

# NeRF-Det: Learning Geometry-Aware Volumetric Representation for Multi-View 3D Object Detection

Chenfeng Xu<sup>1</sup> Bichen Wu<sup>2</sup> Ji Hou<sup>2</sup> Sam Tsai<sup>2</sup> Ruilong Li<sup>1</sup> Jialiang Wang<sup>2</sup> Wei Zhan<sup>1</sup>  
Zijian He<sup>2</sup> Peter Vajda<sup>2</sup> Kurt Keutzer<sup>1</sup> Masayoshi Tomizuka<sup>1</sup>

<sup>1</sup>University of California, Berkeley <sup>2</sup>Meta

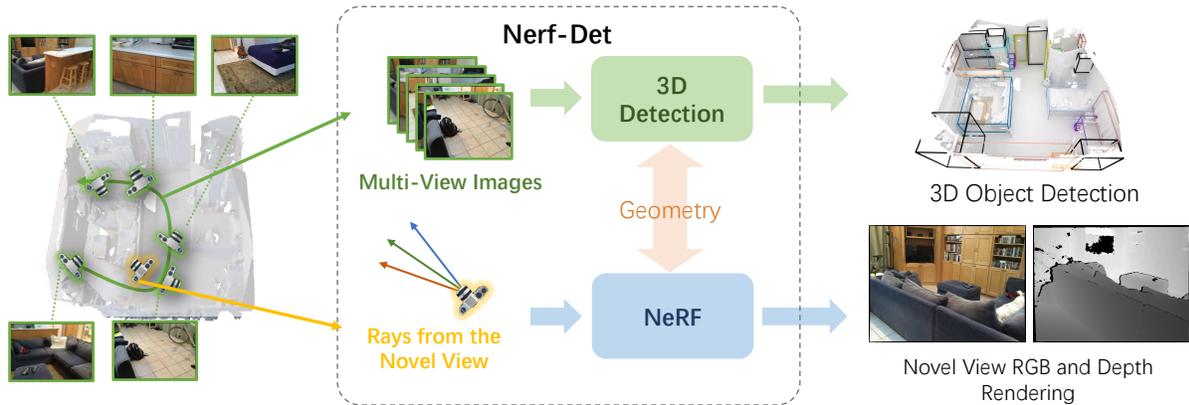


Figure 1: NeRF-Det aims to detect 3D objects with only RGB images as input. To enhance detection, we propose to embed a NeRF branch with our designed synergy. The two joint branches share the same geometry representation and are trained end-to-end, which helps achieve state-of-the-art accuracy on multi-view indoor RGB-only 3D detection, and additionally enables a generalizable novel view synthesis on new scenes without per-scene optimization.

## Abstract

We present *NeRF-Det*, a novel method for indoor 3D detection with posed RGB images as input. Unlike existing indoor 3D detection methods that struggle to model scene geometry, our method makes novel use of NeRF in an end-to-end manner to explicitly estimate 3D geometry, thereby improving 3D detection performance. Specifically, to avoid the significant extra latency associated with per-scene optimization of NeRF, we introduce sufficient geometry priors to enhance the generalizability of NeRF-MLP. Furthermore, we subtly connect the detection and NeRF branches through a shared MLP, enabling an efficient adaptation of NeRF to detection and yielding geometry-aware volumetric representations for 3D detection. Our method outperforms state-of-the-arts by 3.9 mAP and 3.1 mAP on the ScanNet and ARKITScenes benchmarks, respectively. We provide extensive analysis to shed light on how NeRF-Det works. As a result of our joint-training design, NeRF-Det is able to

generalize well to unseen scenes for object detection, view synthesis, and depth estimation tasks without requiring per-scene optimization. Code is available at <https://github.com/facebookresearch/NeRF-Det>.

## 1. Introduction

In this paper, we focus on the task of indoor 3D object detection using posed **RGB images**. 3D object detection is a fundamental task for many computer vision applications such as robotics and AR/VR. The algorithm design depends on input sensors. In the past few years, most 3D detection works focus on both RGB images and depth measurements (depth images, point-clouds, etc.). While depth sensors are widely adopted in applications such as autonomous driving, they are not readily available in most AR/VR headsets and mobile phones due to cost, power dissipation, and form factor constraints. Excluding depth input, however, makes 3D object detection significantly more challenging, since we need to understand not only the semantics, but also the underlying scene geometry from RGB-only images.

This work was done when Chenfeng was an intern at Meta.

To mitigate the absence of geometry, one straightforward solution is to estimate depth. However, depth estimation itself is a challenging and open problem. For example, most monocular depth-estimation algorithms cannot provide accurate metric depth or multi-view consistency [12, 35, 18, 32]. Multi-view depth-estimation algorithms can only estimate reliable depth in textured and non-occluded regions [10, 37].

Alternatively, ImVoxelNet [34] models the scene geometry implicitly by extracting features from 2D images and projecting them to build a 3D volume representation. However, such a geometric representation is intrinsically ambiguous and leads to inaccurate detection.

On the other hand, **Neural Radiance Field** (NeRF) [24, 4, 4] has been proven to be a powerful representation for geometry modeling. However, incorporating NeRF into the 3D detection pipeline is a complex undertaking for several reasons: (i) Rendering a NeRF requires high-frequency sampling of the space to avoid aliasing issues [24], which is challenging in the 3D detection pipeline due to limited resolution volume. (ii) Traditional NeRFs are optimized on a per-scene basis, which is incompatible with our objective of image-based 3D detection due to the considerable latency involved. (iii) NeRF makes full use of multi-view consistency to better learn geometry during training. However, a simple stitch of first-NeRF-then-perception [40, 16, 17] (i.e., reconstruction-then-detection) does not bring the advantage of multi-view consistency to the detection pipeline.

To mitigate the issue of ambiguous scene geometry, we propose NeRF-Det to explicitly model scene geometry as an opacity field by jointly training a NeRF branch with the 3D detection pipeline. Specifically, we draw inspirations from [44, 53] to project ray samples onto the image plane and extract features from the high-resolution image feature map, rather than from the low-resolution volumes, thereby overcoming the need for high-resolution volumes. To further enhance the generalizability of NeRF model to unseen scenes, we augment the image features with more priors as the input to the NeRF MLP, which leads to more distinguishable features for NeRF modeling. Unlike previous works that build a simple stitch of NeRF-then-perception, we connect the NeRF branch with the detection branch through a *shared* MLP that predicts a density field, subtly allowing the gradient of NeRF branches to back-propagate to the image features and benefit the detection branch during training. We then take advantage of the uniform distribution of the volume and transform the density field into an opacity field and multiply it with the volume features. This reduces the weights of empty space in the volume feature. Then, the geometry-aware volume features are fed to the detection head for 3D bounding box regression. It is worth noting that during inference, the NeRF branch is removed, which minimizes the additional overhead to the original detector.

Our experiments show that by explicitly modeling the geometry as an opacity field, we can build a much better volume representation and thereby significantly improve 3D detection performance. Without using depth measurements for training, we improve the state-of-the-art by 3.9 and 3.1 mAP on the ScanNet and the ARKITScenes datasets, respectively. Optionally, if depth measurements are also available for training, we can further leverage depth to improve the performance, while our inference model still does not require depth sensors. Finally, although novel-view synthesis and depth estimation are not our focus, our analysis reveals that our method can synthesize reasonable novel-view images and perform depth prediction without per-scene optimization, which validates that our 3D volume features can better represent scene geometry.

## 2. Related Work

**3D Detection in Indoor Scene.** 3D detection utilizes various methods depending on inputs, and has achieved great success on point cloud [25, 28, 29, 31, 54] and voxel representations [51, 14, 13, 8, 15]. 3D-SIS [13] uses anchors to predict 3D bounding boxes from voxels fused by color and geometric features. The widely-used VoteNet [30] proposes hough voting to regress bounding box parameters from point features. However, depth sensors are not always readily available on many devices due to its huge power consumption, such as on VR/AR headsets. To get rid of sensor depth, Panoptic3D [6] operates on point clouds extracted from predicted depth. Cube R-CNN [3] directly regresses 3D bounding boxes from a single 2D image.

Comparably, the multi-view approach is also not limited by depth sensors and is more accurate. However, the current state-of-the-art multi-view method [34] fuses the image naively by duplicating pixel features along the ray, which does not incorporate a sufficient amount of geometric clues. To address this, we leverage NeRF to embed geometry into the volume for better 3D detection.

**3D Reconstruction with Neural Radiance Field.** Neural Radiance Field (NeRF) [24] is a groundbreaking 3D scene representation that emerged three years ago, and has proven to be powerful for reconstructing 3D geometry from multi-view images [24, 1, 50, 55, 42]. Early works [24, 1, 20, 27, 52] directly optimize for per-scene density fields using differentiable volumetric rendering [23]. Later, NeuS [43] and VolSDF [50] improve the reconstruction quality by using SDF as a replacement of density as the geometry representation. Recently, Ref-NeRF [39] proposes to use reflective direction for better geometry of glossy objects. Aside from aforementioned per-scene optimization methods, there are also works that aim to learn a generalizable NeRF from multiple scenes, such as IBRNet [44] and MVS-NeRF [4], which predict the density and color at each 3D location conditioned on the pixel-aligned image features. Despite this

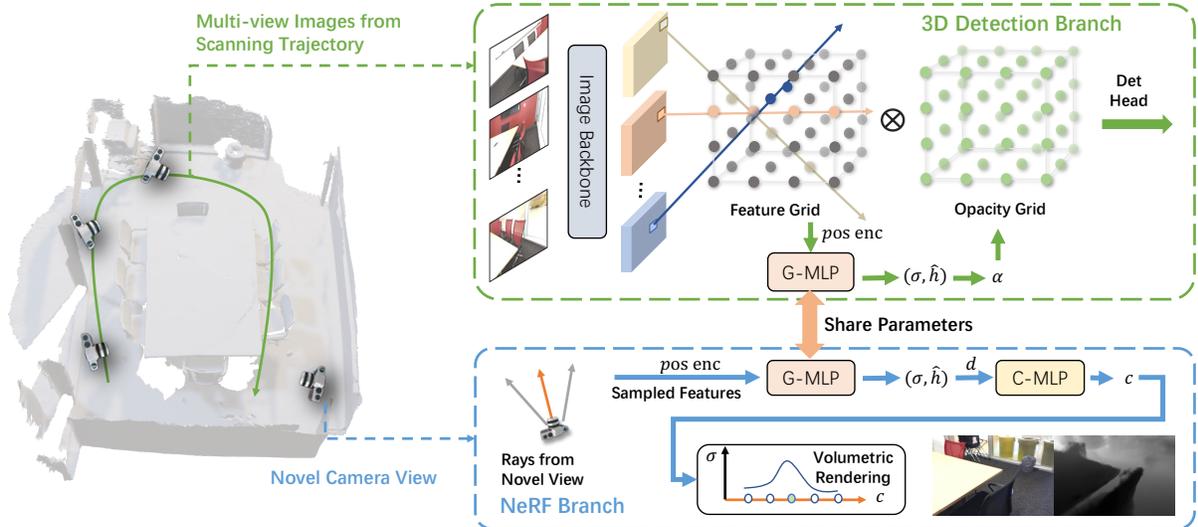


Figure 2: **The framework of NeRF-Det.** Our method leverages NeRF to learn scene geometry by estimating opacity grids. With the shared geometry-MLP (G-MLP), the detection branch can benefit from NeRF in estimating opacity fields and is thus able to mask out free space and mitigate the ambiguity of the feature volume.

amazing progress, all of these methods only focus on a single task – either novel-view synthesis or surface reconstruction. In contrast, we propose a novel method that incorporates NeRF seamlessly to improve 3D detection.

**NeRF for Perception.** Being able to capture accurate geometry, NeRF has gradually been incorporated into perception tasks such as classification [17], segmentation [40, 56], detection [16], instance segmentation [19], and **panoptic** segmentation [9]. However, most of them [17, 16, 40] follow the pipeline of first-NeRF-then-perception, which not only creates extra cost for perception tasks but also does not sufficiently use volumetric renderings to benefit perception during training. Besides, [56, 9] demonstrate that NeRF can significantly improve label efficiency for semantic segmentation by ensuring multi-view consistency and geometry constraints. Our proposed NeRF-Det method incorporates NeRF to ensure multi-view consistency for 3D detection. Through joint end-to-end training for NeRF and detection, no extra overheads are introduced during inference.

### 3. Method

Our method, termed as NeRF-Det, uses posed RGB images for indoor 3D object detection by extracting image features and projecting them into a 3D volume. We leverage NeRF to help infer scene geometry from 2D observations. To achieve this, we **entangle** the 3D object detection and NeRF with a *shared* MLP, with which the multi-view constraint in NeRF can enhance the geometry estimation for the detection, as shown in Fig. 2.

#### 3.1. 3D Detection Branch

In the 3D detection branch, we input posed RGB frames to the 2D image backbone, denoting the images as  $I_i \in \mathbf{R}^{H_i \times W_i \times 3}$  and the corresponding intrinsic matrix and extrinsic matrix as  $K \in \mathbf{R}^{3 \times 3}$  and  $R_i \in \mathbf{R}^{3 \times 4}$ , where  $i = 1, 2, 3, \dots, T$  and  $T$  is the total number of views. We follow [34], which uses an FPN [22] to fuse multi-stage features and use high resolution features, denoted as  $F_i \in \mathbf{R}^{C \times H/4 \times W/4}$ , to generate a 3D feature volume.

We create the 3D feature volume by attaching 2D features from each image to their corresponding positions in 3D. We establish a 3D coordinate system, with the z-axis denoting height, and the x- and y-axis denoting two horizontal dimensions. Then, we build a 3D grid of with  $N_x \times N_y \times N_z$  voxels. For each voxel center with coordinate  $\mathbf{p} = (x, y, z)^T$ , we project it to view- $i$  to obtain the 2D coordinates as

$$(u'_i, v'_i, d_i)^T = K' \times R_i \times (\mathbf{p}, 1)^T, \quad (1)$$

where  $(u_i, v_i) = (u'_i/d_i, v'_i/d_i)$  is to the pixel coordinate of  $\mathbf{p}$  in view- $i$ .  $K'$  is the scaled intrinsic matrix, considering the feature map downsampling. After building this correspondence, we assign 3D features as

$$V_i(\mathbf{p}) = \text{interpolate}((u_i, v_i), F_i), \quad (2)$$

where  $\text{interpolate}()$  looks up the feature in  $F_i$  at location  $(u_i, v_i)$ . Here we use nearest neighbor interpolation. For  $\mathbf{p}$  that are projected outside the boundary of the image, or behind the image plane, we discard the feature and set  $V_i(\mathbf{p}) = \mathbf{0}$ . Intuitively, this is equivalent to shooting a ray

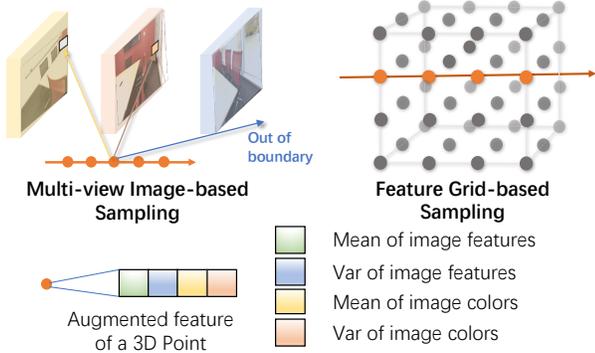


Figure 3: Different feature sampling strategies. Give a ray from a novel view, we can project the 3D points on the ray to the multi-view image features, as presented in the left part, and attach their mean/variance as well as the corresponding RGB to the 3D point. On the other hand, we can also sample features from feature grids, as shown in the right part.

from the origin of camera- $i$  through pixel  $(u_i, v_i)$ , and for all voxels that are on the ray, we scatter image features to the voxel in  $V_i$ . Next, we aggregate multi-view features by simply averaging all *effective* features as done in ImVoxelNet [34]. Letting  $M_p$  denote the number of effective 2D projections, we compute  $V^{avg}(p) = \sum_{i=1}^{M_p} V_i(p)/M_p$ .

However, volume features generated this way “over populate” projection rays without considering empty spaces and other geometric constraints. This makes the 3D representation ambiguous and causes detection errors. To mitigate this issue, we propose to incorporate a NeRF branch to improve learning geometry for the detection branch.

### 3.2. NeRF Branch

**Feature Sampling.** NeRF [24] is a neural rendering method for novel view synthesis. However, previous NeRFs sample features from the 3D volume with high resolution, such as  $128^3$  [24]. In 3D detection, we use a lower grid resolution of  $40 \times 40 \times 16$ , which suffers from the aliasing issue and results in degradation of geometry learning. To mitigate this, we draw inspiration from [44, 53] and sample features from higher resolution 2D image feature maps, typically of size  $160 \times 120$ , as shown in Fig. 3. Specifically, we first sample points along the ray originated from the camera, i.e.,  $\mathbf{r}(t) = \mathbf{o} + t \times \mathbf{d}$  where  $\mathbf{o}$  is the ray origin and  $\mathbf{d}$  is the view direction. For a coordinate  $\mathbf{p}$  sampled on the ray and a viewing direction  $\mathbf{d}$ , we can compute the color  $\mathbf{c}(\mathbf{p}, \mathbf{d})$  and density  $\sigma(\mathbf{p})$  as:

$$\sigma(\mathbf{p}), \hat{\mathbf{h}}(\mathbf{p}) = G\text{-MLP}(\bar{\mathbf{V}}(\mathbf{p}), \gamma(\mathbf{p})), \quad (3)$$

$$\mathbf{c}(\mathbf{p}, \mathbf{d}) = C\text{-MLP}(\hat{\mathbf{h}}(\mathbf{p}), \mathbf{d}). \quad (4)$$

$\bar{\mathbf{V}}(\mathbf{p})$  is ray features aggregated and *augmented* from multi-view features, and  $\gamma(\mathbf{p})$  is the position encoding same as [24] while  $\hat{\mathbf{h}}$  is latent features. The first MLP is termed

*G-MLP* for estimating geometry and the second MLP is termed *C-MLP* for estimating color. For activations, we follow [24] and use ReLU for the density  $\sigma(\mathbf{p})$  and sigmoid for the color  $\mathbf{c}(\mathbf{p}, \mathbf{d})$ .

**Augmenting Features.** Although it is similar to [44, 53] that use image features, it is still difficult to make *G-MLP* estimate accurate geometry across different scenes by simply averaging features from multi-view features as detection branch does. Therefore, we propose to incorporate more priors to help optimize *G-MLP*. Beyond averaging the projected features, we augment the sampled features with the variance from multiple views  $V^{var}(p) = \sum_{i=1}^{M_p} (V_i(p) - V^{avg}(p))^2 / M_p$ . The variance of the color features is able to roughly describe the occupancy of the 3D field, which has been widely used as cost volume in multi-view stereo depth estimation [49]. If the 3D location  $\mathbf{p}$  is occupied, the variance of the observed features should be low under the assumption that the scene only contains Lambertian surfaces. On the other hand, if the location is in free space, different appearances would be observed from different views, and therefore the variance of color features would be high. Compared to naive average of features, variance provides a better prior for estimating scene geometry.

In addition to extracted deep features, which are trained to be invariant to subtle appearance changes, we also augment pixel RGB values into sampled features on the ray. This is inspired by IBRNet [44] where they attach the pixel RGB to the input to the MLP for better appearance modeling. We compute the mean and variance for pixel RGB values in the same way as for deep features. In all, the augmented feature  $\bar{\mathbf{V}}$  is represented as a concatenation of  $\{V^{avg}, V^{var}, RGB^{avg}, RGB^{var}\}$ , as shown in Fig. 3. The sampled augmented features are passed into NeRF MLPs (Eq. 3) to generate the density and color. We use volumetric rendering to produce the final pixel color and depth,

$$\hat{\mathbf{C}}(r) = \sum_{i=1}^{N_p} T_i \alpha_i \mathbf{c}_i, D(r) = \sum_{i=1}^{N_p} T_i \alpha_i t_i, \quad (5)$$

where  $T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_t)$ ,  $\alpha_i = 1 - \exp(-\sigma_i \delta_t)$ ,  $t_i$  is the distance between sampled  $i$ th point between the camera,  $\delta_t$  is the distance between sampled points of rays.

### 3.3. Estimating Scene Geometry

We use an opacity field to model the scene geometry. The opacity field is a volumetric representation that reflects the probability of an object’s presence in a particular area, i.e., if there is an object that cannot be seen through, the opacity field would be 1.0 in that area. To generate the opacity field, we follow the same process of augmenting features in the detection branch as we do in the NeRF branch. A key ingredient to the approach is sharing the G-MLP learned

in the NeRF branch with the detection branch. This enables two important capabilities. Firstly, the shared G-MLP subtly connects the two branches, allowing gradients from the NeRF branch to back-propagate and benefit the detection branch during training. Secondly, during inference, we can directly input the augmented volume features of 3D detection into the shared G-MLP since both inputs from two branches are augmented features. The output of G-MLP is the density represented as  $\sigma(\mathbf{p}) = \text{G-MLP}(\tilde{V}(\mathbf{p}), \gamma(\mathbf{p}))$ , where  $\sigma(\mathbf{p}) \in [0, \infty]$ . Note that  $\mathbf{p}$  is the center position of each voxel in the volume of the detection branch.

Next, we aim to transform the density field into the opacity field by  $\alpha(\mathbf{p}) = 1 - \exp(-\sigma(\mathbf{p}) \times \delta t)$ . However, it is infeasible to calculate  $\delta t$  as we can not decide the ray direction and calculate the point distance within the undirected volume in the detection branch. Here we subtly take advantage of the uniform distribution of the voxels in the space. Thus, the  $\delta t$  in the volumetric rendering equation can be canceled out as it is a constant. So obtaining opacity can be reduced to  $\alpha(\mathbf{p}) = 1 - \exp(-\sigma(\mathbf{p}))$ . After generating the opacity field, we multiply it with the feature grid  $V^{avg}$  for 3d detection, denoted as  $\alpha(\mathbf{p}) \times V^{avg}(\mathbf{p})$ .

### 3.4. 3D Detection Head and Training Objectives

Our geometry-aware volumetric representation can fit to most detection heads. For fair comparison and the simplicity, we use the same indoor detection head as ImVoxelNet [34], in which we select 27 location candidates for each objects and we use three convolution layers to predict the categories, the locations and the centerness.

We jointly train detection and NeRF branches end-to-end. No per-scene optimization for the NeRF branch is performed in test time. For the detection branch, we supervise training with ground truth 3D bounding boxes following ImVoxelNet [34], which computes three losses: focal loss for classification  $L_{cls}$ , centerness loss  $L_{cntr}$ , and localization loss  $L_{loc}$ . For the NeRF branch, we use a photo-metric loss  $L_c = \|\hat{C}(r) - \hat{C}_{gt}(r)\|_2$ . When depth ground truth is used, we can further supervise the expected depth of the scene geometry as  $L_d = \|D(r) - D_{gt}(r)\|$  where  $D(r)$  is computed with Equ. 5. The final loss  $L$  is given by

$$L = L_{cls} + L_{cntr} + L_{loc} + L_c + L_d. \quad (6)$$

Even though we *optionally* use depth during training, it is not required in inference. Also our trained network is generalizable to new scenes which are never seen during training.

## 4. Experimental Results

**Implementation details.** Our detection branch mainly follows ImVoxelNet, including backbones, detection head, resolutions and training recipe *etc.* Please refer to supplemental material for more details.

Our implementation is based on MMDetection3D [5]. To the best of our knowledge, we are the first to implement NeRF in MMDetection3D. We are also the first to conduct NeRF-style novel view synthesis and depth estimation on the whole ScanNet dataset, while prior works only test on a small number of scenes [48, 47]. The code will be released for future research.

### 4.1. Main results

**Quantitative results.** We compare NeRF-Det with point-cloud based methods [46, 13, 30], RGB-D based methods [13, 11] and the state-of-the-art RGB-only method ImVoxelNet [34] on ScanNet, as shown in Tab. 1.

With ResNet50 as the image backbone, we observe that NeRF-Det-R50-1x outperforms ImVoxelNet-R50-1x by 2.0 mAP. On top of that, NeRF-Det with depth supervision, denoted as NeRF-Det-R50-1x\*, further improves the detection performance by 0.6 mAP compared to only RGB supervision NeRF-Det-R50-1x.

We denote the total training iterations of ImVoxelNet from the official code as 1x in the Tab. 1. Yet the 1x setting only iterates through each scene roughly 36 times, which is far from sufficient for optimizing the NeRF branch, which requires thousands of iterations to optimize one scene, as indicated in [24, 44, 1, 20]. Thus, we further conduct experiments with 2x training iterations to fully utilize the potential of NeRF, and we observe that NeRF-Det-R50-2x reaches 52.0 mAP, surpassing ImVoxelNet by 3.6 mAP under the same setting (ImVoxelNet-R50-2x). It is worth mentioning that we do not introduce any extra data/labels to get such an improvement. If we further use depth supervision to train the NeRF branch (NeRF-Det-R50-2x\*), the detection branch is further improved by 1.3 in mAP@.50 as shown in Tab. 3. This validates better geometry modeling (via depth supervision) could be helpful to the 3D detection task. While NeRF-Det provides an efficient method to incorporate depth supervision during the training process, introducing depth supervision in ImVoxelNet is difficult.

Moreover, when substituting ResNet50 with ResNet101, we achieve 52.9 mAP@.25 on ScanNet, which outperforms ImVoxelNet on the same setting over 3.9 points. With depth supervision, NeRF-Det-R101-2x\* reduces the gap between RGB-based method ImVoxelNet [34] and point-cloud based method VoteNet [30] by half (from 10 mAP  $\rightarrow$  5 mAP). Besides, we conduct experiments on the ARKitScenes (see Tab. 2). The 3.1 mAP improvement further demonstrates the effectiveness of our proposed method.

**Qualitative results.** We visualize the predicted 3D bounding boxes from NeRF-Det-R50-2x on scene meshes, as shown in Fig. 4. We observe that the proposed method gets accurate detection prediction even on the ex-

Table 1: 3D Detection with multi-view RGB inputs on ScanNet. The first block of the table includes point-cloud based and RGBD-based methods, and the rest are multi-view RGB-only detection methods. † means our reproduction of ImVoxelNet [34] using the official code. \* indicates the NeRF-Det with depth supervision. 1x and 2x refer that we train the model with the same and two times of the training iteration wrt. the original iterations of ImVoxelNet, respectively.

Methods	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	mAP@.25
Seg-Cluster [46]	11.8	13.5	18.9	14.6	13.8	11.1	11.5	11.7	0.0	13.7	12.2	12.4	11.2	18.0	19.5	18.9	16.4	12.2	13.4
Mask R-CNN [11]	15.7	15.4	16.4	16.2	14.9	12.5	11.6	11.8	19.5	13.7	14.4	14.7	21.6	18.5	25.0	24.5	24.5	16.9	17.1
SGPN [46]	20.7	31.5	31.6	40.6	31.9	16.6	15.3	13.6	0.0	17.4	14.1	22.2	0.0	0.0	72.9	52.4	0.0	18.6	22.2
3D-SIS [13]	12.8	63.1	66.0	46.3	26.9	8.0	2.8	2.3	0.0	6.9	33.3	2.5	10.4	12.2	74.5	22.9	58.7	7.1	25.4
3D-SIS (w/ RGB) [13]	19.8	69.7	66.2	71.8	36.1	30.6	10.9	27.3	0.0	10.0	46.9	14.1	53.8	36.0	87.6	43.0	84.3	16.2	40.2
VoteNet [30]	36.3	87.9	88.7	89.6	58.8	47.3	38.1	44.6	7.8	56.1	71.7	47.2	45.4	57.1	94.9	54.7	92.1	37.2	58.7
FCAF3D [33]	57.2	87.0	95.0	92.3	70.3	61.1	60.2	64.5	29.9	64.3	71.5	60.1	52.4	83.9	99.9	84.7	86.6	65.4	71.5
CAGroup3D [41]	60.4	93.0	95.3	92.3	69.9	67.9	63.6	67.3	40.7	77.0	83.9	69.4	65.7	73.0	100.0	79.7	87.0	66.1	75.12
ImVoxelNet-R50-1x	28.5	84.4	73.1	<b>70.1</b>	51.9	32.2	15.0	34.2	1.6	29.7	66.1	<b>23.5</b>	57.8	43.2	92.4	54.1	74.0	34.9	48.1
ImVoxelNet†-R50-1x	31.6	81.8	74.4	69.3	<b>53.6</b>	29.7	12.9	<b>50.0</b>	1.3	32.6	<b>69.2</b>	12.7	54.6	31.8	<b>93.1</b>	<b>55.8</b>	68.2	31.8	47.5
NeRF-Det-R50-1x	32.7	82.6	74.3	67.6	52.3	34.4	17.3	40.1	<b>2.0</b>	49.2	67.4	20.0	57.2	41.0	90.9	52.3	74.0	<b>33.7</b>	49.5 (+2.0)
NeRF-Det-R50-1x*	<b>32.7</b>	<b>84.7</b>	<b>74.6</b>	62.7	52.7	<b>35.1</b>	<b>17.7</b>	48.4	0.0	<b>49.8</b>	64.6	18.5	<b>60.3</b>	<b>48.3</b>	90.7	51.0	<b>76.8</b>	30.4	<b>50.1</b> (+2.6)
ImVoxelNet†-R50-2x	34.5	83.6	72.6	71.6	54.2	30.3	14.8	42.6	<b>0.8</b>	40.8	65.3	18.3	52.2	40.9	90.4	<b>53.3</b>	74.9	33.1	48.4
NeRF-Det-R50-2x	37.2	<b>84.8</b>	<b>75.0</b>	<b>75.6</b>	51.4	31.8	<b>20.0</b>	40.3	0.1	51.4	69.1	<b>29.2</b>	<b>58.1</b>	<b>61.4</b>	91.5	47.8	<b>75.1</b>	33.6	<b>52.0</b> (+3.6)
NeRF-Det-R50-2x*	<b>37.7</b>	84.1	<b>74.5</b>	71.8	<b>54.2</b>	<b>34.2</b>	<b>17.4</b>	<b>51.6</b>	0.1	<b>54.2</b>	<b>71.3</b>	16.7	54.5	55.0	<b>92.1</b>	50.7	<b>73.8</b>	<b>34.1</b>	51.8 (+3.4)
ImVoxelNet†-R101-2x	30.9	84.0	<b>77.5</b>	73.3	56.7	35.1	<b>18.6</b>	47.5	0.0	44.4	65.5	19.6	58.2	32.8	92.3	40.1	77.6	28.0	49.0
NeRF-Det-R101-2x	36.8	<b>85.0</b>	77.0	73.5	56.9	<b>36.7</b>	14.3	<b>48.1</b>	0.8	<b>49.7</b>	<b>68.3</b>	23.5	54.0	<b>60.0</b>	96.5	49.3	78.4	38.4	52.9 (+3.9)
NeRF-Det-R101-2x*	<b>37.6</b>	84.9	76.2	<b>76.7</b>	<b>57.5</b>	36.4	17.8	47.0	<b>2.5</b>	49.2	52.0	<b>29.2</b>	<b>68.2</b>	49.3	<b>97.1</b>	<b>57.6</b>	<b>83.6</b>	<b>35.9</b>	<b>53.3</b> (+4.3)

Table 2: Comparison experiments "whole-scene" of ARKITScenes validation set.

Methods	cab	fridg	shlf	stove	bed	sink	wshr	tolt	bthb	oven	dshwshr	frplce	stool	chr	tbl	TV	sofa	mAP@.25
ImVoxelNet-R50	32.2	34.3	4.2	0.0	64.7	20.5	15.8	68.9	80.4	9.9	4.1	10.2	<b>0.4</b>	<b>5.2</b>	11.6	3.1	35.6	23.6
NeRF-Det-R50	<b>36.1</b>	<b>40.7</b>	<b>4.9</b>	<b>0.0</b>	<b>69.3</b>	<b>24.4</b>	<b>17.3</b>	<b>75.1</b>	<b>84.6</b>	<b>14.0</b>	<b>7.4</b>	<b>10.9</b>	0.2	4.0	<b>14.2</b>	<b>5.3</b>	<b>44.0</b>	<b>26.7</b> (+3.1)

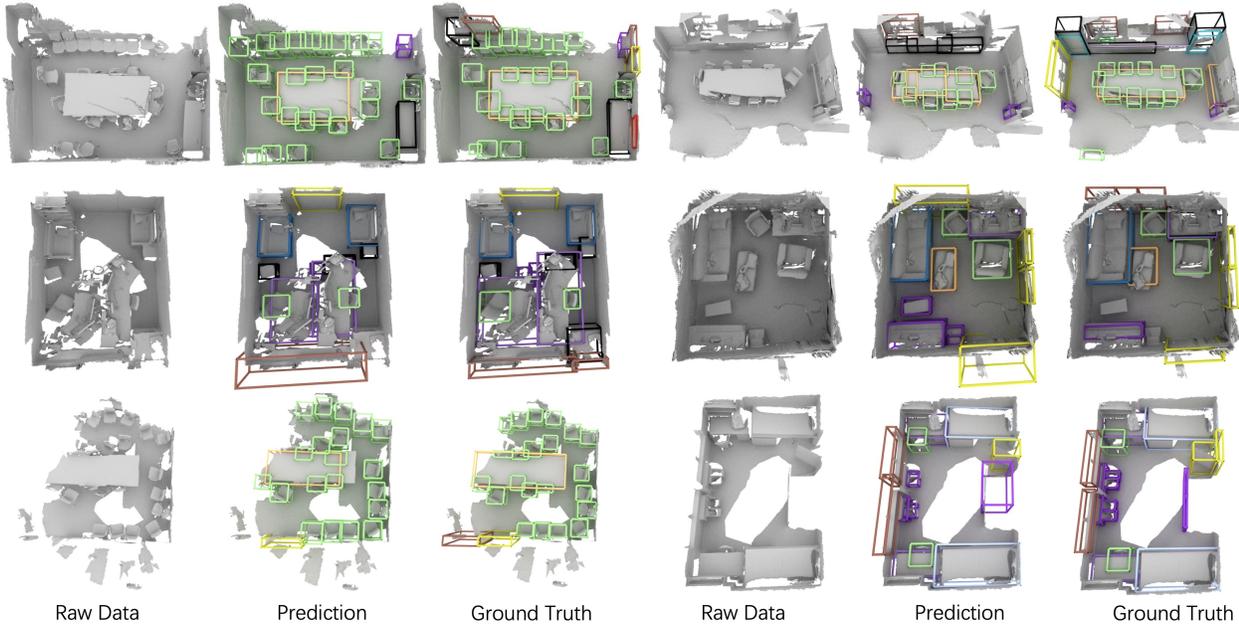


Figure 4: Qualitative results of the predicted 3D bounding box on top of NeRF-Det-R50-2x. Note that our approach takes only posed RGB images as input, the reconstructed mesh is only for visualization purpose.

tremely dense scenes, *e.g.*, the first row and the left of the third row. The chairs are crowded in the room, and some of which are inserted into tables and are heavily occluded. Yet our method is able to detect them accurately. On the other

hand, NeRF-Det can also tackle multiple scales, as shown in the second row and the right side of the third row, in which there are variant objects with difference sizes, like garbage bins, chairs, tables, doors, windows, and sofas *etc.*

Table 3: Ablation on scene geometry modelings. GT-Depth indicates ground truth depth for placing 2D features in 3D volume. NeuralRecon-Depth indicates NeuralRecon [38] pre-trained on ScanNetV2 is used to predict the depth. Depth predictions are used in both training and inference.

Methods	mAP@.25	mAP@.50
ImVoxelNet-R50-2x (baseline)	48.4	23.7
GT-Depth-R50-2x (upper-bound)	54.5 (+6.1)	28.2 (+4.5)
NeuralRecon-Depth-R50-2x	48.8 (+0.4)	21.4 (-2.3)
Cost-Volume-R50-2x (sigmoid)	49.3 (+0.9)	24.4 (+0.7)
NeRF-Det-R50-2x (w/o NeRF)	49.2 (+0.8)	24.6 (+0.9)
NeRF-Det-R50-2x	52.0 (+3.6)	26.1 (+2.5)
NeRF-Det-R50-2x*	51.8 (+3.4)	27.4 (+3.7)

**Analysis of scene geometry modeling.** As stated in the method section, we mitigate the ambiguity of the volume representation by learning an opacity field. Furthermore, we explore different scene geometry modeling methods, from using depth map to cost volume [38, 49], in Tab. 3.

*Using the Depth Map.* In this experiment, we assume we have access to depth maps during *both training and inference*. When building the voxel feature grid, instead of scattering the features on all points along the ray, we only place features to a single voxel cell according to the depth maps. Intuitively, this leads to less ambiguity in the volume representation, so we should observe better performance for detection. As a proof of concept, we first use ground-truth depth that comes with the dataset. This serves as an upper-bound for NeRF-Det as it provides a perfect geometry modeling. It indeed achieves a high detection accuracy of 54.5 mAP@.25 and 28.2 mAP@.50 (see second row), improving the baseline by 6.1 mAP@.25 and 4.5 mAP@.40. But in practice we cannot get access to the ground-truth depth maps. Thus, we try instead to render depth maps using out-of-box geometry reconstruction from NeuralRecon [38].

The results are shown in the third row of Tab. 3. We observe that the depth estimation from NeuralRecon significantly degenerates the detection performance by 2.3 mAP@.50 as compared to plain ImVoxelNet, demonstrating that the estimation error of depth maps propagates that inaccuracy through the detection pipeline.

*Cost Volume.* Next, we compare our method with cost-volume based methods [49, 26]. A common way to compute cost volume is using covariance [49] between source view and reference view. This is similar to our method which calculates the variance of multiple input views. Following [49], we first use several 3D convolution layers to encode the cost volume, then get the probability volume via sigmoid, and multiply it with the feature volume  $V^{avg}$ . The results are in the fourth row of Tab. 3. We can see that the cost-volume based method improves ImVoxelNet by 0.9 mAP@.25 and 0.7 mAP@.50 respectively. It is noteworthy to mention that if we remove the NeRF branch, our method is very similar to a cost-volume-based method with the dif-

Table 4: The first group represents using NeRF-RPN training set, and the second group represents using ScanNet training set. The latency is measured on one V100.

Method	AP25	AP50	Latency
NeRF-RPN-R50[16] (NeRF-then-det)	33.13	5.12	~846.8s
NeRF-Det-R50 (joint NeRF-and-Det)	35.13	7.80	0.554s
NeRF-Det-R50† (joint NeRF-and-Det)	61.48	25.45	0.554s

ferences being: 1) we augment the variance in the cost volume by mean volume and color volume, 2) we use the MLP and opacity function instead of sigmoid to model the scene geometry. The result is shown in the fifth row of Tab. 3. We observe that the result is very close to a cost-volume-based method and that both ours and the cost-volume method lead to improvement over ImVoxelNet.

In contrast to explicitly estimating the depth and cost volume, we leverage NeRF to estimate the opacity field for the scene geometry. As shown in the gray part of Tab. 3, with the opacity field modeled using NeRF, the performance is significantly improved by +3.6 mAP@.25 and +2.5 mAP@.50 compared to the baseline. After using depth supervision, NeRF-Det is able to achieve larger improvement with +3.7 mAP@.50 (the last row). As shown in Tab. 3, our method of using NeRF to model scene geometry is more effective than using predicted depth or cost volume.

**Comparison to NeRF-then-Det method.** We compare our proposed NeRF-Det, a joint NeRF-and-Det method, to NeRF-RPN [16] which is a NeRF-then-Det method, as shown in Tab. 4. We choose 4 scenes as validation set that are not included in both NeRF-RPN and ScanNet train set. We use the official code and provided pre-trained models of NeRF-RPN to evaluate AP. Experiments in the first group demonstrate that our proposed joint-NeRF-and-Det paradigm is more effective than first-NeRF-then-Det method NeRF-RPN, with much faster speed.

The second group of Tab. 4 shows that directly using our model (NeRF-Det-R50-2x in Tab. 1) has drastic improvements relative to NeRF-RPN. Although our model is trained on large-scale dataset, we emphasize that this is our advantages since it is hard to apply NeRF-RPN on the whole ScanNet (~1500 scenes) given the heavy overhead.

**Is NeRF branch able to learn scene geometry?** We hypothesize that the better detection performance comes from better geometry. To verify this, we perform novel view synthesis and depth estimation from the prediction of the NeRF branch. The underlying assumption is that if our model has correctly inferred the scene geometry, we should be able to synthesize RGB and depth views from novel camera positions. We first visualize some of the synthesized images and depth estimation in Fig. 5. The image and depth map quality

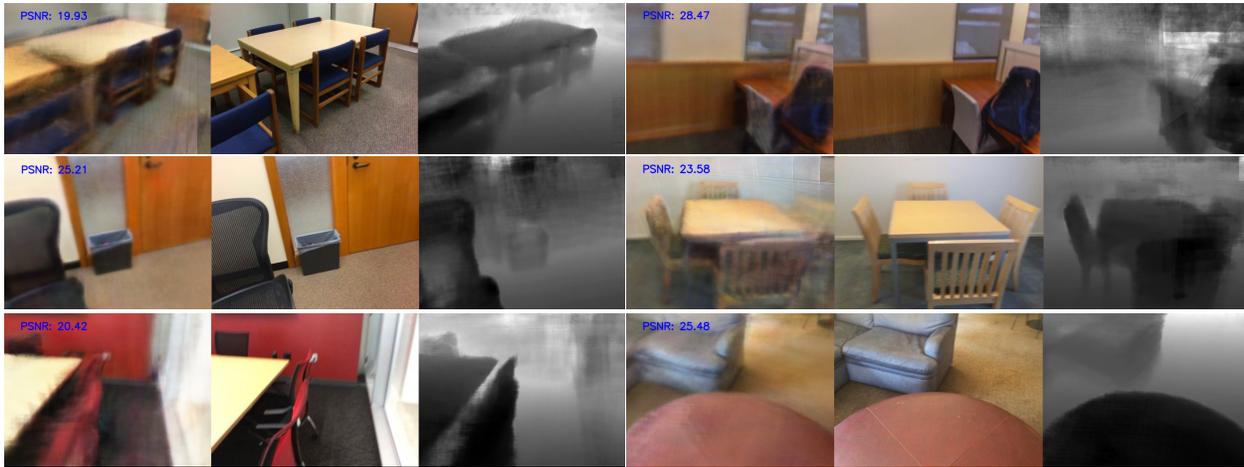


Figure 5: Novel-view synthesis results on top of NeRF-Det-R50-2x\*. For each triplet group, the left figure is the synthesized results, the middle one is the ground truth RGB image, and the right part is the estimated depth map. Note that the visualization is from test set, which is never seen during training.

look reasonable and reflects the scene geometry well.

Quantitatively, we evaluate novel-view synthesis and depth estimation by following the protocol of IBRNet [44]. We select 10 novel views and randomly sample 50 nearby source views. Then, we average the evaluation results of the 10 novel views for each scene, and finally report average results on all 312 scenes in the validation set, as shown in Tab. 8. Although novel-view synthesis and depth estimation are not the main focus of this paper, we achieve an average of 20+ PSNR for novel view synthesis, without per-scene training. For depth estimation, we achieve an RMSE of 0.756 on all scenes. While this performance falls behind state-of-the-art depth estimation methods, we note that [47] reported average RMSE of 1.0125 and 1.0901 using Colmap [36] and vanilla NeRF [24] for depth estimation on selected scenes in ScanNet. This comparison verifies that our method can render reasonable depth.

## 4.2. Ablation Study

We conduct multiple ablation studies on how different components affect the performance of NeRF-Det, including different feature sampling strategies, whether to share G-MLP, different losses and different features feed into the G-MLP. Besides, although novel-view synthesis is not our focus in this paper, we also provide some analysis for the novel-view synthesis results coming out of the NeRF branch, and show how the NeRF branch is influenced by the detection branch during joint training. All experiments in the ablation studies are based on NeRF-Det-R50-1x\*. **Ablation on G-MLP and feature sampling strategy.** As indicated in Sec. 3.2, the key ingredient in our pipeline is a shared G-MLP which enables the constraint of multi-view consistency to be propagated from NeRF branch to detection branch. We conduct an ablation study as shown in Tab. 5. Without shared G-MLP, the performance drops drastically from 50.1 to 48.1 at mAP@.25, shown in the fifth row

Table 5: Ablation study on G-MLP and different ways to sample features onto the ray in the NeRF branch.

Share G-MLP	Sample source	mAP@.25	mAP@.50
✓	3D volume	49.4	24.0
	Multi-view 2D feature	50.1	24.4
	3D volume	48.2	23.8
	Multi-view 2D feature	48.1	23.8

Table 6: Ablation study for loss.

Photo-metric loss	Depth loss	mAP@.25	mAP@.50
✓	✓	50.1	24.4
✓	-	49.4	24.1
-	✓	50.0	24.2
-	-	48.5	23.6

of Tab. 3. In this case, the influence of multi-view consistency is only propagated into image backbone, significantly limiting the improvement created by NeRF.

Moreover, as mentioned in Sec. 3.2, we sample point features along the ray from the multi-view image features instead of the low-resolution volume features. This ablation is shown in Tab. 3. We observe that with shared G-MLP, both approaches outperform the baseline ImVoxelNet, and sampling from image features yields better performance (+0.7 in mAP@0.25) than sampling from volume features. For the novel-view synthesis task using NeRF branch, sampling from image features achieves 20.51 in PSNR comparing to 18.93 with volume sampling.

The fact that the performance of the NeRF branch is proportional to the performance of the detection branch also indicates that better NeRF optimization could lead to better detection results.

**Ablation study on different loss.** We study how different losses work for the NeRF branch, as shown in Tab. 6. It shows that with only photo-metric loss, the performance is closed to purely using depth supervision (third row) in

Table 7: Ablation study on augmented features.

Avg	Var	Color	mAP@.25	mAP@.50
✓	✓	✓	50.1	24.4
✓	✓	-	49.7	24.3
✓	-	-	49.0	23.7

Table 8: Ablation on how detection branch influences novel view synthesis (NVS) and depth estimation (DE) on test set.

Method	PSNR (NVS) ↑	SSIM (NVS) ↑	RMSE (DE) ↓
w/ Det branch	20.51	0.83	0.756
w/o Det branch	20.94	0.84	0.747

term of mAP@.50, indicating that the multi-view RGB consistency already provides sufficient geometry cues to enable the NeRF branch to learn geometry. When using both photo-metric loss and depth loss, the performance can be further improved. When neither photo-metric loss nor depth loss is used (last row), the performance falls back to that of a cost-volume based method. The performance is dropped by 1.2 mAP@.25 and 0.5 mAP@.50, which demonstrates that the NeRF branch is more effective.

**Ablation study on different features.** We then study how different features affect performance, as shown in Tab. 7. The experiment shows that introducing variance features improves performance significantly – over 0.7 mAP@.25 and 0.6 mAP@.50 compared to using only average features, which demonstrates that variance features indeed provide a good geometry prior. Moreover, incorporating image features also improves performance, indicating that low-level color information also provides good geometry cues.

**Ablation Study on detection branch affecting novel view synthesis.** We keep the target views and source views the same with and without the detection branch. Results are shown in Tab. 8. While the NeRF branch significantly improves 3D detection by improving scene geometry modeling, the detection branch does not benefit the NeRF branch. In fact, disabling the detection branch brings a 0.43db improvement. We hypothesize that the detection branch is prone to erase low-level details which are needed for the NeRF, which we aim to address in our future work.

## 5. Conclusion

In this paper, we present NeRF-Det, a novel method that uses NeRF to learn geometry-aware volumetric representations for 3D detection from posed RGB images. We deeply integrate multi-view geometry constraints from the NeRF branch into 3D detection through a subtle shared geometry MLP. To avoid the large overhead of per-scene optimization, we propose leveraging augmented image features as priors to enhance the generalizability of NeRF-MLP. In addition, we sample features from high resolution images instead of volumes to address the need for high-resolution images in NeRF. Our extensive experiments on the ScanNet

and ARKITScene datasets demonstrate the effectiveness of our approach, achieving state-of-the-art performance for indoor 3D detection using RGB inputs. Notably, we observe that the NeRF branch also generalizes well to unseen scenes. Furthermore, our results highlight the importance of NeRF for 3D detection, and provide insights into the key factors for achieving optimal performance in this direction.

## 6. Acknowledgment

We sincerely thank Chaojian Li for the great help on NeuralRecon experiments, Benran Hu for the valuable advice on the NeRF-RPN experiments, Feng (Jeff) Liang and Hang Gao for the insightful discussions, as well as Matthew Yu and Jinhyung Park for the paper proofreading.

## 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, 5
- [2] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. Arkitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. In *NeurIPS*, 2021. 11
- [3] Garrick Brazil, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3d: A large benchmark and model for 3d object detection in the wild. *arXiv preprint arXiv:2207.10660*, 2022. 2
- [4] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021. 2
- [5] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. <https://github.com/open-mmlab/mmdetection3d>, 2020. 5, 12
- [6] Manuel Dahnert, Ji Hou, Matthias Nießner, and Angela Dai. Panoptic 3d scene reconstruction from a single rgb image. *Advances in Neural Information Processing Systems*, 34, 2021. 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 *CVPR*, 2017. 11
- [8] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3D-MPA: Multi-Proposal Aggregation for 3D Semantic Instance Segmentation. In *CVPR*, 2020. 2
- [9] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao.

- Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. *arXiv preprint arXiv:2203.15224*, 2022. [3](#)
- [10] Yasutaka Furukawa and Carlos Hernández. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015. [2](#)
- [11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. [5](#), [6](#)
- [12] Derek Hoiem, Alexei A Efros, and Martial Hebert. Automatic photo pop-up. In *ACM SIGGRAPH 2005 Papers*, pages 577–584. 2005. [2](#)
- [13] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In *CVPR*, 2019. [2](#), [5](#), [6](#), [11](#)
- [14] Ji Hou, Angela Dai, and Matthias Nießner. RevealNet: Seeing Behind Objects in RGB-D Scans. In *CVPR*, 2020. [2](#)
- [15] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15587–15597, 2021. [2](#)
- [16] Benran Hu, Junkai Huang, Yichen Liu, Yu-Wing Tai, and Chi-Keung Tang. Nerf-rpn: A general framework for object detection in nerfs. *arXiv preprint arXiv:2211.11646*, 2022. [2](#), [3](#), [7](#), [13](#)
- [17] Yoonwoo Jeong, Seungjoo Shin, Junha Lee, Chris Choy, Anima Anandkumar, Minsu Cho, and Jaesik Park. Perception: Perception using radiance fields. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. [2](#), [3](#)
- [18] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158, 2014. [2](#)
- [19] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic Neural Fields: A Semantic Object-Aware Neural Scene Representation. In *CVPR*, 2022. [3](#)
- [20] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhofer, Jürgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. *arXiv preprint arXiv:2206.08929*, 2022. [2](#), [5](#)
- [21] Yin hao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with dynamic temporal stereo. *arXiv preprint arXiv:2209.10248*, 2022. [12](#)
- [22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. [3](#)
- [23] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. [2](#)
- [24] 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. [2](#), [4](#), [5](#), [8](#), [12](#)
- [25] Yinyu Nie, Ji Hou, Xiaoguang Han, and Matthias Nießner. Rfd-net: Point scene understanding by semantic instance reconstruction. In *CVPR*, 2021. [2](#)
- [26] Jinyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris Kitani, Masayoshi Tomizuka, and Wei Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. *arXiv preprint arXiv:2210.02443*, 2022. [7](#), [12](#)
- [27] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14314–14323, 2021. [2](#)
- [28] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. Imvotenet: Boosting 3D object detection in point clouds with image votes. In *CVPR*, 2020. [2](#)
- [29] Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. Deep hough voting for 3D object detection in point clouds. *ICCV*, 2019. [2](#)
- [30] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. [2](#), [5](#), [6](#), [11](#)
- [31] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, 2018. [2](#)
- [32] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022. [2](#)
- [33] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Fcaf3d: Fully convolutional anchor-free 3d object detection. In *European Conference on Computer Vision*, pages 477–493. Springer, 2022. [6](#)
- [34] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2397–2406, 2022. [2](#), [3](#), [4](#), [5](#), [6](#), [11](#)
- [35] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008. [2](#)
- [36] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [8](#)
- [37] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006. [2](#)
- [38] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. NeuralRecon: Real-time coherent 3D reconstruction from monocular video. *CVPR*, 2021. [7](#)

- [39] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 2
- [40] Suhani Vora, Noha Radwan, Klaus Greff, Henning Meyer, Kyle Genova, Mehdi SM Sajjadi, Etienne Pot, Andrea Tagliasacchi, and Daniel Duckworth. Nesf: Neural semantic fields for generalizable semantic segmentation of 3d scenes. *arXiv preprint arXiv:2111.13260*, 2021. 2, 3, 13
- [41] Haiyang Wang, Lihe Ding, Shaocong Dong, Shaoshuai Shi, Aoxue Li, Jianan Li, Zhenguo Li, and Liwei Wang. CA-Group3d: Class-aware grouping for 3d object detection on point clouds. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 6
- [42] Jiepeng Wang, Peng Wang, Xiaoxiao Long, Christian Theobalt, Taku Komura, Lingjie Liu, and Wenping Wang. Neuris: Neural reconstruction of indoor scenes using normal priors. *arXiv preprint arXiv:2206.13597*, 2022. 2
- [43] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 2
- [44] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 2, 4, 5, 8, 12
- [45] Tai Wang, Jiangmiao Pang, and Dahua Lin. Monocular 3d object detection with depth from motion. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 386–403. Springer, 2022. 12
- [46] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *CVPR*, 2018. 5, 6
- [47] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5610–5619, 2021. 5, 8
- [48] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5438–5448, 2022. 5
- [49] Jiayu Yang, Wei Mao, Jose M Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4877–4886, 2020. 4, 7
- [50] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 2
- [51] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas Guibas. GSPN: Generative shape proposal network for 3D instance segmentation in point cloud. In *CVPR*, 2019. 2
- [52] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenotrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 2
- [53] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 2, 4
- [54] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *European Conference on Computer Vision*, pages 311–329. Springer, 2020. 2
- [55] Fuqiang Zhao, Yuheng Jiang, Kaixin Yao, Jiakai Zhang, Liao Wang, Haizhao Dai, Yuhui Zhong, Yingliang Zhang, Minye Wu, Lan Xu, et al. Human performance modeling and rendering via neural animated mesh. *arXiv preprint arXiv:2209.08468*, 2022. 2
- [56] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. 3

## A. Dataset and Implementation Details

**Dataset.** Our experiments are conducted on ScanNetV2 [7] and ARKITScenes dataset [2]. ScanNetV2 dataset is a challenging dataset containing 1513 complex scenes with around 2.5 million RGB-D frames and annotated with semantic and instance segmentation for 18 object categories. Since ScanNetV2 does not provide amodal or oriented bounding box annotation, we predict axis-aligned bounding boxes instead, as in [13, 30, 34]. We mainly evaluate the methods by mAP with 0.25 IoU and 0.5 IoU threshold, denoted by mAP@.25 and mAP@.50.

ARKITScenes dataset contains around 1.6K rooms with more than 5000 scans. Each scan includes a series of RGB-D posed images. In our experiments, we utilize the subset of the dataset with low-resolution images. The subset contains 2,257 scans of 841 unique scenes, and each image in the scan is of size  $256 \times 192$ . We follow the dataset setting provided by the official repository<sup>1</sup>. We mainly evaluate the methods by mAP with 0.25 IoU as follow [2].

**Detection Branch.** We follow ImVoxelNet, mainly use ResNet50 with FPN as our backbone and the detection head consists of three 3D convolution layers for classification, location, and centerness, respectively. For the experiment on the ARKITScenes, we additionally predict the rotation. We use the same size  $40 \times 40 \times 16$  of the voxels, with each voxel represents a cube of  $0.16m, 0.16m, 0.2m$ . Besides, we also

<sup>1</sup><https://github.com/apple/ARKitScenes/tree/main/threedod>

keep the training recipe as same as ImVoxelNet. During training, we use 20 images on the ScanNet dataset and 50 images on the ARKITScenes dataset by default. During test we use 50 images and 100 images on the ScanNet dataset and ARKITScenes dataset, respectively. The network is optimized by Adam optimizer with an initial learning rate set to 0.0002 and weight decay of 0.0001, and it is trained for 12 epochs, and the learning rate is reduced by ten times after the 8th and 11th epoch.

**NeRF Branch.** In our NeRF branch, 2048 rays are randomly sampled at each iteration from 10 novel views for supervision. Note that the 10 novel views are ensured to be different with the views input to detection branch for both training and inference. We set the near-far range as (0.2 meter - 8 meter), and uniformly sample 64 points along each ray. During volumetric rendering, if more than eight points on the ray are projected to empty space, then we would throw it and do not calculate the loss of the ray. The geometry-MLP (G-MLP) is a 4-layer MLP with 256 hidden units and skip connections. The color-MLP (C-MLP) is a one-layer MLP with 256 hidden units. Our experiments are conducted on eight V100 GPUs with 16G memory per GPU. We batched the data in a way such that each GPU carries a single scene during training. During training, the two branches are end-to-end jointly trained. During inference, we can keep either one of the two branches for desired tasks. The whole Our implementation is based on MMDection3D [5].

## B. Evaluation Protocol of Novel View Synthesis and Depth Estimation.

To evaluate the novel view synthesis and depth estimation performance, we random select 10 views of each scene as the novel view (indicated as target view in IBRNet [44]), and choose the nearby 50 views as the support views. To render the RGB and depth for the 10 novel views, each points shooting from the pixels of novel views would be projected to the all support views to sample features, and then pass into the NeRF MLP as illustrated in Method section. We keep the same novel view and support view for both setting in Table. 6 of the main text. Note that the evaluation is conducted on the test set of ScanNet dataset, which are never seen during training. The non-trivial results also demonstrate the generazability of the proposed geometry-aware volumetric representation.

## C. Additional Results

**Ablation studies on number of views.** We conducted an analysis of how the number of views affects the performance of 3D detection, as shown in Table 9. Specifically, we used the same number of training images (20 images)

Table 9: Ablation on number of views. Due to the GPU memory limitation, we downsample the image resolution 2x when conduct experiments on 100 views (denoted as ImVoxelNet-R50-2x' and NeRF-Det-R50-2x'). Experiments on each setting run three times. We report the mean and standard deviations of our experiments.

Methods	mAP@.25	mAP@.50
ImVoxelNet-R50-2x (10 views)	37.8±1.2	17.5±1.0
ImVoxelNet-R50-2x (20 views)	46.5±0.5	21.1±0.5
ImVoxelNet-R50-2x (50 views)	48.4±0.3	23.7±0.2
ImVoxelNet-R50-2x'(100 views)	48.1±0.1	24.7±0.1
NeRF-Det-R50-2x (10 views)	41.4 ±1.0 (+3.6)	19.2±0.9 (+1.7)
NeRF-Det-R50-2x (20 views)	50.2 ±0.5 (+3.7)	23.6±0.4 (+2.5)
NeRF-Det-R50-2x (50 views)	51.8 ±0.2 (+3.4)	26.0±0.1 (+2.3)
NeRF-Det-R50-2x'(100 views)	52.2±0.1 (+4.1)	27.4±0.1 (+2.7)

and tested with different numbers of images. Our proposed NeRF-Det-R50-2x showed a significant improvement in performance as the number of views increased. In contrast, the performance of ImVoxelNet-R50-2x had limited improvement, and even worse, the performance decreased when the number of views increased to 100. We attribute the performance improvements of NeRF-Det to its effective scene modeling. NeRF performs better as the number of views increases, typically requiring over 100 views for an object [24]. Our proposed NeRF-Det inherits this advantage, leading to a drastic performance gain of 4.1 mAP@.25 and 2.7 mAP@.50 on 100 views.

Overall, our analysis demonstrates the effectiveness of our proposed NeRF-Det in leveraging multi-view observations for 3D detection and the importance of utilizing a method that can effectively model the scene geometry.

**More Qualitative Results** We provide more visualization results of novel-view synthesis and depth estimation, as shown in Fig. 6. The results come from the test set of ScanNet. We can observe that the proposed method generalizes well on the test scenes. Remarkably, it achieves non-trivial results on the relatively hard cases. For example, the left of the second row presents a bookshelf with full of colorful books, our method can give reasonable novel-view synthesis results. On the other hand, for the left of fifth row, the extremely dense chairs are arranged in the scenes and we can observe the method can predict accurate geometry.

## D. Discussion about outdoor 3D detection

We emphasize the differences of NeRF-Det and the other 3D detection works in outdoor scenes. Our proposed NeRF-Det shares the similar intuition with many of outdoor 3D detection works, such as [26, 45, 21], which try to learn geometric-aware representations. However, the proposed NeRF-Det and the other works differ intrinsically. The outdoor 3D detection works [26, 45, 21] propose to use cost volume or explicitly predicted depth to model the scene geometry. Instead, NeRF-Det leverage the discrepancy of

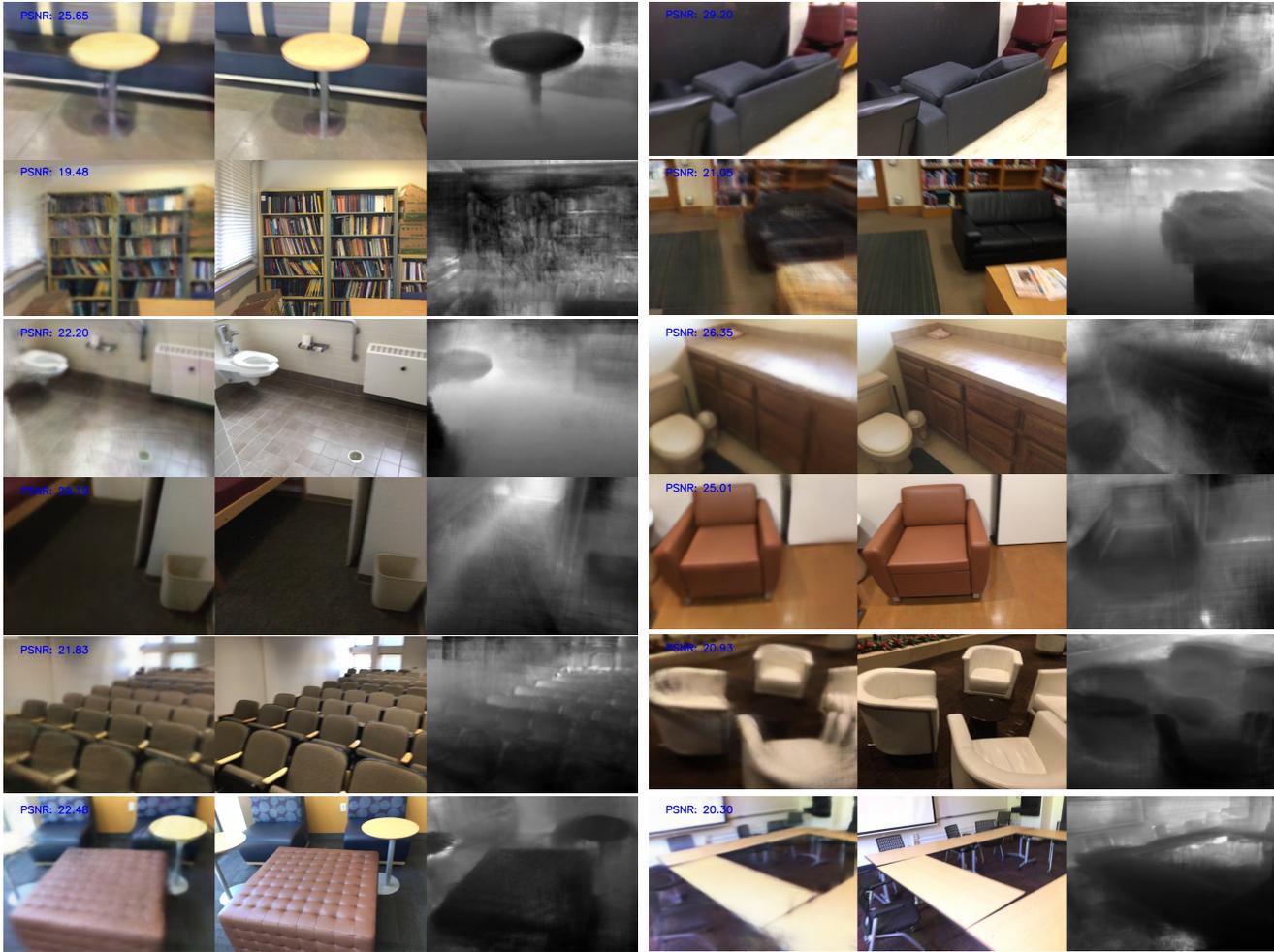


Figure 6: Novel-view synthesis results on top of NeRF-Det-R50-2x\*. For each triplet group, the left figure is the synthesized results, the middle one is the ground truth RGB image, and the right part is the estimated depth map. Note that the visualization is from test set, which is never seen during training.

multi-view observations, i.e., the augmented variance features in our method section, as the priors of NeRF-MLP input. Beyond the cost volume, we step forward to leverage the photo-realistic principle to predict the density fields, and then transform it into the opacity field. Such a geometry representation is novel to the 3D detection task. The analysis in our experiment part also demonstrates the advantages of the proposed opacity field. In addition to the different method of modeling scene geometry, our design of combining NeRF and 3D detection in an end-to-end manner allows the gradient of NeRF to back-propagate and benefit the 3D detection branch. This is also different from previous NeRF-then-perception works [16, 40].

Our NeRF-Det is specifically designed for 3D detection in indoor scenes, where objects are mostly static. Outdoor scenes present unique challenges, including difficulties in

ensuring multi-view consistency due to moving objects, unbounded scene volume, and rapidly changing light conditions that may affect the accuracy of the RGB value used to guide NeRF learning. We plan to address these issues and apply NeRF-Det to outdoor 3D detection in future work.