NeRF-SOS: Any-View Self-supervised Object Segmentation on Complex Scenes

Anonymous authors
Paper under double-blind review

Abstract

Neural volumetric representations have shown the potential that Multi-layer Perceptrons (MLPs) can be optimized with multi-view calibrated images to represent scene geometry and appearance without explicit 3D supervision. Object segmentation can enrich many downstream applications based on the learned radiance field. However, introducing hand-crafted segmentation to define regions of interest in a complex real-world scene is non-trivial and expensive as it acquires per view annotation. This paper carries out the exploration of self-supervised learning for object segmentation using NeRF for complex real-world scenes. Our framework, called *NeRF with Self-supervised Object Segmentation* (**NeRF-SOS**), couples object segmentation and neural radiance field to segment objects in any view within a scene. By proposing a novel collaborative contrastive loss in both appearance and geometry levels, NeRF-SOS encourages NeRF models to distill compact geometry-aware segmentation clusters from their density fields and the self-supervised pre-trained 2D visual features. The self-supervised object segmentation framework can be applied to various NeRF models that both lead to photo-realistic rendering results and convincing segmentation maps for both indoor and outdoor scenarios. Extensive results on the *LLFF*, *BlendedMVS*, *CO3Dv2*, and *Tank & Temples* datasets validate the effectiveness of NeRF-SOS. It consistently surpasses other 2D-based self-supervised baselines and predicts finer object masks than existing supervised counterparts.

1 Introduction

Scene modeling and representation are important and fundamental to the computer vision community. For example, portable Augmented Reality (AR) devices (e.g., the Magic Leap One) reconstruct the scene geometry and further localize users (DeChicchis, 2020). However, despite the geometry, it hardly understands the surrounding objects in the scene, and thus meets difficulty when enabling interaction between humans and environments. The hurdles of understanding and segmenting the surrounding objects can be mitigated by collecting human-annotated data from diverse environments, but that could be difficult in practice due to the costly labeling procedure. Therefore, there has been growing interest in building an intelligent geometry modeling framework without heavy and expensive annotations.

Recently, neural volumetric rendering techniques have shown great power in scene reconstruction. Especially, neural radiance field (NeRF) and its variants (Mildenhall et al., 2020a; Zhang et al., 2020; Barron et al., 2021) adopt multi-layer perceptrons (MLPs) to learn continuous representation and utilize calibrated multi-view images to render unseen views with fine-grained details. Besides rendering quality, the ability of scene understanding has been explored by several recent works (Vora et al., 2021; Yang et al., 2021; Zhi et al., 2021). Nevertheless, they either require dense view annotations to train a heavy 3D backbone for capturing semantic representations (Vora et al., 2021; Yang et al., 2021), or necessitate human intervention to provide sparse semantic labels (Zhi et al., 2021). Recent self-supervised object discovery approaches on neural radiance fields (Yu et al., 2021c; Stelzner et al., 2021) try to decompose objects from givens scenes on the synthetic indoor data. However, still remains a gap to be applied in complex real-world scenarios.

In contrast to previous works, we take one leap further to investigate a more general setting, to segment 3D objects in real-world scenes using general NeRF models. Driven by this motivation, we design a

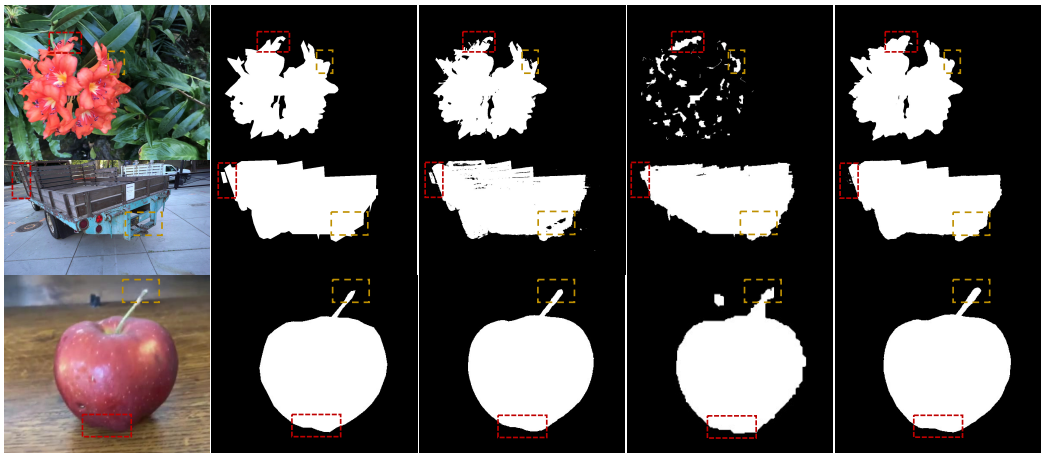


Figure 1: **Visual examples**. From left to right: ground truth color images, annotated object masks, object masks rendered by NeRF-SOS, 2D image co-segmentation using DINO (Amir et al., 2021), and object masks rendered by Semantic-NeRF (Zhi et al., 2021), respectively. Compared to the previous methods, NeRF-SOS generates faithful object masks with finer local details.

new self-supervised object segmentation framework for NeRF using a collaborative contrastive loss: adopting features from a self-supervised pre-trained 2D backbone ("**appearance level**") and distilling knowledge from the geometry cues of a scene using the density field of NeRF representations ("**geometry level**"). To be more concrete, we learn from a pre-trained 2D feature extractor in a self-supervised manner (e.g., DINO-ViT (Caron et al., 2021)), and inject the visual correlations across views to form distinct segmentation feature clusters under NeRF formulation. We seek a geometry-level contrastive loss by formulating a geometric correlation volume between NeRF's density field and the segmentation clusters to make the learned feature clusters aware of scene geometry. The proposed self-supervised object segmentation framework tailored for NeRF, dubbed **NeRF-SOS**, acts as a general implicit framework and can be applied to any existing NeRF models with end-to-end training. We implement and evaluate NeRF-SOS, using vanilla NeRF (Mildenhall et al., 2020a) for real-world forward-facing datasets (LLFF (Mildenhall et al., 2019)), object-centric datasets (BlendedMVS (Yao et al., 2020) and CO3Dv2 (Reizenstein et al., 2021)); and using NeRF++ (Zhang et al., 2020) for outdoor unbounded dataset (Tank and Temples (Riegler & Koltun, 2020)). Experiments show that NeRF-SOS significantly outperforms state-of-the-art 2D object discovery methods and produces view-consistent segmentation clusters: a few examples are shown in Figure 1.

We summarize the main contributions as follows:

- We explore how to effectively apply the self-supervised learned 2D visual feature for 3D representations via an appearance contrastive loss, which forms compact feature clusters for any-view object segmentation in complex real-world scenes.
- We propose a new geometry contrastive loss for object segmentation. By leveraging its density field, our proposed framework can further inject scene geometry into the segmentation field, making the learned segmentation clusters geometry-aware.
- The proposed collaborative contrastive framework can be implemented upon NeRF and NeRF++, for object-centric, indoor and unbounded real-world scenarios. Experiments show that our self-supervised object segmentation quality consistently surpasses 2D object discovery methods and yields finer-grained segmentation results than supervised NeRF counterpart (Zhi et al., 2021).

2 Related Work

Neural Radiance Fields Neural Radiance Fields (NeRF) is first proposed by Mildenhall *et al.* (Mildenhall et al., 2020b), which models the underlying 3D scenes as continuous volumetric fields of color and density via layers of MLP. The input of a NeRF is a 5D vector, containing a 3D location (x, y, z) and a 2D viewing direction (θ, ϕ). Several following works emerge trying to address its limitations and improve the performance, such as unbounded scenes training (Zhang et al., 2020; Barron et al., 2021), fast training (Sun et al., 2021; Deng et al., 2021), efficient inference (Rebain

et al., 2020; Liu et al., 2020; Lindell et al., 2020; Garbin et al., 2021; Reiser et al., 2021; Yu et al., 2021a; Lombardi et al., 2021), better generalization (Schwarz et al., 2020a; Trevithick & Yang, 2020; Wang et al., 2021b; Chan et al., 2020; Yu et al., 2021b; Johari et al., 2021), supporting unconstrained scene (Martin-Brualla et al., 2020; Chen et al., 2021), editing (Liu et al., 2021; Jiakai et al., 2021; Wang et al., 2021a; Jang & Agapito, 2021), multi-task learning (Zhi et al., 2021). In this paper, we treat NeRF as a powerful implicit scene representation and study how to segment objects from a complex real-world scene without any supervision.

Object Co-segmentation without Explicit Learning Our work aims to discover and segment visually similar objects in the radiance field and render novel views with object masks. It is close to the object co-segmentation (Rother et al., 2006) which aims to segment the common objects from a set of images (Li et al., 2018). Object co-segmentation has been widely adopted in computer vision and computer graphics applications, including browsing in photo collections (Rother et al., 2006), 3D reconstruction (Kowdle et al., 2010), semantic segmentation (Shen et al., 2017), interactive image segmentation (Rother et al., 2006), object-based image retrieval (Vicente et al., 2011), and video object tracking/segmentation (Rother et al., 2006). (Rother et al., 2006) first shows that segmenting two images outperforms the independent counterpart. This idea is analogous to the contrastive learning way in later approaches. Especially, the authors in (Hénaff et al., 2022) propose the self-supervised segmentation framework using object discovery networks. (Siméoni et al., 2021) localizes the objects with a self-supervised transformer. (Hamilton et al., 2022) introduces the feature correspondences that distinguish between different classes. Most recently, a new co-segmentation framework based on DINO feature (Amir et al., 2021) has been proposed and achieves better results on object co-segmentation and part co-segmentation.

However, extending 2D object discovery to NeRF is non-trivial as they cannot learn the geometric cues in multi-view images. uORF (Yu et al., 2021c) and ObSuRF (Stelzner et al., 2021) use slot-based CNN encoders and object-centric latent codes for unsupervised 3D scene decomposition. Although they enable unsupervised 3D scene segmentation and novel view synthesis, experiments are on synthetic datasets, leaving a gap for complex real-world applications. Besides, a Gated Recurrent Unit (GRU) and multiple NeRF models are used, making the framework difficult to be applied to other NeRF variants. N3F (Tschernezki et al., 2022) minimizes the distance between NeRF's rendered feature and 2D feature for scene editing. Most recently, (Kobayashi et al., 2022) proposes to distill the visual feature from supervised CLIP-LSeg or self-supervised DINO into a 3D feature field via an element-wise feature distance loss function. It can discover the object using a query text prompt or a patch. In contrast, we design a new collaborative contrastive loss on both appearance and geometry levels to find the objects with a similar appearance and location without any annotations. The collaborative design is general and can be plug-and-play to different NeRF models.

3 METHOD

Overview We show how to extend existing NeRF models to segment objects in both training and inference. As seen in Figure 2, we augment NeRF models by appending a parallel segmentation branch to predict point-wise implicit segmentation features. Specifically, NeRF-SOS can render the depth σ, segmentation s, and color $\boldsymbol{c}$. Next, we feed the rendered color patch $\boldsymbol{c}$ into a self-supervised pre-trained framework (e.g., DINO-ViT (Caron et al., 2021)) to generate feature tensor f, constructing appearance-segmentation correlation volume between f and s. Similarly, a geometry-segmentation correlation volume is instantiated using σ and s. By formulating positive/negative pairs from different views, we can distill the correlation pattern in both visual feature and scene geometry into the compact segmentation field s. During inference, a clustering operation (e.g., K-means) is used to generate object masks based on the rendered feature field.

3.1 PRELIMINARIES

Neural Radiance Fields NeRF (Mildenhall et al., 2020a) represents 3D scenes as radiance fields via several layer MLPs, where each point has a value of color and density. Such a radiance field can be formulated as $F : (\boldsymbol{x}, \boldsymbol{\theta}) \mapsto (\boldsymbol{c}, \sigma)$, where $\boldsymbol{x} \in \mathbb{R}^3$ is the spatial coordinate, $\boldsymbol{\theta} \in [-\pi, \pi]^2$ denotes the viewing direction, and $\boldsymbol{c} \in \mathbb{R}^3, \sigma \in \mathbb{R}_+$ represent the RGB color and density, respectively. To form an image, NeRF traces a ray $\boldsymbol{r} = (\boldsymbol{o}, \boldsymbol{d}, \boldsymbol{\theta})$ for each pixel on the image plane, where $\boldsymbol{o} \in \mathbb{R}^3$ denotes the position of the camera, $\boldsymbol{d} \in \mathbb{R}^3$ is the direction of the ray, and $\boldsymbol{\theta} \in [-\pi, \pi]^2$ is the angular viewing direction. Afterwards, NeRF evenly samples K points $\{t_i\}_{i=1}^K$ between the near-far bound $[t_n, t_f]$ along the ray. Then, NeRF adopts volumetric rendering and numerically evaluates the ray

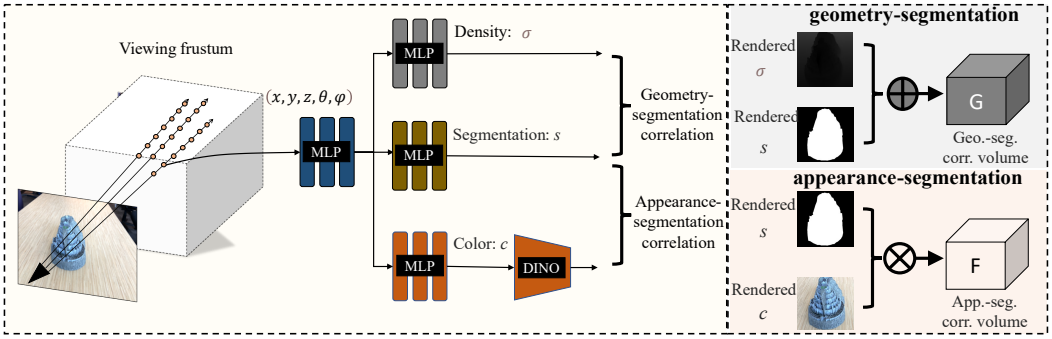


Figure 2: **The overall pipeline of the proposed NeRF-SOS.** Input with rays cast from multiple views, we render the corresponding color patch ($\boldsymbol{c}$), segmentation patch (s), and depth patch (σ). Then, appearance-segmentation correlations and geometry-segmentation correlations are used to formulate a collaborative contrastive loss, enabling NeRF-SOS to render object masks from any viewpoint using the distilled segmentation field.

integration (Max, 1995) by the quadrature rule:

$$\boldsymbol{C}(\boldsymbol{r}) = \sum_{k=1}^{K} T(k)(1 - \exp(-\sigma_k \delta t_k))\boldsymbol{c}_k \quad \text{where } T(k) = \exp\left(-\sum_{l=1}^{k-1} \sigma_l \delta_l\right), \tag{1}$$

where $\delta_k = t_{k+1} - t_k$ are intervals between sampled points, and $(\boldsymbol{c}_k, \sigma_k) = F(\boldsymbol{o} + t_k \boldsymbol{d}, \boldsymbol{\theta})$ are output from the neural network. With this forward model, NeRF optimizes the photometric loss between rendered ray colors and ground-truth pixel colors defined as follows:

$$\mathcal{L}_{photometric} = \sum_{(\boldsymbol{r}, \widehat{\boldsymbol{C}}) \in \mathcal{R}} \left\| \boldsymbol{C}(\boldsymbol{r}) - \widehat{\boldsymbol{C}} \right\|_2^2, \tag{2}$$

where $\mathcal{R}$ defines a dataset collecting all pairs of ray and ground-truth colors from captured images.

3.2 Cross View Appearance Correspondence

Semantic Correspondence across Views Tremendous works have explored and demonstrated the importance of object appearance when generating compact feature correspondence across views (Hénaff et al., 2022; Li et al., 2018). This peculiarity is then utilized in self-supervised 2D semantic segmentation frameworks (Hénaff et al., 2022; Li et al., 2018; Chen et al., 2020) to generate semantic representations by selecting positive and negative pairs with either random or KNN-based rules (Hamilton et al., 2022). Drawing inspiration from these prior arts, we construct the visual feature correspondence for NeRF at the appearance using a heuristic rule. To be more specific, we leverage the self-supervised model (e.g., DINO-ViT (Caron et al., 2021)) learned from 2D image sets to distill the rich representations into compact and distinct segmentation clusters. A four-layer MLP is appended to segment objects in the radiance field parallel to the density and appearance branches. During training, we first render multiple image patches using Equation 1, then we feed them into DINO-ViT to generate feature tensors of shape $H' \times W' \times C'$. They are then used to generate the appearance correspondence volume (Teed & Deng, 2020; Hamilton et al., 2022) across views, measuring the similarity between two regions of different views:

$$F_{hwh'w'} = \sum_{c} \frac{f_{chw}}{|f_{hw}|} \frac{f'_{ch'w'}}{|f'_{h'w'}|}, \tag{3}$$

where f and f' stand for the extracted DINO feature from two random patches in different views, (h, w) and (h', w') denote the spatial location on feature tensor for f and f', respectively, and the c traverses through the feature channel dimension.

Distilling Semantic Correspondence into Segmentation Field The correspondence volume F from DINO has been verified it has the potential to effectively recall true label co-occurrence from image collections (Hamilton et al., 2022). We next explore how to learn a segmentation field s by leveraging F. Inspired by CRF and STEGO (Hamilton et al., 2022) where they refine the initial

predictions using color or feature-correlated regions in the 2D image. We propose to append an extra segmentation branch to predict the segmentation field, formulating segmentation correspondence volume by leveraging its predicted segmentation logits using the same rule with Equation 3. Then, we construct the appearance-segmentation correlation aims to enforce the elements of s and s' closer if f and f' are tightly coupled, where the expression with and without the superscript indicates two different views. The volume correlation can be achieved via an element-wise multiplication between S and F, and thereby, we have the appearance contrastive loss $\mathcal{L}_{app}$:

$$\mathcal{C}_{app}(\boldsymbol{r}, b) = -\sum_{hwh'w'} (F_{hwh'w'} - b) S_{hwh'w'} \tag{4}$$

$$\mathcal{L}_{app} = \lambda_{id}\mathcal{C}_{app}(\boldsymbol{r}_{id}, b_{id}) + \lambda_{neg}\mathcal{C}_{app}(\boldsymbol{r}_{neg}, b_{neg}) \tag{5}$$

where $S_{hwh'w'} = \sum_c \frac{s_{chw}}{|s_{hw}|} \frac{s'_{ch'w'}}{|s'_{h'w'}|}$ indicates the segmentation correspondence volume between two views, $\boldsymbol{r}$ is the cast ray fed into NeRF, b is a hyper-parameter to control the positive and negative pressure. λ_{id} and λ_{neg} indicate loss force between identity pairs (positive) and distinct pairs (negative). The intuition behind the above equation is that minimizing $\mathcal{L}_{app}$ with respect to S, to enforce entries in segmentation field s to be large when $F - b$ are positive items and pushes entries to be small if $F - b$ are negative items.

Discover Patch Relationships To construct Equation 5, we build a cosine similarity matrix to effectively discover the positive/negative pairs of given patches. For each matrix, we take N randomly selected patches as inputs and adopt a pre-trained DINO-ViT to extract meaningful representations. We use the [CLS] token from ViT architecture to represent the semantic features of each patch and obtain N positive pairs by the diagonal entries and N negative pairs by querying the lowest score in each row. An example using three patches from different views is shown in Figure 3. Similar to (Tumanyan et al., 2022), we observe that the [CLS] token from a self-supervised pre-trained ViT backbone can capture high-level semantic appearances and can effectively discover similarities between patches during the proposed end-to-end optimization process.

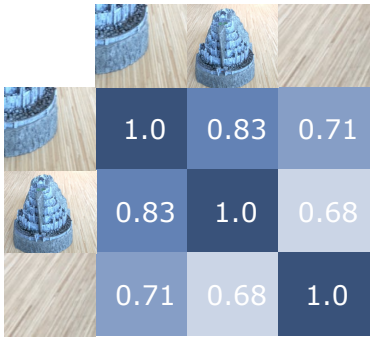


Figure 3: Cosine similarity matrix calculated on scene *Fortress*.

3.3 Cross View Geometry Correspondence

Geometry Correspondence across Views With the appearance level distillation from the DINO feature to the low-dimensional segmentation embedding in the segmentation field s, we can successfully distinguish the salient object with a similar appearance. However, s may mistakenly clusters different objects together, as s may be obfuscated by objects with distinct spatial locations but similar appearances. To solve this problem, we propose to leverage the density field that already exists in NeRF models to formulate a new geometry contrastive loss. Specifically, given a batch of M cast ray $\boldsymbol{r}$ as NeRF's input, we can obtain the density field of size $M \times K$ where K indicates the number of sampled points along each ray. By accumulating the discrete bins along each ray, we can roughly represent the density field as a single 3D point:

$$\boldsymbol{p} = \boldsymbol{r}_o + \boldsymbol{r}_d \cdot D \tag{6}$$

$$D(\boldsymbol{r}) = \sum_{k=1}^{K} T(k)(1 - \exp(-\sigma_k \Delta t_k))t_k \tag{7}$$

where $\boldsymbol{p}$ is the accumulated 3D point along the ray, D is the estimated depth value of the corresponding pixel index. Inspired by Point Transformer (Zhao et al., 2021) which uses point-wise distance as representation, we utilize the estimated point position as a geometry cue to formulate a new geometry level correspondence volume across views by measuring point-wise absolute distance:

$$G_{hwh'w'} = \sum_c \frac{1}{|g_{chw} - g'_{ch'w'}| + \epsilon} \tag{8}$$

where g and g' are the estimated 3D point positions in two random patches of different views, (h, w) and (h', w') denote the spatial location on feature tensor for g and g', respectively.

Table 1: Quantitative evaluation of the novel view synthesis and object segmentation of LLFF dataset on scene *Flower* and *Fortress*, with several 2D object discovery frameworks and the supervised Semantic-NeRF.

Scene "*Flower*"	PSNR ↑	SSIM ↑	LPIPS ↓	NV-ARI ↑	IoU(BG) ↑	IoU(FG) ↑	mIoU ↑
IEM (Savarese et al., 2021)	-	-	-	0.2666	0.7123	0.4267	0.5695
DOCS (Li et al., 2018)	-	-	-	0.0097	0.4824	0.2461	0.3643
DINO+CoSeg (Amir et al., 2021)	-	-	-	0.5946	0.9036	0.5961	0.7498
NeRF-SOS (Ours)	25.96	0.7717	0.1502	0.9529	0.9869	0.9503	0.9686
Semantic-NeRF (Zhi et al., 2021) (Supervised)	25.52	0.7500	0.1739	0.9104	0.9743	0.9090	0.9417
Scene "*Fortress*"	PSNR ↑	SSIM ↑	LPIPS ↓	NV-ARI ↑	IoU(BG) ↑	IoU(FG) ↑	mIoU ↑
IEM (Savarese et al., 2021)	-	-	-	0.3700	0.7799	0.4526	0.6163
DOCS (Li et al., 2018)	-	-	-	0.7412	0.9329	0.7265	0.8297
DINO+CoSeg (Amir et al., 2021)	-	-	-	0.9503	0.9886	0.9395	0.9640
NeRF-SOS (Ours)	29.78	0.8517	0.1079	0.9802	0.9955	0.9751	0.9853
Semantic-NeRF (Zhi et al., 2021) (Supervised)	29.78	0.8578	0.0906	0.9838	0.9963	0.9799	0.9881

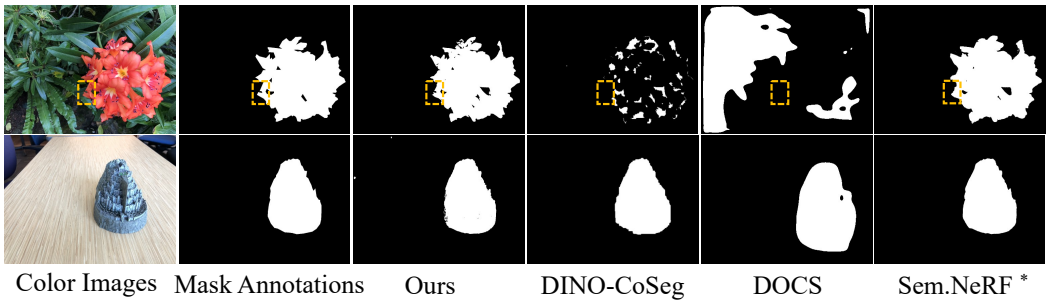


Figure 4: Qualitative results on scene *Flower* and *Fortress* of LLFF dataset. In the fourth column, DINO-CoSeg mistakenly matches several discrete patches, as DINO has higher activation on just a few tokens, which may lead to view-inconsistent and disconnected co-segmentation results. $*$ superscript denotes the supervised method. DOCS and DINO-CoSeg are not able to perform novel view synthesis, and thus we perform rendering before segmentation using a vanilla NeRF. Videos can be viewed in the supplementary materials.

Injecting Geometry Coherence into Segmentation Field To inject the geometry cue from the density field to the segmentation field, we formulate segmentation correspondence volume S and geometric correspondence volume G using the same rule of Equation 4. By pulling/pushing positive/negative pairs for the geometry-segmentation correlation of Equation 9, we come up with a new geometry-aware contrastive loss $\mathcal{L}_{geo}$:

$$\mathcal{C}_{geo}(\boldsymbol{r}, b) = -\sum_{hwh'w'} (G_{hwh'w'} - b) S_{hwh'w'} \tag{9}$$

$$\mathcal{L}_{geo} = \lambda_{id}\mathcal{C}_{geo}(\boldsymbol{r}_{id}, b_{id}) + \lambda_{neg}\mathcal{C}_{geo}(\boldsymbol{r}_{neg}, b_{neg}) \tag{10}$$

Same as appearance contrastive loss, we find positive pairs and negative pairs via the pair-wise cosine similarity of the [CLS] tokens.

3.4 Optimizing with Stride Ray Sampling

We adopt patch-wise ray casting during the training process, while we also leverage a *Stride Ray Sampling* strategy, similar to prior works (Schwarz et al., 2020b; Meng et al., 2021) to handle GPU memory bottleneck. Overall, we optimize the pipeline using a balanced loss function:

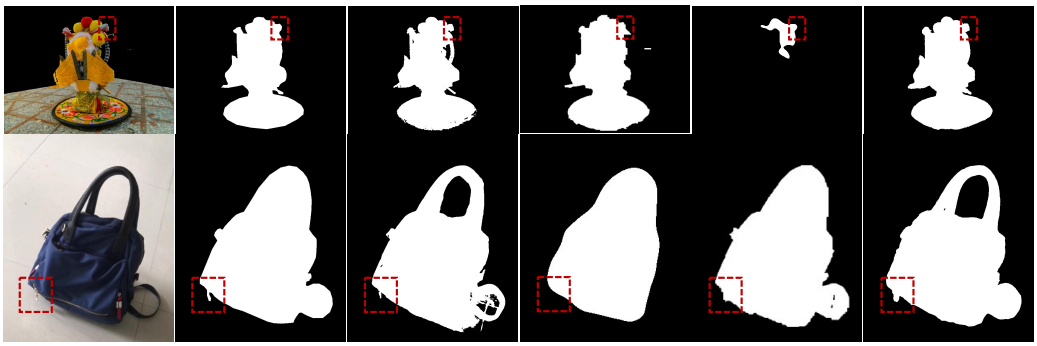
$$\mathcal{L} = \lambda_0\mathcal{L}_{photometric} + \lambda_1\mathcal{L}_{app} + \lambda_2\mathcal{L}_{geo}, \tag{11}$$

where λ_0, λ_1, and λ_2 are balancing weights.

4 Experiments

4.1 Experiment Setup

Datasets We evaluate all methods on four representative datasets: Local Light Field Fusion (LLFF) dataset (Mildenhall et al., 2019), BlendedMVS (Yao et al., 2020), CO3Dv2 (Reizenstein et al., 2021),



Color Images Mask Annotations Ours DINO-CoSeg DOCS Sem.NeRF *

Figure 5: Novel view object segmentation results on object-centric datasets: BlendedMVS (the 1st row) and CO3Dv2 (the 2nd row). NeRF-SOS (the 3rd column) still produces view-consistent masks with finer details. Videos can be viewed in the supplementary materials.

and Tank and Temples (T&T) dataset (Riegler & Koltun, 2020). Particularly, we use the forward-facing scenes {*Flower*, *Fortress*} from LLFF dataset, two object-centric scenes from BlendedMVS dataset, two common objects {*Backpack*, *Apple*} captured by video sequences from CO3Dv2 dataset, and unbounded scene *Truck* from hand-held 360° capture large-scale Tank and Temples dataset. We choose these representative scenes because they contain at least one common object among most views. We manually labeled all views as a binary mask to provide a fair comparison for all methods and used them to train Semantic-NeRF. Foreground objects appearing in most views are labeled as 1, while others are labeled as 0. We train and test all methods with the original image resolutions.

Training Details We first implement the collaborative contrastive loss upon the original NeRF (Mildenhall et al., 2020a). In training, we first train NeRF-SOS without segmentation branch following the NeRF training recipe (Mildenhall et al., 2020b) for 150k iterations. Next, we load the weight and start to train the segmentation branch alone using the stride ray sampling for another 50k iterations. The loss weights λ_0, λ_1, λ_2, λ_{id}, and λ_{neg} are set 0, 1, 0.01, 1 and 1 in training the segmentation branch. The segmentation branch is formulated as a four-layer MLP with ReLU as the activation function. The dimensions of hidden layers and the number of output layers are set as 256 and 2, respectively. The segmentation results are based on K-means clustering on the segmentation logits. We train Semantic-NeRF (Zhi et al., 2021) for 200k in total for fair comparisons. We randomly sample eight patches from different viewpoints (a.k.a batch size N is 8) in training. The patch size of each sample is set as 64×64, with the patch stride as 6. We use the official pre-trained DINO-ViT in a self-supervised manner on ImageNet dataset as our 2D feature extractor. The pre-trained DINO backbone is kept frozen for all layers during training. All hyperparameters are carefully tuned by a grid search, and the best configuration is applied to all experiments. All models are trained on an NVIDIA RTX A6000 GPU with 48 GB memory. We reconstruct N positives and N negatives pairs on the fly during training, given N rendered patches. More details can be found in the appendix.

Metrics We adopt the Adjusted Rand Index in novel views as a metric to evaluate the clustering quality, noted as NV-ARI. We also adopt mean Intersection-over-Union to measure segmentation quality for both object and background, as we set the clusters with larger activation as foreground by DINO. To evaluate the rendering quality, we follow NeRF (Mildenhall et al., 2020a), adopting peak signal-to-noise ratio (PSNR), the structural similarity index measure (SSIM) (Wang et al., 2004), and learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018) as evaluation metrics.

4.2 Comparisons

Self-supervised Object Segmentation on LLFF We first build NeRF-SOS on the vanilla NeRF (Mildenhall et al., 2020a) to validate its effectiveness on LLFF datasets. Two groups of current object segmentation are adopted for comparisons: i. NeRF-based methods, including our NeRF-SOS, and supervised Semantic-NeRF (Zhi et al., 2021) trained with annotated masks; ii. image-based object co-segmentation methods: DINO-CoSeg (Amir et al., 2021) and DOCS (Li et al., 2018); and iii. single-image based unsupervised segmentation: IEM (Savarese et al., 2021) which proposes to partition images into maximally independent sets. As image-based segmentation methods cannot generate novel views, we pre-render the new views using NeRF and construct image pairs between the first image in the test set with others for DINO-CoSeg (Amir et al., 2021) and DOCS (Li et al., 2018). The evaluations on IEM also use the pre-rendered color images.

Quantitative comparisons against other segmentation methods are provided in Table 1, together with

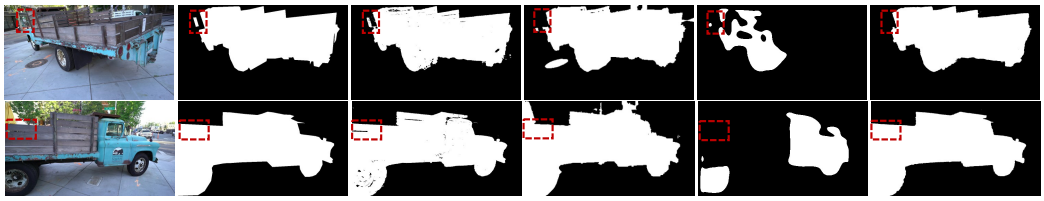


Figure 6: Novel view object segmentation results on unbounded scene *Truck*. NeRF-SOS (the 3rd column) produces more view-consistent masks than other self-supervised methods. It even generates finer details than supervised Semantic-NeRF++ (see the gaps between wooden slats in the top row and the side view mirror in the bottom row). Videos can be viewed in the supplementary materials.

Table 2: Quantitative evaluation of the novel view synthesis and object segmentation on BlendedMVS and CO3Dv2 datasets, with several 2D object discovery frameworks and the supervised Semantic-NeRF. Results on each dataset are averaged on all scenes.

BlendedMVS	PSNR ↑	SSIM ↑	LPIPS ↓	NV-ARI ↑	IoU(BG) ↑	IoU(FG) ↑	mIoU ↑
IEM (Savarese et al., 2021)	-	-	-	0.1339	0.5615	0.3715	0.4665
DOCS (Li et al., 2018)	-	-	-	0.7031	0.9183	0.7030	0.8107
DINO+CoSeg (Amir et al., 2021)	-	-	-	0.9074	0.9692	0.917	0.9431
NeRF-SOS (Ours)	23.86	0.8089	0.1288	0.9280	0.9756	0.9347	0.9552
Semantic-NeRF (Zhi et al., 2021) (Supervised)	23.84	0.8080	0.1339	0.9359	0.9803	0.9391	0.9597
CO3Dv2	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**NV-ARI ↑**	**IoU(BG) ↑**	**IoU(FG) ↑**	**mIoU ↑**
IEM (Savarese et al., 2021)	-	-	-	0.4784	0.7983	0.5708	0.6845
DOCS (Li et al., 2018)	-	-	-	0.8918	0.9684	0.8928	0.9307
DINO+CoSeg (Amir et al., 2021)	-	-	-	0.8199	0.9559	0.8222	0.8891
NeRF-SOS (Ours)	30.37	0.9358	0.073	0.9381	0.9813	0.9401	0.9607
Semantic-NeRF (Zhi et al., 2021) (Supervised)	31.17	0.9405	0.0603	0.9399	0.9821	0.9410	0.9615

Table 3: Quantitative results of the object segmentation results on outdoor unbounded scene *Truck*, with several 2D object discovery frameworks and the supervised Semantic-NeRF.

Scene "*Truck*"	PSNR ↑	SSIM ↑	LPIPS ↓	NV-ARI ↑	IoU(BG) ↑	IoU(FG) ↑	mIoU ↑
IEM (Savarese et al., 2021)	-	-	-	0.3341	0.6791	0.5998	0.6395
DOCS (Li et al., 2018)	-	-	-	0.1517	0.6845	0.2463	0.4654
DINO+CoSeg (Amir et al., 2021)	-	-	-	0.8571	0.9408	0.9080	0.9244
NeRF-SOS (Ours)	22.20	0.7000	0.2691	0.9207	0.9689	0.9455	0.9572
Semantic-NeRF++ (Zhi et al., 2021) (Supervised)	21.08	0.6350	0.4114	0.9674	0.9869	0.9782	0.9826

qualitative visualizations shown in Figure 4. These results convey several observations to us: **1).** NeRF-SOS consistently outperforms image-based co-segmentation in evaluation metrics and view-consistency. **2).** Compared with SoTA supervised NeRF segmentation method (Semantic-NeRF (Zhi et al., 2021)), our method effectively segments the object within the scene and performs on par in both evaluation metrics and visualization.

Self-supervised Object Segmentation on Object-centric Scenes For the object-centric datasets BlendedMVS and CO3Dv2, we uniformly select 12.5% of total images for testing. CO3Dv2 provides coarse segmentation maps using PointRend (Kirillov et al., 2020) while parts of the annotations are missing. Therefore, we manually create faithful binary masks for training the Semantic-NeRF and evaluations. As we can see in Table 2 and Figure 5, our self-supervised NeRF method consistently surpasses other 2D methods. We deliver more details comparisons in our supplementary materials.

Self-supervised Object Segmentation on Unbounded Scene To test the generalization ability of the proposed collaborative contrastive loss, we implement it on NeRF++ (Zhang et al., 2020) to test with a more challenging unbounded scene. Here, we mainly evaluate all previously mentioned methods on scene *Truck* as it is the only scene captured surrounding an object provided by NeRF++. We re-implement Semantic-NeRF using NeRF++ as the backbone model for unbounded setting, termed Semantic-NeRF++. Compared with supervised Semantic-NeRF++, NeRF-SOS achieves slightly worse results on quantitative metrics (see Table 3). Yet from the visualizations, we see that NeRF-SOS yields quite decent segmentation quality. For example, **1).** In the first row of Figure 6, NeRF-SOS can recognize the side view mirror adjacent to the truck. **2).** In the second row of Figure 6, NeRF-SOS can distinguish the apertures between the wooden slats as those apertures have distinct depths than the neighboring slats, thanks to the geometry-aware contrastive loss. Further, we show the 3-center clustering results on the distilled segmentation field in Figure 8.

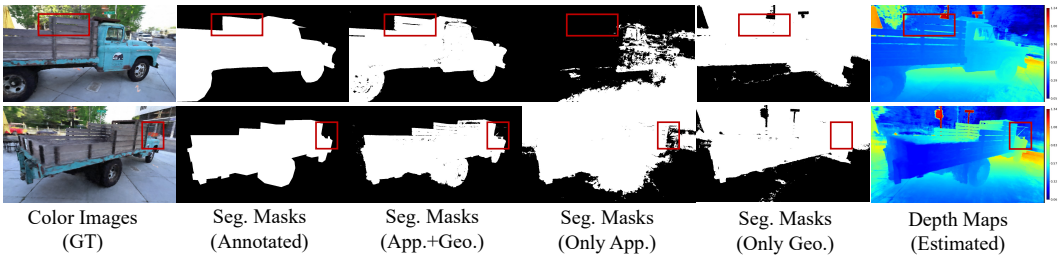


Figure 7: Object segmentations using three loss variants are shown in columns 3, 4, and 5: the collaborative loss (APP.+Geo.), appearance-only loss (App.); geometric-only loss (Geo.).

Figure 8: Qualitative results on scene *Truck* with different cluster centers on its distilled segmentation field. Note that the cross-view visualized colors of multiple-center clustering are not corresponding to the subject ID, as we perform unsupervised clustering.

Table 4: Experiments on multiple NeRF-SOS variants. We show the results of joint training of the NeRF and contrastive loss in the first row, NeRF-SOS with ResNet50 as feature extractor in the second row, and our final model in the last row.

Scene "*Flower*"	PSNR ↑	SSIM ↑	LPIPS ↓	NV-ARI ↑	IoU(BG) ↑	IoU(FG) ↑	mIoU ↑
NeRF-SOS (Joint training)	16.96	0.4585	0.7238	0.1961	0.7951	0.2220	0.5091
NeRF-SOS (ResNet)	25.96	0.7717	0.1502	0.8672	0.9421	0.8827	0.9124
NeRF-SOS (Two-stage training)	25.96	0.7717	0.1502	0.9529	0.9869	0.9503	0.9686

4.3 Ablation Study

Impact of the Collaborative Contrastive Loss To study the effectiveness of the collaborative contrastive loss, we adopt two baseline models by only using appearance contrastive loss or geometric contrastive loss on NeRF++ backbone. As shown in Figure 7, we observe that the segmentation branch failed to cluster spatially continuous objects without geometric constraints. Similarly, without visual cues, the model lost the perception of the central object.

Joint Training with NeRF Optimization To demonstrate the advantages of two-stage training, we conduct an ablation study by jointly optimizing vanilla NeRF rendering loss and the proposed two-level collaborative contrastive loss. As shown in Table 4, both the novel view synthesis quality and the segmentation quality significantly decreased when we optimize the two losses together. We conjecture the potential reason to be the fact that the optimization process of NeRF training is affected by the conflicting update directions, the reconstruction loss and the contrastive loss, which remains a notorious challenge in the multi-task learning area (Yu et al., 2020).

CNN-based Backbone for Feature Extraction DINO-ViT firstly concludes that ViT architecture can extract stronger semantic information than ConvNets when being self-supervised trained. To study its effect on discovering the semantic layout of scenes, we apply self-supervised ResNet50 (He et al., 2020) as backbones. The results in the second row of Table 4 imply that the ViT architecture is more suitable for our NeRF object segmentation in both expressiveness and pair-selection perspectives.

5 Conclusion, Discussion of Limitation

We present NeRF-SOS, a framework that learns object segmentation for any view from complex real-world scenes. A self-supervised framework, NeRF-SOS leverages a collaborative contrastive loss in appearance-segmentation and geometry-segmentation levels are included. Comprehensive experiments on four different types of datasets are conducted, in comparison to SoTA image-based object (co-) segmentation frameworks and fully supervised Semantic-NeRF. NeRF-SOS consistently performs better than image-based methods, and sometimes generates finer segmentation details than its supervised counterparts. Similar to other scene-specific NeRF methods, one limitation of NeRF-SOS is that it cannot segment across scenes, which we will explore in future works.

REFERENCES

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

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

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*, pp. 9650–9660, 2021.

Eric Chan, Marco Monteiro, Peter Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. *https://arxiv.org/abs/2012.00926*, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Xingyu Chen, Qi Zhang, Xiaoyu Li, Yue Chen, Feng Ying, Xuan Wang, and Jue Wang. Hallucinated neural radiance fields in the wild, 2021.

Joseph DeChicchis. Semantic understanding for augmented reality and its applications. 2020.

Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. *arXiv preprint arXiv:2107.02791*, 2021.

Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. *https://arxiv.org/abs/2103.10380*, 2021.

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. *arXiv preprint arXiv:2203.08414*, 2022.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Olivier J Hénaff, Skanda Koppula, Evan Shelhamer, Daniel Zoran, Andrew Jaegle, Andrew Zisserman, João Carreira, and Relja Arandjelović. Object discovery and representation networks. *arXiv preprint arXiv:2203.08777*, 2022.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12949–12958, 2021.

Zhang Jiakai, Liu Xinhang, Ye Xinyi, Zhao Fuqiang, Zhang Yanshun, Wu Minye, Zhang Yingliang, Xu Lan, and Yu Jingyi. Editable free-viewpoint video using a layered neural representation. In *ACM SIGGRAPH*, 2021.

Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. *arXiv preprint arXiv:2111.13539*, 2021.

Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9799–9808, 2020.

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

Adarsh Kowdle, Dhruv Batra, Wen-Chao Chen, and Tsuhan Chen. imodel: interactive co-segmentation for object of interest 3d modeling. In *European Conference on Computer Vision*, pp. 211–224. Springer, 2010.

labelme. labelme. `https://github.com/wkentaro/labelme`.

Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. In *Asian Conference on Computer Vision*, pp. 638–653. Springer, 2018.

David Lindell, Julien Martel, and Gordon Wetzstein. AutoInt: Automatic integration for fast neural volume rendering. *https://arxiv.org/abs/2012.01714*, 2020.

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields, 2021.

Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering, 2021.

Ricardo Martin-Brualla, Noha Radwan, Mehdi Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. *https://arxiv.org/abs/2008.02268*, 2020.

Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 1995.

Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6351–6361, 2021.

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.

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 *European conference on computer vision*, pp. 405–421. Springer, 2020a.

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 *European conference on computer vision*, pp. 405–421. Springer, 2020b.

Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. DeRF: Decomposed radiance fields. *https://arxiv.org/abs/2011.12490*, 2020.

Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10901–10911, 2021.

Gernot Riegler and Vladlen Koltun. Free view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pp. 993–1000. IEEE, 2006.

Pedro Savarese, Sunnie SY Kim, Michael Maire, Greg Shakhnarovich, and David McAllester. Information-theoretic segmentation by inpainting error maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4029–4039, 2021.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3D-aware image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020a.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33: 20154–20166, 2020b.

Tong Shen, Guosheng Lin, Lingqiao Liu, Chunhua Shen, and Ian Reid. Weakly supervised semantic segmentation based on co-segmentation. In *BMVC*, 2017.

Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. *arXiv preprint arXiv:2109.14279*, 2021.

Karl Stelzner, Kristian Kersting, and Adam R Kosiorek. Decomposing 3d scenes into objects via unsupervised volume segmentation. *arXiv preprint arXiv:2104.01148*, 2021.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *arXiv preprint arXiv:2111.11215*, 2021.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Alex Trevithick and Bo Yang. GRF: Learning a general radiance field for 3D scene representation and rendering. *https://arxiv.org/abs/2010.04595*, 2020.

Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. *arXiv preprint arXiv:2209.03494*, 2022.

Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. *arXiv preprint arXiv:2201.00424*, 2022.

Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *CVPR 2011*, pp. 2217–2224. IEEE, 2011.

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

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

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021b.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Bangbang Yang, Yinda Zhang, Yinghao Xu, Yijin Li, Han Zhou, Hujun Bao, Guofeng Zhang, and Zhaopeng Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13779–13788, 2021.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1790–1799, 2020.

Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *IEEE International Conference on Computer Vision (ICCV)*, 2021a.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021b.

Hong-Xing Yu, Leonidas J Guibas, and Jiajun Wu. Unsupervised discovery of object radiance fields. *arXiv preprint arXiv:2107.07905*, 2021c.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33: 5824–5836, 2020.

Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16259–16268, 2021.

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

A APPENDIX

A.1 ADDITIONAL TRAINING DETAILS

Implementation of the Patch Selection We reconstruct positive and negative pairs on the fly during training. Given N rendered patches from N different viewpoints in training, we fed the patches into the DINO-ViT and obtained the [CLS] tokens. Next, we compute a $N \times N$ similarity matrix using the cosine similarity with the [CLS] tokens. The negative pairs are selected from the pair with the lowest similarity in each row; the positive pairs are set as the identity pairs. Overall, $2N$ pairs (N positives + N negatives) are formulated per iteration to compute the collaborative contrastive loss.

Implementation of Stride Ray Sampling Neural radiance field casts a number of rays (typically not adjacent) from the camera origin, intersecting the pixel, to generate input 3D points in the viewing frustum. Our model requires patch-wise rendering of size (P, P) to formulate the collaborative contrastive loss. However, we can only render a patch less than 64 $\times$ 64 in each view due to GPU memory bottleneck (Garbin et al., 2021). Thus, it hardly covers a sufficient receptive field to capture the global context, using the pre-trained DINO. To solve this problem, we adopt a *Strided Ray Sampling* strategy (Schwarz et al., 2020b; Meng et al., 2021), to enlarge the receptive field of the patches while keeping computational cost fixed. Specifically, instead of sampling a patch of adjacent locations $P \times P$, we sample rays with an interval k, resulting in a receptive field of $(P \times k) \times (P \times k)$.

Hyperparameters Selection The hyperparameters of NeRF-SOS on different datasets are shown in Table 5. We adopt the number of b^{knn} and b^{self} in (Hamilton et al., 2022) as the hyperparameter for our appearance level (b^{neg} and b^{id}). We share the appearance level hyperparameters across all datasets. We set the weights of b^{neg} and b^{id} in the geometry level loss with physical intuition. For example, since the radius of the foreground object in LLFF datasets is roughly 0.5 meters, we set the b^{neg} and b^{id} to be 0.5 and 3, respectively. Analogously, for the unbounded scene (e.g., scene *Truck*), the b^{neg} and b^{id} are set to be 1 and 5, respectively.

Table 5: Hyperparameters of NeRF-SOS on different datasets. We share the hyperparameters of appearance level for all datasets while we set the b^{neg} and b^{id} for geometry level loss with physical intuition.

parameter name	LLFF	BlendedMVS	CO3Dv2	Tank and Temples
$b_{id}(\mathcal{L}_{geo})$	0.50	0.12	0.25	1.00
$b_{neg}(\mathcal{L}_{geo})$	3.00	0.60	1.00	5.00

Self-supervised Learned 2D Representations We adopt DINO-ViT (Caron et al., 2021) as our feature extractor for distillation. The training process of DINO-ViT largely simplifies self-supervised learning by applying a knowledge distillation paradigm (Hinton et al., 2015) with a momentum encoder (He et al., 2020), where the model is simply updated by a cross-entropy loss.

A.2 HUMAN ANNOTATION DETAILS

We use the publicly available annotation tool: (labelme) for the foreground and background annotation. To be specific, we annotate all training and testing views of different scenes using multiple-polygon, extract the polygons and convert them to binary masks. The masks of scene *Flower* are included in our supplementary and we provide usage guidelines in README. All annotated data will be made public.

A.3 ADDITIONAL VISUALIZATIONS

Qualitative Visualization on More Views Qualitative comparisons on LLFF dataset can be found in Figure 9 and Figure 10, respectively. Qualitative comparisons on BlendedMVS dataset can be found in Figure 11 and Figure 12, respectively. Qualitative comparisons on CO3Dv2 dataset can be found in Figure 13 and Figure 14, respectively. Qualitative comparisons on Tank and Temple

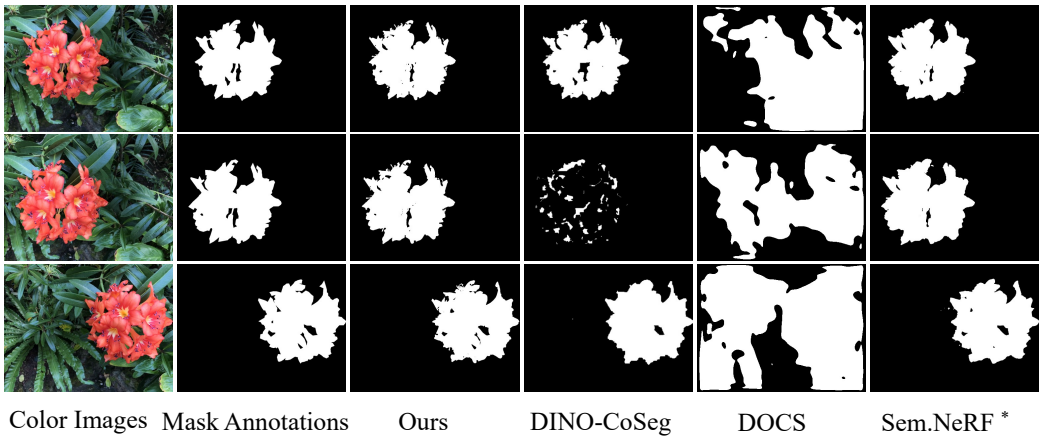


Figure 9: Novel view object segmentation results on scene *Flower* of LLFF dataset.

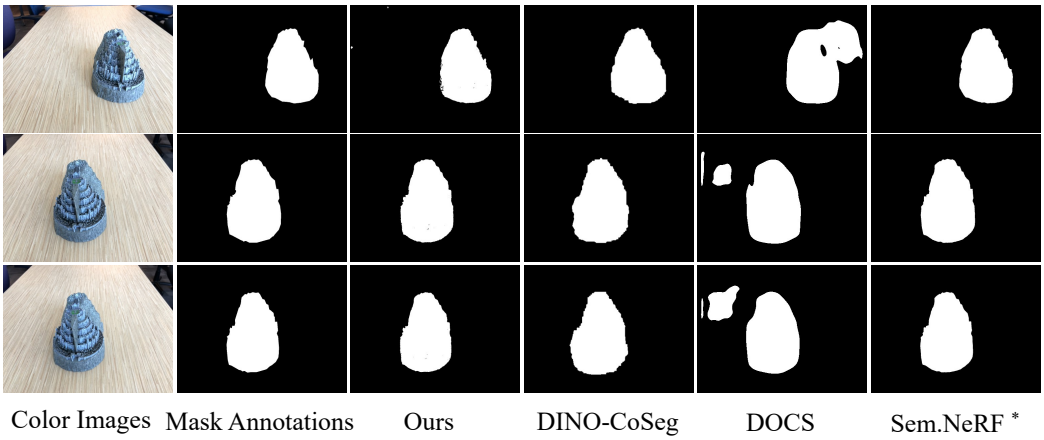


Figure 10: Novel view object segmentation results on scene *Foretress* of LLFF dataset.

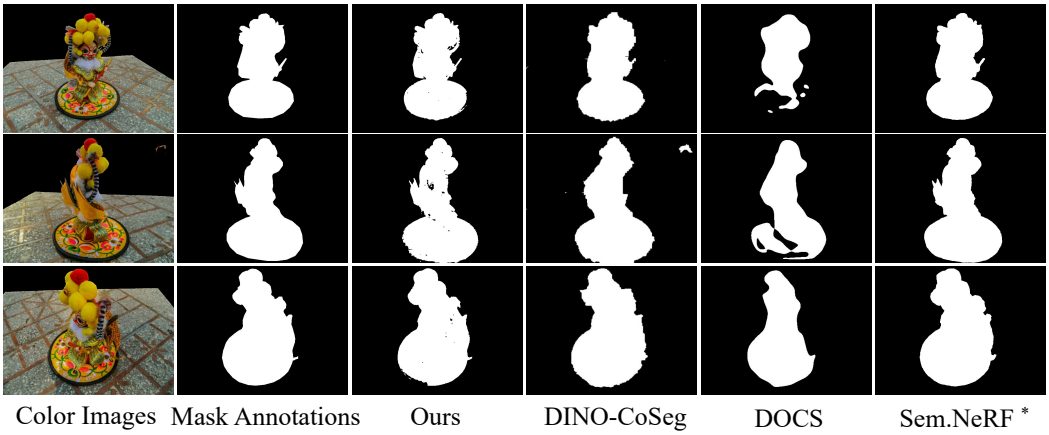


Figure 11: Novel view object segmentation results on scene *5a3* of BlendedMVS dataset.

dataset can be found in Figure 15. Here, we visualize three different views to show the segmentation consistency across views. Visualized video can be found in the supplementary.

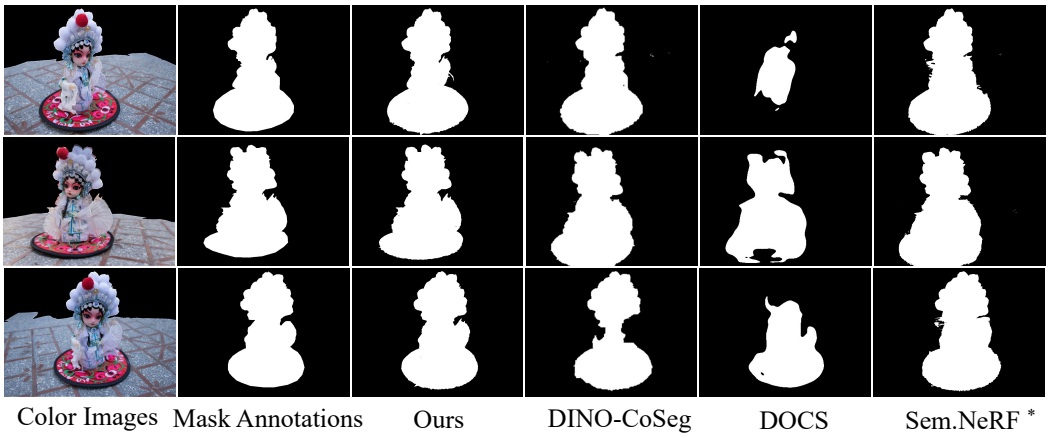


Figure 12: Novel view object segmentation results on scene *5a6* of BlendedMVS dataset.

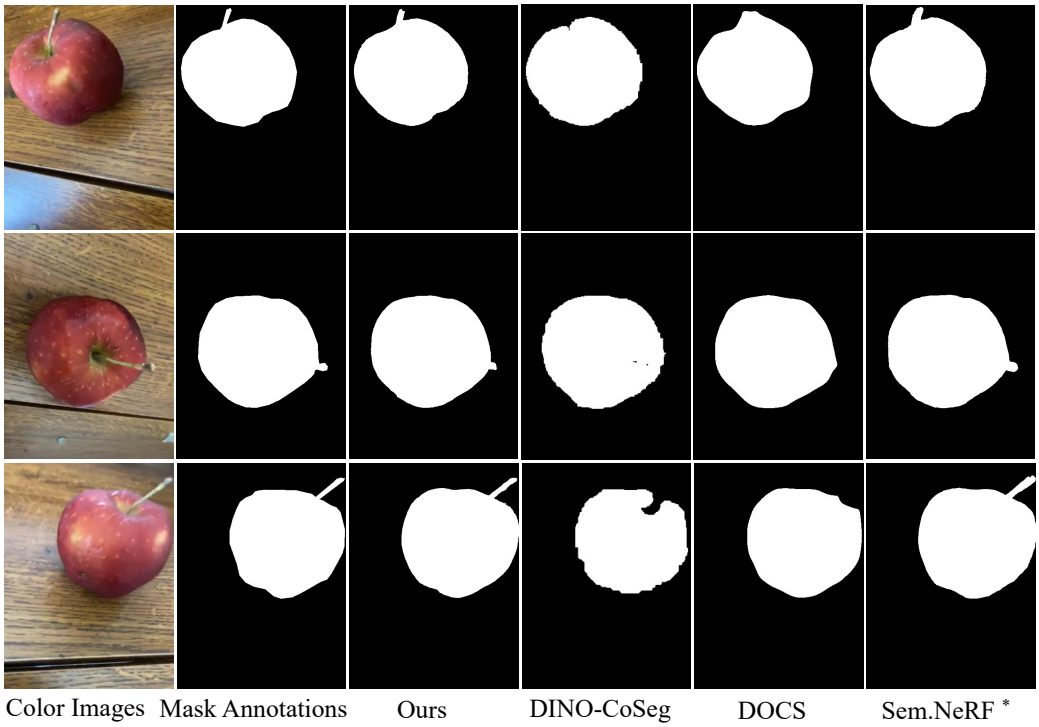


Figure 13: Novel view object segmentation results on scene *Apple* of CO3Dv2 dataset.

Figure 14: Novel view object segmentation results on scene *Backpack* of CO3Dv2 dataset.

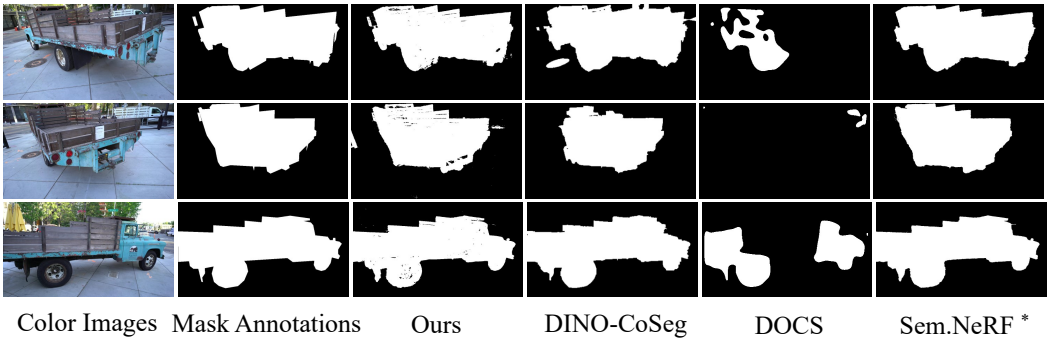


Figure 15: Novel view object segmentation results on scene *Truck* of Tank and Temples dataset.