CAAI Transactions on Intelligence Technology

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

Improving 3D Object Detection in Neural Radiance Fields With Channel Attention

Minling Zhu[1] | Yadong Gong[1] | Dongbing Gu[2] | Chunwei Tian[3]

[1]College of Computer Science, Beijing Information Science and Technology University, Beijing, China | [2]School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK | [3]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

Correspondence: Chunwei Tian (chunweitian@hit.edu.cn)

Received: 27 May 2024 | **Revised:** 27 February 2025 | **Accepted:** 17 March 2025

Handling Editor: Ying Fu

Funding: This research was funded by the Qiyuan Innovation Foundation (Grant S20210201067) and sub-themes (No. 9072323404).

Keywords: 3-D | feature extraction | neural network | pattern recognition

ABSTRACT

In recent years, 3D object detection using neural radiance fields (NeRF) has advanced significantly, yet challenges remain in effectively utilising the density field. Current methods often treat NeRF as a geometry learning tool or rely on volume rendering, neglecting the density field's potential and feature dependencies. To address this, we propose NeRF-C3D, a novel framework incorporating a multi-scale feature fusion module with channel attention (MFCA). MFCA leverages channel attention to model feature dependencies, dynamically adjusting channel weights during fusion to enhance important features and suppress redundancy. This optimises density field representation and improves feature discriminability. Experiments on 3D-FRONT, Hypersim, and ScanNet demonstrate NeRF-C3D's superior performance validating MFCA's effectiveness in capturing feature relationships and showcasing its innovation in NeRF-based 3D detection.

1 | Introduction

3D object detection plays a crucial role in computer vision and has important research value for efficient understanding and interpretation of real-world scenes. By integrating spatial information, 3D object detection can capture the position, pose and dimensions of objects more accurately, providing a more comprehensive perspective for in-depth scene understanding. This not only contributes to improving the robustness of perception systems in complex environments but also offers critical foundational support for applications in fields, such as autonomous driving, robot perception, virtual reality and augmented reality.

Although there have been notable advances, there remains ample opportunity for exploration and innovation within 3D object detection. Image-based methods use single- or multi-view images as input to the network. However, due to the lack of significant information, such as depth in images, which indicates spatial geometric characteristics, many methods attempt to compensate for this deficiency through manual design. Monocular-based methods Wang et al. [1], Weng and Kitani [2], utilise depth maps or pseudo-Lidar images to improve detection performance. For another, multi-view-based methods Liu et al. [3], Li et al. [4], Philion and Fidler [5], leverage Transformers or camera calibration to map multi-view 2D image features into 3D space for 3D object detection. Despite significant advancements achieved by image-based methods, they are still constrained by the inherent limitation of images. Point cloud-based methods Misra et al. [6], Qi et al. [7], Sheng et al. [8], which provide abundant 3D geometric information, have demonstrated their effectiveness on datasets. However,

these methods heavily depend on precise data captured by sensors. Meanwhile, multimodal methods Bai et al. [9], Yan et al. [10], effectively combine the advantages of the two aforementioned methods. However, they inevitably require complex design and extensive computational resources.

The arrival of neural radiance fields (NeRF) Mildenhall et al. [11] presents a novel alternative by extracting radiance information (RGB) and density features of the original 3D scene from multi-view 2D images as input to the network. Hu et al. proposed the first 3D object detection framework in NeRFs, NeRF-RPN Hu et al. [12], which can directly predict 3D region proposals from radiance and density grid in neural fields. However, radiance information and density feature are not equally important for different tasks in neural fields. View-dependent radiance information can achieve high-quality scene reconstruction and rendering. For 3D object detection task where only objects and backgrounds are concerned, density features that can reflect geometric information are crucial. Hu et al. [12] overlook the representation of the density field within the neural field and does not fully utilise the dependency relationships of the features. Meanwhile, channel attention serves as an improvement that can help networks learn features that are more important to the target task, thereby enhancing the representation ability of these features in the network and reducing interference with irrelevant information. Therefore, we introduce channel attention for 3D object detection in NeRFs for the first time, and propose NeRF-C3D. In detail, building upon the NeRF-RPN, we incorporated channel attention into the process of multi-scale feature fusion. Compared to the original method, the processed features are more refined, with each layer better learning and representing density weights. And our method does not need to lose any radiance features, which makes our method more suitable for other downstream tasks, that is, instance segmentation task, object classification tasks and secondary object detection, while improving 3D object detection accuracy.

The main contributions of this paper are summarised as follows:

- *Identification of a Key Limitation in NeRF-RPN*: We critically analyse the NeRF-RPN framework, identifying a significant drawback: its neglect in effectively representing the density grid feature and under-utilisation of feature dependency relationships within neural fields. This oversight limits the performance and robustness of 3D object detection in complex scenes. Addressing this limitation is crucial for advancing the accuracy and efficiency of NeRF-based detection models.
- *Proposal of MFCA and NeRF-C3D*: To overcome the identified limitation, we propose the multi-scale feature fusion module with channel attention (MFCA), designed to better capture and integrate multi-scale features by enhancing feature dependency relationships within neural fields. Channel Attention within MFCA plays a crucial role in dynamically reweighting feature volumes, allowing the model to focus on the most informative features and suppress the less relevant ones. Furthermore, we propose NeRF-C3D, a novel framework specifically tailored for 3D object detection in NeRF. NeRF-C3D takes advantage of the strengths of MFCA to refine feature representation, thereby significantly improving detection accuracy and reliability in NeRF-based 3D object detection tasks.
- *Comprehensive Evaluation and Validation*: We conducted extensive experiments on established 3D object detection datasets within NeRFs, including 3D-FRONT, Hypersim, and ScanNet. Our method consistently outperforms the NeRF-RPN, demonstrating substantial improvements in detection performance. This validates the effectiveness of our method and highlights its potential to set new benchmarks in NeRF-based 3D object detection.

2 | Related Work

2.1 | NeRF

NeRF is a deep learning-based method employed for 3D reconstruction in novel views, which uses multi-layer perceptrons (MLP) to model radiance information and density, for each point in the scene. By training neural networks, spatial structure and colour texture information of the 3D scene can be inferred from a limited number of observation angles and positions, yielding a 3D scene with high-quality details. Since then, great progress has been made in 3D reconstruction based on NeRF. Barron et al. [13], Verbin et al. [14] all share the goal of improving NeRF's view synthesis and 3D scene representation by enhancing both the photometric and geometric aspects. This collective effort aims to enhance the image quality in synthesised views. Müller et al. [15], Garbin et al. [16], focus on endeavours to accelerate NeRF training and inference speed. A. Chen et al. [17], Yu et al. [18] leverage pre-trained image feature extraction networks, to significantly reduce the required number of training samples for successful NeRF training. Although these methods may differ slightly in the details of their model designs, fundamentally, they model the input *xyz* coordinates, and 3D camera poses as RGB colours and volume densities for each point in the scene, thus generating high-quality 3D scene images. Such NeRF-based method, which can generate high-quality 3D scene details through simple neural network training, also demonstrates its potential for 3D object detection.

2.2 | 3D Object Detection

We classify current 3D object detection methods into three categories based on the modalities of the input data: image-based methods, point cloud-based methods, and multi-modal-based methods. Image-based methods, whether based on monocular, stereo, or multi-view images, are popular and widely applied in autonomous driving technology due to the ability to obtain rich texture features. Monocular 3D object detection methods typically rely on prior information. For instance, R. Zhang et al. [19] utilises a pre-predicted foreground depth map as a guidance signal and extracts features under the guidance of a Transformer adaptively. ImVoxelNet Rukhovich et al. [20] is also a monocular-based method that maps image features to 3D voxels through 2D CNN. For each voxel, its features come from element wise averaging of several images, and the final prediction is made based on these voxel. Stereo-

based methods leverage stereo matching techniques to provide additional geometric constraints. You et al. [21] converts the predicted disparity map of stereo images into depth images, and then convert them into pseudo-LiDAR point clouds for 3D object detection. Methods based on multi-view images often employ techniques like Philion and Fidler [5], Li et al. [4] to map 2D plane features into 3D space and perform 3D object detection from a bird's eye view. Liu et al. [3], Liu et al. [22] utilise the Transformer's Query mechanism to interact features between 2D image information and 3D spatial points. Additionally, Li et al. [4], Yang et al. [23], Zhang et al. [24] integrate spatio-temporal features on this basis, enhancing performance of 3D object detection.

Point cloud-based methods can directly capture the geometric features of 3D objects and can be primarily classified into two types based on the data representation: point-based and voxel-based methods. Chen et al. [25], He et al. [26] directly process on raw points employing Transformer or 3D Convolutional Neural Networks (3D CNN) for feature extraction or enhancement. Mao et al. [27], Liu et al. [28] convert raw point clouds into voxel grids and apply methods from 2D object detection for feature extraction and subsequent object detection. FCAF3D Rukhovich et al. [29], a simple and effective method for indoor 3D object detection, leverages traditional convolutional structures, using voxel representations of point clouds and sparse convolutions to process the voxels. Multi-modal-based 3D object detection methods often integrate data features from point clouds and RGB images, and use the fused features for 3D object detection. Bai et al. [9], Li et al. [30] utilise Transformer architecture to interact features from multi-modal data, greatly improving detection performance. Although these methods have demonstrated outstanding performance across diverse datasets and offer valuable insights for 3D object detection within NeRF, they cannot be directly applied to NeRF-based 3D object detection in its entirety.

2.3 | 3D Object Detection With NeRFs

Several researchers have dedicated their efforts to advancing the field of 3D object detection with NeRF. NeRF-Det Xu et al. [31] embeds a NeRF module in the collaborative design of the model to learn the geometric features of the scene, thus enhancing detection performance. NeRF-Det++ Huang et al. [32] introduces innovative enhancements, including semantic refinement, perspective-aware sampling, and ordinal residual deep supervision techniques. These advancements address crucial challenges such as semantic ambiguity, inadequate sampling strategies, and underutilisation of deep supervision in the original model. By incorporating these enhancements, NeRF-Det++ improves the detection performance of the base model to a certain extent, providing a more robust and effective solution for 3D scene understanding and reconstruction. The NeRF-Det series methods can effectively leverage the density information in NeRF to improve image-based 3D object detection performance, but they fail to utilise the radiance information at each point in the scene, and are not suitable for directly participating in tasks like instance segmentation within NeRF. In monocular 3D object detection, MonoNeRD Xu et al. [33] models the scene using a Signed Distance Function (SDF), which helps generate dense 3D representations and treats these representations as NeRF. This method can effectively infer 3D geometric structure and occupancy from a single image, demonstrating the important potential of implicit reconstruction for basic 3D perception of images.

On the other hand, NeRF-RPN Hu et al. [12] is the first 3D object detection framework directly operating in neural fields. The 3D volumetric features obtained in the neural field are represented by using a novel 3D voxel feature. This method showcases the ability to directly predict 3D bounding boxes for objects within the NeRF framework. The method uses 3D CNN to extract features preliminarily from the input high-dimensional semantics, then employs Feature Pyramid Network (FPN) Lin, Dollár et al. [34] to fuse multi-scale features. Finally, through 3D Region Proposal Network (3D RPN) and non-maximum suppression (NMS), it regresses 3D region proposals. NeRF-MAE Irshad et al. [35], inspired by NeRF-RPN, is the first method to employ large-scale self-supervised pretraining using NeRF radiance and density grids as input modalities. It utilises a standard 3D Swin-Transformer encoder and a voxel decoder to develop a robust representation. The method incorporates an opacity-aware dense volume masked self-supervised learning objective. When fine-tuned on a small subset of data, this representation demonstrates significant improvements in various 3D downstream tasks, such as 3D object detection, voxel super-resolution, and voxel labelling. Although NeRF-RPN establishes a solid baseline model for 3D object detection in NeRF, the method itself overlooks the representation of density grid feature and failing to fully utilise the feature dependency relationships in neural fields, impacting the detection performance of the model in NeRF.

3 | Methodology

The overall structure of the network is shown in Figure 1. Our method uses the RGB and density features obtained through NeRF as the model's inputs. Following the preliminary feature extraction through a 3D backbone network, the extracted features are then utilised to generate the 3D Feature Pyramid Network (3D FPN). We propose a novel multi-scale feature fusion module, denoted as MFCA, which incorporates Channel attention for multi-scale feature fusion. This enables the feature pyramid to obtain more refined features during the process of downward feature volume construction. Subsequently, 3D RPN uses these obtained features to predict region proposals.

3.1 | Input Sampling and Feature Extraction

Despite the existence of numerous excellent NeRFs representations, they all have the same characteristics for input sampling: the ability to query radiance information (RGB) and density based on view directions and spatial positions Gao et al. [36]. Our method still utilises these radiance and density feature volumes as inputs, independent of the specific NeRF structure. In the process of input sampling, we adhere to the design from

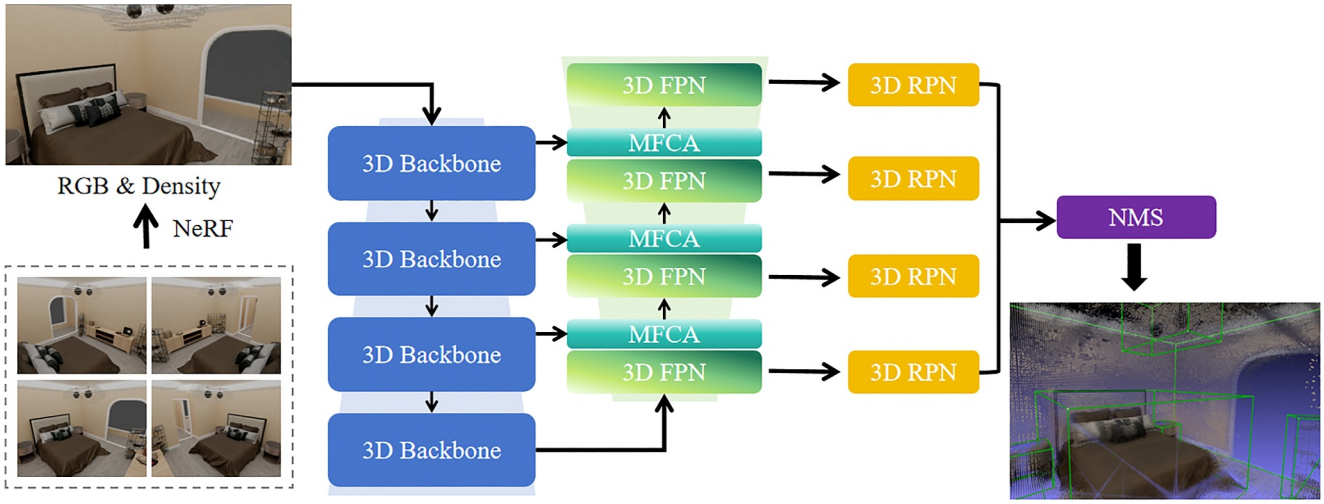

FIGURE 1 | Overview of NeRF-C3D. Our method uses RGB and density features that are extracted from NeRF as inputs to NeRF-C3D. We propose the MFCA module to fuse features from the 3D CNN and upper-level features from the 3D FPN. These features are then input into the 3D RPN head to generate precise 3D region proposals.

Hu et al. [12]. Using a 3D mesh that is large enough to adaptively change the resolution to contain the resulting NeRF model, and adopt a uniform sampling strategy for the entire NeRF model to obtain the radiance and density information. Typically, each sample can be represented as (r, g, b, α), where (r, g, b) represents the average radiance, α converted through density η, as shown in Equation (1):

$$\alpha = \text{np.clip}(1 - \exp(-\eta\gamma), 0, 1), \tag{1}$$

where $\gamma = 0.01$ is the preset distance, and the value is then constrained within the range of 0 and 1 using the np.clip function, ensuring that α never becomes less than 0 or greater than 1. For the feature extraction component, we conducted comparative experiments in ablation studies using two backbone networks: VGGNet Simonyan and Zisserman [37] and ResNet He et al. [38]. In these networks, the relevant 2D convolutions, poolings and normalisations were replaced with 3D modules, with minimal additional modifications.

3.2 | Multi-Scale Feature Fusion Module With Channel Attention

Although FPN Lin et al. [34], Tan et al. [39], Li et al. [40] perform well in multi-scale feature fusion, there are still some shortcomings in the context of NeRF-based 3D object detection tasks. The original FPN primarily relies on a simple feature aggregation strategy and fails to adequately consider the importance of different channel features. Additionally, FPN may struggle to effectively suppress redundant information when facing complex scenes, leading to a decline in the accuracy of feature representation. These issues are particularly pronounced in 3D object detection within NeRF, as the geometric complexity and variations in viewpoint can impair the discriminative ability of features. To address these shortcomings, we propose the MFCA module, which integrates channel attention to achieve dynamic channel recalibration and adaptive multi-scale fusion. This design enables MFCA to focus on important features, enhancing feature representation and making it better suited for 3D object detection in NeRF. MFCA not only overcomes the limitations of the original FPN but also significantly improves detection performance.

The MFCA module, as illustrated in Figure 2, introduces a channel attention mechanism based on the 3D FPN and is inserted as a computational module between various output layers during the top-down process of the 3D FPN. The feature volumes generated from bottom to top in 3D FPN are denoted as: $Q = [Q_1, Q_2, ..., Q_n], Q \in \mathbb{R}^{C\times W\times L\times H}$. The feature volumes obtained during the top-down process of the 3D FPN are denoted as: $P = [P_1, P_2, ..., P_n], P \in \mathbb{R}^{C\times W\times L\times H}$. In MFCA, the preliminary fusion of features follows the design of FPN, where the upsampled high-level feature volumes are fused with the bottom-up generated low-level feature volumes. This fusion can be expressed by Equation (2):

$$P_{n-1} = F_{\text{upsampled}}(P_n) + Q_{n-1}. \tag{2}$$

Subsequently, the obtained feature volumes P_{n-1} are fed into the 3D Channel attention module. The design of the 3D Channel attention module is inspired by Hu et al. [41] and Woo et al. [42], and does not adopt a Transformer-based method. Utilising Transformer-based Channel attention would introduce significant computational overhead, particularly for a structure already employing 3D CNN as its backbone, negatively impacting the model's training speed. The 3D Channel attention module is based on a purely convolutional method, ensuring ease of implementation. Firstly, the spatial dimensions of the preliminarily fused feature volumes P_{n-1} are reduced using 3D adaptive average pooling F_{ap3d}. This can be expressed as shown in Equation (3):

$$Z_{n-1} = F_{\text{ap3d}}(P_{n-1}) = \frac{1}{W\times L\times H}\sum_{i=1}^{W}\sum_{j=1}^{L}\sum_{k=1}^{H} P_{n-1}(i, j, k). \tag{3}$$

The purpose of this is to retain only the Channel information of the feature volumes, using a relatively simple 3D adaptive average

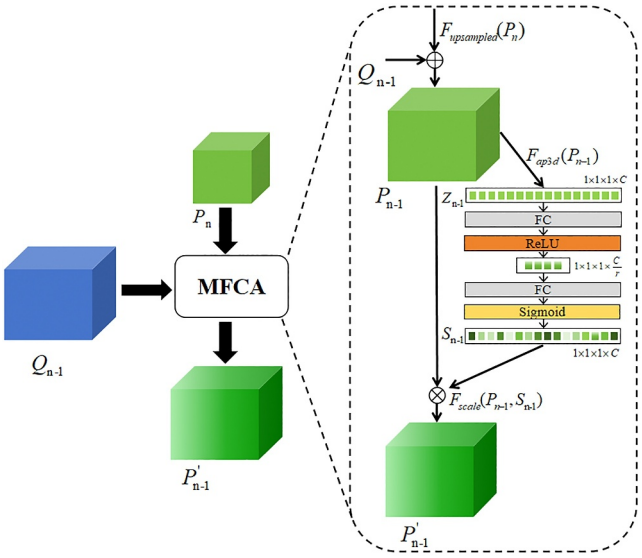

FIGURE 2 | Architecture of MFCA module.

pooling to roughly preserve the original features of the entire feature volume. After obtaining the feature volume through spatial dimension average pooling, the goal is to capture inter-channel dependencies at minimal cost, maximising the utilisation of feature information. Inspired by Hu et al. [41], we perform adaptive re-calibration F_{AR} on the obtained feature volume Z_{n-1}, specifically designed as shown in Equation (4):

$$S_{n-1} = F_{AR}(Z_{n-1}, W) = \sigma(\tau(Z_{n-1}, W)) = \sigma(W_2\delta(W_1 Z_{n-1})), \quad (4)$$

where δ and σ are the ReLU and Sigmod activation function, respectively. $W_1 \in \mathbb{R}^{\frac{C}{r}\times C}$, $W_2 \in \mathbb{R}^{c\times\frac{C}{r}}$. To control the complexity of the model and improve generalisation, we introduced a gating mechanism parameterised by creating a bottleneck around nonlinearity using two Fully Connected (FC) layers. The gating mechanism specifically includes a dimension reduction layer with a reduction ratio of r, followed by a ReLU, and subsequently an expansion layer that restores the channel dimension of the transformed output P_{n-1}. The final output of this block P'_{n-1} is obtained through Equation (5):

$$P'_{n-1} = F_{scale}(P_{n-1}, S_{n-1}) = S_{n-1}P_{n-1}, \quad (5)$$

where S_{n-1} is the feature weight obtained through learning and F_{scale} is the input feature volume multiplied by the channel weights. We utilise channel attention to refine the expression of transformed features, thereby optimising their representation before propagation to the subsequent layer. This meticulous process ensures the comprehensive representation of both low-level and high-level features, thereby facilitating the nuanced encoding of weights for each channel prior to their transmission to the 3D RPN detection head.

3.3 | 3D RPNs

The 3D RPNs leverages features extracted from 3D FPN as inputs, generating a set of 3D region proposals with corresponding object scores as output. In accordance with ablation

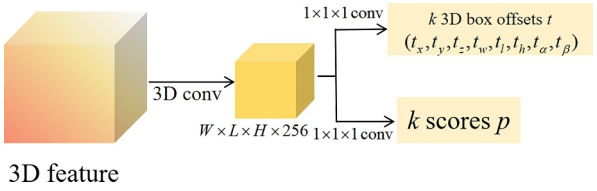

FIGURE 3 | Architecture of anchor-based 3D RPN head.

experiments, we conducted experiments on two types of RPN heads: an anchor-based method and an anchor-free method.

As shown in Figure 3, the anchor-based method involves the addition of k layers of 3D convolutional layers following the feature pyramid. Over these convolutional layers, two independent 1 × 1 × 1 3D convolutional layers are employed for predictions. These layers are dedicated to determine the possibility p of object presence and the bounding box offsets t for each anchor point. In essence, this architecture processes features through multiple layers of 3D convolutional operations and makes final predictions for object presence probability and bounding box offsets through specific 1 × 1 × 1 convolutional layers. The parameters for the offsets of bounding box, denoted as $t = (t_x, t_y, t_z, t_w, t_l, t_h, t_\alpha, t_\beta)$, are described as follows in Equation (6):

$$\begin{aligned} &t_x = (x - x_a)/w_a, t_y = (y - y_a)/l_a, t_z = (z - z_a)/h_a, \\ &t_w = \log(w/w_a), t_i = \log(l/l_a), t_h = \log(h/h_a), \\ &t_\alpha = \Delta\alpha/w, t_\beta = \Delta\beta/l, \end{aligned} \quad (6)$$

where $x, y, w, l, \Delta\alpha, \Delta\beta$ describe the representation of the 3D bounding box in the xy-plane, z, and h represent dimensions in height. On the other hand, $x_a, y_a, z_a, w_a, l_a, h_a$ provide the position and size of the bounding box.

The anchor-free method is primarily based on the fully convolutional one-stage (FCOS) Tian et al. [43] method, extended to its 3D form. Compared with anchor-based methods, the RPN based on FCOS predicts a single object probability p, a set of bounding box offset values $t = (x_0, y_0, z_0, x_1, y_1, z_1, \Delta\alpha, \Delta\beta)$ and a objectness scores c for each voxel as shown in Figure 4. In FCOS, an extension has been applied to the encoding of bounding box offsets, and the regression objects are defined as follows in Equation (7):

$$\begin{aligned} &x_0^* = x - x_0^{(i)}, x_1^* = x_1^{(i)} - x, \\ &y_0^* = y - y_0^{(i)}, y_1^* = y_1^{(i)} - y, \\ &z_0^* = z - z_0^{(i)}, z_1^* = z_1^{(i)} - z, \\ &\Delta\alpha^* = v_x^{(i)} - x, \Delta\beta^* = v_y^{(i)} - y, \end{aligned} \quad (7)$$

where, x, y, and z denote the position of the voxel, while $x_0^{(i)}, x_1^{(i)}$ represents the lateral boundary of the ith ground truth bounding box, similarly in the y and z directions. The variables $v_x^{(i)}$ and $v_y^{(i)}$ denote the x and y coordinates of the vertices of the bounding box in the xy-plane. x_0^*, x_1^* represent the offsets of the bounding box along the x-axis, similarly in the y and z directions.$\Delta\alpha^*$, $\Delta\beta^*$ are two additional offsets used to represent variations in other directions. The centre of the actual bounding box is computed by the following Equation (8):

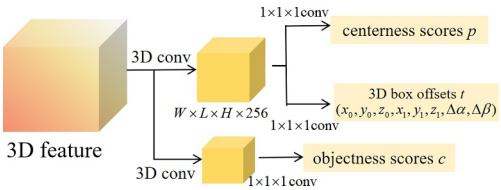

FIGURE 4 | Architecture of anchor-free 3D RPN head.

$$c^* = \sqrt{\frac{\min(x_0^*, x_1^*)}{\max(x_0^*, x_1^*)} \times \frac{\min(y_0^*, y_1^*)}{\max(y_0^*, y_1^*)} \times \frac{\min(z_0^*, z_1^*)}{\max(z_0^*, z_1^*)}}. \quad (8)$$

To learn p, t, and c in FCOS, k 3D convolutional layers are added after the FPN for classification and regression. An additional convolutional layer is added on the classification and regression branches to obtain the centrality score $p$and 3D box offset t, and a convolutional layer is added in the regression branch to predict the objectness score c.

3.4 | Loss Functions

In the loss function section, due to the adoption of different RPN detection heads in subsequent experiments, the loss function also has different designs depending on the 3D RPNs, which are Anchr-based and Anchor-free. According to Hu et al. [12], for the Anchor-based method, a predicted bounding box is labelled as positive if its Intersection over Union (IoU) exceeds 0.35 with the ground truth or has extremely high IoU overlap with its associated bounding box. Conversely, IoU below 0.2 is labelled as negative and the overall loss function is shown in Equation (9):

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{\text{cls}}} \sum_i L_{\text{cls}}(p_i, p_i^*) + \frac{\lambda}{N_{\text{reg}}} \sum_i p_i^* L_{\text{reg}}(t_i, t_i^*), \quad (9)$$

where p_i, t_i represent the predicted objectness and offset of bounding box, p_i^*, t_i^* are the targets for the ground truth bounding box. N_{cls}, N_{reg} represent the number of anchors used in the loss computation, and λ is used to balance the weights between the two losses. L_{cls} is the binary cross-entropy loss, and L_{reg} is the smooth L_1 loss. Regression loss is only computed anchors which are positive. The overall loss function for the Anchor-free method is shown in Equation (10):

$$\begin{aligned} L(\{p_i\}, \{t_i\}, \{c_i\}) = & \frac{1}{N_{\text{pos}}} L_{\text{cls}}(p_i, p_i^*) + \frac{\lambda}{N_{\text{pos}}} p_i^* L_{\text{reg}}(t_i, t_i^*) \\ & + \frac{1}{N_{\text{pos}}} p_i^* L_{\text{ctr}}(c_i, c_i^*), \end{aligned} \quad (10)$$

where L_{cls}, L_{reg} and L_{ctr} are Focal loss from Lin et al. [44], IoU loss for rotated boxes from Zhou et al. [45] and binary cross-entropy loss, respectively. $p_i^* \in \{0, 1\}$ is used to represent the truth label for each voxel. λ is still used to balance the weights between the three losses, and N_{pos} is the number of voxels with $p_i^* = 1$. c_i represents the predicted objectness scores. Regression and centrality losses are considered only for positive voxels.

4 | Experiment

4.1 | Datasets and Evaluation Metrics

We use the dataset built by Hu et al. [12] for 3D object detection within NeRF in our experiments. The datasets mainly include 3D-FRONT Fu et al. [46], Hypersim Roberts et al. [47] from synthetic datasets as well as ScanNet Dai et al. [48] dataset from real scenes. Based on the performance of the model on these three datasets, we verify the effectiveness of our proposed method. Figure 5 shows some examples of datasets used for NeRF training and input NeRFs. We adopt common evaluation metrics including *Recall*, Average Precision (*AP*), and IoU to assess the performance of our proposed method. *Recall* and *AP* are computed at IoU thresholds of 0.25 and 0.5, respectively.

4.1.1 | 3D-FRONT

3D-FRONT is a novel, large-scale, and comprehensive synthetic indoor scene dataset focusing on professionally designed layouts. Numerous rooms are filled with high-quality textured 3D models with style-compatible features. The dataset comprises 18,797 rooms with various 3D objects, including 7302 pieces of furniture with high-quality texture features. After processing by Hu et al. [12], 159 usable rooms were selected, cleaned, and annotated. A total of 1191 real bounding boxes were labelled, with an average of 7.5 real bounding boxes per scene.

4.1.2 | Hypersim

Hypersim is a synthetic dataset for indoor scene understanding. Using a vast library of synthetically created scenes by professional artists, the dataset generates up to 70,000 images for over 400 indoor scenes. Each indoor scene has detailed pixel-level labels and corresponding ground-truth structures. And, each scene includes full scene geometry, material information, and lighting details, as well as dense pixel-level semantic pose segmentation and full camera pose for each image. After cleaning and annotation, Hu et al. [12] retained approximately 250 scenes from the original dataset, resulting in 4798 labelled real bounding boxes. On average, each scene contains 19.2 real bounding boxes.

4.1.3 | ScanNet

ScanNet is a video dataset containing over 2500 + view images with detailed camera pose information, surface structure information as well as instance-level semantic segmentation labels. It is important for understanding of indoor complex scenes. For 3D object detection in NeRFs, Hu et al. [12] randomly selected 90 scenes from ScanNet for training, and selected the most appropriate few frames for each scene according to the clarity of the video. Annotations for scene objects underwent cleaning and modifications. In total, 1086 real bounding boxes were labelled, with an average of 12.1 real bounding boxes per scene.

FIGURE 5 | Examples of datasets and input NeRFs. We utilise Instant-NGP Müller et al. [15] to obtain input NeRFs based on the raw images from datasets. The first four columns represent the raw images, and the last column represents the input NeRFs. These images are from Hypersim.

4.2 | Training and Testing

All our experiments are run with python = 3.9, pytorch = 1.12.1, torchvision = 0.13.1 and cudatoolkit = 11.3. All experiments are conducted using 2 NVIDIA A5000 (24 GB) GPUs for training, including the reproduction of the baseline model. The overall training detail is similar to NeRF-RPN. The AdamW optimizer is used for network optimisation. In terms of hyperparameter design during training, according to Hu et al. [41], we set the value of r in MFCA to 16 to achieve the best performance. The Anchor-based loss function is assigned $\lambda = 5.0$, and for the Anchor-free loss function, $\lambda = 1.0$. We employ a 4-layer FPN and 13 different sizes of anchors for the Anchor-based method. The shortest side of all Anchors on the same feature volume level has the same size, ranging from fine to coarse scales at {8, 16, 32, 64}. Similar to the previous RPN method, 256 anchors are randomly sampled from each scene in each iteration, with a 1:1 ratio of positive to negative anchors. For the Anchor-free method, all output solutions are used for the loss calculation. Different parameter settings are adopted during training on different datasets due to experimental constraints. For example, we reduced the batch size on the Hypersim dataset and adjusted the learning rate and weight decay based on the batch size as shown in Table 1. All experimental results are obtained by training 200 epochs from scratch.

For model testing, all our experiments are conducted with a single NVIDIA A5000 (24 GB) GPU, and the batch size is set to 2. Upon obtaining the region of interest (ROI) with objectiveness scores. Subsequently, the top 2500 proposals at each level of the feature volume are selected independently. The threshold of rotated IoU is set to 0.1 and NMS is applied to eliminate redundant proposals.

4.3 | Ablation Studies and Result Analysis

We conducted ablation studies on the three datasets mentioned above. The experiment compared the detection performance of our method with the baseline model in different backbone networks and RPN detection heads as shown in Tables 2–4. Our method outperforms NeRF-RPN significantly on the 3D-FRONT dataset. For instance, our NeRF-C3D outperforms NeRF-RPN with VGG-16 and anchor-based RPN heads by +0.7%, +4.4%, +2.0% and +5.7% in terms of $Recall_{25}$, $Recall_{50}$, AP_{25}, and AP_{50}, respectively. Our method also has significant advantages over NeRF-RPN on the Hypersim and ScanNet datasets. On the Hypersim dataset, our proposed method can significantly enhance the performance of object detection, leading to more objects being successfully detected and thus improving *Recall* value. However, these additional detection results may contain more false positives, leading to a decrease in precision and affecting *AP*. Moreover, our method can significantly improve detection accuracy but the improvement in *Recall* is not significant on ScanNet dataset. We believe that there are two main reasons for such results. Firstly, the baseline model itself has strong detection performance on 3D-FRONT, which may be greatly related to the reconstruction quality of the dataset. Our method can bring more significant improvements on methods with certain detection performance. Secondly, the different parameter settings of the dataset in the experiment have a certain impact, which is evident on the Hypersim dataset.

TABLE 1 | Setting of experimental parameters on different datasets.

Datasets	Batch size	Learning rate	Weight decay
3D-FRONT	8	3e-4	1e-3
Hypersim	4	4e-4	2e-3
Scannet	8	3e-4	1e-3

Experiments on different backbone networks and RPNs have shown that the combination of VGG-16 and Anchor-free can maximise the detection performance of the model, followed by ResNet-50. Anchor-based methods are slightly inferior. We believe that the performance of anchor-free methods often surpasses that of anchor-based methods primarily due to the following reasons: (1) Anchor-based methods rely on predefined anchors to regress the position and size of object bounding boxes. This method assumes certain distributions of object locations and scales, which may perform suboptimally when faced with high-variance objects, such as those with varying scales and shapes. (2) For small objects, the limited scale of anchors can result in inaccurate localisation or missed detections. In contrast, anchor-free methods directly predict the object centre, offering a more efficient method with higher representational capacity, particularly when dealing with complex scenes. Moreover, the network characteristics of VGG-16 can better perform in such tasks, while ResNet-50 may achieve more outstanding performance in large datasets.

TABLE 2 | *Recall* and *AP* of NeRF-RPN and NeRF-C3D with different backbones and RPN heads on 3D-FRONT.

Methods	Backbones	Heads	$Recall_{25}$	$Recall_{50}$	AP_{25}	AP_{50}
NeRF-RPN	ResNet-50	Anchor-based	97.8	67.7	66.5	37.4
NeRF-C3D (Ours)	ResNet-50	Anchor-based	97.1	67.7	**69.0**	**41.0**
NeRF-RPN	ResNet-50	Anchor-free	97.1	67.7	79.3	48.3
NeRF-C3D (Ours)	ResNet-50	Anchor-free	97.1	**69.1**	**80.3**	**52.2**
NeRF-RPN	VGG-16	Anchor-based	97.1	73.5	67.9	42.3
NeRF-C3D (Ours)	VGG-16	Anchor-based	**97.8**	**77.9**	**69.9**	**48.0**
NeRF-RPN	VGG-16	Anchor-free	95.6	67.7	83.6	53.4
NeRF-C3D (Ours)	VGG-16	Anchor-free	95.6	**73.5**	**85.4**	**59.5**
NeRF-RPN	Swin transformer	Anchor-based	98.5	63.2	51.8	26.6
NeRF-C3D (Ours)	Swin transformer	Anchor-based	98.5	**69.1**	**57.7**	**34.8**
NeRF-RPN	Swin transformer	Anchor-free	96.3	62.5	78.7	41.0
NeRF-C3D (Ours)	Swin transformer	Anchor-free	96.3	**64.0**	77.1	**42.8**

Note: Bold values indicate the performance metrics where our proposed model outperforms the baseline model under identical experimental conditions.

TABLE 3 | *Recall* and *AP* of NeRF-RPN and NeRF-C3D with different backbones and RPN heads on Hypersim.

Methods	Backbones	Heads	$Recall_{25}$	$Recall_{50}$	AP_{25}	AP_{50}
NeRF-RPN	ResNet-50	Anchor-based	39.4	7.6	5.2	0.3
NeRF-C3D (Ours)	ResNet-50	Anchor-based	**41.3**	**9.8**	**5.4**	**0.5**
NeRF-RPN	ResNet-50	Anchor-free	61.9	10.5	13.7	1.0
NeRF-C3D (Ours)	ResNet-50	Anchor-free	**64.1**	**14.0**	**13.9**	**2.0**
NeRF-RPN	VGG-16	Anchor-based	36.5	7.9	3.4	0.1
NeRF-C3D (Ours)	VGG-16	Anchor-based	**42.2**	**9.8**	3.3	**0.3**
NeRF-RPN	VGG-16	Anchor-free	61.3	13.7	17.3	3.2
NeRF-C3D (Ours)	VGG-16	Anchor-free	61.0	**15.2**	**19.2**	3.0
NeRF-RPN	Swin transformer	Anchor-based	43.2	10.2	5.2	0.8
NeRF-C3D (Ours)	Swin transformer	Anchor-based	**56.8**	**13.7**	**10.7**	**1.1**
NeRF-RPN	Swin transformer	Anchor-free	65.4	17.8	21.4	4.4
NeRF-C3D (Ours)	Swin transformer	Anchor-free	**68.9**	**24.8**	**29.7**	**10.8**

Note: Bold values indicate the performance metrics where our proposed model outperforms the baseline model under identical experimental conditions.

TABLE 4 | *Recall* and *AP* of NeRF-RPN and NeRF-C3D with different backbones and RPN heads on ScanNet.

Methods	Backbones	Heads	$Recall_{25}$	$Recall_{50}$	AP_{25}	AP_{50}
NeRF-RPN	ResNet-50	Anchor-based	80.8	16.8	18.1	0.6
NeRF-C3D (Ours)	ResNet-50	Anchor-based	**85.2**	**23.2**	**21.5**	**2.1**
NeRF-RPN	ResNet-50	Anchor-free	90.2	28.1	46.2	7.4
NeRF-C3D (Ours)	ResNet-50	Anchor-free	87.7	**29.6**	**50.7**	**8.8**
NeRF-RPN	VGG-16	Anchor-based	87.2	24.6	22.1	3.6
NeRF-C3D (Ours)	VGG-16	Anchor-based	**87.7**	24.6	**27.3**	**4.6**
NeRF-RPN	VGG-16	Anchor-free	91.6	34.0	49.0	10.5
NeRF-C3D (Ours)	VGG-16	Anchor-free	89.7	29.6	**51.9**	**11.8**
NeRF-RPN	Swin transformer	Anchor-based	88.7	33.0	31.5	4.2
NeRF-C3D (Ours)	Swin transformer	Anchor-based	**90.6**	**34.5**	**32.0**	**4.5**
NeRF-RPN	Swin transformer	Anchor-free	90.2	30.5	44.0	12.0
NeRF-C3D (Ours)	Swin transformer	Anchor-free	**92.1**	**33.0**	**48.0**	**16.1**

Note: Bold values indicate the performance metrics where our proposed model outperforms the baseline model under identical experimental conditions.

Additionally, we conducted ablation experiments using the Swin Transformer as the backbone network. The Swin Transformer, a spatial attention mechanism based on the Transformer architecture, effectively extracts contextual information from scenes and objects, and has been shown to significantly enhance the detection performance of models. Experimental results demonstrate that our proposed method retains significant advantages. When compared with the baseline model utilising the Swin Transformer on three different datasets, our optimised method achieves notable improvements across nearly all metrics, with some metrics performing on par with the baseline.

Moreover, we integrated the Swin Transformer backbone with our proposed method, achieving outstanding experimental results. Notably, on the Hypersim dataset, the Anchor-free method attained state-of-the-art performance, demonstrating substantial improvements over the baseline. Similarly, on the ScanNet dataset, our method delivered competitive performance, further validating its effectiveness.

Overall, NeRF-C3D achieves a certain degree of performance improvement compared to NeRF-RPN in various settings, indicating its effectiveness and potential. It is worth noting that our proposed MFCA can be integrated as a plugin on different backbone networks and RPN detection heads, and it will achieve certain improvements. However, there are significant differences in performance on different datasets. This highlights a limitation in the generalisation ability of our method. How to achieve better generalisation and eliminate differential performance on the dataset is the focus of our future research.

4.4 | Comparison With Other NeRF-Based Methods

NeRF-MAE Irshad et al. [35], the first large-scale self-supervised pretraining utilising NeRF radiance and density grid as an input modality. This method leverages MAE He et al. [49] for self-supervised pretraining across multiple datasets, allowing the model to acquire more generalised features and patterns. Consequently, it demonstrates exceptional generalisation capabilities and has achieved remarkable performance across a range of downstream tasks. Comparing our method with NeRF-MAE presents a considerable challenge. For a comprehensive comparison, refer to the results detailed in Table 5. We conducted a comparison using VGG-16 as the backbone network and two different detection head methods. The first row presents the Anchor-based methods, while the second row features the Anchor-free methods. NeRF-MAE uses Swin-Transformer Liu et al. [50] as the backbone network and similarly employs Anchor-free methods for detection. We marked the training methods of NeRF-MAE with different notations. The results indicate that our method can achieve comparable or even slightly superior performance in some metrics. For example, in the 3D-FRONT dataset, our Anchor-based method attained a higher *Recall*, and our Anchor-free method also demonstrated competitive *AP* scores. However, due to the advantages of NeRF-MAE's pretraining, including its strong generalisation capability and ability to learn from large datasets, it outperforms our method on the ScanNet dataset. Notably, NeRF-MAE requires eight A100 GPUs for 6 days of pre-training on the 3D-FRONT dataset, which significantly exceeds the training time of our method.

4.5 | Qualitative Comparison

We use MeshLab Cignoni et al. [52] to visualise the results of proposals. Figure 6 shows the visualisation results produced by NeRF-RPN and NeRF-C3D. The proposals for these scenarios are all generated by the anchor-free RPN head. Compared with NeRF-RPN, our method can more accurately regress object proposals. We have identified the areas with missing predictions on the visualisation results of NeRF-RPN using red circular boxes. For the first, second, and fourth scenarios, our method corrects the erroneous predictions and missed predictions made by NeRF-RPN. In the second and third scenarios, we reduced duplicate predictions, and in the second scenario, we predicted a proposal that was closer to the real scene object compared to Ground Truth. In addition, in the third, fifth, and sixth scenarios, our method overcome the difficulty of detecting small objects. From the visualisation results, our method performs poorly in detecting multi-object overlap in the scene. For instance, in the third scenario, the overlap prediction of multiple chairs is not good enough. When facing such a scenario, only partial object predictions can be made. Improving solutions for such challenges stands as one of our future research directions. Based on the visualisation results, we consider the performance of 3D object detection heavily depends on the reconstruction quality of NeRF. Poor reconstruction quality will greatly affect the detection ability of the model.

TABLE 5 | Quantitative comparisons with NeRF-MAE on 3D-FRONT and ScanNet.

Methods	Backbones	Heads	3D-FRONT				ScanNet			
			$Recall_{25}$	$Recall_{50}$	AP_{25}	AP_{50}	$Recall_{25}$	$Recall_{50}$	AP_{25}	AP_{50}
NeRF-MAE ⋆	Swin transformer	Anchor-free	96.2	67.5	78.0	54.3	89.7	36.1	51.0	14.5
NeRF-MAE ∘	Swin transformer	Anchor-free	96.3	74.3	83.0	59.1	90.5	39.1	54.3	15.5
NeRF-MAE △	Swin transformer	Anchor-free	97.2	74.5	85.3	63.0	92.0	39.5	57.1	17.0
NeRF-C3D (Ours)	Swin transformer	Anchor-based	98.5	69.1	57.7	34.8	90.6	34.5	32.0	4.5
NeRF-C3D (Ours)	Swin transformer	Anchor-free	96.3	64.0	77.1	42.8	92.1	33.0	48.0	16.1

Note: ⋆ represents no self-supervised pretraining with the 3D-FRONT dataset, ∘ represents pretraining with the 3D-FRONT dataset, and △ represents pretraining with the 3D-FRONT, HM3D Ramakrishnan et al. [51], and Hypersim datasets, with the latter two methods involving data augmentation prior to pretraining. The experimental data for NeRF-MAE are obtained from Irshad et al. [35].

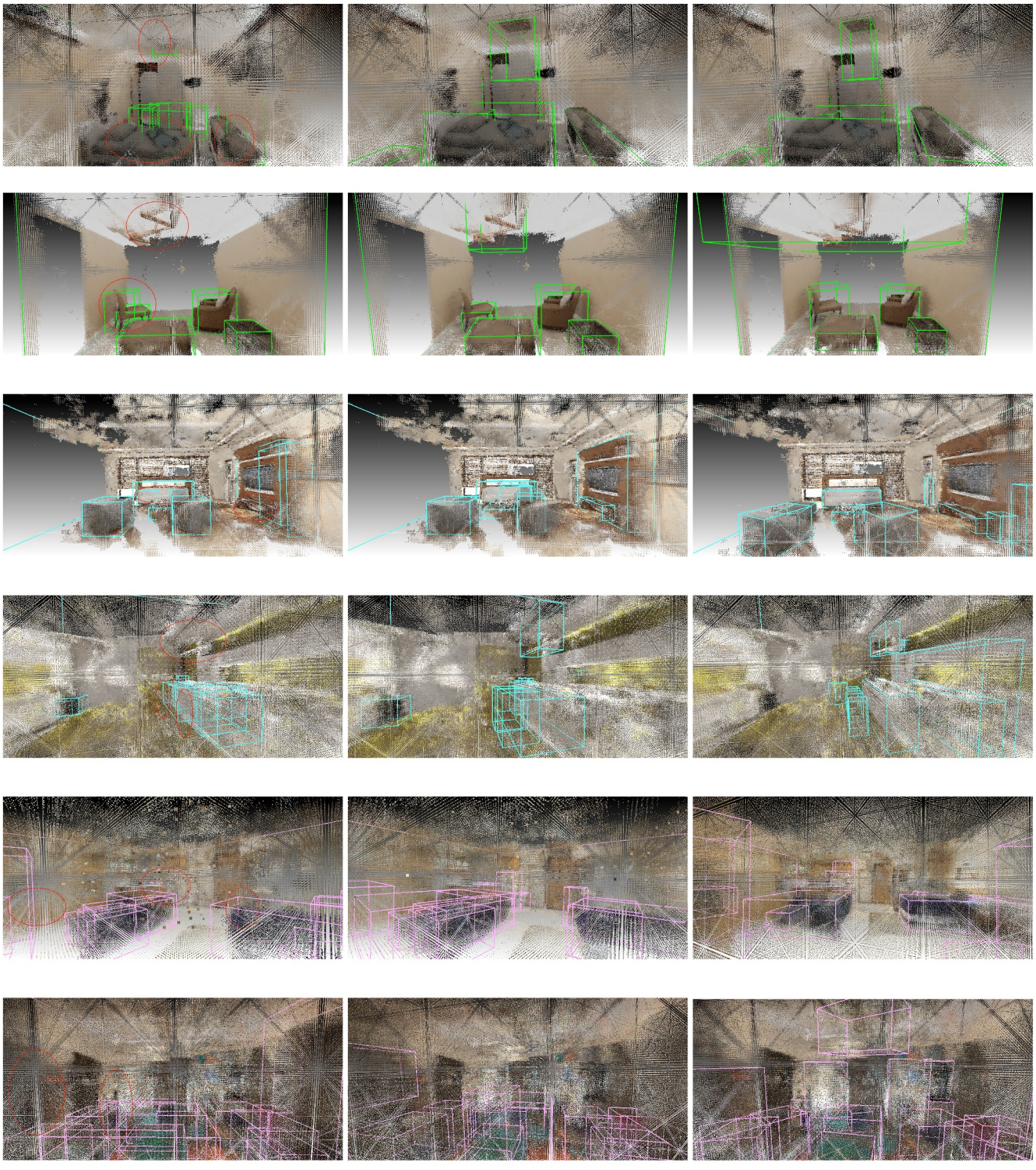

FIGURE 6 | Qualitative comparison between NeRF-RPN and NeRF-C3D. The first column shows the prediction proposals generated by NeRF-RPN; the second column displays the prediction proposals generated by the improved NeRF-C3D; the third column is Ground Truth. From top to bottom, scene 1–2, 3–4, and 5–6 are from 3D-FRONT, Hypersim and ScanNet, respectively. Red circles highlight the significant omissions of the baseline model.

4.6 | Quantitative Comparison

In order to prove that our proposed method can not only improve the performance of the original baseline model, but also achieve competitive ability with other method in the field of 3D object detection, we compare with two advanced methods, ImVoxelNet [20] and FCAF3 [29]. We use the raw images used for NeRF training and the point cloud obtained from the depth map as substitutes. In our experiments, we opted for VGG-16 as the backbone network and employed Anchor-free RPN heads.

As depicted in Table 6, the performance of our proposed NeRF-C3D significantly outperforms ImVoxelNet across both the Hypersim and ScanNet datasets. However, on the 3D-FRONT dataset, our method exhibits slightly inferior performance compared to ImVoxelNet. Additionally, when compared with FCAF3D, NeRF-C3D still maintains a leading position on the FRONT-3D dataset. Nevertheless, its performance on Hypersim and ScanNet datasets, is slightly less competitive. Overall, our NeRF-C3D method surpasses FCAF3D, particularly maintaining a leading position on the 3D-FRONT dataset, while significantly outperforming ImVoxelNet on the Hypersim and ScanNet datasets but slightly underperforming on the FRONT-3D dataset.

5 | Limitations

Based on experimental data and visualisation results, NeRF-C3D demonstrates relative advantages across several benchmark datasets but also reveals certain limitations.

5.1 | Generalisation Capabilities of NeRF-C3D

NeRF-C3D shows noticeable performance variability across different datasets. While it outperforms NeRF-RPN on 3D-FRONT and ScanNet, its improvements in the Hypersim dataset, especially in AP_{50}, are minimal. This suggests that NeRF-C3D may have issues with adaptability to varying dataset characteristics, leading to inconsistent performance. When compared to NeRF-MAE, NeRF-C3D does not surpass its performance on 3D-FRONT and ScanNet, particularly under extensive pretraining across multiple datasets. This suggests that NeRF-C3D has limitations in its generalisation capabilities, particularly in more complex scenarios, where its performance could be further improved.

5.2 | Limited Adaptability to Complex Scenes

Although NeRF-C3D demonstrates advantages on several benchmark datasets, it has limitations in complex scenes. In particular, on ScanNet, it experiences a decline in $Recall_{50}$ despite improvements in other metrics. Visualisation results further indicate that NeRF-C3D performs moderately in object detection within complex environments, struggling to balance recall and precision under high-precision conditions, which impacts its effectiveness in intricate scenarios.

TABLE 6 | Quantitative comparison with other types of 3D object detection methods.

	3D-FRONT		Hypersim		ScanNet	
Methods	AP_{25}	AP_{50}	AP_{25}	AP_{50}	AP_{25}	AP_{50}
ImVoxelNet	86.1	66.4	9.7	2.3	37.3	9.8
FCAF3D	73.1	35.2	30.7	8.8	63.7	18.5
NeRF-C3D (Ours)	85.4	59.5	19.2	3.0	51.9	11.8

While NeRF-C3D demonstrates potential as an improved method across multiple datasets, the issues of cross-dataset stability, adaptability to complex scenes, and comparative disadvantages against other advanced methods indicate areas where its generalisation capability and performance optimisation could be further refined. These challenges offer valuable insights and directions for future research and development.

6 | Conclusion

In this paper, we propose a 3D object detection method NeRF-C3D in NeRFs with a Channel attention architecture. Our method aims to refine representation of density field features and fully utilise the feature dependency relationships in neural fields by introducing Channel attention in the process of multi-scale feature fusion. Through experiments combining different datasets, backbones, and RPN heads, we demonstrate that our proposed NeRF-C3D can improve 3D object detection capabilities in NeRFs. Furthermore, our proposed method introduces only modest modifications to the baseline model, yet demonstrates a noticeable improvement in 3D object detection tasks. We hope our research provides valuable insights for future studies in 3D object detection with NeRF.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Data available on request from the authors.

References

1. L. Wang, L. Du, X. Ye, et al., "Depth-Conditioned Dynamic Message Propagation for Monocular 3D Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), 454–463.

2. X. Weng and K. Kitani, "Monocular 3D Object Detection With Pseudo-Lidar Point Cloud," in *IEEE International Conference on Computer Vision (ICCV) Workshops* (2019).

3. Y. Liu, T. Wang, X. Zhang, and J. P. Sun, "Position Embedding Transformation for Multi-View 3D Object Detection," in *European Conference on Computer Vision* (Springer, 2022), 531–548.

4. Z. Li, W. Wang, H. Li, et al., "Bevformer: Learning Bird's-Eye-View Representation From Multi-Camera Images via Spatiotemporal Transformers," in *European Conference on Computer Vision* (Springer, 2022), 1–18.

5. J. Philion and S. Fidler, "Lift, Splat, Shoot: Encoding Images From Arbitrary Camera Rigs by Implicitly Unprojecting to 3D," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16* (Springer, 2020), 194–210.

6. I. Misra, R. Girdhar, and A. Joulin, "An End-to-End Transformer Model for 3D Object Detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 2906–2917.

7. C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep Hough Voting for 3D Object Detection in Point Clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), 9277–9286.

8. H. Sheng, S. Cai, Y. Liu, et al., "Improving 3D Object Detection With Channel-Wise Transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 2743–2752.

9. X. Bai, Z. Hu, X. Zhu, et al., "Transfusion: Robust Lidar-Camera Fusion for 3D Object Detection With Transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 1090–1099.

10. J. Yan, Y. Liu, J. Sun, et al., "Cross Modal Transformer: Towards Fast and Robust 3D Object Detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), 18268–18278.

11. B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing Scenes as Neural Radiance Fields for View Synthesis," *Communications of the ACM* 65, no. 1 (2021): 99–106.

12. B. Hu, J. Huang, Y. Liu, Y. W. Tai, and C. K. Tang, "NeRF-RPN: A General Framework for Object Detection in NeRFs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), 23528–23538.

13. J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 5470–5479.

14. D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2022), 5481–5490.

15. T. Müller, A. Evans, C. Schied, and A. Keller, "Instant Neural Graphics Primitives With a Multiresolution Hash Encoding," *ACM Transactions on Graphics (ToG)* 41, no. 4 (2022): 1–15, https://doi.org/10.1145/3528223.3530127.

16. S. J. Garbin, M. Kowalski, M. Johnson, J. Shotton, and J. Valentin, "FastNeRF: High-Fidelity Neural Rendering at 200FPS," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 14346–14355.

17. A. Chen, Z. Xu, F. Zhao, et al., "MVSNeRF: Fast Generalizable Radiance Field Reconstruction From Multi-View Stereo," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 14124–14133.

18. A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelNeRF: Neural Radiance Fields From One or Few Images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), 4578–4587.

19. R. Zhang, H. Qiu, T. Wang, et al., "MonoDETR: Depth-Guided Transformer for Monocular 3D Object Detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), 9155–9166.

20. D. Rukhovich, A. Vorontsova, and A. 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* (2022), 2397–2406.

21. Y. You, Y. Wang, W. L. Chao, et al., "Pseudo-Lidar++: Accurate Depth for 3D Object Detection in Autonomous Driving," *arXiv preprint arXiv:1906.06310* (2019).

22. Y. Liu, J. Yan, F. Jia, et al., "PETRv2: A Unified Framework for 3D Perception From Multi-Camera Images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), 3262–3272.

23. C. Yang, Y. Chen, H. Tian, et al., "BEVFormer V2: Adapting Modern Image Backbones to Bird's-Eye-View Recognition via Perspective Supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), 17830–17839.

24. H. Zhang, H. Li, X. Liao, et al., "DA-BEV: Depth Aware Bev Transformer for 3D Object Detection," *arXiv preprint arXiv:2302.13002* (2023).

25. X. Chen, H. Zhao, G. Zhou, and Y. Q. Zhang, "PQ-Transformer: Jointly Parsing 3D Objects and Layouts From Point Clouds," *IEEE Robotics and Automation Letters* 7, no. 2 (2022): 2519–2526, https://doi.org/10.1109/lra.2022.3143224.

26. C. He, R. Li, S. Li, and L. Zhang, "Voxel Set Transformer: A Set-to-Set Approach to 3D Object Detection From Point Clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 8417–8427.

27. J. Mao, Y. Xue, M. Niu, et al., "Voxel Transformer for 3D Object Detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 3164–3173.

28. Z. Liu, X. Zhao, T. Huang, R. Hu, Y. Zhou, and X. T. Bai, "Robust 3D Object Detection From Point Clouds With Triple Attention," *Proceedings of the AAAI Conference on Artificial Intelligence* 34, no. 7 (2020): 11677–11684, https://doi.org/10.1609/aaai.v34i07.6837.

29. D. Rukhovich, A. Vorontsova, and A. Konushin, "FCAF3D: Fully Convolutional Anchor-Free 3D Object Detection," in *European Conference on Computer Vision* (Springer, 2022), 477–493.

30. Y. Li, A. W. Yu, T. Meng, et al., "DeepFusion: Lidar-Camera Deep Fusion for Multi-Modal 3D Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 17182–17191.

31. C. Xu, B. Wu, J. Hou, et al., "NeRF-Det: Learning Geometry-Aware Volumetric Representation for Multi-View 3D Object Detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (2023), 23320–23330.

32. C. Huang, Y. Hou, W. Ye et al., "NeRF-Det++: Incorporating Semantic Cues and Perspective-Aware Depth Supervision for Indoor Multi-View 3D Detection," *IEEE Transactions on Image Processing* 34 (2024): 2575–2587, https://doi.org/10.1109/TIP.2025.3560240.

33. J. Xu, L. Peng, H. Cheng, et al., "MonoNeRD: NeRF-Like Representations for Monocular 3D Object Detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), 6814–6824.

34. T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), 2117–2125.

35. M. Z. Irshad, S. Zakharov, V. Guizilini, A. Gaidon, Z. Kira, and R. Ambrus, "NeRF-MAE: Masked Autoencoders for Self-Supervised 3d Representation Learning for Neural Radiance Fields," in *European Conference on Computer Vision (ECCV)* (2024).

36. K. Gao, Y. Gao, H. He, D. Lu, L. Xu, and J. Li, "NeRF: Neural Radiance Field in 3D Vision, a Comprehensive Review," *arXiv preprint arXiv:2210.00379* (2022).

37. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556* (2014).

38. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), 770–778.

39. M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 10781–10790.

40. C. Li, Z. Qu, and S. Wang, "An Object Detection Approach With Residual Feature Fusion and Second-Order Term Attention Mechanism," *CAAI Transactions on Intelligence Technology* 9, no. 2 (2024): 411–424, https://doi.org/10.1049/cit2.12236.

41. J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), 7132–7141.

42. S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), 3–19.

43. Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully Convolutional One-Stage Object Detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), 9627–9636.

44. T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision* (2017), 2980–2988.

45. D. Zhou, J. Fang, X. Song, et al., "IoU Loss for 2D/3D Object Detection," in *2019 International Conference on 3D Vision (3DV)* (IEEE, 2019), 85–94.

46. H. Fu, B. Cai, L. Gao, et al., "3D-Front: 3D Furnished Rooms With Layouts and Semantics," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 10933–10942.

47. M. Roberts, J. Ramapuram, A. Ranjan, et al., "Hypersim: A Photo-realistic Synthetic Dataset for Holistic Indoor Scene Understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 10912–10922.

48. A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), 5828–5839.

49. K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked Autoencoders Are Scalable Vision Learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 16000–16009.

50. Z. Liu, Y. Lin, Y. Cao, et al., "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 10012–10022.

51. S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, et al., "Habitat-Matterport 3D Dataset (HM3D): 1000 Large-Scale 3D Environments for Embodied AI," *arXiv preprint arXiv:2109.08238* (2021).

52. P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: An Open-Source Mesh Processing Tool," in *Eurographics Italian Chapter Conference*, ed. V. Scarano, R. D. Chiara, and U. Erra (Eurographics Association, 2008).