SANeRF-HQ: Segment Anything for NeRF in High Quality

Yichen Liu[1] Benran Hu[2] Chi-Keung Tang[1] Yu-Wing Tai[3]
[1]The Hong Kong University of Science and Technology
[2]Carnegie Mellon University [3]Dartmouth College

Abstract

Recently, the Segment Anything Model (SAM) has showcased remarkable capabilities of zero-shot segmentation, while NeRF (Neural Radiance Fields) has gained popularity as a method for various 3D problems beyond novel view synthesis. Though there exist initial attempts to incorporate these two methods into 3D segmentation, they face the challenge of accurately and consistently segmenting objects in complex scenarios. In this paper, we introduce the Segment Anything for NeRF in High Quality (SANeRF-HQ) to achieve high quality 3D segmentation of any object in a given scene. SANeRF-HQ utilizes SAM for open-world object segmentation guided by user-supplied prompts, while leveraging NeRF to aggregate information from different viewpoints. To overcome the aforementioned challenges, we employ density field and RGB similarity to enhance the accuracy of segmentation boundary during the aggregation. Emphasizing on segmentation accuracy, we evaluate our method quantitatively on multiple NeRF datasets where high-quality ground-truths are available or manually annotated. SANeRF-HQ shows a significant quality improvement over previous state-of-the-art methods in NeRF object segmentation, provides higher flexibility for object localization, and enables more consistent object segmentation across multiple views. Additional information can be found at https://lyclyc52.github.io/SANeRF-HQ/.

1. Introduction

Neural Radiance Field (NeRF) [41] has produced state-of-the-art results in novel view synthesis for intricate real-world scenes. NeRF encodes a given scene using Multi-Layer Perceptrons (MLP) and supports queries of density and radiance given 3D coordinates and view directions, which are used to render photo-realistic images from any view points. Moreover, during training, NeRF only requires RGB images with camera poses, which directly links 3D to 2D. The simple but ingenious architecture with its continuous representation quickly starts challenging traditional representations using explicit discrete structures, such as RGB-D images or point clouds. As a result, NeRF is poised to tackle more challenging tasks in 3D vision.

One important downstream task that can benefit from NeRF representations is 3D object segmentation, which is fundamental in 3D vision and widely used in many applications. To address object segmentation in NeRF, researchers have investigated various methods. Semantic-NeRF [65], targeting semantic segmentation, is one of the first works in this direction. DFF [33] distills the knowledge of pre-trained features such as DINO [9] into a 3D feature field for unsupervised object decomposition. Supervised approaches, such as [47], utilize Mask2Former [14] to obtain initial 2D masks and lifts them to 3D with a panoptic radiance field. Although these methods demonstrate impressive results, their performance is constrained by the pre-trained models used to produce features.

Recently, large vision models, such as Segment Anything Model (SAM) [32], have emerged with strong zero-shot generalization performance and can be adopted as the backbone component for many downstream tasks. Specifically, SAM proposes a new paradigm for segmentation tasks which can accept a wide variety of prompts as input, and produce segmentation masks of different semantic levels as output. The versatility and generalizability of SAM thus shed light on a new way to perform promptable object segmentation in NeRF. While there exist some investigations [10, 13, 21] into this area, the mask quality in novel views is still unsatisfactory.

In view of this, we propose a new general framework to achieve prompt-based 3D segmentation in NeRF. Our framework, termed Segment Anything for NeRF in High Quality, or SANeRF-HQ, leverages existing 2D foundation models such as Segment Anything to allow various prompts as input, and produces 3D segmentations with high accuracy and multi-view consistency. The major contributions of our paper are:

- We propose SANeRF-HQ, which is one of the first attempts in producing high-quality 3D object segmentation in NeRF in terms of more accurate segmentation boundaries and better multi-view consistency.
- We validate our method by assembling and evaluating

quantitatively on a challenging dataset with high-quality ground-truths.
- We present a more general framework to embed foundation 2D image models into Neural Radiance Field and extend it to different 3D segmentation tasks in NeRFs.

Comparing with [21] and [10], SANeRF-HQ can produce more accurate segmentation results and is more flexible to a variety of segmentation tasks. Similar to SAM [32] in 2D segmentation, SANeRF-HQ can automatically segment a given entire 3D scene in high quality without any user inputs. SANeRF-HQ inherits the zero-shot performance from SAM, instead of being bounded by pre-trained models with limited generalizability [5, 38, 47]. SANeRF-HQ is a general framework which can be readily extended to 4D dynamic NeRFs, where temporal consistency can be naturally handled in a similar way as our multi-view consistency. Our preliminary results confirm the potential extension to dynamic scenes.

2. Related Work

2.1. Object Segmentation

2D image segmentation is a thoroughly studied area where a lot of progress has been made, including semantic segmentation [12, 22, 63, 64], instance segmentation [6, 7, 23, 51, 55, 56], and panoptic segmentation [18, 28, 31, 35, 36, 58]. With the emergence of Transformer-based models [8], Vision Transformer (ViT) [19, 39, 60, 61] has become increasingly popular as a backbone structure, and pre-trained features from large models such as MAE [24, 52, 57] have demonstrated great power in segmentation tasks. Moreover, significant research effort [14, 28, 62] focuses on universal models to accomplish multiple segmentation tasks under different training configuration. However, the performance of these models on open-world images is not satisfactory, as they cannot go beyond the limits prescribed by the underlying training datasets.

Thus, latest research has been focusing on open-world segmentation, aiming to generalize segmentation models to unseen data. The recent advancement in visual foundation models have attracted great attention. DINOv2 [44] leverages self-supervised distillation and produces visual features across domains without any fine-tuning. Segment Anything Model (SAM) [32], as a more significant breakthrough, shows promising results on promptable segmentation. Given diverse and plentiful training data, the prompt-based architecture can enable zero-shot generalization, which extends relevant tasks to wider data categories. Recent works [16, 45, 66] have already adopted SAM in many downstream tasks, enhancing the capability of different segmentation models.

In spite of the great success of 2D segmentation, 3D segmentation is relatively underexplored. Traditional methods are usually based on RGB-D images [25] or point clouds [53, 54]. However, they require explicit depth or 3D representations as input, and the generalizability is highly restricted by the scarcity of the dataset and the expensive computational cost. Therefore, exploiting the vast 2D image datasets and performing 3D segmentation directly from 2D multi-view images warrants attention from the community.

2.2. Segmentation in Neural Radiance Field

The family of Neural Radiance Fields (NeRF) [2–4, 11, 41, 42] has become the state-of-the-art for novel view synthesis. Beyond that, requiring only multi-view images with camera parameters during training, NeRF can also be regarded as an implicit 3D representation since it captures the 3D structural details of the scenes. Due to the capability of linking 2D images to 3D volumes, NeRF shows potential impact in various 3D visual tasks, and numerous research works have been focusing on 3D object segmentation and scene decomposition in NeRF.

Semantic-NeRF [65] extends NeRF with an additional branch encoding 3D semantic labels for semantic-level segmentation. Other research investigated 3D instance segmentation and make great efforts in solving inconsistency across views. For example, [38] adopts an 3D object detection method [26] to resolve 2D mask correspondences. [47] utilize linear assignment, while [5] employs a contrastive loss to optimize 3D instance embeddings. Besides, there are unsupervised methods [43, 48, 59], which can separate foreground objects from background. However, these approaches are mostly limited to simple scenes and often struggle to generalize to complex open-world problems. To fully utilize 2D features from pre-trained models, some research [21, 30, 33] introduces an extra feature field to the vanilla NeRF model, which can fuse 2D features from pre-trained models into 3D. For instance, LERF [30] splits images into patches of different sizes to obtain multi-scale CLIP [27] feature maps that can supervise the neural field training. ISRF [21] uses DINO [9] features and K-Means clustering to separate user-selected regions from background. However, lacking a powerful decoder, the segmentation based on these features is not accurate and sharp along the boundaries.

With the proposal of SAM, SA3D [10] employs SAM on the NeRF-rendered images and achieve 3D segmentation from a single-view 2D mask by self-prompting. Nonetheless this pipeline relies on the first-view mask and is susceptible to the ambiguity inherent in SAM in delineating intricate structures during its self-prompting. SAN [13] chooses to distill the SAM encoder with a neural field to render SAM feature maps from novel views, which are supplied to the SAM decoder to produce the segmentation. However, it can produce inconsistent masks in different views, and

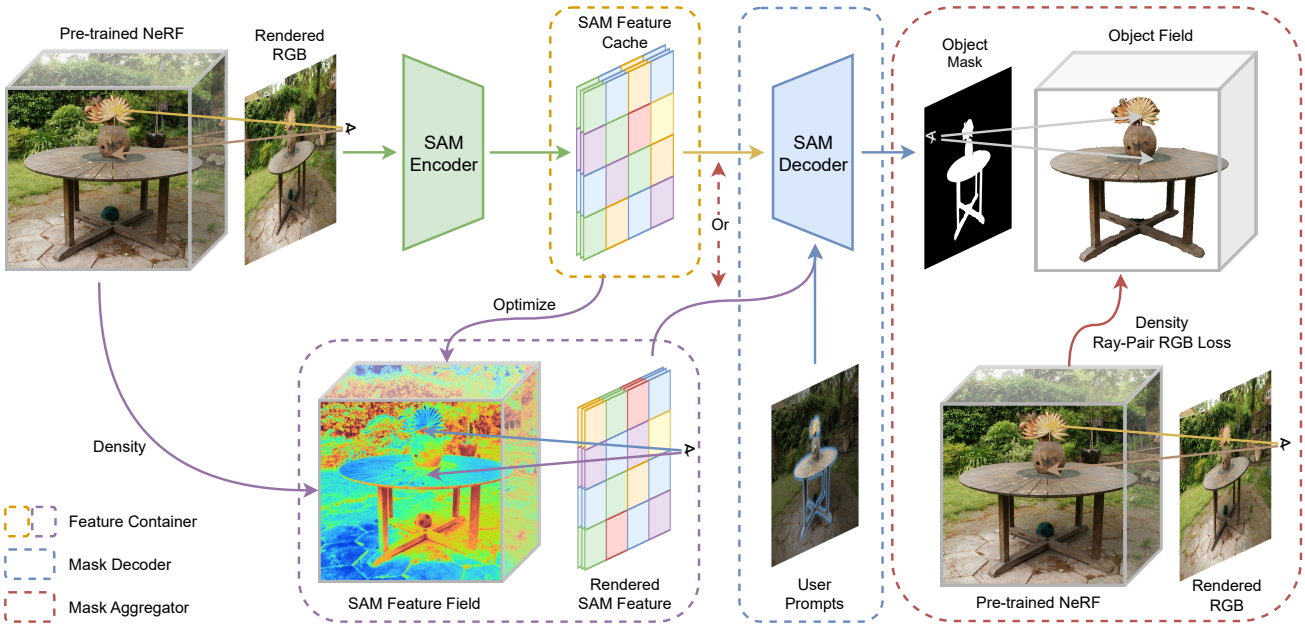

Figure 1. **SANeRF-HQ Pipeline.** Our method is composed of three parts: a feature container (feature cache or feature field), a mask decoder, and a mask aggregator (object field). It first renders a set of images using a pre-trained NeRF and encodes their SAM features, which are cached or used to optimize a feature field. SAM decoder takes the feature maps from the cache or the feature field, and generates 2D masks based on user prompts. The aggregator fuses 2D masks from different views to produce an object field.

distilling low resolution SAM feature maps results in aliasing in the output masks, in the form of jagged mask edges. Contrary to these approaches, our proposed method directly aggregates masks for neural field training, which naturally addresses the aliasing and consistency issues.

3. Method

Given a pre-trained NeRF, our method aims to segment any kind of object in 3D, conditioning on the manual prompts and/or other user supplied inputs. The SANeRF-HQ pipeline in Figure 1 consists of three major components: a feature container, a mask decoder, and a mask aggregator. The feature container encodes the SAM features of images. The mask decoder propagates the user supplied hints between different views and generates intermediate mask outputs using the SAM features from the container. Finally, the mask aggregator integrates the resulted 2D masks into 3D space and utilizes the color and density fields from NeRF models to achieve high-quality 3D segmentation.

3.1. Feature Container

The first step of utilizing SAM is to encode the images into 2D features using the SAM feature encoder. These features can be used repeatedly when predicting and propagating masks, thus they can be precomputed or distilled for a scene and reused for different input prompts.

We consider two different methods for the feature container. The first method is to compute and cache the features of multiple views. This allows us to reuse ground-truth SAM features for different user prompts and generate accurate 2D mask when decoding. However, the cache size is constrained by the memory available. It also requires extra time to run the encoder if users choose to supply prompts on any of the uncached novel views.

Another method is to distill the SAM features using a neural field, which is similarly done in SAN [13] and in [21, 30], where SAM, DINO, or CLIP features are lifted into 3D. Instead of radiance or density, 3D SAM embeddings are encoded in a neural field and the same volumetric rendering equation is applied to render 2D feature maps. Specifically, vanilla NeRF [41] is formulated as $f(\mathbf{x}, \mathbf{d}; \Theta_N) = (\sigma, \mathbf{c})$, where $\mathbf{x} = (x, y, z)$ is the position of the point, $\mathbf{d} = (\theta, \phi)$ is the view direction, and Θ_N is the set of parameters of the color and density field. The RGB color at each pixel is estimated through a ray casting process:

$$\begin{aligned} &\hat{\mathbf{C}}(\mathbf{r}) = \textstyle\sum_{k=1}^{K} \hat{T}\left(t_k\right) \alpha\left(\sigma\left(t_k\right) \delta_k\right) \mathbf{c}\left(t_k\right), \\ &\hat{T}\left(t_k\right) = \exp\left(-\textstyle\sum_{a=1}^{k-1} \sigma\left(t_a\right) \delta_a\right), \\ &\alpha(x) = 1 - \exp(-x), \\ &\delta_k = t_{k+1} - t_k, \end{aligned} \tag{1}$$

where $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ is the ray emitted from the camera center passing through that pixel, and $\sigma\left(t_k\right)$ and $\mathbf{c}\left(t_k\right)$ are the volume density and color at the point $\mathbf{o} + t_k\mathbf{d}$ along the ray.

To encode SAM features, the SAM embedding at $(\mathbf{x}, \mathbf{d})$ is defined as $\mathbf{f} = f(\mathbf{x}, \mathbf{d}; \Theta_f)$, where Θ_f is the set of pa-

rameters of the feature field. The feature $\hat{\mathbf{F}}$ integrated over ray $\mathbf{r}$ is given as:

$$\hat{\mathbf{F}}(\mathbf{r}) = \sum_{k=1}^{K} \hat{T}(t_k)\,\alpha(\sigma(t_k)\,\delta_k)\,\mathbf{f}(t_k). \tag{2}$$

And the feature field is optimized with the MSE loss $\mathcal{L}_f$:

$$\mathcal{L}_f = \sum_{\mathbf{r}\in\mathcal{R}(\Phi)} \left\| \hat{\mathbf{F}}(\mathbf{r}) - \mathbf{F}(\mathbf{r}) \right\|_2^2, \tag{3}$$

where $\mathcal{R}(\Phi)$ is the set of rays from the feature map Φ and $\mathbf{F}(\mathbf{r})$ is the ground-truth feature value of the ray $\mathbf{r}$.

The feature field enables feature map rendering from any viewpoint efficiently, as aggregating features in the neural field is typically faster than running the original SAM encoder. However, the feature maps produced by the SAN encoder has a relatively low resolution, which can cause severe aliasing in the rendered feature maps. While this can be alleviated by augmenting the input camera views and sample more rays during distillation, the rendered features still deteriorate after distillation. The rendered SAN features usually fail to retain accurate high frequency spatial information along boundaries, which consequently leads to jagged mask boundaries after decoding.

Noting the complementary advantages and disadvantages of the caching method and the feature distillation method, we conducted experiments on both methods in the ablation study.

3.2. Mask Decoder

The Mask Decoder $\mathbf{Dec}$ takes as input the feature map from the feature container and generates 2D masks based on the input prompts (e.g., 2D or 3D points, texts). Figure 2 illustrates the architecture of the decoder, which is similar to the SAM decoder. The 2D mask decoding can be formulated as

$$\mathbf{M} = \mathbf{Dec}(\hat{\Phi}, \mathit{prompts}), \tag{4}$$

where $\hat{\Phi}$ is the feature map. NeRF can estimate depth with Equation 5,

$$\hat{\mathbf{D}}(\mathbf{r}) = \sum_{k=1}^{K} \hat{T}(t_k)\,\alpha(\sigma(t_k)\,\delta_k)\,t_k, \tag{5}$$

so 3D points can be easily obtained by projecting 2D prompts from users back to 3D with camera poses. Given a 2D point (w, h), its depth $d(p)$, the camera intrinsic matrix $\mathbf{K}$, and extrinsic matrix $\mathbf{P}$, the corresponding 3D point $\mathbf{p} = (x, y, z)^T$ in the world space is

$$\mathbf{p} = \mathbf{P}^{-1}\mathbf{K}^{-1} \begin{pmatrix} w \cdot d(p) \\ h \cdot d(p) \\ d(p) \end{pmatrix}. \tag{6}$$

The equation to project 3D points in world space to 2D pixel coordinates in other camera views can be derived likewise.

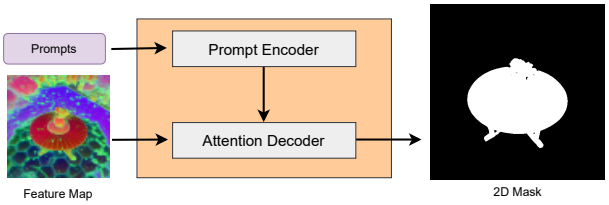

Figure 2. **Mask Decoder Architecture.** The decoder consists of a prompt encoder and an attention decoder. First, the prompts are fed into the prompt encoder. The attention decoder takes the encoded prompts and the feature map from the Feature Container, and uses attention to produce 2D masks for the given view.

3.3. Mask Aggregator

The decoder cannot produce correct 2D masks if the 3D points after projection are not visible at certain viewpoints. Furthermore, despite good performance in most cases, the predicted masks may include artifacts. The innate semantic ambiguity of SAM predictions can also cause inconsistency across views. Hence, we aggregate these imperfect 2D masks in the 3D space to generate high-quality and consistent 3D masks.

3.3.1 Object Field

Given the decoder output from different views, the object field can fuse all 2D images and generate accurate 3D masks. The mask is represented by a L-dimensional object identity vector i. To represent the identity value, an additional branch is introduced into the pre-trained NeRF model, parameterized as Θ_o. Different from the function of RGB color, which is view-dependent, object identity function is defined by $\mathbf{i} = f(\mathbf{x}; \Theta_o)$, where the view direction vector is not included in the inputs due to the view invariance of the object masks in 3D. The equation for mask rendering is similar to that of image rendering:

$$\hat{\mathbf{M}}(\mathbf{r}) = \mathit{Softmax}\left(\sum_{k=1}^{K} \hat{T}(t_k)\,\alpha(\sigma(t_k)\,\delta_k)\,\mathbf{i}(t_k) \right), \tag{7}$$

where $\sigma(t_k)$ is inherited from the pre-trained NeRF model. Volume density σ which interprets the 3D geometry in NeRF makes the object field aware of the structural information. The object field is trained with the cross-entropy loss $\mathcal{L}_o$:

$$\mathcal{L}_o(\mathcal{R}) = -\frac{1}{|\mathcal{R}|L} \sum_{\mathbf{r}\in\mathcal{R}} \sum_{l=1}^{L} m^l(\mathbf{r}) \log \hat{m}^l(\mathbf{r}), \tag{8}$$

where $\mathcal{R}$ is a set of rays, and $m^l(\mathbf{r})$ and $\hat{m}^l(\mathbf{r})$ are the l-th entry of the ground-truth mask $\mathbf{M}(\mathbf{r})$ and the predicted mask $\hat{\mathbf{M}}(\mathbf{r})$, respectively.

3.3.2 Ray-Pair RGB Loss

Segmentation errors in both 3D and 2D are more likely to occur at object boundaries. One observation is that humans usually distinguish object boundaries by the color and texture difference on the two sides. Here we introduce the Ray-Pair RGB loss, aiming to incorporate color and spatial information to improve the segmentation quality.

Given a batch of rays $\mathcal{R}$, we sample a subset of rays $\mathcal{K}$ from $\mathcal{R}$ as references. For each ray $\mathbf{r}_k \in \mathcal{K}$, we calculate the RGB similarity between $\mathbf{r}_k$ and other rays $\mathbf{r} \in \mathcal{R}$, denoted by $g(\mathbf{c}(\mathbf{r}), \mathbf{c}(\mathbf{r}_k))$, where $\mathbf{c}(\mathbf{r})$ is the rendered RGB color along $\mathbf{r}$. Next, a subset $\mathcal{S}_k$ is selected from $\mathcal{R} \setminus \mathbf{r}_k$, where for all $\mathbf{r}_s \in \mathcal{S}_k$, $g(\mathbf{c}(\mathbf{r}_s), \mathbf{c}(\mathbf{r}_k)) \geq \tau$, $\tau \in \mathbb{R}$ is a threshold. The RGB loss is defined as

$$\mathcal{L}_{RGB}(\mathcal{R}) = \frac{1}{|\mathcal{K}|} \sum_{\mathbf{r}_k \in \mathcal{K}} \frac{1}{|\mathcal{S}_k|} \sum_{\mathbf{r}_s \in \mathcal{S}_k} f(\hat{\mathbf{M}}(\mathbf{r}_k), \hat{\mathbf{M}}(\mathbf{r}_s)), \tag{9}$$

where f is a distance function of two probability vectors. This loss function encourages those rays with similar RGB colors to have similar object identity predictions. $\hat{\mathbf{M}}(\mathbf{r}_k)$ is detached from the compute graph, so gradients from $\mathcal{L}_{RGB}$ only flow through $\hat{\mathbf{M}}(\mathbf{r}_s)$. In our implementation, g, f are defined as:

$$g(\mathbf{c}_0, \mathbf{c}_1) = \|\mathbf{c}_0 - \mathbf{c}_1\|_2, \tag{10}$$

$$f(\mathbf{M}_0, \mathbf{M}_1) = \exp\left(-w \frac{\mathbf{M}_0 \cdot \mathbf{M}_1}{\max(\|\mathbf{M}_0\|_2^2,\ \|\mathbf{M}_1\|_2^2)} - \epsilon\right), \tag{11}$$

where w and ϵ are hyperparameters.

Sampling Strategy. At the beginning, only $\mathcal{L}_o$ is used to optimize the object field. Concurrently, an error map $\mathbf{E}_t$ is updated to record the difference between the rendered mask and the ground-truth mask for each training view:

$$\mathbf{E}_t(\mathbf{r}) = f(\mathbf{M}(\mathbf{r}), \hat{\mathbf{M}}(\mathbf{r})), \tag{12}$$

where f is the function in Eq. 11. In practice, the resolution of error maps is smaller than that of training images to reduce memory usage and increase update efficiency, so $\mathbf{E}_t(\mathbf{r})$ is approximated by $f(\mathbf{M}(\mathbf{r}'), \hat{\mathbf{M}}(\mathbf{r}'))$, where $\mathbf{r}'$ is the sample nearest to $\mathbf{r}$ in the low-resolution error map. After k iterations of training, we include the Ray-Pair RGB loss $\mathcal{L}_{RGB}$ in training, which is only applied on local regions sampled according to the error maps. Specifically, a pixel p from a certain viewpoint with a large error is sampled and reprojected to different viewpoints in a set $\mathcal{V}$, forming a set of pixels $\{p_v \,|\, v \in \mathcal{V}\}$ in different views. From each p_v, we cast a set of rays $R_{v,p}$ in the local $N \times N$ image patch around p_v. The entire set of rays relevant to p, denoted by $\mathcal{R}_p$, is defined as:

$$\mathcal{R}_p = \bigcup_{v \in \mathcal{V}} R_{v,p}, \tag{13}$$

on which we compute the $\mathcal{L}_{RGB}$ for p. This allows us to enforce the loss to rays in different views that are relevant to the same high-error region.

To maintain the global segmentation results while refining local regions, we combine Ray-Pair RGB loss with the loss function in Eq. 8 and adopt mixed sampling: the cross entropy loss is applied to the rays sampled gloablly while the RGB loss is only applied to certain local regions. The final loss function $\mathcal{L}$ is

$$\mathcal{L} = \mathcal{L}_o(\mathcal{R}) + \frac{1}{|\mathcal{T}|} \sum_{p \in \mathcal{T}} \mathcal{L}_{RGB}(\mathcal{R}_p), \tag{14}$$

where $\mathcal{R}$ is a set of rays sampled randomly from all training views, and $\mathcal{T}$ is a set of points sampled based on error maps.

4. Experiments

4.1. Datasets

We evaluate our performance on data from multiple datasets, mixed with synthetic and real-world scenes. We compare the masks projected on 2D images with ground-truth masks to evaluate our results quantitatively. For data without ground-truth, we manually annotate object masks. We categorize the dataset used into the following 5 groups:

- Mip-NeRF 360: a dataset widely used in NeRF research including synthetic and real-world examples. In our experiments, we use the data in [3].
- LERF: a set of scenes captured by [30], which contains complex real-world samples.
- LLFF: first used in [40], the dataset contains scenes with only front views. We use the masks released with [21].
- 3D-FRONT: a synthetic indoor scene dataset created in [20], further curated for NeRF training and scene understanding in Instance-NeRF [38].
- Others: the rest of our evaluation set is composed of the data used in Panoptic Lifting [47] and Contrastive Lift [5]. The former uses scenes from existing datasets like Hypersim [46], Replica [49] and ScanNet [17], while the latter created a new dataset called Messy Rooms.

For each group above, we select scenes with representative objects and segmentation outcomes that can be clearly identified by humans. In total, 24 scenes are selected, each containing 1 to 3 object segmentation. For those without ground-truth masks, we use SAM and CascadePSP [15] with manual annotation to create ground-truth.

4.2. Metrics

We evaluate the segmentation performance of SANeRF-HQ using mean Intersection over Union (mIoU) and Accuary (Acc). For each object, we render the object masks in several novel test views, then combine the pixels in all test views to calculate the per-object IoU and Acc. The overall mIoU and Acc are averaged over all objects.

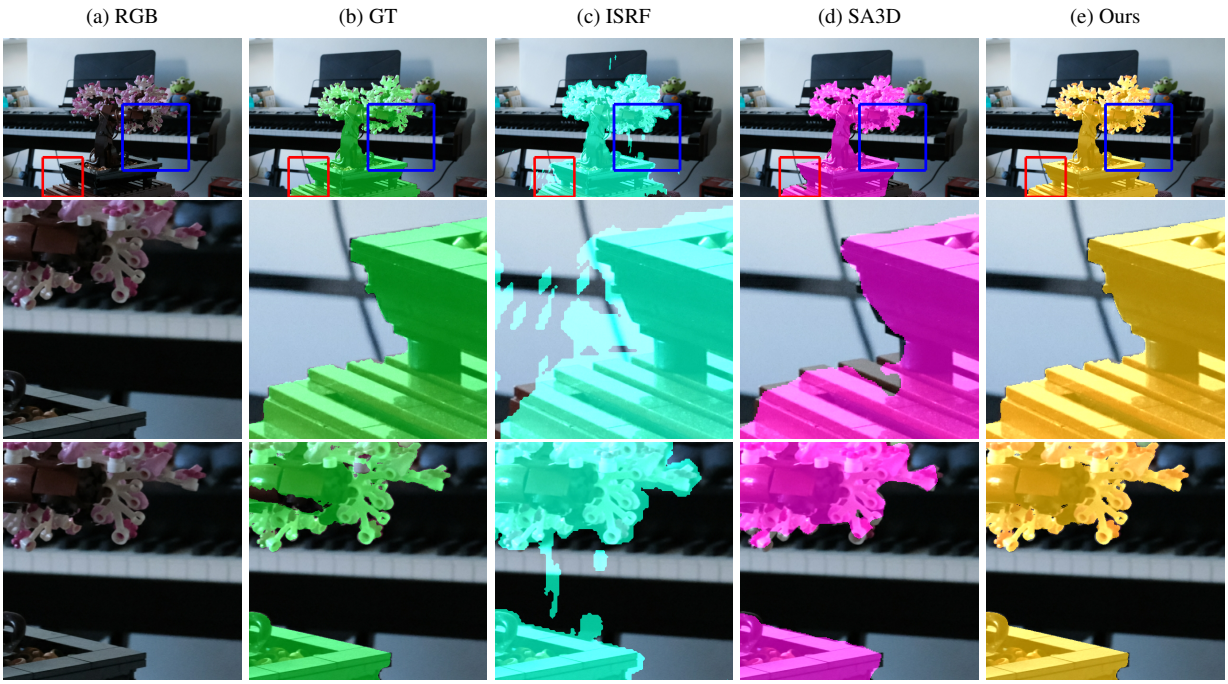

Figure 3. **Comparison with SA3D and ISRF on the Bonsai.** SANeRF-HQ can produce accurate segmentation around boundaries.

Figure 4. **Comparison with SA3D and ISRF on the Table.** SANeRF-HQ can preserve structure details of the table.

4.3. Comparison

We provide the comparison with some zero-shot segmentation methods mentioned in Section 2.2. These methods leverages large vision models such as SAM [32] or DINO [9] and can achieve zero-shot segmentation on general scenes when user prompts are provided. Table 1 shows the quantitative comparison with 4 methods on 5 different datasets mentioned in Section 4.1. We use point prompts as they are more unequivocal.

To ensure fair comparison and eliminate bias in prompt-

Figure 5. **Ablation Study on the Mask Aggregator.** The red points in the RGB images represent the prompts we use in the experiments. By leveraging the 3D geometry, SANeRF-HQ can produce more accurate segmentation (the first two rows). Moreover, our method can maintain the consistency since multi-view information is fused in the object field (the last two rows).

Methods	Mip-NeRF 360		LERF		LLFF		3D-FRONT		Others	
	Acc.↑	mIoU.↑	Acc.↑	mIoU.↑	Acc.↑	mIoU.↑	Acc.↑	mIoU.↑	Acc.↑	mIoU.↑
SA3D	99.0	88.8	94.7	52.3	98.7	90.6	97.3	78.7	**99.6**	88.8
ISRF	95.2	65.7	88.5	27.9	96.7	80.0	92.4	68.5	86.5	23.6
SAM	97.9	80.4	98.0	82.9	99.1	93.7	97.4	77.7	99.2	83.6
SAN	97.6	77.2	98.1	71.0	96.7	83.0	97.0	76.8	98.4	73.0
Ours	**99.2**	**91.0**	**99.0**	**90.7**	**99.3**	**95.2**	**98.6**	**89.9**	**99.6**	**91.1**

Table 1. **Quantitative Results on Different Datasets.**

ing, we use the same point prompts to get the initial masks for SA3D [10]. In addition, since SA3D requires a single-view mask as input, we manually select a view containing the major component of the target object, and pick the mask that best matches the ground-truth, instead of using the predicted scores from SAM to select the initial mask. To our best extent, we make sure that SA3D is provided with good initialization. For ISRF [32], strokes are used as prompts, so we manually connect the point prompts to create strokes. Figure 3 and 4 demonstrate the qualitative comparison. SA3D uses a self-prompting strategy and iteratively inverse renders the 2D masks to a voxel grid, whereas our method uses a set of global prompts and collectively optimize the object field. Despite the use of IoU rejection, self-prompting may incorrectly include occluded regions in novel views into prompts, which can accumulate errors, especially in initial iterations. ISRF lifts DINO features [9] into a neural field, but its clustering and searching process produces less accurate mask boundaries compared to SAM.

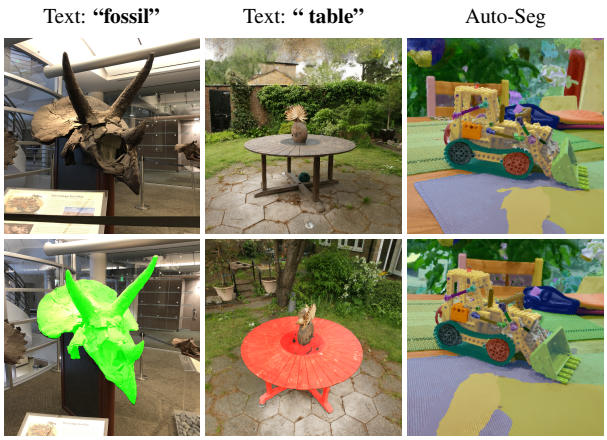

Figure 6. **Qualitative Results of text prompts and auto segmentation.**

4.4. More Qualitative Results

Figure 6 illustrates the qualitative results of our methods on other segmentation tasks. Utilizing Grounding-DINO [37], our method is also capable of segmenting objects based on text prompts. Additionally, our method can perform automatic 3D segmentation, utilizing the video rendered by NeRF, the auto-segmentation function of SAM, and incorporating [16] into the mask decoder. These results show the potential of our method in readily extending to *dynamic* NeRF.

4.5. Ablation Study

Mask Aggregator. Based on point prompts, we perform ablation study on the mask aggregator, comparing the intermediate results from the feature container, i.e., directly decoding the SAM features from the encoder or the feature field, to the final outputs of the mask aggregator. The quantitative results are in Table 1.

Directly propagating point prompts between different views and applying SAM cannot guarantee cross-view consistency. When prompts are sparse, some 3D point prompts may be occluded at certain viewpoints, and no mask can be produced. Even when a large number of prompts are provided, SAM masks may still cover different objects across views, and this naive approach fail to utilize masks from other views to collectively refine the results. The same issue also exists in SAN [10], which distills SAM features with another neural field and later decodes the rendered features using the decoder. In addition to the consistency issue, the rendered feature maps from SAN further suffer from aliasing, and fine spatial semantics in the SAM features along object boundaries can be lost during interpolation. This can lead to less accurate segmentation results on the object boundaries, like jagged mask edges. To ensure fairness in comparison, we introduce enough point prompts to guarantee masks can be generated from every viewpoint where the target object is visible.

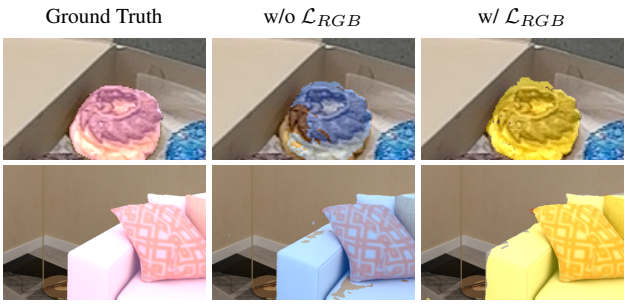

Figure 7. **Qualitative Results of Ray-Pair RGB Loss.** The Ray-Pair RGB loss can help to recover local regions and make the results more solid.

Metrics	$\mathcal{L}_{RGB}$	Mip-NeRF	LERF	LLFF	3D-FRONT	Others
mIoU.↑	w/o	91.0	88.3	95.2	89.3	90.9
	w/	**91.3**	**90.7**	**95.8**	**89.9**	**91.1**
Acc.↑	w/o	**99.2**	98.9	99.4	98.6	**99.6**
	w/	**99.2**	**99.0**	**99.5**	**98.7**	**99.6**

Table 2. **Ablation Results of the Ray-Pair RGB Loss.**

Figure 5 illustrates the qualitative comparison results. Although SANeRF-HQ uses the potentially inconsistent segmentation from SAM as input, by aggregating multi-view information and integrating 3D geometry captured by NeRF, it can still produce a underlying 3D mask close to the ground-truth geometry, which guarantees consistent multi-view masks and usually comes with higher quality.

Ray-Pair RGB Loss. We perform ablations on the Ray-Pair RGB loss in Section 3.3.2. Table 2 shows the quantitative results, where the Ray-Pair RGB loss slightly enhances the mask quality. The visual improvement of the masks illustrated in Figure 7 is more significant. The loss helps fill the missing interior and boundaries of the masks by enforcing a local match between the similarity in labels, and the similarity in appearance.

For more ablation study, please refer to our supplementary materials.

5. Conclusion

In this paper, we introduced the Segment Anything for NeRF in High Quality (SANeRF-HQ) framework. By combining the strengths of the Segment Anything Model (SAM) for open-world object segmentation and NeRF for aggregating information from multiple viewpoints, SANeRF-HQ represents a significant advancement in high-quality 3D segmentation. Our method was quantitatively and qualitatively evaluated on various NeRF datasets, which demonstrates SANeRF-HQ's advantages over previous state-of-the-art methods. Furthermore, we demonstrate the potential of extending our work to 4D dynamic NeRF object segmentation (see supplementary materials). SANeRF-HQ holds promise for contributing significantly to the evolving landscape of 3D computer vision and segmentation techniques.

References

[1] Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O'Toole, and Changil Kim. HyperReel: High-fidelity 6-DoF video with ray-conditioned sampling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 12

[2] 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. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[4] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2

[5] Yash Bhalgat, Iro Laina, João F Henriques, Andrew Zisserman, and Andrea Vedaldi. Contrastive lift: 3d object instance segmentation by slow-fast contrastive fusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2, 5, 12

[6] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[7] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact++: Better real-time instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 2

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, pages 213–229. Springer, 2020. 2

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 6, 7

[10] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Chen Yang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, and Qi Tian. Segment anything in 3d with nerfs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1, 2, 7, 8

[11] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[12] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision (ECCV)*, pages 801–818, 2018. 2

[13] Xiaokang Chen, jiaxiang Tang, Diwen Wan, Jingbo Wang, and Gang Zeng. Interactive segment anything nerf with feature imitation. *arXiv preprint arXiv:2211.12368*, 2023. 1, 2, 3

[14] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *arXiv*, 2021. 1, 2

[15] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. CascadePSP: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5

[16] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1316–1326, 2023. 2, 8

[17] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[18] Daan De Geus, Panagiotis Meletis, and Gijs Dubbelman. Panoptic segmentation with a joint semantic and instance segmentation network. *arXiv preprint arXiv:1809.02110*, 2018. 2

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[20] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10933–10942, 2021. 5

[21] Rahul Goel, Dhawal Sirikonda, Saurabh Saini, and P.J. Narayanan. Interactive Segmentation of Radiance Fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 3, 5

[22] Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:1140–1156, 2022. 2

[23] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 2

[24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, 2022. 2

[25] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[26] Benran Hu, Junkai Huang, Yichen Liu, Yu-Wing Tai, and Chi-Keung Tang. Nerf-rpn: A general framework for object detection in nerfs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[27] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. 2

[28] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[29] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 12

[30] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 3, 5

[31] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6399–6408, 2019. 2

[32] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1, 2, 6, 7

[33] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1, 2

[34] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5521–5531, 2022. 13

[35] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7026–7035, 2019. 2

[36] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, Tong Lu, and Ping Luo. Panoptic segformer: Delving deeper into panoptic segmentation with transformers, 2021. 2

[37] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 8

[38] Yichen Liu, Benran Hu, Junkai Huang, Yu-Wing Tai, and Chi-Keung Tang. Instance neural radiacne field. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 5, 12

[39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021. 2

[40] 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 TOG*, 2019. 5

[41] 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 (ECCV)*, 2020. 1, 2, 3

[42] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 2, 12

[43] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[44] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2

[45] Frano Rajič, Lei Ke, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Segment anything meets point tracking. *arXiv preprint arXiv:2307.01197*, 2023. 2

[46] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 5

[47] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3d scene understanding with neural fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9043–9052, 2023. 1, 2, 5, 12

[48] Karl Stelzner, Kristian Kersting, and Adam R Kosiorek. Decomposing 3d scenes into objects via unsupervised volume segmentation. *arXiv:2104.01148*, 2021. 2

[49] 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 preprint arXiv:1906.05797*, 2019. 5

[50] Jiaxiang Tang. Torch-ngp: a pytorch implementation of

instant-ngp, 2022. https://github.com/ashawkey/torch-ngp. 12

[51] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *European Conference on Computer Vision (ECCV)*, pages 282–298. Springer, 2020. 2

[52] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:10078–10093, 2022. 2

[53] Khoi Nguyen Tuan Duc Ngo, Binh-Son Hua. Isbnet: a 3d point cloud instance segmentation network with instance-aware sampling and box-aware dynamic convolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[54] Thang Vu, Kookhoi Kim, Tung M. Luu, Xuan Thanh Nguyen, and Chang D. Yoo. Softgroup for 3d instance segmentation on 3d point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[55] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. SOLO: Segmenting objects by locations. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[56] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[57] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14668–14678, 2022. 2

[58] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8818–8826, 2019. 2

[59] Hong-Xing Yu, Leonidas J. Guibas, and Jiajun Wu. Unsupervised discovery of object radiance fields. In *ICLR*, 2022. 2

[60] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 558–567, 2021. 2

[61] Xiaoyu Yue, Shuyang Sun, Zhanghui Kuang, Meng Wei, Philip HS Torr, Wayne Zhang, and Dahua Lin. Vision transformer with progressive sampling. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 387–396, 2021. 2

[62] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-Net: Towards unified image segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[63] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017. 2

[64] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6881–6890, 2021. 2

[65] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[66] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *arXiv preprint arXiv:2304.06718*, 2023. 2

A. Implementation Details

We use torch-ngp [50] as our initial NeRF implementation. When we use 3D points as prompts in evaluation, the views containing less than k visible points get filtered out automatically and will not be used to train the object field, where k is a hyperparameter, depending on the total number of input points.

For both the SAM feature field and the object field, we use a hash grid as in [42] with 16 levels and feature dimension of 8 per level. The lowest and highest level are of resolution 16 and 2^{19}, respectively. We use a 5-layer 256-hidden dimensional MLP with skip connections and Layer Normalization after the feature field hash grid, and a 3-layer 256 hidden dimensional MLP with skip connections after the object field hash grid. In addition to the features from their respective hash grid, both MLPs also take the features from the density field as input, where feature MLP also takes the viewing directions as input. The initial radiance and density field, the SAM feature field, and the object field are trained for 15,000, 5,000, and 600 iterations, respectively. All models are trained on an NVIDIA RTX 4090 GPU.

Ray-Pair RGB loss is included after 300 iterations of warm-up. We use error maps downsampled by 4 times compared to original training images for Ray-Pair RGB loss sampling. In each iteration, we update the error maps using the training ray batch, and for every 200 iterations, we perform a full update for all error map pixels. During sampling, we independently sample initial rays on each error map weighted by their errors, reproject them onto each view, and subsequently sample 32 additional rays in each $N \times N$ patch centered at the reprojected pixels randomly. Here we choose $N = 8$ or 16. A subset of 20 rays are then sampled from each set as references in the Ray-Pair RGB loss.

B. More Ablation Study

SAM Backbones. We perform ablation study on different types of backbones for feature extracting and decoding, which are SAM and SAM-HQ [29].

Feature Container Method. We propose two methods to store the pre-computed features in feature container Section 3.1. The first one is to store the feature maps in a cache. The second one is to distill the features into a feature field and render the feature map during inference. We investigate the performance of these two methods on our evaluation set.

Table C.3 demonstrates our the quantitative results on the ablation study. SAM-HQ with a cache container performs the best in most cases, while SAM-HQ with a distillation container does not produce similar results as others on our evaluation set. Nevertheless, our method demonstrates advantages over other 3D segmentation methods in general. Notice that we do not choose the best results from all of them in our comparison to ensure the fairness. Considering that all the previous methods leverage SAM, we use SAM with feature distillation as our feature container in comparison.

Metrics	Ours	Instance-NeRF
Acc.↑	98.7	99.2
mIoU.↑	89.9	92.8

Table C.1. Comparison with Instance-NeRF on 3D-FRONT.

Metrics	Ours	Panoptic Lifting	Contrastive Lift
Acc.↑	99.6	94.3	94.1
mIoU.↑	91.1	84.5	81.5

Table C.2. Comparison with Panoptic Lifting and Contrastive Lift on the data mentioned in their papers.

C. Comparison with Instance Segmentation Methods

We also compare our method with some instance segmentation methods. The instance segmentation methods in NeRF mentioned in our related works do not require user prompts and can automatically generate segmentation of salient objects in NeRF. These methods also leverage 2D segmentation methods for NeRF training but they mainly focus on the challenge of 3D consistency. Despite their different configurations and issues of concern, we still provide the comparison with these automatic end-to-end pipelines, showing that our prompt-based method can produce comparable results to these state-of-the-art auto-segmentation methods. Instance-NeRF [38] is a training-based methods so we only compare with it on 3D-FRONT dataset. Figure C.1 and Table C.1 illustrates the visual results and quantitative comparison respectively. For Panoptic Lifting [47] and Contrastive Lift [5], we also compare on the scenes they mentioned in the papers to ensure the fairness. Results are shown in Figure C.2 and Table C.2.

We use the objects in our evaluation sets as targets and choose the object that has the largest IoU with the target object as the predicted results of the instance segmentation methods. Notice that we only compare with those methods on the datasets mentioned in their papers, since they do not leverage SAM to achieve zero-shot generalization.

D. Extending to Dynamic NeRFs

We present a preliminary demonstration in Figure D.1 on the easy extension of our method to 4D dynamic NeRF representations. We use HyperReel [1] as our reference NeRF representation and only supply user prompts for the first frame of each camera. The prompts are fed into SAM to retrieve initial masks, whose bounding boxes are used as

Backbone	Container	Mip-NeRF 360		LERF		LLFF		3D-FRONT		Others	
		Acc.↑	mIoU.↑	Acc.↑	mIoU.↑	Acc.↑	mIoU.↑	Acc.↑	mIoU.↑	Acc.↑	mIoU.↑
SAM	Cache	99.3	93.6	98.9	90.3	99.5	95.8	97.6	84.8	**99.6**	**91.1**
	Distillation	99.2	91.0	99.0	90.7	99.3	95.2	98.6	89.9	**99.6**	**91.1**
SAM-HQ	Cache	**99.4**	**94.4**	**99.2**	**93.1**	**99.6**	**96.8**	**98.9**	**91.8**	99.5	90.1
	Distillation	98.9	88.1	98.0	86.3	98.8	89.8	96.3	79.4	99.4	89.4

Table C.3. Quantitative Results on different backbones and feature containers.

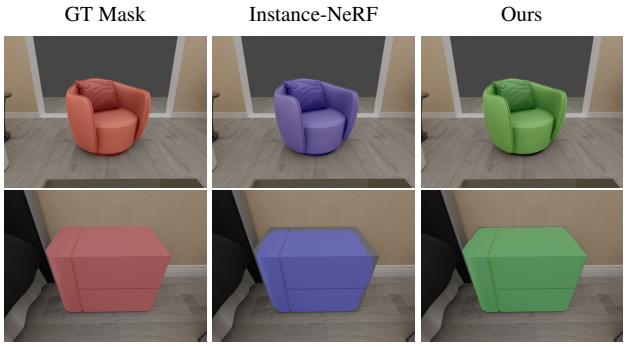

Figure C.1. **Qualitative Comparison with Instance-NeRF.** Zoom in for details especially along the segmentation boundaries.

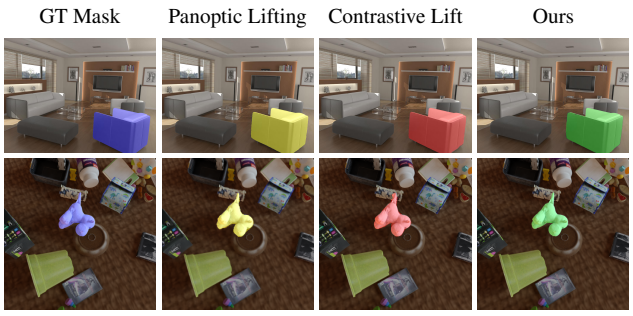

Figure C.2. **Qualitative Comparison with Panoptic Lifting and Contrastive Lift**. Zoom in for details.

the prompts for the next frame. This process repeats until masks are acquired from all video frames, after which we proceed to object field training as in previous static scene cases. The scene is from the Neural 3D Video dataset [34].

E. More Qualitative Results

We demonstrate extra qualitative comparisons between our method and other zero-shot 3D segmentation methods as mentioned in the main paper. The results are given in Figures E.1, E.2, E.3, E.4. Please watch the video for more qualitative results.

(a) $t = 0s$ (b) $t = 0.25s$ (c) $t = 0.5s$ (d) $t = 0.75s$ (e) $t = 1s$ (f) $t = 1.25s$ (g) $t = 1.5s$

Figure D.1. **Demonstration of Applying SANeRF-HQ to Dynamic NeRFs.** The first row are the NeRF RGB images over time, and the second row are the masks from SANeRF-HQ, which is also dynamic. Our method can be easily adapted to dynamic NeRFs and still retains reasonable performance. The implementation is based on HyperReel, and the *cook spinach* scene shown is from the Neural 3D Video dataset.

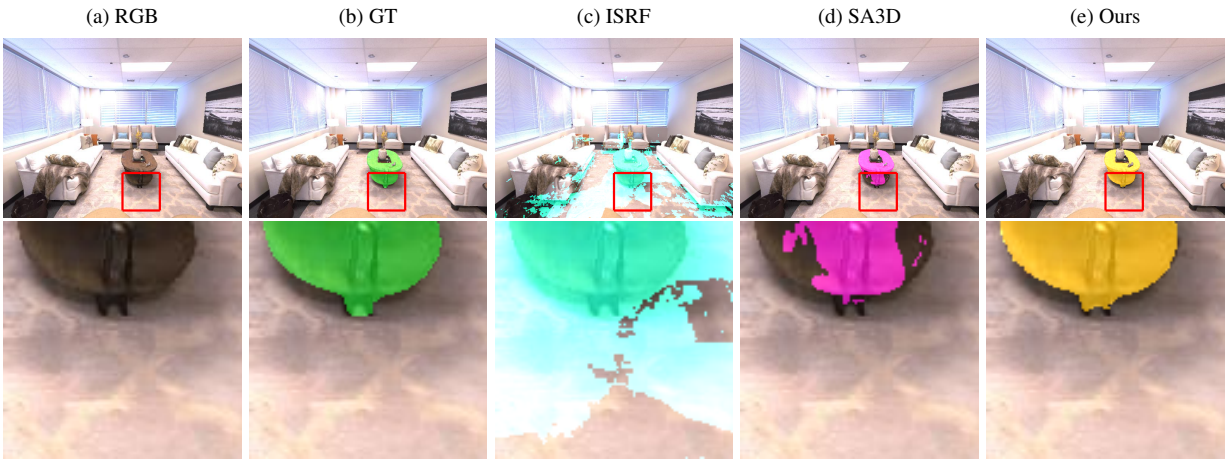

Figure E.1. **Comparison with SA3D and ISRF on the Replica Room.** Data is from the Others subset. SANeRF-HQ can maintain the object structure while excludes the background.

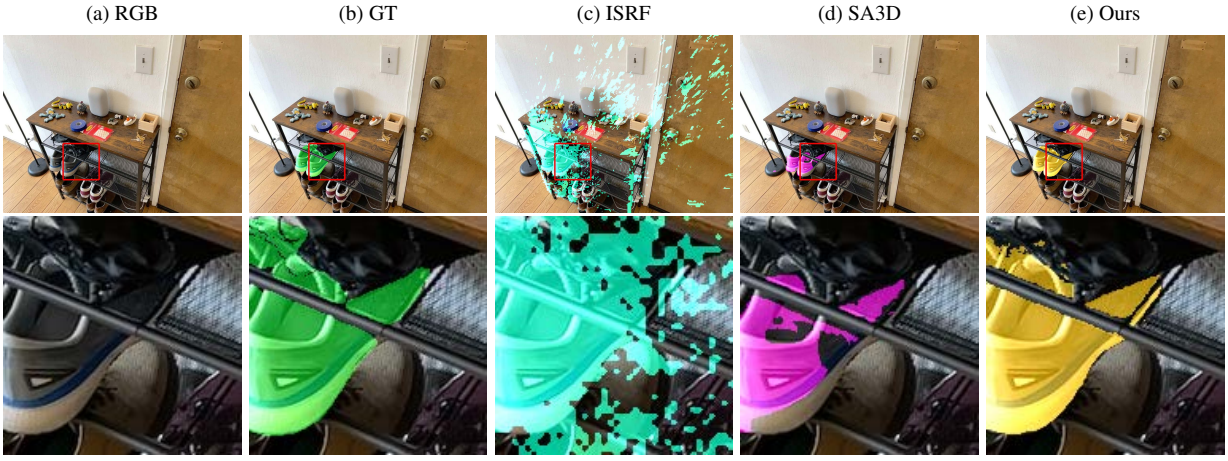

Figure E.2. **Comparison with SA3D and ISRF on the Shoe Rack.** Data is from the LERF subset. Our method can reproduce the segmentation details even with some occlusion.

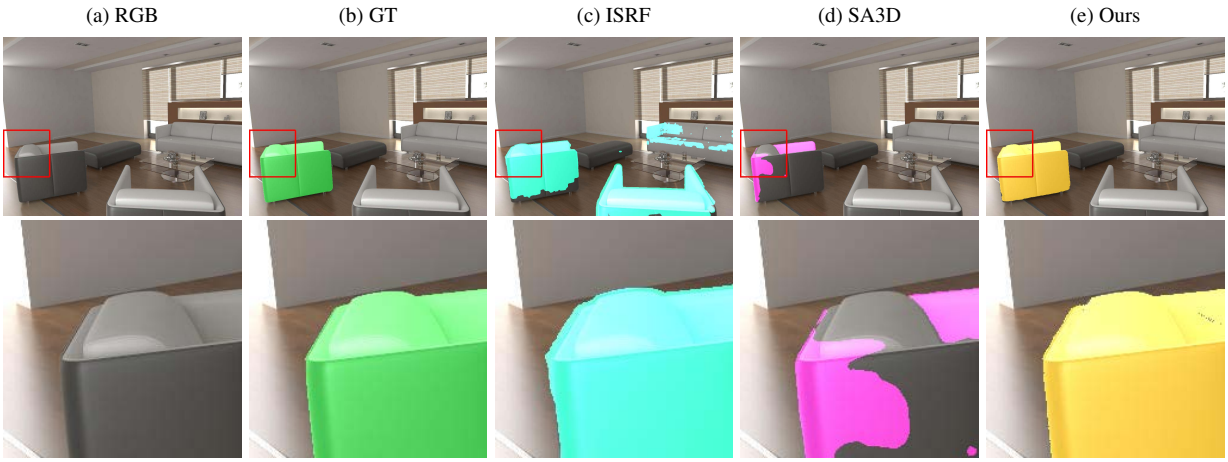

Figure E.3. **Comparison with SA3D and ISRF on the Hypersim.** Data is from the Others subset. ISRF contains too many false positives, while SA3D cannot cover the whole object.

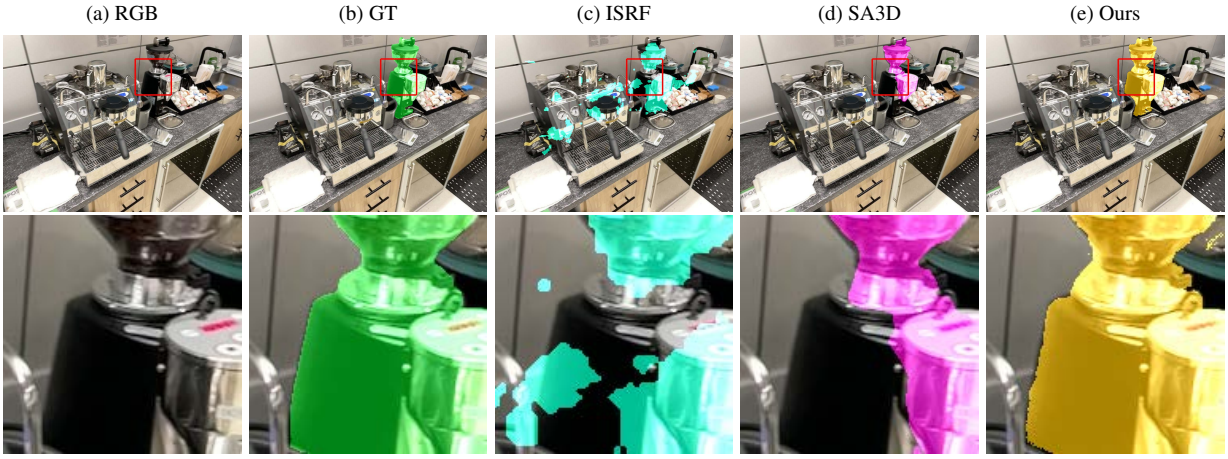

Figure E.4. **Comparison with SA3D and ISRF on the Espresso.** Data is from the LERF subset. Our method produces the most reasonable segmentation in the distant, complex setting.