

# Interactive Segment Anything NeRF with Feature Imitation

XIAOKANG CHEN, School of Intelligence Science and Technology, Peking University

JIAXIANG TANG, School of Intelligence Science and Technology, Peking University

DIWEN WAN, School of Intelligence Science and Technology, Peking University

JINGBO WANG, The Chinese University of Hong Kong

GANG ZENG, School of Intelligence Science and Technology, Peking University



Fig. 1. Our pipeline allows click- or text-based user interaction to perform zero-shot semantic segmentation in 3D space. We further investigate single-object mesh extraction by projecting semantic masks onto mesh surface, leading to applications like texture editing and model composition.

This paper investigates the potential of enhancing Neural Radiance Fields (NeRF) with semantics to expand their applications. Although NeRF has been proven useful in real-world applications like VR and digital creation, the lack of semantics hinders interaction with objects in complex scenes. We propose to imitate the backbone feature of off-the-shelf perception models to achieve zero-shot semantic segmentation with NeRF. Our framework reformulates the segmentation process by directly rendering semantic features and only applying the decoder from perception models. This eliminates the need for expensive backbones and benefits 3D consistency. Furthermore, we can project the learned semantics onto extracted mesh surfaces for real-time interaction. With the state-of-the-art Segment Anything Model (SAM), our framework accelerates segmentation by 16 times with comparable mask quality. The experimental results demonstrate the efficacy and computational advantages of our approach. Project page: <https://me.kiui.moe/san/>.

Additional Key Words and Phrases: NeRF, Interactive 3D Segmentation

## 1 INTRODUCTION

Neural Radiance Fields (NeRF) [45] has recently garnered significant attention as a promising method for synthesizing photo-realistic images of complex 3D environments. By providing a theoretically sound approach to scene reconstruction from 2D images, NeRF

demonstrates the potential to bridge the gap between captured images and the corresponding 3D world with more comprehensive information. Despite their strengths, the output of NeRFs remains limited to geometry and appearance, devoid of any explicit semantic information. This lack of interpretability can hamper the development of flexible applications aimed at interacting with content in reconstructed scenarios, such as editing the appearance of specific objects in a complex 3D environment or extracting the mesh of this object.

In this paper, we address the challenge of incorporating explicit semantic information into the otherwise purely geometric representation of reconstructed scenes provided by Neural Radiance Fields. The goal is to enhance the flexibility and potential of NeRF for interactive applications by leveraging off-the-shelf semantic models, such as pre-trained large-scale perception [32, 79] and language models [29], to infuse semantically meaningful information into their output. To this end, we propose to use segmentation mask-based approaches that enable pixel-wise classification of objects in the rendered images of the reconstructed scene, and thus allow for object-level human scene interactions. As shown in Figure 1, our method demonstrates the effectiveness of using pre-existing

\*X. Chen and J. Tang have made equal contributions in the technical aspect. D. Wan and J. Wang have made equal contributions in the writing. The authorship order is determined based on the alphabetical order of the authors' names.

semantic knowledge to enhance the output of NeRF with semantics, thereby improving their applicability in complex real-world scenarios.

Our proposed framework is built upon the grid-based NeRF [46, 55, 60] representation, which has gained significant popularity across various fields [22, 30, 36]. Although one plausible method for NeRF to cooperate with off-the-shelf perception and large language models is to extract semantics directly from the rendered images, this approach can often be computationally prohibitive for flexible interactive applications, due to the heavy-weight backbones associated with such models. To overcome this challenge, we introduce a novel approach that is known as semantic feature imitation processing. This process enables direct rendering of semantic-aware features similar to rendering colors in NeRF. By leveraging this process, the pretrained NeRF model can learn and incorporate meaningful semantic patterns from the output of the semantic model backbones, allowing it to accurately segment rendered images using lightweight decoders of the semantic models efficiently. For instance, with SAM [32], which is the state-of-the-art image segmentation model, our framework could accelerate the segmentation process by 16 times with comparable mask quality. By leveraging this innovative approach, our framework unlocks tremendous performance potential for powering real-time interactive applications with ease.

Our proposed framework offers several key advantages. First, it eliminates the need for expensive segmentation backbones, resulting in a significant speed up of segmentation processing and facilitating human-scene interaction. Second, our semantic imitation module is pluggable and independent of the original NeRF module, thus preserving the rendering quality without compromise. Finally, our approach is model-agnostic to both NeRF and perception models, enabling it to seamlessly integrate with advanced models in the future. Overall, our method presents a practical and flexible solution for enhancing NeRF-based real-world applicability and effectiveness.

## 2 RELATED WORK

### 2.1 NeRF for 3D representation

Neural Radiance Fields (NeRF) [45] has gained significant attention and led to rapid progress for photo-realistic novel view synthesis. Various works are proposed to enhance different aspects of NeRF, like improving the quality of rendering image [1, 3, 65], decreasing training and inference time [24, 46, 52], expanding the applicable range [2, 50, 56], and generalizing with few-shot settings [72, 74]. NeRF has also been widely used in a variety of applications, including 3D-aware image generation [48, 59, 70], text-to-3D generation [36, 49], and 3D shape generation [28, 43], pose estimation [4, 37]. Our work aims to enhance semantic understanding ability of NeRF, especially on user interaction.

### 2.2 2D Semantic Understanding

Amazing progress has been made in image semantic understanding in the past few years. Detection Transformer (DETR) [6] and follow-ups [7–10, 12, 16, 27, 34, 42, 76] employ transformer-based architecture [63] and have made significant advance in semantic understanding. Combined with large-scale vision-language pretraining models like CLIP [29], open-vocabulary segmentation methods [21,

26, 51, 67, 68] can perform segmentation from human language prompts. To facilitate user interactions, some works [14, 39, 69] propose to use strokes or clicks as input for segmentation. More recently, Segment Anything Model (SAM) [32] trains on large-scale datasets and achieves strong zero-shot performance given various visual prompts like clicks or boxes. X-Decoder [79] proposes a unified approach to support various types of segmentation and vision-language tasks including open-vocabulary segmentation. SEEM [80] further includes visual and audio prompts into a joint visual-semantic space and enables composition of different types of prompts. However, these methods usually rely on heavy-weight segmentation backbones to extract semantic features from images, which slows down the inference speed. We propose to bypass the segmentation backbones with our feature imitation module, thus accelerating various semantic tasks in 3D space.

### 2.3 3D Semantic Understanding

Compared to semantic understanding in 2D images, 3D semantic understanding is more complex. Existing methods [11, 13, 17, 25, 57, 58, 64, 71, 78] mainly focus on closed set segmentation where scenes are represented by point clouds or voxels. Recently, some works explore to apply NeRF in 3D semantic understanding, including object segmentation [19, 35, 40, 62], panoptic segmentation [20, 54], 3D semantic segmentation [77], part segmentation [75], text-based segmentation [30, 33], and interactive segmentation [22, 53, 61]. In particular, NVOS [53] segments objects in neural volumetric representations using positive and negative user strokes inputs. ISRF [22] also uses user strokes as input and distills 2D semantic features from a large self-supervised pretrained model to NeRF and use nearest neighbor feature matching to segment objects. By fusing CLIP embedding into a NeRF, LERF [30] supports using human language to localize a wide variety of queries in 3D scenes. Our work is able to perform both click- and text-based 3D segmentation using different 2D perception backbones, while also being much faster compared to previous works and achieving real-time interaction.

## 3 PRELIMINARIES

### 3.1 NeRF

The Neural Radiance Fields (NeRF) approach, introduced by Mildenhall et al. [45], employs a 5D function  $f_\Theta$  to depict a 3D volumetric scene. This function takes a 3D coordinate  $\mathbf{x} = (x, y, z)$  and a 2D viewing direction  $\mathbf{d} = (\theta, \phi)$  as inputs, and outputs a volume density  $\sigma$  with an emitted color  $\mathbf{c} = (r, g, b)$ . For a ray  $\mathbf{r}$  that starts at  $\mathbf{o}$  and follows the direction  $\mathbf{d}$ , we sample points  $\mathbf{x}_i = \mathbf{o} + t_i \mathbf{d}$  along the ray in sequence, and use  $f_\Theta$  to retrieve densities  $\sigma_i$  and colors  $\mathbf{c}_i$ . The color of the pixel associated with the ray can then be estimated using numerical quadrature:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_i T_i \alpha_i \mathbf{c}_i, T_i = \prod_{j < i} (1 - \alpha_j), \alpha_i = 1 - \exp(-\sigma_i \delta_i), \delta_i = t_{i+1} - t_i \quad (1)$$

Here,  $\delta_i$  denotes the step size,  $\alpha_i$  is the opacity, and  $T_i$  is the transmittance. The process of volume rendering is differentiable, which allows NeRF to be optimized using only 2D image supervision.

This is done by minimizing the L2 difference between the predicted color of each pixel  $\hat{C}(\mathbf{r})$  and the actual color  $C(\mathbf{r})$  from the image:

$$\mathcal{L}_{\text{NeRF}} = \sum_{\mathbf{r}} \|C(\mathbf{r}) - \hat{C}(\mathbf{r})\|_2^2 \quad (2)$$

### 3.2 Mesh Extraction from NeRF

The implicit volumetric representations of NeRF differ significantly from the widely-adopted polygonal meshes and lack support from common 3D software and hardware, making their rendering and manipulation inefficient. Recent works [15, 47, 60, 73] explore using surface meshes for accelerated rendering. For example, NVdiffeq [47] uses differentiable rendering to optimize a tetrahedron grid for geometry, while the texture can be encoded with a multi-scale hashgrid [46]. NeRF2Mesh [60] further extends mesh representation into unbounded scenes by optimizing a coarse mesh extracted from NeRF. Our method is agnostic to the underlying 3D representation and can be used in such mesh-based setting too, which leads to faster RGB rendering and also enables mesh segmentation.

### 3.3 Large Perception Models

Recently, large scale vision datasets and transformer architecture have enabled training of strong perception models like SAM [32] and X-Decoder [79] for 2D dense semantic understanding. These models typically consist of an image backbone and a prompt decoder. The image backbone Enc usually adopts a heavy-weight transformer (*e.g.*, Vit-Huge [18]) to encode semantic features from input image  $I$ , while the prompt decoder Dec embeds different input prompts  $p$  such as point clicks or text descriptions to predict semantic masks  $M$ :

$$M = \text{Dec}(\text{Enc}(I), p). \quad (3)$$

## 4 METHOD

In this section, we introduce our pipeline to perform interactive 3D segmentation. First, we propose a semantic feature imitation module to replace heavy-weight segmentation backbones by directly rendering semantic features in Section 4.1. Next, we explore different loss functions for single-scale and multi-scale feature imitation in Section 4.2. Then, we discuss several key designs including camera augmentation and caching to enhance feature imitation quality in Section 4.3. Finally, we show how to perform user interaction in our graphic user interface and potential applications in Section 4.4.

### 4.1 Semantic Feature Imitation

We propose to render the semantic features from neural radiance fields directly, eliminating the need for the forward process of the segmentation backbone. The overall training procedure is shown in Figure 2. Our method is model-agnostic and could be applied to a broad range of NeRF. In implementation, we choose grid-based NeRF [46, 60] to enhance efficiency. Assume we have a trained NeRF model that predicts the density  $\sigma$  and color  $c$  at each 3D location  $x$ :

$$\sigma = \Phi(\text{MLP}(E^{\text{geo}}(x))), \quad (4)$$

$$c = \Psi(\text{MLP}(E^{\text{rgb}}(x))), \quad (5)$$

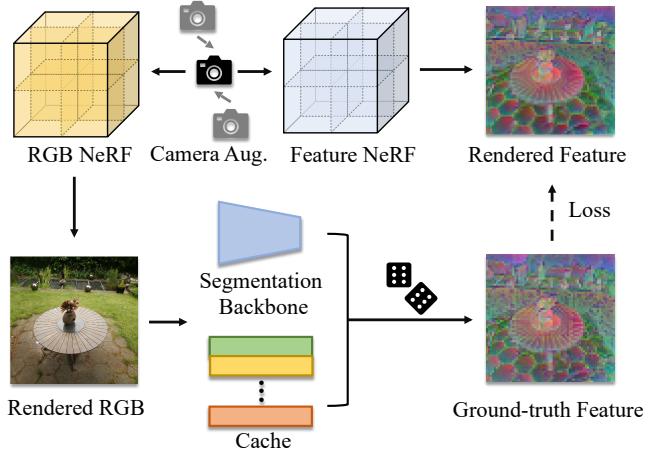


Fig. 2. **Semantic feature imitation training.** We visualize the high-dimensional semantic features by rendering the first three channels as RGB.

where  $\Phi$  is the exponential activation [46] that promotes sharper surface and  $\Psi$  refers to the sigmoid activation.  $E^{\text{geo}}$  and  $E^{\text{rgb}}$  are learnable feature grids to represent the 3D field.

Our semantic feature imitation module is totally pluggable to the original RGB and density field in NeRF, which are fixed during semantic feature imitation training. Given a camera view, we first render the RGB image  $I$  with NeRF. We then use the perception model's backbone to extract semantic features  $F = \text{Enc}(I) \in \mathbb{R}^{C \times h \times w}$ , where  $C$  is the feature dimension and  $h, w$  are the feature height and width (usually smaller than image resolution  $H, W$ ). These 2D feature maps can be used to supervise 3D feature grid just like the RGB information. We reuse the density information and render the semantic feature along each ray  $r$  using numerical quadrature:

$$\hat{F}(r) = \text{MLP}\left(\sum_i T_i \alpha_i E^{\text{sem}}(x_i)\right) \quad (6)$$

where  $\hat{F}(r)$  is the imitated feature and  $E^{\text{sem}}$  is the learnable semantic feature grid. As the feature channels  $C$  are usually much higher than 3-channel RGB, we move the non-linear MLP after performing quadrature following [23, 52]. Also, we perform feature rendering directly at the smaller feature resolution  $h \times w$  after RGB rendering, which further saves the computation. In cases when we can extract a mesh surface from the NeRF [60], we only need to sample one surface point  $x_s$  per ray and simplify Equation 6 to:

$$\hat{F}(r) = \text{MLP}(E^{\text{sem}}(x_s)) \quad (7)$$

Different perception models may require different number of features maps for the decoder. For single-scale decoder as used in SAM [32], we only need to imitate one feature map  $F$ . Other models like X-Decoder [79] use multi-scale feature maps  $\{F_i\}, i \in [0, 3]$ . In such cases, we share the feature grid  $E^{\text{sem}}$  and use different heads  $\text{MLP}_i, i \in [0, 3]$  to predict multi-scale features, which helps to exploit the cross-scale correlation and reduce total parameters.

During inference, we render all the necessary semantic feature maps  $\hat{F}$  of the test image, and apply the perception model's decoder to generate mask predictions  $\hat{M} = \text{Dec}(\hat{F})$ .

## 4.2 Loss Function

**Single-Scale Decoder.** For single-scale decoder models, we directly minimize the MSE loss between the rendered semantic features  $\hat{\mathbf{F}}$  and the ground-truth semantic features  $\mathbf{F}$ :

$$\mathcal{L}_{\text{single}} = \text{MSE}(\hat{\mathbf{F}}, \mathbf{F}) = \frac{1}{N} \sum_{\mathbf{r}} \|\mathbf{F}(\mathbf{r}) - \hat{\mathbf{F}}(\mathbf{r})\|_2^2 \quad (8)$$

where  $N$  is the number of rays per training step.

**Multi-Scale Decoder.** For multi-scale decoder models, the feature maps at different scales are usually correlated to represent the same underlying image [38, 41]. Based on this observation, we introduce an additional loss term called the cross-scale correlation loss. This loss term promotes to learn the correlation between semantic features at different scales and aids in capturing contextual information. We first calculate the cosine similarity map between the  $i$ -th and the  $j$ -th multi-scale feature maps and obtain a similarity maps  $S_{ij} \in \mathbb{R}^{C \times s_i \times s_j}$ , where  $s_i, s_j$  represent the number of pixels in the two feature maps. This cosine similarity map can then be used as an extra supervision:

$$\mathcal{L}_{\text{multi}} = \mathcal{L}_{\text{single}} + \mathcal{L}_{\text{cross}} \quad (9)$$

$$\mathcal{L}_{\text{single}} = \sum_i \text{MSE}(\hat{\mathbf{F}}_i, \mathbf{F}_i) \quad (10)$$

$$\mathcal{L}_{\text{cross}} = \sum_i \sum_{j > i} \text{MSE}(\hat{S}_{ij}, S_{ij}) \quad (11)$$

$\hat{S}_{ij}$  is the similarity map of predicted feature maps. In the experiment, we find this cross-scale correlation loss could improve feature imitation quality and accelerate convergence.

## 4.3 Training Details

**Camera Augmentation.** Since we are able to synthesize RGB images from arbitrary camera pose with the pretrained NeRF, we augment the training dataset by interpolating between the original training camera poses for feature imitation training. This technique is widely used to distill NeRF [5, 66], which shares a similar pipeline with ours. By capturing a more diverse range of viewpoints, we found that camera augmentation helps to enhance the robustness of our semantic feature imitation and results in more smooth segmentation predictions. During training, we perform on-the-fly rendering of the novel RGB images and then extract the corresponding semantic features by the preception model as supervision.

**Caching Mechanism.** The above on-the-fly feature extraction training leads to heavy burden as we run the heavy perception model per training step. To mitigate the large computational cost associated with the segmentation backbone, we employ a caching mechanism. During training, we maintain a cache that stores the camera view, RGB image, and the corresponding feature map. For the initial steps, we forward novel view images to the segmentation backbone and cache the resulting feature maps. In the following steps, we adopt a randomized approach to determine whether to use a cached feature map or sample a new feature map as the supervision. By randomly choosing between the cached feature map and the online-generated feature map, we strike a balance between reusing cached information and incorporating fresh data during training.

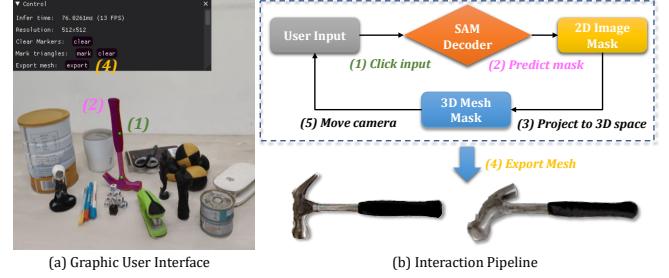


Fig. 3. **GUI and interaction pipeline.** We design a GUI that allows user interaction in real-time to perform 2D image segmentation and 3D mesh segmentation.

To maintain efficiency and manage cache memory, we make the cache obey the first-in-first-out (FIFO) rule, where the oldest entries in the cache are replaced with new ones when the cache reaches its capacity. By employing this caching mechanism, we significantly reduce the computational cost and accelerate the training, while maintaining similar performance.

## 4.4 User Interactivity

To demonstrate the effectiveness of our method, we implement a GUI for user interaction as shown in Figure 3. The GUI allows users to drag and view the 3D scene in real-time. For any camera view, the user can input a prompt (e.g., click a point on the viewport), and our model renders the 2D segmentation mask overlayed on the RGB image. Then the user can choose to project the 2D mask onto 3D mesh surfaces, so the mask can be rendered from other camera views. For click points, we also project it to 3D so it can be automatically tracked when changing the camera view. By repeating the above process from several camera views, it's easy to get the targeted object fully segmented from the 3D space. Finally, the user can export the segmented mesh and also the texture maps.

## 5 EXPERIMENTS

### 5.1 Implementation Details

**Training Setting.** We use NeRF2Mesh [60] as the framework for our NeRF training and Mesh extraction. The training of NeRF takes 10,000 steps, with each step containing approximately  $2^{18}$  points. An exponentially decayed learning rate schedule ranging from 0.01 to 0.001 is employed. A coarse mesh is then extracted and finetuned for additional 5,000 steps. Then, the training of feature grid takes 10,000 steps. At each step, we have 75% chance to use a cached camera view, and 25% chance to sample new camera view. The cache size is set to 256. We use the Adam [31] optimizer in all stages. All experiments are conducted on a single NVIDIA V100 GPU.

**Datasets.** We choose the widely used Mip-NeRF 360 [2] dataset and LLFF [44] dataset to evaluate our method. Mip-NeRF 360 [2] dataset contains 4 indoor scenes and 3 outdoor scenes captured in 360 degree. LLFF [44] dataset contains 8 scenes captured in a forward-facing manner. Furthermore, we use some self-captured custom data to demonstrate the generalization ability of our method.

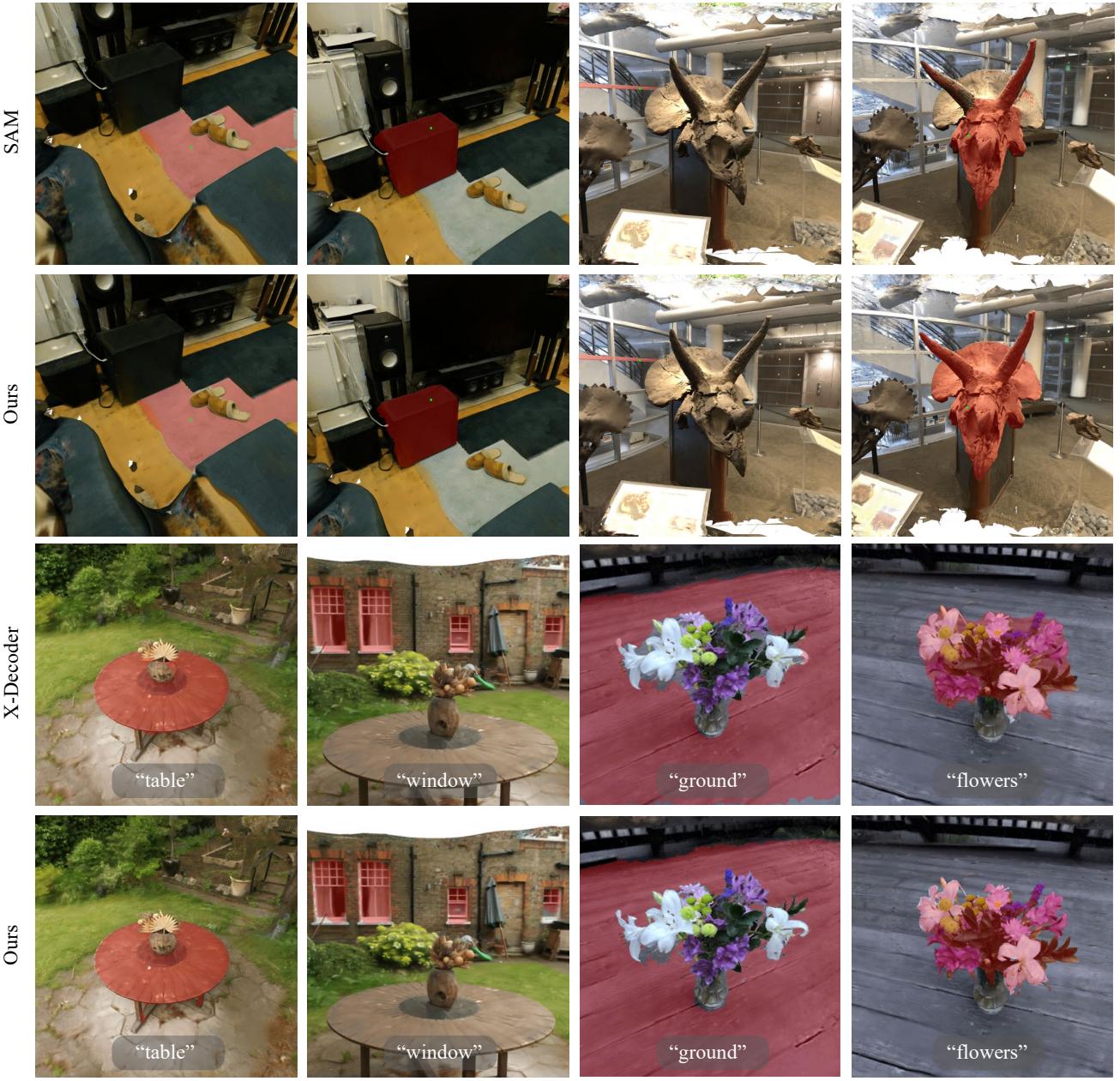


Fig. 4. **Semantic segmentation results.** The click input is shown in green dot and the segmentation mask in red. The text input is shown in gray box. Our method achieves comparable segmentation quality compared to the perception models. We observe in some cases, our results achieve more robust and desirable mask benefits from the 3D consistency.

## 5.2 Segmentation Efficiency

We first analyse the segmentation efficiency in Table 1. Compared to SAM [32], our imitation model delivers a  $52\times$  ( $624\text{ ms} \rightarrow 12\text{ ms}$ ) increase in feature encoding speed, leading to a  $16\times$  ( $1.53\text{ FPS} \rightarrow 24.39\text{ FPS}$ ) boost in overall rendering speed. This makes our method the first real-time 3D click-based segmentation model capable of facilitating smooth user interaction at a  $512 \times 512$  resolution with a

contemporary NVIDIA GPU. In prompt-based segmentation with X-decoder [79], our model still manages to double the feature encoding speed, despite that X-Decoder’s image backbone is relatively small. However, the oversized decoder of X-Decoder has become a bottleneck, limiting the FPS to around 5. Still, our method is considerably faster compared to other prompt-based models like LeRF [30] which runs around 1 FPS. Our successful validation with prompted-based

	Method	SAM	X-Decoder
RGB Rendering (ms)	-	17	17
Feature Encoding (ms)	Original Ours	624 <b>12</b>	73 <b>38</b>
Feature Decoding (ms)	-	8	155
FPS	Original Ours	1.53 <b>24.39 (16x)</b>	4.07 <b>4.71 (1.2x)</b>

Table 1. **Inference efficiency.** We report the inference time of three major steps in milliseconds (ms), and overall FPS.

X-Decoder suggests that our method has the potential to perform well given the availability of more suitable prompt-based segmentation models. An ideal model might feature a larger backbone and a more lightweight decoder.

### 5.3 Qualitative Evaluation

We demonstrate the feature imitation quality of our method through performing segmentation in challenging realistic 3D scenes. In Figure 4, we present the results of click- and text-based segmentation. Our imitation model achieves a segmentation quality comparable to that generated by pretrained segmentation models, even in intricate scenarios such as foliage and slender areas. Our model also demonstrates proficiency in segmenting different regions based on language prompts. Notably, we find our results tend to be more robust in capturing the entirety of the object than X-Decoder, due to the inherent 3D consistency of the features.

We also compare our approach with recent methods [22, 30, 53] in Figure 5. Unlike LeRF [30], our method is capable of generating masks with distinct boundaries. Compared with ISRF [22] and NVOS [53] that utilize strokes as input, our method achieves better or comparable quality only using one point click as input.

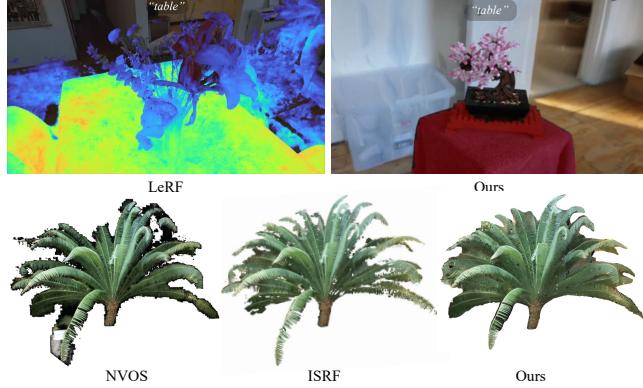


Fig. 5. **Comparison with other methods.** We compare against other methods using prompts or strokes as input for 3D semantic understanding. Since the data used by LeRF [30] is not publicly available, we choose a similar scene from the Mip-NeRF 360 dataset [2].



Fig. 6. **Mesh segmentation results.** Our method allows mesh segmentation by projecting the 2D masks to 3D surfaces. With simple interactions in our GUI, users can get ready-to-use single object meshes easily.

### 5.4 Quantitative Evaluation

To further evaluate the quality of feature imitation beyond qualitative analysis, we conduct a quantitative evaluation. We report the Intersection over Union (IoU) between the predicted mask and the target mask (produced by the pretrained segmentation model). For SAM, we uniformly sample  $5 \times 5$  points in the image, and each point corresponds to a segmentation result. For X-Decoder, we design several prompts for each scene based on the objects present, and each prompt corresponds to a segmentation result. For example, the test prompts for the scene ‘garden’ in Mip-NeRF 360 dataset [2] are ‘table’, ‘window’ and ‘grass’. Our imitation model achieves 82.2% and 74.9% IoU for SAM and X-Decoder, respectively.

### 5.5 Mesh Segmentation

We showcase the results of mesh segmentation in Figure 6. In a complex scene composed of multiple objects, we project the 2D segmentation masks onto 3D surfaces from various viewpoints to achieve mesh segmentation. This approach allows us to extract single object meshes with simple, controllable user interaction. Please check our demonstration video for a practical usage example. Since our pipeline’s ultimate output representation is a mesh, we pave the way for a variety of downstream applications based on the extracted single-object meshes. For instance, we can carry out texture editing and model compositions using common 3D software tools, as demonstrated in Figure 1.

### 5.6 Ablations

We perform ablation studies to verify the designs of our method. We report two metrics on the test set: (1) MSE loss between the imitated feature and the ground-truth feature from the segmentation backbone. (2) Mask IoU that introduced in section 5.4.

	Camera Aug.	Correlation	Caching	Feature Loss ↓	Mask IoU ↑	Training Time ↓
SAM	✓	-	✓	0.0026	0.822	90 min
	-	-	✓	0.0039	0.746	89 min
	✓	-	-	0.0025	0.824	303 min
X-Decoder	✓	✓	✓	0.744	0.749	30 min
	-	✓	✓	0.751	0.742	30 min
	✓	-	✓	0.759	0.741	28 min
	✓	✓	-	0.743	0.751	42 min

Table 2. **Ablation studies.** We report feature MSE loss value, mask IoU, and training time to compare different settings. Please note that the feature value ranges of SAM and X-Decoder are different, so the loss of these two methods cannot be directly compared.

**Camera Augmentation.** We generate novel views through interpolation, thereby expanding the dataset during training. It can be observed that camera augmentation are helpful in reducing feature loss and improving final mask IoU for both SAM and X-decoder backend, demonstrating its effectiveness.

**Caching.** We could reduce the cost of running segmentation backbone by reutilizing cached ground-truth features, which is the major training time bottleneck. In Table 2, we find that our caching mechanism does not markedly affect the feature imitation quality, while significantly accelerating training. This enhancement is particularly conspicuous in models with a heavy backbone such as SAM [32], where we can decrease the training time by a factor of 3.

**Cross-scale Correlation Loss.** We also verify the effectiveness of the proposed cross-scale correlation loss. By constraining the correlations between multi-scale feature maps, we observe smaller feature loss for X-Decoder [79] and also improved IoU.

## 6 LIMITATIONS

In Figure 7, we show some failure cases of our method. For click-based segmentation, the mask could be imperfect and produce multiple unconnected regions. These are also the problems with the original SAM and can be solved by using multiple positive points and also negative points, or other input form like bounding boxes [32]. For prompt-based segmentation, our performance is dependent on the original model’s capability. We found that X-Decoder [79] could fail to correctly recognize concept like “umbrella” but select the round table, or distinguish between close concepts like “lily” or “flowers”. For mesh segmentation, we also rely on the NeRF backbone for mesh extraction. These problems may be solved using more powerful perception models and NeRF backbones.



Fig. 7. **Failure cases.** We show some cases when our method cannot produce satisfactory segmentation masks for both click- and text-based prompts.

## 7 CONCLUSIONS

In this paper, we introduce a novel feature imitation pipeline designed to enhance NeRF with 2D perception models and accomplish 3D perception tasks. By substituting the heavy backbone of SAM with a feature rendering module, our model operates at 16× the speed of its counterparts while maintaining comparable quality. We devise several techniques to enhance imitation quality, including camera augmentation and cached training. Our pipeline has been validated using two state-of-the-art 2D models for click- and text-based segmentation, respectively. In addition, we have developed a graphical user interface to allow user interaction, which demonstrates the practical applicability of our method.

## REFERENCES

- [1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*. 5855–5864.
- [2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *CVPR*. 5460–5469.
- [3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2023. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. *arXiv preprint arXiv:2304.06706* (2023).
- [4] Wenjing Bian, Zirui Wang, Kejie Li, Jiawang Bian, and Victor Adrian Prisacariu. 2023. NoPe-NeRF: Optimising Neural Radiance Field with No Pose Prior. In *CVPR*.
- [5] Junli Cao, Huan Wang, Pavlo Chemerys, Vladislav Shakhrai, Ju Hu, Yun Fu, Denys Makoviichuk, Sergey Tulyakov, and Jian Ren. 2022. Real-Time Neural Light Field on Mobile Devices. *arXiv preprint arXiv:2212.08057* (2022).
- [6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-End Object Detection with Transformers. In *ECCV*. 213–229.
- [7] Qiang Chen, Xiaokang Chen, Jian Wang, Haocheng Feng, Junyu Han, Errui Ding, Gang Zeng, and Jingdong Wang. 2022. Group detr: Fast detr training with group-wise one-to-many assignment. *arXiv preprint arXiv:2207.13085* 1, 2 (2022).
- [8] Qiang Chen, Jian Wang, Chuchu Han, Shan Zhang, Zexian Li, Xiaokang Chen, Jiahui Chen, Xiaodi Wang, Shuming Han, Gang Zhang, et al. 2022. Group detr v2: Strong object detector with encoder-decoder pretraining. *arXiv preprint arXiv:2211.03594* (2022).
- [9] Xiaokang Chen, Jiahui Chen, Yan Liu, and Gang Zeng. 2022. D<sup>3</sup>ETR: Decoder Distillation for Detection Transformer. *arXiv preprint arXiv:2211.09768* (2022).
- [10] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. 2022. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026* (2022).
- [11] Xiaokang Chen, Kwan-Yee Lin, Chen Qian, Gang Zeng, and Hongsheng Li. 2020. 3d sketch-aware semantic scene completion via semi-supervised structure prior. In *CVPR*. 4193–4202.
- [12] Xiaokang Chen, Fangyun Wei, Gang Zeng, and Jingdong Wang. 2022. Conditional detr v2: Efficient detection transformer with box queries. *arXiv preprint arXiv:2207.08914* (2022).
- [13] Xiaokang Chen, Yajie Xing, and Gang Zeng. 2020. Real-time semantic scene completion via feature aggregation and conditioned prediction. In *ICIP*. IEEE, 2830–2834.

- [14] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. 2022. FocalClick: Towards Practical Interactive Image Segmentation. In *CVPR*. 1290–1299.
- [15] Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. 2022. MobileNeRF: Exploiting the Polygon Rasterization Pipeline for Efficient Neural Field Rendering on Mobile Architectures. *arXiv preprint arXiv:2208.00277* (2022).
- [16] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. 2022. Masked-attention Mask Transformer for Universal Image Segmentation. In *CVPR*. 1280–1289.
- [17] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. 2023. PLA: Language-Driven Open-Vocabulary 3D Scene Understanding. In *CVPR*. 7010–7019.
- [18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.
- [19] Zhiwen Fan, Peihao Wang, Yifan Jiang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. 2022. NeRF-SOS: Any-View Self-supervised Object Segmentation on Complex Scenes. *arXiv preprint arXiv:2209.08776* (2022).
- [20] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. 2022. Panoptic NeRF: 3D-to-2D Label Transfer for Panoptic Urban Scene Segmentation. In *3DV*.
- [21] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. 2022. Scaling Open-Vocabulary Image Segmentation with Image-Level Labels. In *ECCV*. 540–557.
- [22] Rahul Goel, Dhawal Sirikonda, Saurabh Saini, and P.J. Narayanan. 2023. Interactive Segmentation of Radiance Fields. In *CVPR*.
- [23] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. 2021. Baking Neural Radiance Fields for Real-Time View Synthesis. In *ICCV*. 5855–5864.
- [24] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul E. Debevec. 2021. Baking Neural Radiance Fields for Real-Time View Synthesis. In *ICCV*. 5855–5864.
- [25] Qiangui Huang, Weiye Wang, and Ulrich Neumann. 2018. Recurrent Slice Networks for 3D Segmentation of Point Clouds. In *CVPR*. 2626–2635.
- [26] Dat Huynh, Jason Kuen, Zhe Lin, Jiuxiang Gu, and Ehsan Elhamifar. 2022. Open-Vocabulary Instance Segmentation via Robust Cross-Modal Pseudo-Labeling. In *CVPR*. 7010–7021.
- [27] Jitesh Jain, Jiacheng Li, Man Chun Chiu, Ali Hassani, Nikita Orlov, and H. Shi. 2022. OneFormer: One Transformer to Rule Universal Image Segmentation. *arXiv preprint arXiv:2211.06220* (2022).
- [28] Wonbong Jang and Lourdes Agapito. 2021. CodeNeRF: Disentangled Neural Radiance Fields for Object Categories. In *ICCV*. 12929–12938.
- [29] Chao Jia, Yinfai Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *ICML*. 4904–4916.
- [30] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. 2023. LERF: Language Embedded Radiance Fields. *arXiv preprint arXiv:2303.09553* (2023).
- [31] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.
- [32] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. 2023. Segment Anything. *arXiv preprint arXiv:2304.02643* (2023).
- [33] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. 2022. Decomposing NeRF for Editing via Feature Field Distillation. In *NeurIPS*.
- [34] Feng Li, Hao Zhang, Hu-Sheng Xu, Siyi Liu, Lei Zhang, Lionel Ming shuan Ni, and Heung yeung Shum. 2022. Mask DINO: Towards A Unified Transformer-based Framework for Object Detection and Segmentation. *arXiv preprint arXiv:2206.02777* (2022).
- [35] Shengnan Liang, Yichen Liu, Shangzhe Wu, Yu-Wing Tai, and Chi-Keung Tang. 2022. ONEfR: Unsupervised 3D Object Segmentation from Multiple Views. *arXiv preprint arXiv:2211.12038* (2022).
- [36] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2022. Magic3D: High-Resolution Text-to-3D Content Creation. *arXiv preprint arXiv:2211.10440* (2022).
- [37] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. 2021. BARF: Bundle-Adjusting Neural Radiance Fields. In *ICCV*. 5721–5731.
- [38] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *CVPR*. 2117–2125.
- [39] Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. 2022. SimpleClick: Interactive Image Segmentation with Simple Vision Transformers. *arXiv preprint arXiv:2210.11006* (2022).
- [40] Xinhang Liu, Jiaben Chen, Huai Yu, Yu-Wing Tai, and Chi-Keung Tang. 2022. Unsupervised Multi-View Object Segmentation Using Radiance Field Propagation. In *NeurIPS*.
- [41] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*. 3431–3440.
- [42] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. 2021. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3651–3660.
- [43] Lu Mi, Abhijit Kundu, David A. Ross, Frank Dellaert, Noah Snavely, and Alireza Fathi. 2022. im2nerf: Image to Neural Radiance Field in the Wild. *arXiv preprint arXiv:2209.04061* (2022).
- [44] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM TOG* 38, 4 (2019), 29:1–29:14.
- [45] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*. 405–421.
- [46] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM TOG* 41, 4 (2022), 102:1–102:15.
- [47] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. 2022. Extracting Triangular 3D Models, Materials, and Lighting From Images. In *CVPR*. 8270–8280.
- [48] Michael Niemeyer and Andreas Geiger. 2021. GIRAFFE: Representing Scenes As Compositional Generative Neural Feature Fields. In *CVPR*. 11453–11464.
- [49] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).
- [50] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *CVPR*. 10318–10327.
- [51] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. 2021. DenseCLIP: Language-Guided Dense Prediction with Context-Aware Prompting. In *CVPR*. 18061–18070.
- [52] Christian Reiser, Richard Szeliski, Dor Verbin, Pratul P. Srinivasan, Ben Mildenhall, Andreas Geiger, Jonathan T. Barron, and Peter Hedman. 2023. MERF: Memory-Efficient Radiance Fields for Real-time View Synthesis in Unbounded Scenes. *arXiv preprint arXiv:2302.12249* (2023).
- [53] Zhongzheng Ren, Aseem Agarwala, Bryan C. Russell, Alexander G. Schwing, and Oliver Wang. 2022. Neural Volumetric Object Selection. In *CVPR*. 6123–6132.
- [54] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. 2022. Panoptic Lifting for 3D Scene Understanding with Neural Fields. *arXiv preprint arXiv:2212.09802* (2022).
- [55] Cheng Sun, Min Sun, and Hwann-Tzong Chen. 2022. Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. In *CVPR*. 5449–5459.
- [56] Jiaxiang Tang, Xiaokang Chen, Jingbo Wang, and Gang Zeng. 2022. Compressible-composable NeRF via Rank-residual Decomposition. In *NeurIPS*.
- [57] Jiaxiang Tang, Xiaokang Chen, Jingbo Wang, and Gang Zeng. 2022. Not all voxels are equal: Semantic scene completion from the point-voxel perspective. In *AAAI*, Vol. 36. 2352–2360.
- [58] Jiaxiang Tang, Xiaokang Chen, Jingbo Wang, and Gang Zeng. 2022. Point scene understanding via disentangled instance mesh reconstruction. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Springer, 684–701.
- [59] Jiaxiang Tang, Kaisiyuan Wang, Hang Zhou, Xiaokang Chen, Dongliang He, Tianshu Hu, Jingtuo Liu, Gang Zeng, and Jingdong Wang. 2022. Real-time Neural Radiance Talking Portrait Synthesis via Audio-spatial Decomposition. *arXiv preprint arXiv:2211.12368* (2022).
- [60] Jiaxiang Tang, Hang Zhou, Xiaokang Chen, Tianshu Hu, Errui Ding, Jingdong Wang, and Gang Zeng. 2023. Delicate Textured Mesh Recovery from NeRF via Adaptive Surface Refinement. *arXiv preprint arXiv:2303.02091* (2023).
- [61] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. [n. d.]. Neural Feature Fusion Fields: 3D Distillation of Self-Supervised 2D Image Representations. In *3DV*.
- [62] Vadim Tschernezki, Diane Larlus, and Andrea Vedaldi. 2021. NeuralDiff: Segmenting 3D objects that move in egocentric videos. In *3DV*.
- [63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *NeurIPS*. 6000–6010.
- [64] Thang Vu, Kookhoi Kim, Tung Minh Luu, Thanh Nguyen, and Chang D. Yoo. 2022. SoftGroup for 3D Instance Segmentation on Point Clouds. In *CVPR*. 2698–2707.
- [65] Chen Wang, Xian Wu, Yuanchen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. 2022. NeRF-SR: High Quality Neural Radiance Fields using Supersampling. In *MM*. 6445–6454.

- [66] Huan Wang, Jian Ren, Zeng Huang, Kyle Olszewski, Menglei Chai, Yun Fu, and Sergey Tulyakov. 2022. R2l: Distilling neural radiance field to neural light field for efficient novel view synthesis. In *ECCV*. Springer, 612–629.
- [67] Wenhui Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. 2023. VisionLLM: Large Language Model is also an Open-Ended Decoder for Vision-Centric Tasks. *arXiv preprint arXiv:2305.11175* (2023).
- [68] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas M. Breuel, Jan Kautz, and Xiaolong Wang. 2022. GroupViT: Semantic Segmentation Emerges from Text Supervision. In *CVPR*. 18113–18123.
- [69] Ning Xu, Brian L. Price, Scott Cohen, Jimei Yang, and Thomas S. Huang. 2016. Deep Interactive Object Selection. In *CVPR*. 373–381.
- [70] Yang Xue, Yuheng Li, Krishna Kumar Singh, and Yong Jae Lee. 2022. GIRAFFE HD: A High-Resolution 3D-aware Generative Model. In *CVPR*. 18419–18428.
- [71] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. 2019. Learning Object Bounding Boxes for 3D Instance Segmentation on Point Clouds. In *NeurIPS*. 6737–6746.
- [72] Jiawei Yang, Marco Pavone, and Yue Wang. 2023. FreeNeRF: Improving Few-shot Neural Rendering with Free Frequency Regularization. *arXiv preprint arXiv:2303.07418* (2023).
- [73] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P.Srinivasan, Richard Szeliski, Jonathan T.Barron, and Ben Mildenhall. 2023. BakedSDF: Meshing Neural SDFs for Real-Time View Synthesis. *arXiv preprint arXiv:2302.14859* (2023).
- [74] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. 2021. pixelNeRF: Neural Radiance Fields From One or Few Images. In *CVPR*. 4578–4587.
- [75] Jesus Zarzar, Sara Rojas, Silvio Giancola, and Bernard Ghanem. 2022. SegNeRF: 3D Part Segmentation with Neural Radiance Fields. *arXiv preprint arXiv:2211.11215* (2022).
- [76] Xinyu Zhang, Jiahui Chen, Junkun Yuan, Qiang Chen, Jian Wang, Xiaodi Wang, Shumin Han, Xiaokang Chen, Jimin Pi, Kun Yao, et al. 2022. CAE v2: Context Autoencoder with CLIP Target. *arXiv preprint arXiv:2211.09799* (2022).
- [77] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. 2021. In-Place Scene Labelling and Understanding with Implicit Scene Representation. In *ICCV*. 15818–15827.
- [78] Min Zhong, Xinghao Chen, Xiaokang Chen, Gang Zeng, and Yunhe Wang. 2023. MaskGroup: Hierarchical Point Grouping and Masking for 3D Instance Segmentation. In *ICME*.
- [79] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Singh Behl, Jianfeng Wang, Lu Yuan, Nanyun Peng, Lijuan Wang, Yong Jae Lee, and Jianfeng Gao. 2022. Generalized Decoding for Pixel, Image, and Language. *arXiv preprint arXiv:2212.11270* (2022).
- [80] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. 2023. Segment Everything Everywhere All at Once. *arXiv preprint arXiv:2304.06718* (2023).