Obj-NeRF: Extract Object NeRFs from Multi-view Images

Zhiyi Li[1], Lihe Ding[2], and Tianfan Xue[2]
[1]Tsinghua University, [2]The Chinese University of Hong Kong
lizhiyi20@mails.tsinghua.edu.cn, {dl023,tfxue}@ie.cuhk.edu.hk

Figure 1. Proposed Obj-NeRF: indicate prompts on an image, then Obj-NeRF will output the segmented NeRF for the target object. With the segmented object NeRF, some applications including NeRF editing can be realized.

Abstract

Neural Radiance Fields (NeRFs) have demonstrated remarkable effectiveness in novel view synthesis within 3D environments. However, extracting a radiance field of one specific object from multi-view images encounters substantial challenges due to occlusion and background complexity, thereby presenting difficulties in downstream applications such as NeRF editing and 3D mesh extraction. To solve this problem, in this paper, we propose Obj-NeRF, a comprehensive pipeline that recovers the 3D geometry of a specific object from multi-view images using a single prompt. This method combines the 2D segmentation capabilities of the Segment Anything Model (SAM) in conjunction with the 3D reconstruction ability of NeRF. Specifically, we first obtain multi-view segmentation for the indicated object using SAM with a single prompt. Then, we use the segmentation images to supervise NeRF construction, integrating several effective techniques. Additionally, we construct a large object-level NeRF dataset containing diverse objects, which can be useful in various downstream tasks. To demonstrate the practicality of our method, we also apply Obj-NeRF to various applications, including object removal, rotation, replacement, and recoloring. The project page is at https://objnerf.github.io/.

1. Introduction

Neural Radiance Fields (NeRFs) have attracted enormous academic interest in 3D scene representation, due to their remarkable ability to produce high-quality synthesized views in diverse 3D environments [17]. Recent works have concentrated on enhancing the performance and practicality of NeRF, thereby broadening its applicability with higher reconstruction quality and faster training speed [8, 19, 27].

Moreover, NeRF is also widely used in many downstream applications, including 3D editing and novel view synthesis [14]. With that, the demand for object-specific NeRF datasets has risen. Nevertheless, the inherent limitation of NeRF, which provides only color and density information, presents a challenge for extracting specific objects from multi-view images.

In order to extract object-level NeRF from multi-view images, recent works have primarily explored the utilization of 2D visual models like CLIP [23] or DINO [4] along with additional feature images provided by modified NeRFs, such as LERF [11] and Interactive Segment Anything NeRF [6]. However, this approach has limitations, including the absence of required 3D object meshes, excessive additional training costs for the complete scene NeRF, and suboptimal reconstruction quality [6].

To address these limitations, our objective is to extract specific object NeRFs from multi-view images representing a 3D scenario. Although there have been continuous advancements in 2D image segmentation, such as the recently proposed Segment Anything Model (SAM) [12], segmenting the required 3D object NeRF from an original scenario encounters numerous challenges. Notably, training a 3D segmentation model similar to SAM for zero-shot segmentation tasks remains a formidable undertaking [5]. While

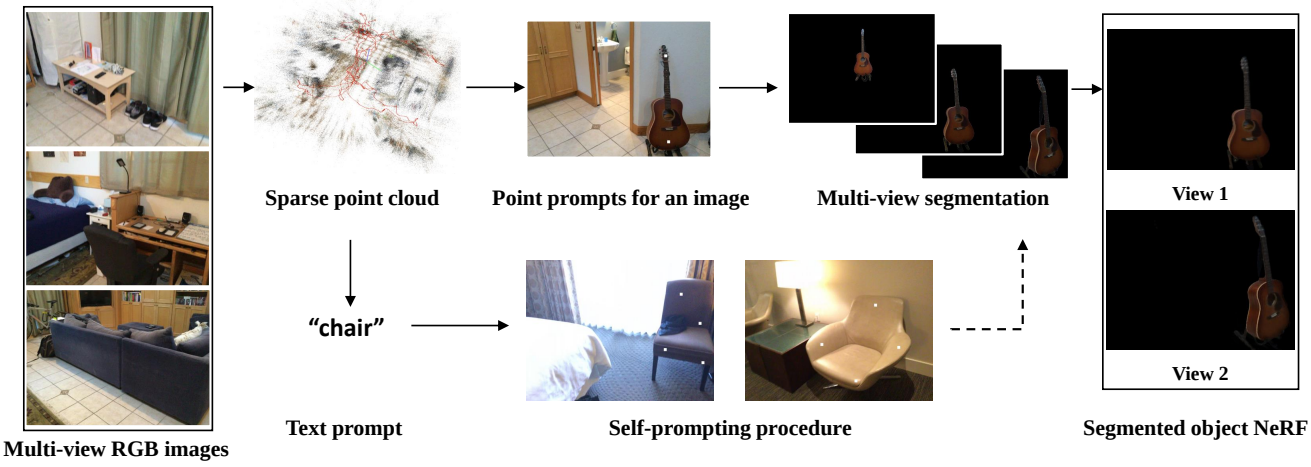

Figure 2. The overall pipeline for Obj-NeRF. Starting with multi-view RGB images, a COLMAP sparse point cloud can be constructed, which provides multi-view consistency for segmentation. After initializing several prompts for the first image, we can automatically obtain multi-view segmented images quickly, which are used to construct the required segmented NeRF. For large datasets, the indicated objects will be prompted for each scene.

extending segmentation capabilities directly into 3D scenarios presents formidable challenges, combining the 2D segmentation proficiency of SAM with the 3D representation capabilities of NeRF is a feasible and promising approach.

Based on this idea, in this work, we propose Segment Object NeRF (Obj-NeRF) to extract a certain object and reconstruct its geometry, from a few user prompts. The effectiveness of Obj-NeRF is depicted in Figure 1. Obj-NeRF initially receives prompts indicating a specific object within a single image, selected from a set of multi-view images representing a 3D scenario. Subsequently, Obj-NeRF generates multi-view segmented images through the utilization of the SAM, thereby supervising the construction of the segmented target object NeRF. After acquiring the object NeRF, we further evaluate it on various 3D editing applications. The main contributions of this paper can be summarized as follows:

- Firstly, we present a comprehensive pipeline for constructing a segmented NeRF targeting a specific object, with the input consisting of initial prompts from a single image. Our pipeline eliminates the need for pre-trained full-scene NeRFs, thereby avoiding unnecessary training expenses and enhancing reconstruction quality.
- Next, to obtain multi-view segmented images of the target object, we introduce a multi-view segmentation algorithm with high quality and efficiency. Leveraging a 3D sparse point cloud, we rapidly disseminate the initial prompts to all images and extract segmented masks from SAM.
- Additionally, in order to create massive object NeRFs from large multiview datasets, we propose an automatic self-prompting mechanism with only simple textual input. It will enable the identification of the desired object for each scene, thereby constructing a dataset of multi-view segmented objects.
- Finally, to enhance the quality of novel view synthesis using NeRFs, we propose several methods, including supervision with sparse and dense depth priors, bounding box calculation, and ray pruning for improving performance. Additionally, we validate the effectiveness of segmented object NeRFs by modifying existing NeRFs.

2. Related Works

NeRF for Novel View Synthesis. Neural Radiance Fields (NeRFs) have gained numerous research interests in novel view reconstruction [17]. Recently, researchers have been working on improving the effectiveness of NeRFs, including enhancing reconstruction quality via various methods [1, 8, 24, 31], increasing training speed with high synthesis effect [19, 27, 28], and expanding their application scenarios [22, 29]. There are also many downstream works on NeRFs, such as NeRF editing [18, 33, 34], 3D mesh extraction [20], and 3D generation tasks [13, 14, 21].

Segmentation on NeRFs. Significant progress has been made in 2D semantic fields, including DETR [3] and CLIP [10]. Recently, models trained on extremely large-scale datasets, such as the Segment Anything Model (SAM) [12] and SEEM [35], have shown strong ability on zero-shot image segmentation. Based on these, many researchers have made some progress to expand on 3D segmentation fields, by training an extra semantic feature on modified NeRF [11] and distilling segmentation backbone with NeRF [6]. However, these works cannot provide a segmented object NeRF, which is essential in many downstream applications like 3D scenario editing. SA3D [5] has

proposed a method to construct a segmented object NeRF from multi-view images with SAM. Nonetheless, SA3D requires a pre-trained original full-scene NeRF, which is impractical and brings extra training costs and low reconstruction quality, especially for large-scale scenes.

3. Preliminaries

In this section, some preliminaries for Obj-NeRF will be introduced briefly, including the Neural Radiance Fields (NeRFs) [17] and the Segment Anything Model (SAM) [12].

Neural Radiance Field. NeRF presents an effective way to synthesize novel views in 3D scenarios. Specifically, NeRF defines an underlying continuous volumetric scene function $\mathcal{F}_{\boldsymbol{\Theta}} : (\boldsymbol{x}, \boldsymbol{d}) \Longrightarrow (\boldsymbol{c}, \sigma)$, which outputs the color $\boldsymbol{c} \in \mathbb{R}^3$ and the volume density $\sigma \in \mathbb{R}^+$ with a given spacial location $\boldsymbol{x} \in \mathbb{R}^3$ and viewing direction $\boldsymbol{\theta} \in \mathbb{S}^2$. In this way, the rendering color $C(\boldsymbol{r})$ for a specific camera ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ can be expressed by a volume rendering algorithm as follows:

$$C(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(\boldsymbol{r}(t))\boldsymbol{c}(\boldsymbol{r}(t), \boldsymbol{d})\, dt, \tag{1}$$

where t_n and t_f are the near and far bounds, and the accumulated transmittance $T(t)$ can be calculated as:

$$T(t) = \exp - \int_{t_n}^{t} \sigma(\boldsymbol{r}(s))\, ds. \tag{2}$$

With these definitions, NeRFs can be optimized using the loss between the ground-truth color $C(\boldsymbol{r})$ and the calculated color $\hat{C}(\boldsymbol{r})$ for any image I:

$$\mathcal{L}_I = \sum_{\boldsymbol{r} \in I} \left\| C(\boldsymbol{r}) - \hat{C}(\boldsymbol{r}) \right\|^2. \tag{3}$$

Segment Anything Model (SAM) SAM, training by numerous 2D images, has been proved to achieve a state-of-art efficiency in zero-shot segmentation tasks [12]. With an image $\boldsymbol{I}$ and some prompts $\mathcal{P}$, including points (positive or negative) and boxes, SAM can provide a mask for the indicated object mask $= \mathcal{S}(\boldsymbol{I}, \mathcal{P})$.

4. Methods

In this section, we will introduce details of Obj-NeRF. First, the overall pipeline will be demonstrated in Sec. 4.1. Then, a one-shot multi-view segmentation method will be presented in Sec. 4.2. After that, we will introduce a self-prompting method to construct an object NeRF dataset including massive objects using the segmentation method above in Sec. 4.3. In the end, some strategies for novel views synthesizing by NeRFs will be provided in Sec. 4.4.

4.1. Overall Pipeline

We consider a set of multi-view images $\boldsymbol{I} = I_1, ..., I_n$ for one specific scenario with known camera poses. If not, structure-from-motion methods like COLMAP [25] can be utilized to estimate them. The objective is to acquire the 3D representation for any object segmented from this scenario with few prompts. To achieve this, a pre-trained full-scene NeRF for the scenario is not required due to the unnecessary training cost and relatively poor quality. Thus, we propose a method to acquire the segmented NeRF for the object from multi-view images directly.

The overall pipeline is shown in Fig. 2. To start with, users will first provide a few prompts for one image on the object that is expected to be segmented. Based on this, a COLMAP sparse point cloud [25] can be constructed, which provides the correspondence between 2D images and point clouds used in the next step. Then, the multi-view segmentation procedure with SAM will provide multi-view masks for the object. For large datasets, such as ScanNet [7], ScanNet++ [32], and 3RScan [30], the self-prompting procedure will quickly generate a series of prompts of a kind of objects, which can be used to acquire multi-view segmented for each scene with the method above. In the end, the segmented NeRF of the indicated object will be trained with these multi-view segmented images, which provides novel view synthesizing abilities.

4.2. Multi-view Segmentation

4.2.1 Multi-view Segmentation Algorithm

The first step is quickly obtaining multi-view segmented images from initial prompts on one specific image. It is easy to get the mask for the initial image M_0 with SAM. However, multi-view consistency should be utilized here in order to segment the indicated object on each image. A similar approach is also used by Yin et al. [33] to find a 2D-3D geometry match relationship with prompts spreading, but here we use it for a different task with some effective methods mentioned as follows.

More specifically, a sparse point cloud can be easily constructed from input images using COLMAP, a 3D reconstruction toolbox [25]. These sparse point clouds provide the correspondence between feature points on each image and the 3D points in the point cloud. In this way, we can construct a 3D point list $\boldsymbol{D}$, which contains 3D points belonging to the indicated object. After initializing the list with 3D points that correspond to the feature points on $I_0[M_0]$, the 3D point list and the masks of remnant images can be updated iteratively. Specifically, for a new image I_i, the point prompts $\boldsymbol{p}_i$ can be selected from the feature points that correspond to 3D points in the list. Then, the mask M_i can be obtained using SAM segmentation model $\mathcal{S}(I_i, \boldsymbol{p}_i)$. After that, all feature points on $I_i[M_i]$ can be added to the

Algorithm 1 Multi-view Segmentation

Input: A set of images $I_0, I_1, I_2, ..., I_n$; Initial point prompts $\boldsymbol{p}_0$; SAM model $\mathcal{S}$; COLMAP sparse point cloud $\mathcal{C}$.

Output: Multi-view segmented masks $M_1, M_2, ..., M_n$.

Get the mask for the initial image $M_0 = \mathcal{S}(I_0, \boldsymbol{p}_0)$
Find all feature points $\boldsymbol{X} = \mathcal{C}[I_0] \cup I_0[M_0]$
Init 3D point list $\boldsymbol{D} = \mathcal{C}(\boldsymbol{X})$.
for $i = 1, 2, ..., n$ **do**
    Find point prompts $\boldsymbol{p}_i$ for I_i from $\boldsymbol{D}$ and $\mathcal{C}$
    $M_i \leftarrow \mathcal{S}(I_i, \boldsymbol{p}_i)$
    $\boldsymbol{X}_i \leftarrow \mathcal{C}[I_i] \cup I_i[M_i]$
    $\boldsymbol{D} \leftarrow \boldsymbol{D} \cup \mathcal{C}(\boldsymbol{X}_i)$
end for
return $M_1, M_2, ..., M_n$

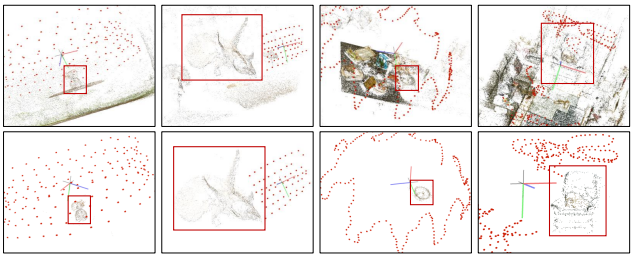

Figure 3. Segmented point cloud of target objects; Images in the first row are original sparse point clouds; Images in the second row are segmented point clouds; Red points on these images indicate the position of cameras.

list $\boldsymbol{D}$, which finishes an iterating step. The multi-view segmentation procedure can be summarized in Algorithm 1.

After executing the algorithm above, an interesting byproduct will be obtained from the list $\boldsymbol{D}$. As the definition of $\boldsymbol{D}$, it is consisted of those 3D points belonging to the indicated object, which provides its sparse point cloud. Fig. 3 shows the segmented sparse point cloud of these indicated objects. This can be used in the next step and will offer some priors to the novel views synthesizing procedure in Subsection 4.4.

4.2.2 Obstruction Handling

There is a thorny problem that when the target object is obstructed by other things, it will be easy for the procedure leading to a wrong result. In [33], the method based on projecting 3D points directly as the prompts will severely suffer

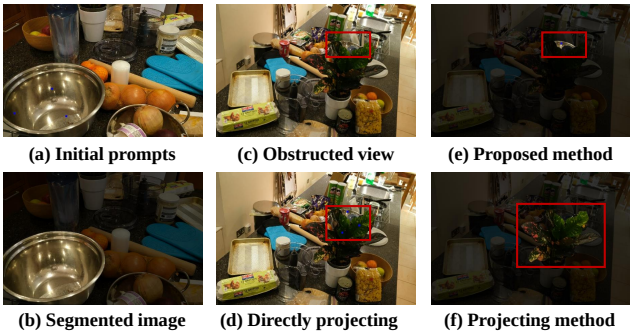

Figure 4. Showing the obstruction effect; (a) Initial point prompts for the target bowl; (b) Segmented image for the first image; (c) A view where the target bowl is obstructed by the plant; (d) and (f) Directly projecting method leading to a wrongly segmented object; (e) Proposed method to get the correct segmentation.

from it. Here, our proposed multi-view segmentation procedure overcomes the wrongly prompting effect, by using the feature point correspondence instead of the projection method. However, these partially obstructed segmented images bring multi-view inconsistency for later NeRF training procedures. As Fig. 4 shows, although the target bowl indicated in Fig. 4 (a) has been correctly segmented in Fig. 4 (e), the inconsistency between Fig. 4 (b) and Fig. 4 (e) will lead to performance degradation during novel views synthesizing.

In order to eliminate the inconsistency brought by obstructed images, it is important to identify them first. With the segmented sparse point cloud $\boldsymbol{D}$, we can first project these 3D points to each image to get the 2D coordinates for these feature points. Then, we can construct the concave hall for these 2D points as the mask of the target object using the alpha-shape method [9]. After that, the estimated mask can be smoothed by a Gaussian filter. Fig. 5 shows the procedures above to identify the obstructed images. The IoU between Fig. 5 (d) and Fig. 5 (f) is 0.096, which means the segmented image should be discarded. It should be noted that we cannot simply calculate the convex hull and regard it as the mask, for there are usually some outlier points which will extremely affect it.

4.2.3 Multi-object Segmentation

The proposed segmentation algorithm can be extended to k-object segmentation tasks. After giving initial prompts for each target object, we can construct k 3D point lists $\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_k$ and update them with masks separately with almost little increase in time consumption. In this way, several target objects can be segmented for only one time. The performances of the multi-object segmentation method will be shown in Subsection 5.2.

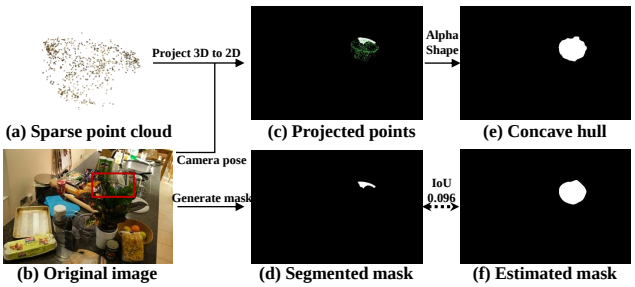

Figure 5. Procedures to identify obstructed images; (a) Segmented sparse point cloud of the target object; (b) Original RGB image waiting to be masked; (c) Projecting 3D points to 2D image with known camera pose; (d) Generating mask for the original image; (e) Alpha-shape concave hull for the points; (f) Estimated mask after Gaussian filtering.

4.3. Large Dataset Self-prompting

In Sec. 4.2, we propose a method which receives point prompts as input and outputs multi-view segmented views. Thus, if we can segment every object in a large dataset like ScanNet [7], which contains 1000+ scenes, a large multi-view 3D object dataset can be constructed and is useful for many downstream works including generative tasks, like zero123 [14]. However, manually labeling the prompts for each object is tedious and unrealistic, which pushes us to find a feasible way to generate prompts for each scene quickly from a text prompt.

First, a text prompt should be converted to something SAM can utilize and then generate a proper mask. Here, we use an object detector named Grounding DINO model [15], which receives text input and outputs boxes and scores that indicate the position and the probability of the target object. Then, the box with the highest score can be considered as an input to SAM, which provides a proper mask for the target object.

The next step is generating point prompts to fit the requirements of the segmentation algorithm proposed in Subsection 4.2. These point prompts should fulfill the conditions below: (1) They stay away from each other and represent all parts of the object; (2) They cannot stay too close to the edge. Thus, we can first calculate the distance to their mask for each point on the mask. Then these points near the edge are selected, for the interior points will interfere with the next step. Finally, point prompts can be generated through the k-means method [26]. Fig. 6 shows the steps which provide point prompts from the mask.

In this way, we create an object NeRF dataset including a large number of objects with just a few textual inputs. Details are discussed in 5.2.

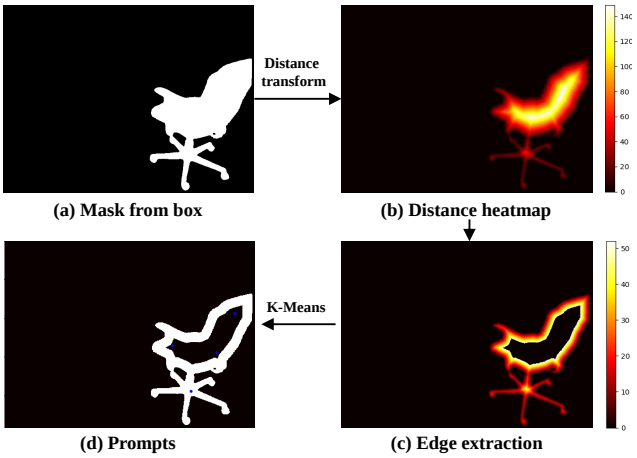

Figure 6. The procedure from mask to point prompts; (a) Mask from SAM and the box prompt; (b) Distance heatmap showing the distance to the edge for each point; (c) Extracted points which are near the edge; (d) Point prompts from k-means method.

4.4. Novel View Synthesizing

With the multi-view segmented images from Subsection 4.2, it is practicable to synthesize novel views for the target object after training a NeRF. However, simply constructing NeRF with segmented images only will not lead to a perfect performance. In this subsection, we will introduce some methods which significantly increase the quality of synthesizing.

4.4.1 Sparse and Dense Depth-Supervised NeRF

In order to acquire better performance and faster convergence, Deng et al. [8] have proposed a method that adds depth information to supervise the NeRF training procedure. Specifically, the segmented object sparse point cloud $\boldsymbol{D}$ mentioned in Subsection 4.2 will provide their 3D coordinate information. Thus, for each image, the depth of feature points corresponding to the 3D point cloud can be calculated respectively. In this way, the sparse depth supervised NeRF training can be realized with the loss as follows,

$$\mathcal{L}_{\text{NeRF}} = \mathcal{L}_{\text{rgb}} + \lambda_d \mathcal{L}_{\text{depth}}, \tag{4}$$

where $\mathcal{L}_{\text{depth}} = \|\boldsymbol{d} - \hat{\boldsymbol{d}}\|^2$ indicates the mean square error of depth. It should be noticed that sparse depth supervision makes better performance in extremely few multi-view images like less than 10. For more input images, it will also improve the quality for reconstruction 3D mesh but may not for the reconstructed RGB images [8].

Sparse depth supervision brings limited performance enhancement due to the scarcity of depth information. To achieve higher reconstruction, dense depth information should be included in NeRF training. Many large multi-view datasets include depth image for each RGB image,

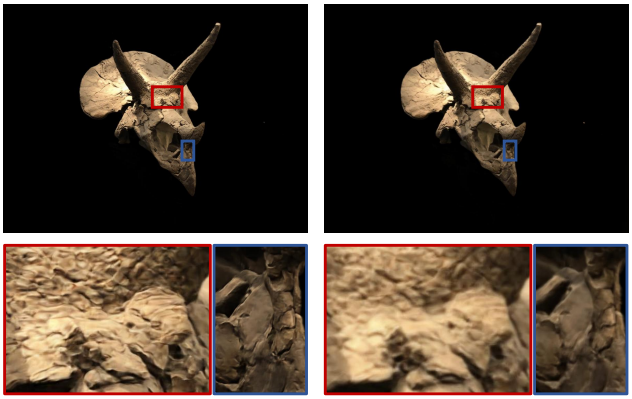

Figure 7. Comparison of reconstruction performance with different resolution training images; Left: No down-sampling with ray pruningS; Right: Down-sampling.

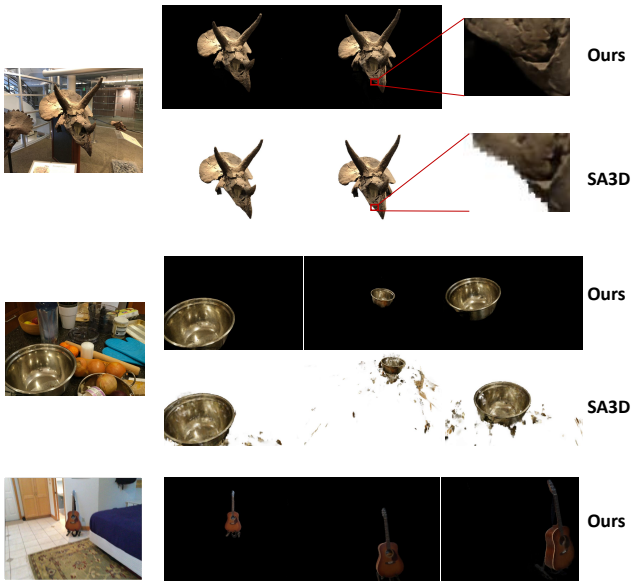

Figure 8. Novel view synthesis performance for indicated objects compared with SA3D [5].

such as ScanNet [7], ScanNet++ [32], and 3RScan [30], which will provide required dense depth information. Fig. 9 shows the novel-view reconstruction performance comparison with and without dense depth supervision. Comparing Fig. 9 (d) and Fig. 9, reconstruction with depth supervision will provide a significantly higher quality 3D mesh.

4.4.2 Bounding Box and Ray Pruning

After Segmenting the indicated object from each whole image, there are three notable advantages of the reconstruction as follows: (1) Eliminate the extra components thereby reducing the additional NeRF training cost; (2) Reduce the world size leading to augmented ray sampling density and voxel density; (3) Pruning of rays unrelated to the object, significantly conserving CUDA memory and enabling the utilization of higher resolution images. In order to achieve these advantages above, some methods will be introduced during the NeRF training procedure.

According to the segmented sparse point cloud in Subsection 4.2, a bounding box $\boldsymbol{B}$ can be calculated from the known 3D point coordinates, which provides the scale of the world size in NeRF settings. With the much smaller bounding box, the density of ray sampling and the voxel grid increase accordingly (e.g. in the first column of Fig. 3, the overall voxel grid size decreases to 1% of the original one). Additionally, any rays which not intersect with the bounding box, i.e. out of the box, will be pruned and not be used to supervise the training. In this way, the number of effective training rays is reduced by an order of magnitude, which makes the utilization of higher resolution possible. As shown in Fig. 7, with higher resolution input RGB images, the reconstruction performance increases accordingly.

5. Experiments

5.1. Implementation Details

Dataset. In order to verify the generality of our proposed comprehensive pipeline, we evaluate the Obj-NeRF on various multi-view datasets, including face-forwarding LLFF dataset [16], Mip NeRF 360 dataset [2], LERF dataset [11], and large indoor datasets such as 3RScan [30], ScanNet [7], and ScanNet++ [32]. Obj-NeRF will provide an indicated object NeRF with only a single prompt input for any scenario in these datasets. For large indoor datasets, the self-prompting procedure mentioned in Subsection 4.3 can be used. It will eventually provide a large multi-view object dataset including thousands of objects.

Novel-view Synthesis. In our process of synthesizing novel views, we utilize the framework of DVGO NeRF [27]. It is important to note that our method is not limited to DVGO, other implementations of NeRF such as Instant NGP [19] or NeRF Studio [28] can also be used. Moreover, we have improved the quality of reconstruction by adopting the depth-supervision method from DS-NeRF [8]. To achieve object NeRF applications like object removal, replacement, rotation, and color-changing, we have used Blender to generate appropriate camera poses.

5.2. Results

Multi-view Segmentation Consistency. As Fig. 10 shown, the proposed multi-view segmentation algorithm demonstrates strong robustness in various datasets, including face-forwarding, 360° panoramic, and large indoor scenes. Especially in the third row in Fig. 10, images in ScanNet dataset [7] have relatively low resolution and sometimes loss of focus, our proposed procedure also works.

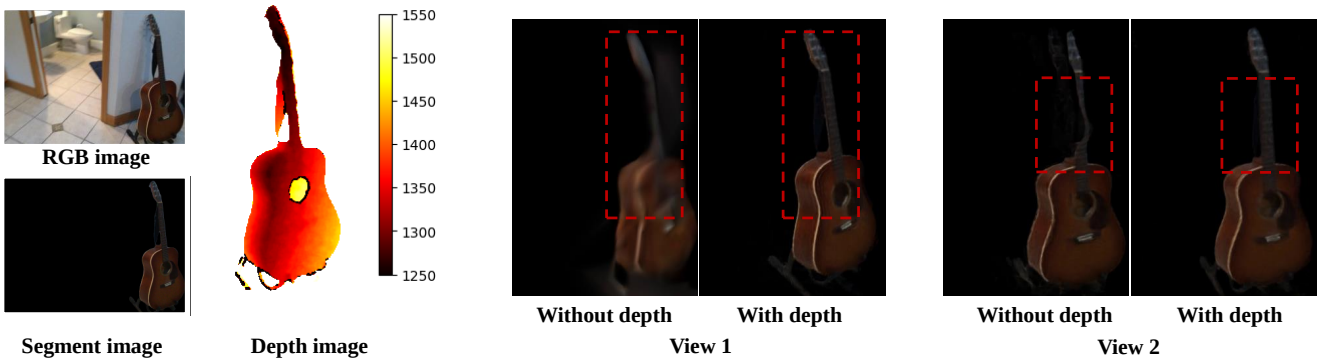

Figure 9. Comparison of novel view reconstruction performance with and without dense depth supervision.

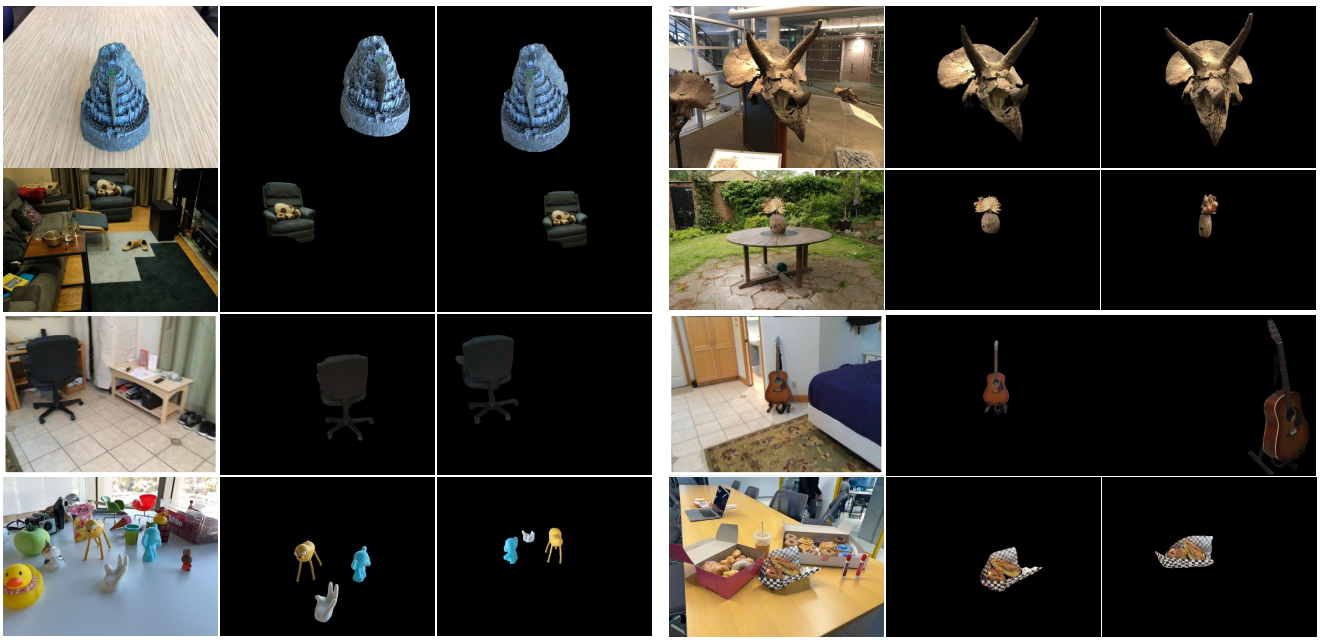

Figure 10. Performance of the multi-view segmentation procedure; First row: LLFF dataset; Second row: Mip NeRF 360 dataset; Third row: ScanNet dataset; Last row: LERF dataset.

Multi-view Object Dataset. We utilize our proposed self-prompting method on some large indoor datasets in order to construct a multi-view object dataset. As Fig. 11 has shown, after indicating a textual input like "chair" or "table", it will automatically generate initial prompts for the target object in each scene. After that, the multi-view segmentation and NeRF training procedures are followed, constructing an object NeRF for each object.

Novel-view Synthesis. We construct the object NeRF under the supervision of the multi-view segmentation images mentioned above. With the methods introduced in Section 4, our novel view synthesis procedure overcomes the obstruction effect, enables multi-object reconstruction, and utilizes techniques that improve the reconstruction performance. As shown in Fig. 8, we compare our proposed method to SA3D [5] segmenting foreground NeRF from an pre-trained full-scene NeRF, which suffers from low resolution and floaters. Our proposed method achieves relatively high reconstruction quality across various scenarios, especially for large indoor datasets like the last row of Fig. 8, where the full-scene NeRF required for SA3D is impractical and low-quality.

5.3. Applications

In order to verify the effectiveness of the object NeRF dataset, we utilize the extracted object NeRF in various applications as shown in Fig. 12, including object removal, replacement, rotation, and color changing.

Add-on. We can integrate the segmented object NeRF into any existing NeRF to realize the add-on task. Dur-

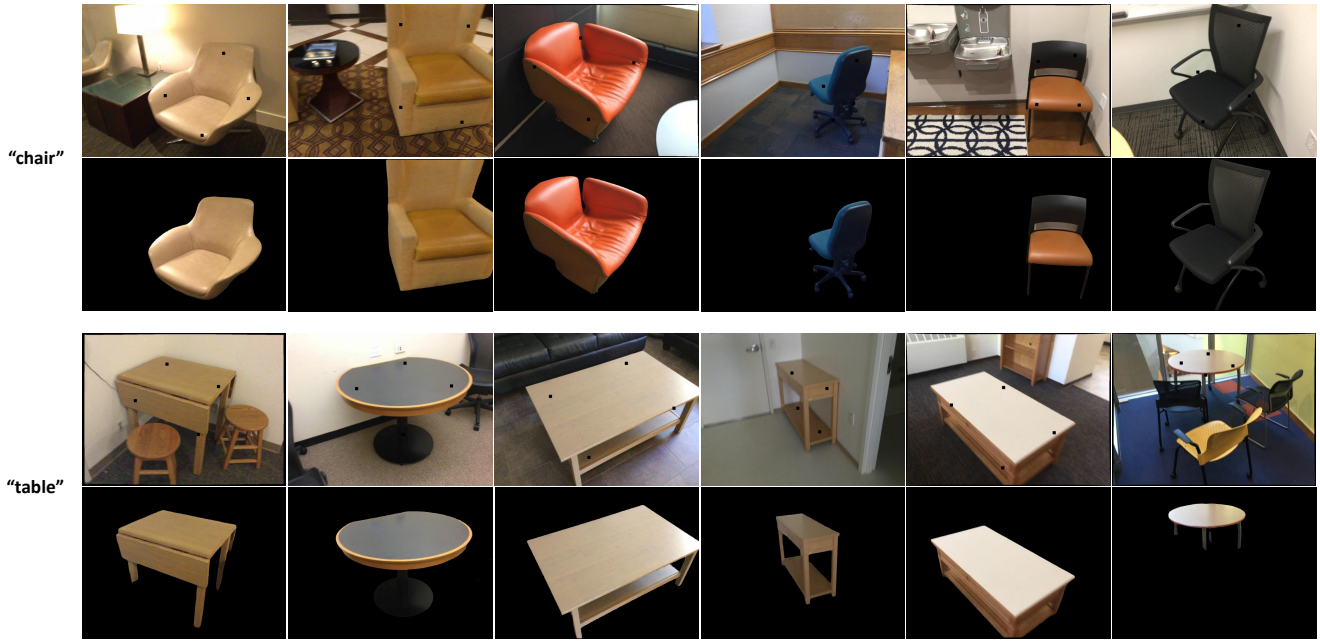

Figure 11. Construction of the multi-view objects dataset; With a textual input like "chair" or "table", the initial prompts are generated automatically for each scene.

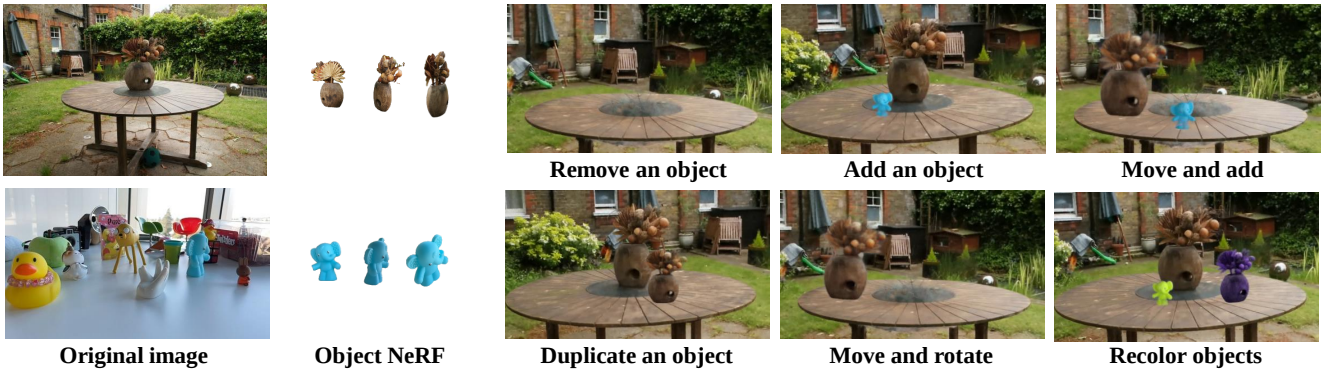

Figure 12. Applications of Obj-NeRF: editing NeRFs with object removal, add-on, movement, rotation, and color changing.

ing this process, we can also apply the rotation, resize, and other transformations to the object NeRF. Nerfstudio and blender [28] provides a user-friendly way to construct the required camera poses during the editing procedure.

Removal. After obtaining the multi-view segmentation for each image, we can add a reverse alpha channel to the original image, representing the background environment without the foreground object. During the NeRF training procedure, the obstructed areas by the foreground object in one view can be inferred by other views. In this way, the object removal NeRF can be realized.

6. Conclusions

In this paper, we propose a comprehensive pipeline for constructing segmented object NeRFs, combining the 2D segmentation proficiency of SAM and the 3D reconstruction ability of NeRF. Without dependence on full-scene NeRF, our proposed Obj-NeRF is widely applicable to various scenarios. Compared to existing works, our method outperforms on reconstruction quality and the extensiveness of application environments. Additionally, we provide a feasible way to construct a large object NeRF dataset, which is verified in some applications like NeRF editing tasks. For future works, the constructed object NeRF dataset can be extended to 3D generation tasks.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 2

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 6

[3] 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*, pages 213–229. Springer, 2020. 2

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 1

[5] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Xiaopeng Zhang, and Qi Tian. Segment anything in 3D with NeRFs. *arXiv preprint arXiv:2304.12308*, 2023. 1, 2, 6, 7

[6] Xiaokang Chen, Jiaxiang Tang, Diwen Wan, Jingbo Wang, and Gang Zeng. Interactive segment anything NeRF with feature imitation. *arXiv preprint arXiv:2305.16233*, 2023. 1, 2

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 3, 5, 6

[8] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 1, 2, 5, 6

[9] Kaspar Fischer. Introduction to alpha shapes. *Department of Information and Computing Sciences, Faculty of Science, Utrecht University*, 17, 2000. 4

[10] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 2

[11] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. LERF: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 1, 2, 6

[12] 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, 3

[13] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 2

[14] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023. 1, 2, 5

[15] 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. 5

[16] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 6

[17] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3

[18] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G Derpanis, Jonathan Kelly, Marcus A Brubaker, Igor Gilitschenski, and Alex Levinshtein. SPIn-NeRF: Multiview segmentation and perceptual inpainting with neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20669–20679, 2023. 2

[19] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 1, 2, 6

[20] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022. 2

[21] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[22] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 2

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language super-

vision. In *Proceedings of International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1

[24] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12892–12901, 2022. 2

[25] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 3

[26] Kristina P Sinaga and Miin-Shen Yang. Unsupervised k-means clustering algorithm. *IEEE Access*, 8:80716–80727, 2020. 5

[27] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022. 1, 2, 6

[28] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 2, 6, 8

[29] Jiaxiang Tang, Xiaokang Chen, Jingbo Wang, and Gang Zeng. Compressible-composable NeRF via rank-residual decomposition. *Advances in Neural Information Processing Systems*, 35:14798–14809, 2022. 2

[30] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. RIO: 3D object instance relocalization in changing indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7658–7667, 2019. 3, 6

[31] Chen Wang, Xian Wu, Yuan-Chen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. NeRF-SR: High quality neural radiance fields using supersampling. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6445–6454, 2022. 2

[32] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023. 3, 6

[33] Youtan Yin, Zhoujie Fu, Fan Yang, and Guosheng Lin. OR-NeRF: Object removing from 3D scenes guided by multiview segmentation with neural radiance fields. *arXiv preprint arXiv:2305.10503*, 2023. 2, 3, 4

[34] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. NeRF-editing: geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364, 2022. 2

[35] 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