encoding is used). The size of a training image is 320×240 for the Replica dataset and 1008×756 for the other datasets. The batchsize of training rays is 1024 for Replica and 2048 for the others. During finetuning of feature fields or radiance fields, Gaussian noise is removed, and the learning rate is set to 1e-4. See appendix A and C for further training details.

5.1 3D Semantic Segmentation

We construct a 3D semantic segmentation benchmark from four scenes in the Replica dataset [86] with data split and posed images provided by [112]. See appendix D for further details of the dataset. We train DFF to reconstruct each scene with radiance and feature fields from training images and evaluate the quality of novel view synthesis and 3D segmentation of the annotated point clouds. We use LSeg as a teacher network. The LSeg text encoder encodes each label, and the probability of each point is calculated by Equation 2[2]. Note that the training uses only the photometric and feature losses (Equation 4) and does not access any supervision via semantic labels.

Semantic Segmentation Results. First, we show evaluation metrics of 3D semantic segmentation, mean intersection-over-union (mIoU) and accuracy in Table 1. For comparison, we also experiment with a sparse 3D convolution-based segmentation model, MinkowskiNet42 [18] taking a colored

[2]While LSeg-DFF can perform zero-shot inference using text labels that are not seen during training, we do not focus on thoroughly evaluating the zero-shot ability. The evaluation has been conducted in the original paper on the teacher network, and DFF's ability is expected to follow it due to distillation. Please refer to Li et al. [44] for the detail of the zero-shot ability of LSeg.

Table 2: Performance of novel view synthesis on Replica dataset. PSNR, SSIM, and LPIPS are metrics of image synthesis. $\delta < 1.25$ and absrel are metrics of geometry (depth estimation).

	PSNR↑	SSIM↑	LPIPS↓	$\delta < 1.25$↑	absrel↓
basic NeRF	32.87	0.934	0.148	0.993	0.018
DFF	32.85	0.932	0.150	0.993	0.017
DFF (overweighting λ)	32.68	0.927	0.162	0.993	0.018

point cloud as input. It has a standard state-of-the-art architecture for point cloud segmentation and is trained on the ScanNet dataset [20], the largest annotated training dataset of 3D semantic segmentation[3]. Results demonstrate that DFF, taught by LSeg, achieves promising performance, even better than the supervised model. This indicates that DFF succeeds at distilling 3D semantic segmentation from the 2D teacher network.

Impact of Sampling on Semantic Segmentation. NeRF employs two MLPs for hierarchical sampling, where the coarse MLP performs volume rendering with fewer points (64) using stratified sampling, and the fine MLP works with importance sampling (192 in total). So, we have two sampling options to train a feature field. Although fine sampling is critical for training accurate radiance fields, segmentation is of significantly lower spatial frequency than texture. We thus analyze the impact of coarse and fine training in Fig. 2. As expected, the coarse model produces smooth segmentations, while the fine version introduces high-frequency artifacts. This smoothness property is important for natural editable novel view synthesis and is discussed again later.

Compatibility with View Synthesis. We also check and compare the quality of novel view synthesis with NeRF, which does not learn feature fields. Because the feature branch partially shares the layers with the radiance field (as shown in Fig. 1), learning feature fields could possibly harm the radiance field. Despite this concern, as shown in Tab. 2, the performance of view synthesis is not degraded. Thus, we can train and use the branch-based DFF with small computational and parameter overhead compared to the original NeRF. If we excessively increased the weight of the feature loss, $\lambda \times 10$, it hurt view synthesis while not improving segmentation performance further. We further confirm that training independent, light-weight feature-field MLP, instead of a branch of the radiance-field MLP, achieves semantic segmentation results competitive with the branch-based approach (see appendix Tab. 3 for the result of all variants)[4]. This option is useful especially when we want to introduce DFF decomposition into arbitrary 3D scene representations, including off-the-shelf NeRF models, dynamic NeRFs [26, 66, 45], or meshes, without re-training of the radiance field.

5.2 Editable Novel View Synthesis

In the previous section, we quantitatively validated the ability of DFF to perform semantic decomposition. We now discuss the capability for editable view synthesis on real-world scenes, including the LLFF dataset [55] and our own dataset. Our method can be used even for LLFF scenes based on normalized device coordinates. Please see the supplemental web page for further results, including videos. In addition to LSeg using a text query, we also experiment with self-supervised DINO [12] as another teacher network to enable query-based decomposition using image patch queries. Here, we use thresholded cosine similarity to directly compute the probability of a query instead of softmax with negative queries in Eq. 5 and set $\mathbf{p} = 1$ if the similarity exceeds the threshold [5], and $\mathbf{p} = 0$ otherwise for hard decomposition. We first train NeRFs without a feature branch for each scene for 200K iterations (L_p) and then finetune them with a feature branch via distillation for 5K iterations ($L_p + \lambda L_f$) since we found that the feature loss converged significantly faster than the photometric loss and short training was thus sufficient. We use coarse sampling for training feature branches and use it for edited rendering with fine sampling. See appendix A for the details.

[3]For a fair comparison, the label set follows the ScanNet dataset. We also manually tune the range and scale of input point clouds to maximize the performance of MinkowskiNet42 on the Replica dataset.

[4]Note that, even if we are interested in the feature-field MLP, it is important to simultaneously prepare the radiance-field MLP for reconstructing geometry well and training the feature field via geometrically plausible volume rendering.

[5]The results of the LLFF room scene used a label set, {whiteboard, ceiling, light, television, wall, bin, table, cabinet, cable, chair, box, floor}, for avoiding tuning of thresholds.

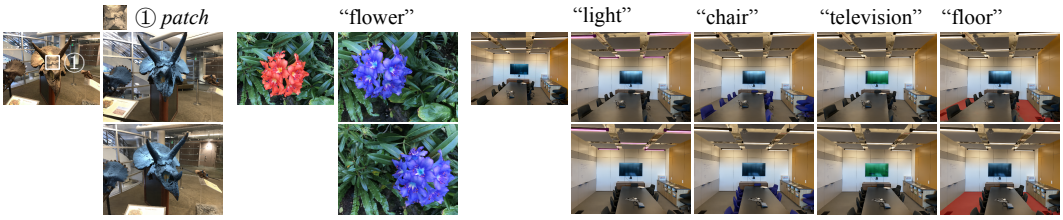

Figure 3: Appearance edits of specific objects via different query modalities: an image patch or text.

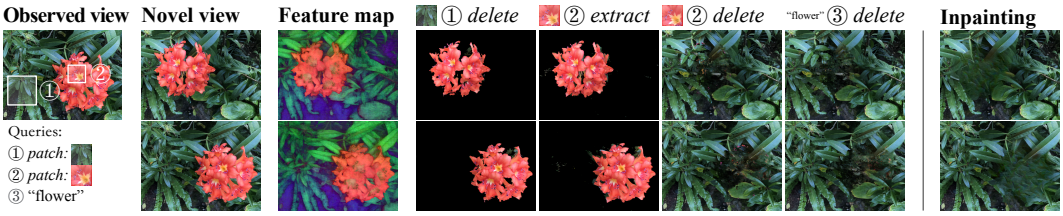

Figure 4: Extraction and deletion of specific objects via different query modalities, an image patch or text. The edited views are 3D consistent, unlike an image inpainting baseline [87]

Appearance Editing, Deletion, Extraction. We show qualitative evaluations of novel view synthesis in Fig. 3 and Fig. 4. Specific 3D regions in these scenes are identified and locally edited via decomposition depending on various query modalities. In these experiments, we use a text query for LSeg-DFF as in Section 5.1 and use an image patch query for DINO-DFF. Because DINO features capture the similarity and correspondences of regions well thanks to self-supervised learning [12, 3], image patch queries help select all semantically similar areas at once. The patch feature is then calculated by averaging the features of all pixels in the patch.

In Fig. 3, we demonstrate that the DFF enables convincing selective appearance edits. Because our focus is region selection via decomposition, we use simple color transformation for clarity here (e.g., flip RGB to BGR, blend colors). One might think that the MLP of a radiance field by the original NeRF also has hidden layers, and their features could possibly be used for decomposition. We confirm that the naive usage of NeRF features is not robust to decomposition, as shown in Fig. 5, especially in a complex multi-object scene. We use the 8th hidden layer of the fine radiance field network (i.e., the layer just before branching in Fig. 1)[6]. NeRF features cannot clearly decompose even objects with simple shapes and colors. The region selections are leaked to other parts with similar colors, geometry, or positions while they do not entirely cover the targets. For example, floor

[6]Other layers or the coarse MLP of the NeRF also indicated similar behaviors but a little worse qualitatively.

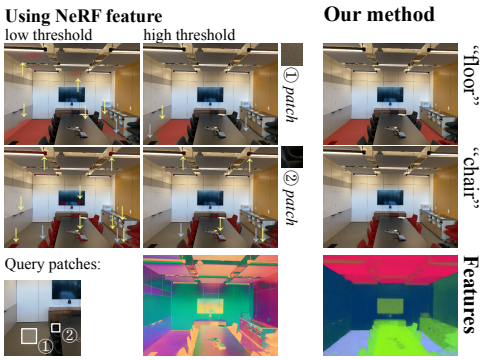

Figure 5: Appearance edits of specific objects, compared with decomposition using features of a NeRF hidden layer. For reference, we also show PCA-based visualizations of the features.

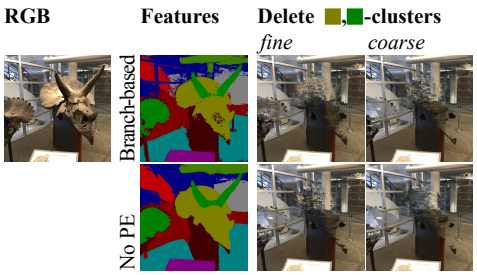

Figure 6: Comparison of predictions by a branch-based feature field MLP and independent MLP with no positional encoding, each of which is trained with coarse and fine sampling.

selection is leaked to walls, a table, bins, or ceilings. Chair selection is leaked to irrelevant black parts like television, cables, lighting equipment, or shadows. This indicates that the feature space of the original NeRF does not learn semantic similarity well and is entangled with unpredictable and more low-level factors like color or spatial adjacency.

In Fig. 4, we demonstrate that the DFF also works well on deletion or extraction of objects, using two patch queries (query-① for leaves and ground, query-② for flowers) and a text query-③ "flower". For comparison with a baseline editing method, we show the results by a state-of-the-art image inpainting model, LaMa [87]. Because the model requires masks for inpainting regions, we manually annotate the views for evaluation. As shown in the figure, the image inpainting model cannot generate clear and realistic images, and the different views are inconsistent. On the other hand, DFF produces multi-view consistent plausible results, especially succeeding at extracting foreground objects. Although the performance on deleting foreground objects is high, a remaining shortcoming is the existence of floating artifacts and blurred volumes in the far distance behind the deleted object.

Priors for Smooth Decomposition We can organize the challenges of editable NeRFs into several categories: surface decomposition, volume decomposition, lighting decomposition, and estimation of less or never observed parts. If we edit appearances only, it practically requires decomposing regions only near the surface of objects, i.e., surface decomposition, because the color of a ray is determined mostly in a condensed interval around the surface. On the other hand, geometric transformations often require a higher level of decomposition. As shown in the deletion examples, geometric transformation may move or remove some surfaces and expose the space behind them. This forces models to render unknown regions less or never observed due to occlusions, including even the inside of objects. Thus, it is desirable to decompose volumes smoothly while synthesizing their inside and back[7]. Although these include the same challenges as novel view synthesis tackles, editability further highlights their importance.

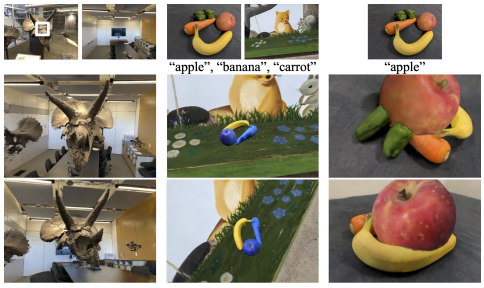

Figure 7: Editing with warping, deformation, shift, and rotation.

Apart from lighting decomposition discussed in prior work [8, 9, 111], we further investigate the new challenge of smooth volume decomposition by experimenting with different DFF setups. As discussed in Section 5.1, DFF has two sampling options to train feature fields. The coarse training may introduce smoothness regularization and help cohesive decomposition and smoother in-painting of unobserved regions. Another reasonable smoothness regularizer is to eliminate the high-frequency positional encoding (PE). We thus train an independent MLP network for a feature field without PE. We compare four combinations of renderings in Fig. 6. To better understand their behavior, we use the DINO-DFF, show k-means clusters of the rendered feature map, and delete the head of the Triceratops by a query choosing its corresponding clusters. As expected, coarsely trained models and no-PE models succeed in smoother volume decomposition, and this combination can minimize high-frequency floating artifacts. A side effect is the lack of high-frequency representation power, which sometimes deletes disparate background regions and misses to represent features of complex structures (e.g., see the cluster visualization of the thin frames of the window). Towards the best of both worlds, developing proper priors or inductive biases is an important direction for future work [70]. Otherwise, surface-aware representations like IDR [106, 98] could avoid problems with floating artifacts. Note that not all geometric edits suffer from these problems. For example, it is often less problematic to move objects closer to the camera, enlarge them, or warp them to other scenes, as shown in Fig. 7.

Localizing Optimization-based Editing. Finally, we show a combination with an optimization-based editing method. CLIPNeRF [97] optimizes the parameters of a radiance field so that its rendered images match with a text prompt via CLIP. While it is mainly designed for a single-object scene of specific categories, it is possible to apply to other real-world NeRFs. However, because it cannot control the scope of editing, a prompt like "white flower" may change the color of unintentional

[7]Note that this problem also arises when Yang et al. [104] used *ground-truth* instance segmentation masks and trained multiple networks, although the authors did not investigate this issue.

Figure 8: Comparison of appearance editing by CLIPNeRF and our extension.

targets like leaves. Our DFF-based decomposition can upgrade such an optimization-based method to render a scene via the composition of a CLIP-optimized NeRF scene and the original NeRF scene. We show the results in Fig. 8[8]. Although the naive CLIPNeRF edits unintentional parts, our method helps it to locally edit intentional parts only. In addition to switching rendering, we can also use the decomposition for controlling training signals during backpropagation. The additional experiment is shown in Appendix F. These extensions broaden the application of CLIPNeRF or other optimization-based editing methods to complex scenes.

6 Discussion, Limitations, and Conclusions

In this work, we propose distilled feature field (DFF), a novel method of NeRF scene decomposition for selective editing. We present quantitative evaluations of segmentation and extensive qualitative evaluations of editable novel view synthesis. In addition to these promising results, DFF-based models will benefit from future improvements to self-supervised 2D foundation models. We also clarify future directions on editable view synthesis through our experiments, especially for smoothness priors and estimation of unobserved regions. Furthermore, while this work focuses on editable view synthesis, it is also intriguing to transfer DFF to other applications, including 3D registration of text queries [16, 1, 47, 4] or robot teaching [32, 80].

The limitations of the DFF framework are two-fold. The first one is the upper bduround of the performance due to distillation. The student model of distillation cannot largely outperform the teacher model[9]. If the resolution of teacher encoders is low, the corresponding DFFs also becomes coarse-grained. If the LSeg cannot understand a text query, the LSeg-DFF also cannot. Secondly, the DFF uses volume rendering depending on the 3D reconstruction by NeRF. A NeRF model is sometimes optimized to geometrically wrong solutions (e.g., floaters). Such geometry errors of the radiance fields would make supervision to DFFs noisy.

As a possible negative societal impact, one might use our method for making realistic but fake content by editing NeRFs as desired. Automatic fake detection methods may help in preventing such misuse. NeRFs are further computation-intense, leading to high electricity usage. Recent work on efficient NeRFs [23, 57, 15] may alleviate this concern.

Acknowledgements

We thank Tsukasa Takagi, Toshiki Nakanishi, Hiroharu Kato, and Masaaki Fukuda for their helpful feedback.

References

[1] Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas J. Guibas. ReferIt3D: Neural listeners for fine-grained 3d object identification in real-world scenes. In *16th European Conference on Computer Vision (ECCV)*, 2020.

[2] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38 (7):1425–1438, 2016.

[8] Because the official implementation is not available, we implemented CLIPNeRF by ourselves to reproduce the experiment in Figure 14 of the paper to the best of our abilities. See appendix F for the details.

[9] While the DFF may outperform the teacher a little because the DFF denoises and improves the feature via multi-view fusion [112, 93], such an improvement is limited.

[3] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2021.

[4] Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoki Kawanabe. Scanqa: 3d question answering for spatial scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[5] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[6] Sagie Benaim, Frederik Warburg, Peter Ebert Christensen, and Serge Belongie. Volumetric disentanglement for 3d scene manipulation. *arXiv*, 2022.

[7] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. On the opportunities and risks of foundation models. *arXiv*, 2021. URL https://arxiv.org/abs/2108.07258.

[8] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P. A. Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. URL http://arxiv.org/abs/2012.03918v4.

[9] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik Lensch. Neural-PIL: Neural pre-integrated lighting for reflectance decomposition. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=fATZNtA1-V0.

[10] Maxime Bucher, Tuan-Hung VU, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/file/0266e33d3f546cb5436a10798e657d97-Paper.pdf.

[11] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

[12] 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 (ICCV)*, pages 9650–9660, October 2021.

[13] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. URL http://arxiv.org/abs/2003.10983v3.

[14] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. URL http://arxiv.org/abs/2103.15595v2.

[15] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[16] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. *16th European Conference on Computer Vision (ECCV)*, 2020.

[17] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9640–9649, 2021.

[18] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[19] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[20] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Nasa: Neural articulated shape approximation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. URL http://arxiv.org/abs/1912.03207v4.

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[23] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022.

[24] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc' Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/7cce53cf90577442771720a370c3c723-Paper.pdf.

[25] 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*, 2022. URL https://arxiv.org/abs/2203.15224.

[26] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. *arXiv preprint arXiv:2105.06468*, 2021. URL http://arxiv.org/abs/2105.06468v1.

[27] Jonathan Granskog, Till N Schnabel, Fabrice Rousselle, and Jan Novák. Neural scene graph rendering. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021.

[28] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503*, 2020. URL http://arxiv.org/abs/2012.08503v1.

[29] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4338–4364, 2021. doi: 10.1109/TPAMI.2020.3005434.

[30] Huy Ha and Shuran Song. Semantic abstraction: Open-world 3D scene understanding from 2D vision-language models. In *Proceedings of the 2022 Conference on Robot Learning*, 2022.

[31] Zekun Hao, Arun Mallya, Serge Belongie, and Ming-Yu Liu. Gancraft: Unsupervised 3d neural rendering of minecraft worlds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. URL http://arxiv.org/abs/2104.07659v1.

[32] Jun Hatori, Yuta Kikuchi, Sosuke Kobayashi, Kuniyuki Takahashi, Yuta Tsuboi, Yuya Unno, Wilson Ko, and Jethro Tan. Interactively picking real-world objects with unconstrained spoken language instructions. In *Proceedings of International Conference on Robotics and Automation*, 2018.

[33] Yong He, Hongshan Yu, Xiaoyan Liu, Zhengeng Yang, Wei Sun, Yaonan Wang, Qiang Fu, Yanmei Zou, and Ajmal Mian. Deep learning based 3d segmentation: A survey. *arxiv*, 2021. URL https://arxiv.org/abs/2103.05423.

[34] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL http://arxiv.org/abs/1503.02531.

[35] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5885–5894, October 2021.

[36] Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2112.01455.

[37] Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. URL http://arxiv.org/abs/2109.01750v1.

[38] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)*, 2021. URL http://arxiv.org/abs/2109.11418v1.

[39] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arxiv*, 2020. URL https://arxiv.org/abs/2006.12057.

[40] Amit Kohli, Vincent Sitzmann, and Gordon Wetzstein. Semantic implicit neural scene representations with semi-supervised training. In *International Conference on 3D Vision (3DV)*. IEEE, 2020. URL http://arxiv.org/abs/2003.12673v2.

[41] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic Neural Fields: A Semantic Object-Aware Neural Scene Representation. In *CVPR*, 2022.

[42] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: A composite dataset for multi-domain semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2879–2888, 2020.

[43] Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, and Gerard Pons-Moll. Control-nerf: Editable feature volumes for scene rendering and manipulation. *arXiv preprint arXiv:2204.10850*, 2022.

[44] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RriDjddCLN.

[45] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, and Zhaoyang Lv. Neural 3d video synthesis. *arXiv preprint arXiv:2103.02597*, 2021. URL http://arxiv.org/abs/2103.02597v1.

[46] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[47] Haolin Liu, Anran Lin, Xiaoguang Han, Lei Yang, Yizhou Yu, and Shuguang Cui. Refer-it-in-rgbd: A bottom-up approach for 3d visual grounding in rgbd images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6032–6041, 2021.

[48] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. URL http://arxiv.org/abs/2007.11571v2.

[49] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. *arXiv preprint arXiv:2105.06466*, 2021. URL http://arxiv.org/abs/2105.06466v2.

[50] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer V2: scaling up capacity and resolution. *arxiv*, abs/2111.09883, 2021. URL https://arxiv.org/abs/2111.09883.

[51] Sebastian Loeschcke, Serge Belongie, and Sagie Benaim. Text-driven stylization of video objects. *arXiv*, 2022. URL https://arxiv.org/abs/2206.12396.

[52] Timo Lüddecke and Alexander Ecker. Prompt-based multi-modal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2112.10003.

[53] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. URL http://arxiv.org/abs/1812.03828v2.

[54] Björn Michele, Alexandre Boulch, Gilles Puy, Maxime Bucher, and Renaud Marlet. Generative zero-shot learning for semantic segmentation of 3d point clouds. In *2021 International Conference on 3D Vision (3DV)*, pages 992–1002, 2021. doi: 10.1109/3DV53792.2021.00107.

[55] 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.

[56] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. URL http://arxiv.org/abs/2003.08934v2.

[57] 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, July 2022. doi: 10.1145/3528223.3530127. URL https://doi.org/10.1145/3528223.3530127.

[58] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017.

[59] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proc. ICCV*, pages 7588–7597, 2019.

[60] Thu H Nguyen-Phuoc, Christian Richardt, Long Mai, Yongliang Yang, and Niloy Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6767–6778, 2020. URL https://proceedings.neurips.cc/paper/2020/file/4b29fa4efe4fb7bc667c7b301b74d52d-Paper.pdf.

[61] 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 (CVPR)*, 2021. URL http://arxiv.org/abs/2011.12100v2.

[62] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[63] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. URL http://arxiv.org/abs/2104.03110v2.

[64] 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 (CVPR)*, pages 2856–2865, June 2021.

[65] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. URL http://arxiv.org/abs/1901.05103v1.

[66] 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. *arXiv preprint arXiv:2106.13228*, 2021. URL http://arxiv.org/abs/2106.13228v2.

[67] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[68] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. URL https://arxiv.org/abs/2209.14988.

[69] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

[70] Sameera Ramasinghe, Lachlan E. MacDonald, and Simon Lucey. On regularizing coordinate-mlps. *arxiv*, 2022.

[71] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. URL https://arxiv.org/abs/2103.13413.

[72] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[73] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. Derf: Decomposed radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. URL http://arxiv.org/abs/2011.12490v1.

[74] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural*

Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3982–3992, November 2019. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410.

[75] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, November 2020. doi: 10.18653/v1/2020.emnlp-main.365. URL https://aclanthology.org/2020.emnlp-main.365.

[76] Zhongzheng Ren, Aseem Agarwala†, Bryan Russell†, Alexander G. Schwing†, and Oliver Wang†. Neural volumetric object selection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. († alphabetic ordering).

[77] Barbara Roessle, Jonathan T. Barron, Ben Mildenhall, Pratul P. Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[78] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[79] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv*, 2022. URL https://arxiv.org/abs/2205.11487.

[80] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B. Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se(3)-equivariant object representations for manipulation. *arXiv preprint arXiv:2112.05124*, 2021.

[81] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019.

[82] Vincent Sitzmann, Michael Zollhofer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2019. URL http://arxiv.org/abs/1906.01618v2.

[83] Cameron Smith, Hong-Xing Yu, Sergey Zakharov, Fredo Durand, Joshua B. Tenenbaum, Jiajun Wu, and Vincent Sitzmann. Unsupervised discovery and composition of object light fields. *arXiv*, 2022.

[84] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.

[85] Karl Stelzner, Kristian Kersting, and Adam R. Kosiorek. Decomposing 3d scenes into objects via unsupervised volume segmentation. *arxiv*, 2021. URL http://arxiv.org/abs/2104.01148.

[86] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arxiv*, 2019.

[87] 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 (WACV)*, pages 2149–2159, January 2022. URL https://arxiv.org/abs/2109.07161.

[88] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. URL http://arxiv.org/abs/2101.10994v1.

[89] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. *arXiv*, 2022.

[90] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, Z. Xu, T. Simon, M. Nießner, E. Tretschk, L. Liu, B. Mildenhall, P. Srinivasan, R. Pandey, S. Orts-Escolano, S. Fanello, M. Guo, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, D. B Goldman, and M. Zollhöfer. Advances in neural rendering. In *ACM SIGGRAPH 2021 Courses*, SIGGRAPH '21, 2021. doi: 10.1145/3450508.3464573. URL https://arxiv.org/abs/2111.05849.

[91] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 10347–10357, 2021.

[92] Vadim Tschernezki, Diane Larlus, and Andrea Vedaldi. Neuraldiff: Segmenting 3d objects that move in egocentric videos. *arXiv preprint arXiv:2110.09936*, 2021.

[93] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3D distillation of self-supervised 2D image representations. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2022. URL https://arxiv.org/pdf/2209.03494.pdf.

[94] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. *arxiv*, 2021. URL https://arxiv.org/abs/2112.10703.

[95] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV Jawahar. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *2019 IEEE Winter Conference on Applications of Computer Vision*, pages 1743–1751. IEEE, 2019.

[96] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[97] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[98] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*. International Joint Conferences on Artificial Intelligence Organization, 2021. URL http://arxiv.org/abs/2106.10689v1.

[99] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2111.15174.

[100] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. Sep 2021. URL http://arxiv.org/abs/2109.01129v2.

[101] Xiaoshi Wu, Hadar Averbuch-Elor, Jin Sun, and Noah Snavely. Towers of babel: Combining images, language, and 3d geometry for learning multimodal vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 428–437, October 2021.

[102] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond, 2021. URL https://neuralfields.cs.brown.edu/.

[103] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for zero-shot semantic segmentation with pre-trained vision-language model. *arxiv*, 2021. URL https://arxiv.org/abs/2112.14757.

[104] 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 *International Conference on Computer Vision (ICCV)*, October 2021. URL https://arxiv.org/abs/2109.01847.

[105] Bangbang Yang, Yinda Zhang, Yijin Li, Zhaopeng Cui, Sean Fanello, Hujun Bao, and Guofeng Zhang. Neural rendering in a room: Amodal 3d understanding and free-viewpoint rendering for the closed scene composed of pre-captured objects. *ACM Trans. Graph.*, 41(4):101:1–101:10, July 2022. doi: 10.1145/3528223.3530163. URL https://doi.org/10.1145/3528223.3530163.

[106] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2020. URL http://arxiv.org/abs/2003.09852v3.

[107] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. *Proc. ICCV*, 2021.

[108] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multi-task learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[109] Hong-Xing Yu, Leonidas Guibas, and Jiajun Wu. Unsupervised discovery of object radiance fields. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=rwE8SshAlxw.

[110] Jason Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. Ners: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29835–29847. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/f95ec3de395b4bce25b39ef6138da871-Paper.pdf.

[111] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *arXiv preprint arXiv:2106.01970*, 2021. URL http://arxiv.org/abs/2106.01970v1.

[112] 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 International Conference on Computer Vision (ICCV)*, 2021. URL http://arxiv.org/abs/2103.15875v2.

[113] Shuaifeng Zhi, Edgar Sucar, Andre Mouton, Iain Haughton, Tristan Laidlow, and Andrew J. Davison. iLabel: Interactive neural scene labelling. *arXiv*, 2021.

[114] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://arxiv.org/abs/2112.09106.

[115] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The claims are empirically demonstrated in Section 5, and described in the whole paper.
   (b) Did you describe the limitations of your work? [Yes] See mainly Section 5, and 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] Mathematical formulations of the models are written in Section 3 and 4.
   (b) Did you include complete proofs of all theoretical results? [N/A] Mathematical formulations of the models are written in Section 3 and 4.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The complete code for the reproduction of all the experimental results is not publicly available. Because the code is a modification from a public code by Zhi et al. [112], reproduction is also easier than from scratch. We will make our scene dataset publicly available.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Some of them are further described in the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Error bars are not reported because it would be computationally expensive and results are expected to be stable. Note that most existing studies on NeRF have not reported the bars too.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] It is difficult to completely track and sum the total amount of computing in the experiments. Instead, we reported the setup of the main experiments.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Codebase and datasets are appropriately cited, mainly in Section 5.
   (b) Did you mention the license of the assets? [No] We refer the readers to the original source instead of mentioning them in this paper.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include generated video by models in the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] No data about people.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] No such data.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] No such data or experiment.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] No such data or experiment.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] No such data or experiment.

A Training and Model Architectures

In the experiments, during the training of 200K iterations, the loss L in Equation 4 is minimized by Adam with a linearly decaying learning rate (5e-4 to 8e-5). During training, Gaussian noise for density is also applied. The number of coarse and fine samplings is 64 and 128, respectively. The MLP of the neural radiance field consists of eight ReLU layers with 256 dimensions, followed by a linear layer for density, three layers for color, and three layers for feature, as shown in Fig. 1. Positional encoding of length 10 is used for the input coordinate and its skip connection, and that of length 4 is for viewing direction. If an independent MLP is prepared for the feature field, it consists of four layers (with a skip connection at the third layer if the positional encoding is used). The size of a training image is 320×240 for the Replica dataset and 1008×756 for the other datasets. The batchsize of training rays is 1024 for Replica and 2048 for the others. During finetuning of feature fields or radiance fields, Gaussian noise is removed, and the learning rate is set to 1e-4.

For segmentation, unless otherwise stated, we use thresholded cosine similarity to directly compute the probability of a query instead of softmax with negative queries in Eq. 5 and set $\mathbf{p} = 1$ if the similarity exceeds the threshold, and $\mathbf{p} = 0$ otherwise for hard decomposition. In Fig. 3 and 5, the results of the room scene used a label set, {whiteboard, ceiling, light, television, wall, bin, table, cabinet, cable, chair, box, floor}, for skipping tuning of thresholds.

B Editing Procedure

Editing for colorization, translation, and deletion proceed as follows:

(1) Sample points $[..., \mathbf{x}_i, ...]$ on a ray as usual in NeRF.
(2) For each point, we query the DFF. We can now calculate the probability of the coordinate being matched with a set of queries as $p \in [0, 1]$. We now define "the coordinate is selected by the query" if p is above a user-defined threshold, otherwise "not selected". We can also use the mixture of selected and not-selected results in the proportion of p without the threshold.
(3) If *not* selected, we calculate density $\sigma(\mathbf{x})$ and color $\mathbf{c}(\mathbf{x})$ at $\mathbf{x}$ via the vanilla NeRF.
(4) If selected, we may apply the following transforms:
(4-A) Deletion (Fig. 4): We set the density $\sigma(\mathbf{x})$ of the point to zero.
(4-B) Color editing: We query the NeRF for density $\sigma(\mathbf{x})$ and color $\mathbf{c}(\mathbf{x})$ by querying the NeRF. The color is then edited by a colorization function $\mathbf{b}$, i.e., it is transformed to $\mathbf{b}(\mathbf{c}(\mathbf{x}))$.
(4'-C) Translation / rescaling: Geometric transformation needs another step before performing (2). We first compute a deformed point coordinate $\mathbf{x}'$: $\mathbf{x}'$ is computed by applying the inverse of the editing transformation; that is, $\mathbf{x}' = \mathbf{g}^{-1}(\mathbf{x})$. For translation, $\mathbf{g}$ would be a simple addition with a vector. If $\mathbf{x}'$ is selected by the query, $\mathbf{x}'$ is used instead of $\mathbf{x}$ for calculating color and density. If both $\mathbf{x}$ and $\mathbf{x}'$ are selected and have non-zero density (e.g., the boundary between the deformed apple and others in Fig. 7), we mix their colors $\mathbf{c}(\mathbf{x})$ and $\mathbf{c}(\mathbf{x}')$ in the ratio of their alphas at the point for simplicity.
(5) Finally, as usual, we perform volume rendering with the series of (density, color) tuples.

C Feature Encoders

We investigate two teacher networks, LSeg and DINO, which are pre-trained and publicly available[10]. Each training image is encoded by the encoders of the networks and used as target feature maps, $\mathbf{f}_{\text{img}}(I, r)$, defined in Equation 4. Because the feature maps are of reduced sizes, we use them after resizing to the original image size via interpolation.

For LSeg, we use the official demo model, which has the ViT-L/16 image encoder and CLIP's ViT-B/32 text encoder. Inference follows the official script and uses multi-scale inference (scales = [0.75, 0.83, 0.92, 1.0, 1.08, 1.17, 1.25] for Replica, [0.75, 1.0, 1.25, 1.5, 1.75, 2.0, 2.25] for the others)[11]. The model is trained on seven different datasets [42], including ADE20K [115], BDD [108],

[10] https://github.com/isl-org/lang-seg https://github.com/facebookresearch/dino

[11] Multi-scale inference stabilizes features and alleviates artifacts due to discrete patch processing in ViT. Inference with large scale means inference with a zoomed and cropped image. Although it may increase the effective resolution of feature maps, it loses context information and may produce noisy features with wrong semantic understanding.

Cityscapes [19], COCO-Panoptic [46, 11], IDD [95], Mapillary Vistas [58], and SUN RGBD [84]. Possibly because the mixed dataset is biased, especially to traffic scenes, we experimentally confirmed that LSeg does not work well in out-of-distribution regions and queries. For example, small objects or parts are often not discriminative by the network. More careful training will improve its zero-shot ability in various domains and that of student neural feature fields.

For DINO, we use the extended implementation by Amir et al. [3][12]. It uses overlapping patches with stride 4 to enlarge the feature map's size. We use the feature at the 11th layer of the dino_vits8 model taking a 448x448-resized image as input. We average it with the features of the horizontally flipped image. While DINO cannot accept text queries, its feature captures more fine-grained information than LSeg's.

We visualize features of teacher networks, LSeg and DINO, in Fig. 9. We also visualize features of NeRF in Fig. 5. For visualizing feature spaces, we use scikit-learn's `sklearn.decomposition.PCA` [67]. We calculate 3-dimensional PCA components using the teacher's feature map of the first frame of the training view images, and visualize features by the components as RGB, normalized with min-max values with outlier removal. Although the 3-dimensional PCA cannot perfectly visualize the feature space, we show the results for reference and understanding.

As explained, although the features of both models are sufficiently view-consistent, their semantic resolution seems different. LSeg tends to show boundaries with coarse-grained categories, possibly due to the bias in the training datasets. The result will be changed if we train LSeg with other datasets focusing on fine-grained parts or regions. In contrast to LSeg, DINO produces fine-grained features where we can feel most of the original edges. This enables a wide range of decomposition based on various queries as we confirmed in the experiments. However, even with DINO, the finest-grained details in high resolution are still challenging to capture, e.g., the ribs of T-rex. The improvement of feature encoders will push the boundary of the decomposition quality of complex objects.

D Replica Dataset Experiment

We experiment distilled feature field on 3D semantic segmentation in Section 5.1 for demonstrating basic ability of segmentation of a 3D space and text queries. Because 3D zero-shot semantic segmentation is a novel task except for some studies Michele et al. [54], no standard benchmark datasets exist. Furthermore, even for 3D semantic segmentation with closed label sets, there is no dataset with high-quality images, accurate point clouds[13], and reasonable annotations. For proof-of-concept of our work, we created a dataset from the existing Replica dataset [86]. It is a moderate-quality room reconstruction dataset with semantic segmentation labels. We use four scenes, room_0, room_1, office_3, and office_4 with posed images rendered by Zhi et al. [112]. Because the pose trajectory was randomly generated, some images are unrealistically rendered (e.g., rendered from a camera inside furniture). We filtered such images from training and evaluation. Near and far of NeRF follow their setting, (0.1, 10.0), except for office_3 (0.1, 15.0). Although the quality of reconstructions is better than other datasets like ScanNet [20], it still suffers from collapsed geometry, less photo-realistic appearances, and defective annotations. Some objects are difficult to predict their labels due to bad appearance, label ambiguity, or both. Because this evaluation is not intended to measure the ability to overcome such unrealistic biases or artifacts in a dataset, we semi-automatically fix the label set of the dataset. We first apply LSeg to predict the semantic label map of each RGB image and evaluate the accuracy against Replica's original label map. In the evaluation, the accuracy of some labels is almost or exactly zero. We ignore points with such labels from the evaluation. In addition, noticeable label ambiguity is also fixed manually by ignoring the label or re-labeling (e.g., 'rug' and 'floor' are merged as they are intrinsically nonexclusive and often indistinguishable in the dataset). For reference, we show a visualization of predictions by LSeg and Replica's ground truth in Fig. 10, and the confusion matrix in Fig. 11. Note that it is not guaranteed that training images cover all the regions of the scene and its point cloud. Hence, the evaluation also measures the ability of generalization to unobserved regions via propagation.

[12] https://github.com/ShirAmir/dino-vit-features

[13] If the input point clouds exist at geometrically wrong coordinates, in reality, their annotated labels could be wrong. It is problematic, especially for evaluating point-cloud-agnostic models like our model.

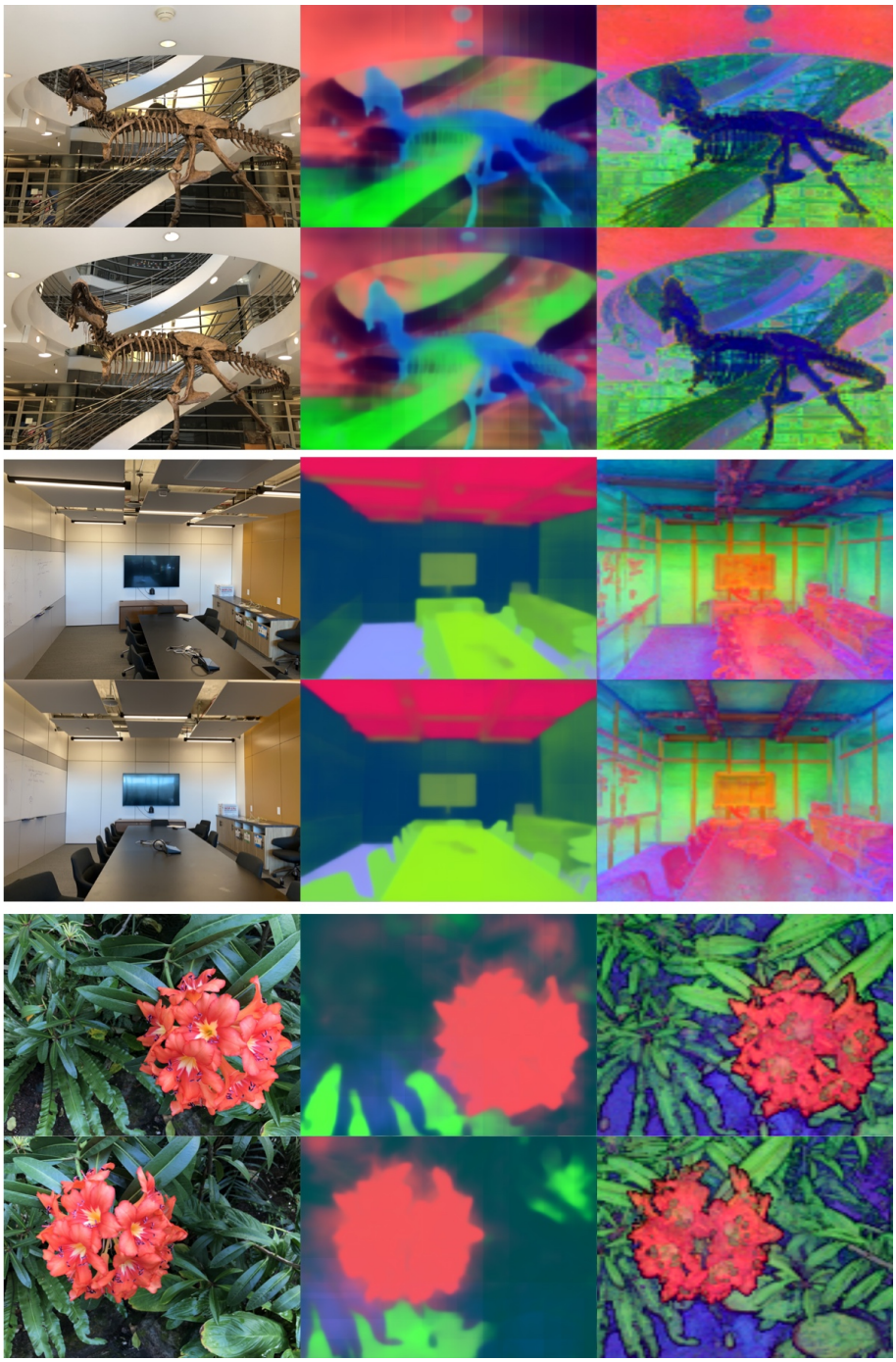

Figure 9: Visualization of features of LSeg (center) and DINO (right). Note that it is invalid to compare colors themselves between different scenes or models.

E Ablation Experiments of Variants

We experiment with some variants of DFF architectures and show the result in Table 3. While they could indicate different behavior for each scene, the average performance difference was marginal. In total, using MLP trained with volume rendering of coarse sampling performs better than MLP with fine sampling based on hierarchical sampling. The coarse sampling might implicitly regularize MLP

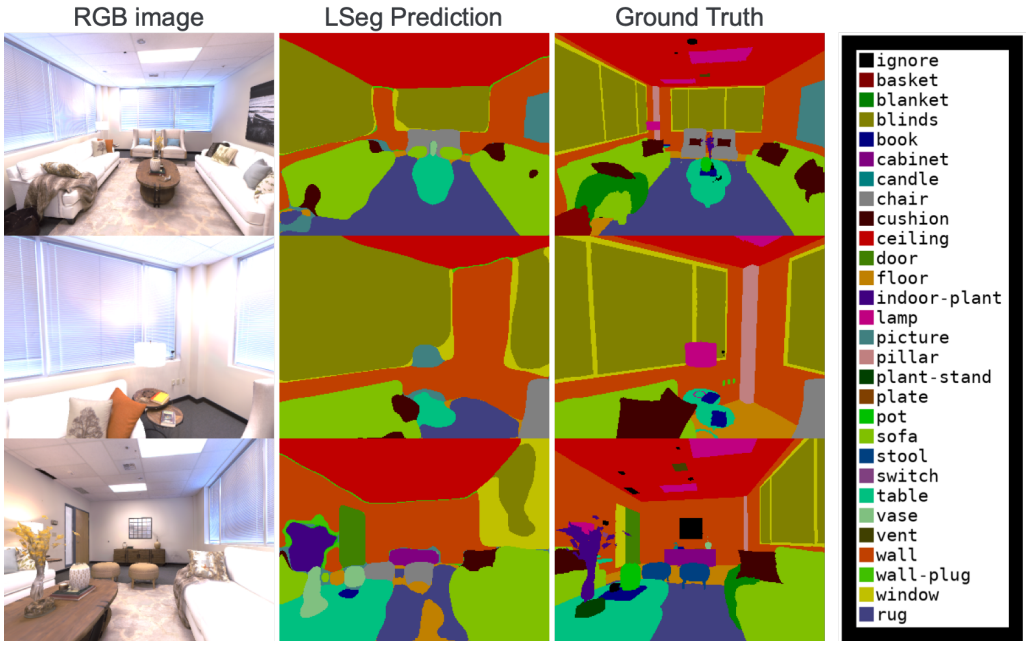

Figure 10: Visualization of prediction by the LSeg (teacher network) on the image semantic segmentation task on Replica's room_0.

Table 3: Performance of zero-shot semantic segmentation and novel view synthesis on Replica. mIoU is calculated on the 3D point cloud. @2 allows the matching of labels with the second-highest probability. 2D accuracy to teacher indicates the agreement ratio with the labels predicted by the LSeg teacher network. PSNR, SSIM, and LPIPS are metrics of image synthesis. $\delta < 1.25$ and absrel are metrics of geometry (depth estimation).

	mIoU	-@2	2D acc. to teacher	PSNR↑	SSIM↑	LPIPS↓	$\delta < 1.25$↑	absrel↓
basic NeRF	-	-	-	32.87	0.934	0.148	0.993	0.018
Coarse MLP								
indep. freq4	0.544	0.760	-	-	-	-	-	-
branch	0.562	0.779	-	-	-	-	-	-
branch ($\lambda \times 10$)	0.565	0.790	-	-	-	-	-	-
Fine MLP								
indep. freq4	0.553	0.774	0.941	32.87	0.934	0.148	0.993	0.018
branch	0.553	0.784	0.942	32.85	0.932	0.150	0.993	0.017
branch ($\lambda \times 10$)	0.543	0.770	0.942	32.68	0.927	0.162	0.993	0.018

and help to obtain smoothness properties. Even if increasing the weight of feature loss, λ, it did not improve the performance and hurt the quality of view synthesis.

F Implementation of CLIPNeRF Experiment

Because the official implementation of CLIPNeRF [97] is not available, we implement it from the description by ourselves. In addition, the experiment for Figure 14 in their paper (i.e., editing of the LLFF scene) unfortunately lacks significant descriptions to be required for reproduction. Thus, there might be nuanced differences while we confirmed that our result shows similar behaviors. We follow the description in the paper as much as possible. For 500 iterations, we optimize the parameters of the NeRF while freezing the sub-parameters related to density. It minimizes negative cosine similarity between a text prompt and a rendered image from a randomly sampled pose in the training dataset, calculated by CLIP ViT-B/32, with Adam of learning rate 0.0001. Because a naive differentiable rendering of high-resolution images is intractable due to memory constraints, we use an efficient

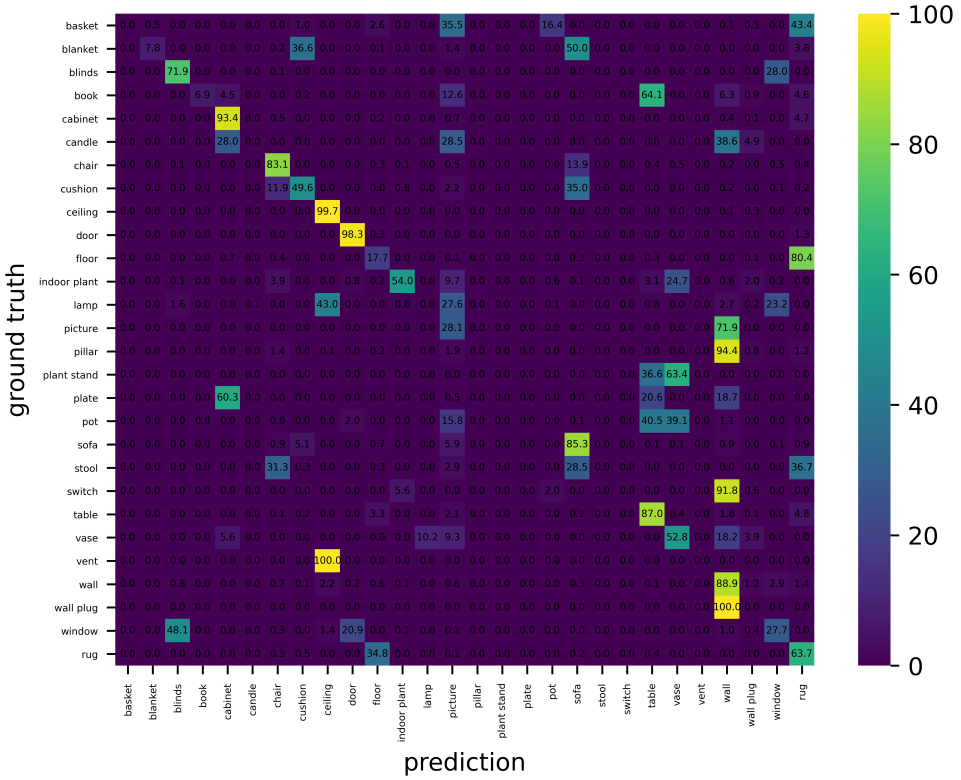

Figure 11: Confusion matrix of the LSeg on the image semantic segmentation task on Replica room_0 (full label setting).

method. We render a patch of 128×128 size, followed by bilinear interpolation for resizing and feed it to CLIP. For making the patch cover the view widely, rays in 128×128 patch are sampled with stride interval 5; thus, CLIP's receptive field is 640×640 of the original image. For memory efficiency, during the rendering of a patch, we reduce the number of importance sampling to 16; fine MLP uses $64 + 16 = 80$ in total. While it is out of the scope of this paper, it is possible to further increase the patch resolution if we use a recomputation technique, where NeRF first computes depth without the need for gradient, and then, renders a patch with a few samples nearby the depth surface per ray in a differentiable manner.

For demonstrating that distilled feature field can be used with other 3D scene representations, we additionally experimented with the InstantNGP [57] and the LSeg-DFF, followed by CLIPNeRF editing. The implementation is derived from `https://github.com/kwea123/ngp_pl`. For computational efficiency, we do not perform the complete process of volume rendering for feature rendering. We first compute the pseudo depth of the surface as a weighted average of the depth of each sampled point in volume rendering. We simply compute the feature in five points around the pseudo surface point by adding small perturbations during training and inference. This method is very efficient but works for training the distilled feature field. When optimizing the scene with CLIPNeRF, instead of backpropagating gradients to all the rays, we backpropagate the gradients to only the rays selected by "apple". The result of the "rainbow apple"-edit is shown in Fig. 13.

G CLIP-inspired Segmentation Models

We use 2D vision(-and-language) models as teacher networks for distillation. In particular, for processing text queries, we use LSeg [44] as a teacher network. We can use other models for teacher

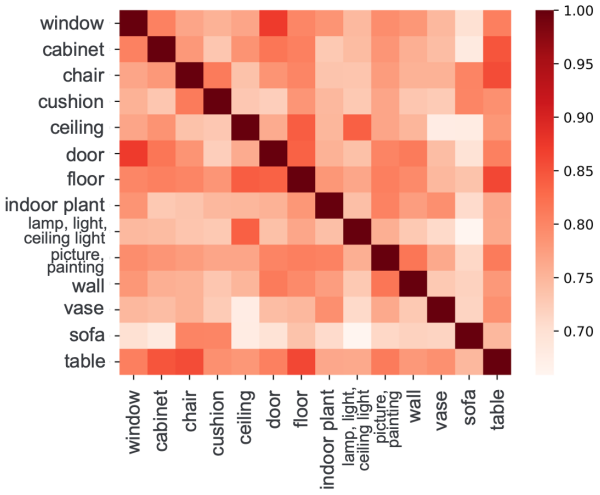

Figure 12: Similarity matrix of text label features by LSeg of Replica room_0.

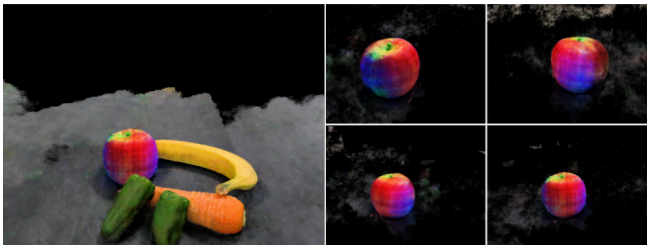

Figure 13: Editing of an InstantNGP scene with CLIPNeRF and extraction.

networks. As introduced in Section 2, within a few months[14], many studies [44, 52, 99, 103, 72] concurrently tackled image semantic segmentation tasks with similar approaches using CLIP [69]. They use the pretrained CLIP models as a text encoder and an image feature encoder in freezing or finetuning manners and train zero-shot image semantic segmentation models. Because they were concurrent, differences in various choices and performances are missing so far. While we use LSeg by Li et al. [44] for simplicity of implementation, it is possible to use other models, especially text-unconditional encoders, where the image encoder produces a feature map without using text labels and calculates a score map by pixel-level calculations with text labels. Although the publicly available LSeg model can accept any text queries thanks to the CLIP's text encoder, it is not so robust to out-of-distribution data (e.g., "fossils", "T-Rex", "Triceratops") due to their training strategy overfitting to the bias in its training datasets (e.g., traffic scenes). Using other concurrent models or the ongoing improvement of training and architectures [50] will alleviate such issues and improve distilled feature field.

[14] Li et al. [44] appeared anonymously at OpenReview on September 29th, 2021. The others appeared at arXiv on November or December, 2021.