CasA: A Cascade Attention Network for 3D Object Detection from LiDAR point clouds

Hai Wu, Jinhao Deng, Chenglu Wen, *Senior Member, IEEE*, Xin Li, *Senior Member, IEEE*, Cheng Wang, *Senior Member, IEEE*, and Jonathan Li, *Senior Member, IEEE*

***Abstract*—3D object detection from LiDAR point clouds has gained great attention in recent years due to its wide applications in smart cities and autonomous driving. Cascade framework shows its advancement in 2D object detection but is less investigated in 3D space. Conventional cascade structures use multiple *separate* sub-networks to sequentially refine region proposals. Such methods, however, have limited ability to measure proposal quality in all stages, and hard to achieve a desirable performance improvement in 3D space. This paper proposes a new cascade framework, termed CasA, for 3D object detection from LiDAR point clouds. CasA consists of a Region Proposal Network (RPN) and a Cascade Refinement Network (CRN). In CRN, we designed a new Cascade Attention Module that uses multiple sub-networks and attention modules to aggregate the object features from different stages and progressively refine region proposals. CasA can be integrated into various two-stage 3D detectors and improve their performance. Extensive experiments on KITTI and Waymo datasets with various baseline detectors demonstrate the universality and superiority of our CasA. In particular, based on one variant of Voxel-RCNN, we achieve state-of-the-art results on the KITTI dataset. On the KITTI online 3D object detection leaderboard, we achieve a high detection performance of 83.06%, 47.09%, and 73.47% Average Precision (AP) in the moderate Car, Pedestrian, and Cyclist classes, respectively.**

***Index Terms*—LiDAR point clouds, 3D object detection, deep learning, cascade network.**

I. Introduction

3D object detection is one of the key tasks in scene understanding. Light Detection and Ranging (LiDAR) technology provides accurate 3-D spatial information in the form of point clouds. In recent years, 3D object detection from LiDAR point clouds has gained more and more attention due to its wide application in smart cities [1], urban planning [2], and autonomous vehicles [3]. Compared with the well-studied 2D detection problem, 3D object detection from LiDAR point clouds is challenging as the data collected by LiDAR typically exhibit sparse and irregular distribution.

This work was supported in part by the National Natural Science Foundation of China under Grant 62171393, and the National Key R&D Program of China under Grant 2021YFF070460. (Corresponding author: Chenglu Wen.)

Hai Wu, Jinhao Deng, Chenglu Wen and Cheng Wang are with the Fujian Key Laboratory of Sensing and Computing for Smart Cities and the School of Informatics, Xiamen University, Xiamen, FJ 361005, China (e-mail: wuhai@stu.xmu.edu.cn; jinhaodeng@stu.xmu.edu.cn; clwen@xmu.edu.cn; cwang@xmu.edu.cn).

Xin Li is with the School of Electrical Engineering and Computer Science, Louisiana State University, USA. (e-mail: xinli@cct.lsu.edu)

Jonathan Li is with the Departments of Geography and Environmental Management /Systems Design Engineering, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. (e-mail: junli@uwaterloo.ca)

Existing methods follow single-stage or two-stage 3D detection frameworks. Single-stage methods directly perform object detection using encoded features of point clouds [4]–[6], while two-stage methods follow the Region-based Convolutional Neural Networks (RCNN) framework presented in [7]. The latter approaches generate a set of candidate bounding boxes and then further classify and refine each candidate box. Many recent studies adopt the two-stage framework because of its higher accuracy. MV3D [8] generates object proposals from 2D multi-view feature maps, then refines proposals by using 2D features. Point RCNN [9] and Voxel RCNN [10] generate and refine region proposals using point and voxel features, respectively. These two-stage approaches usually refine region proposals using a single sub-network. However, the LiDAR scanning in 3D space is uneven. There exists a huge distribution gap between nearby and distant objects. The single sub-network has limited ability to learn objects under such diverse conditions, hindering the detection performance.

Some 2D object detectors address a similar problem by integrating multiple separate sub-networks. For example, the multi-region detector [11] proposes an iterative bounding box regression that uses the same RCNN model several times to refine detection bounding boxes. Cascade RCNN [12], [13] cascades a series of RCNN heads and uses the output of one stage to train the next. In 3D space, as a pioneering work, Meng et al. [14] presented a weakly supervised detector that directly applies the cascade framework for point clouds. Nevertheless, so far, the multi-stage framework for fully supervised 3D object detection is still under-explored. Besides, directly applying the aforementioned cascade structures to 3D object detection using a separate sub-network only enhances/refines the object from each stage to a limited extent, and often fails to achieve performance gain.

This paper proposes **CasA**, a new **Cas**cade **A**ttention-based multi-stage framework that performs 3D object detection from vehicle-borne LiDAR point clouds. CasA consists of a Region Proposal Network (RPN) and a Cascade Refinement Network (CRN). The RPN uses a 3D backbone network to encode voxels into 3D feature volumes. Then a 2D detection head is adopted to generate region proposals. Unlike most state-of-the-art two-stage 3D detectors that refine region proposals using a single sub-network, our CRN progressively refines and complements predictions from a series of sub-networks into high-quality predictions. Besides, different from conventional cascade structures that use the output of one stage to train the

next, we designed a new Cascade Attention Module (CAM) to aggregate features from different stages for comprehensive region proposal refinement. In addition, CasA integrates a part-aided scoring that considers object completeness of parts as Structure Aware Single-stage 3D Object Detection (SA-SSD) [15] to better estimate proposal confidence. Compared with recently released high-performance detectors [16], [17], our method uses only LiDAR points while the above two methods use both LiDAR points and images. Moreover, the most advantage of our method is the expandability. Our cascade attention design can be extended to any two-stage 3D detectors and greatly improves their detection performance. Experimental results demonstrated the CasA's consistent performance improvement over multiple baseline detectors. For example, on the widely used KITTI [18] validation set (moderate car class), CasA improves PV-RCNN [19], Voxel-RCNN [10] and CT3D [20] with 2.27%, 2.06% and 0.57% Average Precision (AP)(R11) respectively. On the KITTI test set (moderate car class), CasA improves the Voxel-RCNN with 1.44%, and achieves the state-of-the-art of 83.06% AP(R40). We believe this effective design can be of interest to many 3D downstream tasks.

Our contributions are summarized as follows:

- We propose a new cascade framework, CasA, for 3D object detection from LiDAR point clouds, which progressively refines and complements predictions through multiple sub-networks to obtain high-quality predictions. CasA can significantly improve the performance of various state-of-the-art 3D object detectors.
- We propose a Cascade Attention Module (CAM) to aggregate object features from different stages. CAM comprehensively considers the quality of proposals from all of previous stages, significantly boosting the accuracy of proposal refinement.

II. Related Work

A. 3D Object Detection from LiDAR Point Clouds

Most recent 3D detectors follow a single-stage or two-stage framework. Single-stage methods directly perform object detection using encoded features of point clouds. SECOND [4], SIEV-Net [3], SA-SSD [15] and SE-SSD [21] first convert the point cloud into voxels, then abstract voxel features using 3D backbone network, and finally predict object confidences and boxes from the encoded features. 3DSSD [5] extracts point features by set abstraction layers to perform single-stage object detection. Two-stage methods usually first generate region proposals using an RPN, then extract the region features of the proposals by employing a Region of Interest (RoI) pooling method, and finally refine the proposal using the extracted features. Some previous methods generate and refine region proposals using Multi-view [8] or BEV [22] features. PointRCNN [9] generates and refines region proposals using point features. Multiple methods [19], [23] encode point clouds into 3D feature volumes using 3D sparse convolution, compress the features into BEV representation to generate object proposals, and then refine the object proposals. Recent methods applied transformers for 3D object detection. Votr [24] uses a Voxel Transformer to encode point clouds and CT3D [20] applies a channel-wise transformer to refine object proposals.

B. Multi-Stage Network for Object Detection

Multi-stage (beyond two-stage) methods have been more widely explored and demonstrated effective in 2D object detection. Iterative bounding box regression [25] uses the same R-CNN model several times to refine detection bounding boxes. IntegralLoss [26] integrates the results from multiple proposal refinement networks to perform object detection. Cascade R-CNN [12], [13] cascades several detection networks, taking the output of the former as the input to train the latter. In this way, it progressively obtains results with raising Intersection over Unions (IoUs) and finally achieves significant performance gain. IoU-Net [27] improves Cascade R-CNN by predicting the IoU between the detected object and the matched ground truth. Gong et al. [28] introduced an LSTM-based proposal refinement module that iteratively refines bounding box proposals, further improving the 2D detection performance. In 3D space, as a pioneering work, Meng et al. [14] proposed a weakly supervised object detector that directly applies a cascade framework for point clouds.

However, multi-stage methods for fully supervised 3D object detection on point clouds are still under-explored. In this work, we propose CasA, an effective multi-stage 3D object detection framework that refines and complements predictions from multiple sub-networks into high-quality detections.

C. Attention-Based Network

Recently, the attention-based approach has been widely applied in the field of computer vision. For image, Dosovitskiy et al. [29] proposed a Vision Transformer that directly applies the attention mechanism to sequences of image patches. DETR [30] views object detection as a direct set prediction problem and builds an end-to-end object detector based on a transformer. Zhu et al. proposed a spectral-spatial attention network for hyperspectral image classification [31]. CFCA-Net [32] proposed an cascaded feature attention method for cloud detection from satellite images. For point cloud, Zhao et al. [33] leverages the attention mechanism to process point cloud data and proposes the point transformer architecture as a general backbone. P4Transformer [34] uses self-attention to capture the appearance and motion information in point cloud video effectively. Inspired by these works, this paper adopts an attention-based method to aggregate features from multiple sub-networks to achieve more accurate 3D object detection.

III. Method

CasA is a multi-stage detection framework and can be integrated into various two-stage 3D detectors. Current multi-stage methods and cascade structures [12] use a series of separate sub-networks to refine object proposals. Generally, these methods can learn object features under various difficult conditions. However, in these separate sub-networks, a later stage has limited ability to measure proposal quality in all previous stages. This hinders the effectiveness of proposal refinement in the 3D scene.

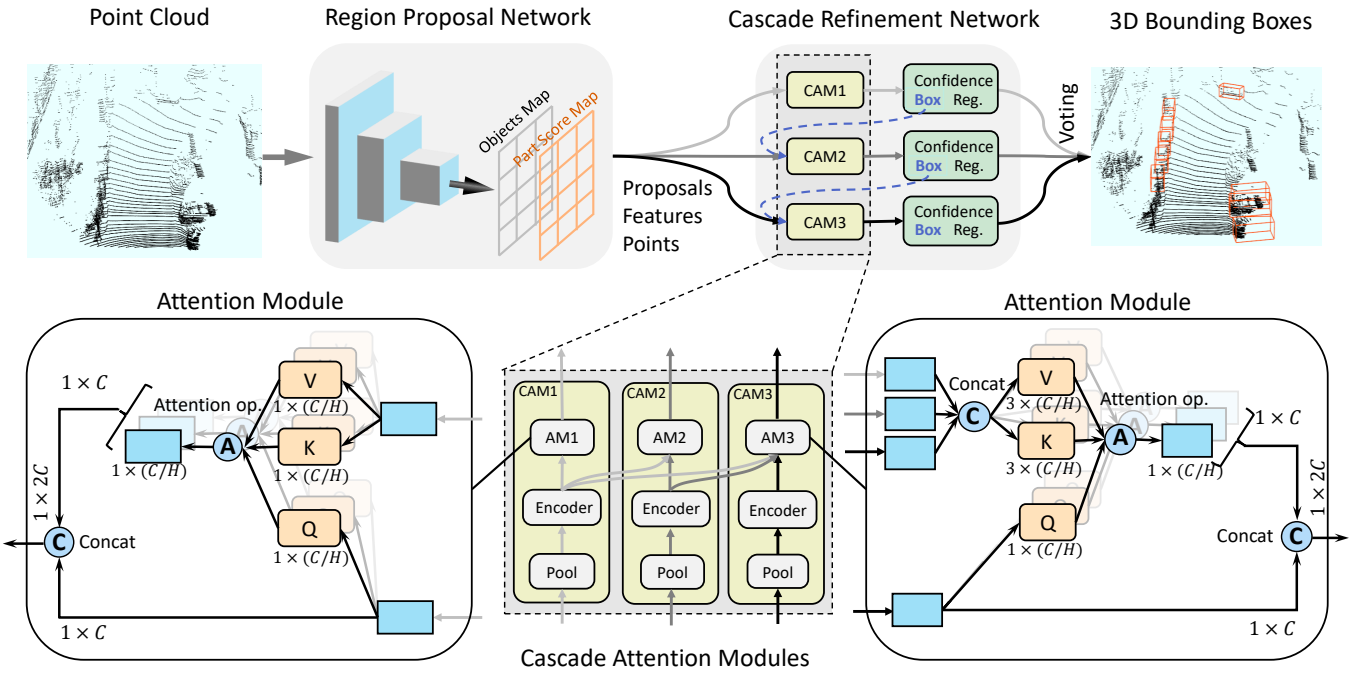


Fig. 1. Our CasA architecture. CasA generates proposals of 3D bounding boxes by using a voxel-based RPN, which consists of a 3D backbone and a 2D detection head. The proposals are progressively refined by a CRN, in which the features from different stages are aggregated by Cascade Attention Modules.

Our idea is to aggregate the features from all the stages in a cascade attention manner. As illustrated in Fig. 1, CasA consists of an RPN and a CRN. The RPN first uses a 3D backbone network and a 2D detection head to generate region proposals. The CRN consists of multiple sub-networks that progressively refine the proposals. In this CRN, we develop a new cascade attention scheme, which aggregates the proposal features from different stages for a more comprehensive bounding box prediction. We elaborate on this cascade attention design first.

A. Cascade Attention for Proposal Refinement

1) Vanilla Cascade Structure: The cascade detection framework is well studied for 2D images. Cascade R-CNN [12] uses a vanilla cascade structure, which employs a series of separate sub-networks with raising Intersection over Union (IoU) thresholds to refine region proposals. Such a vanilla cascade structure consists of N_r refiners. The j-th refiner takes a region proposal B^{j-1} from the previous stage as input, extracts object feature F^j with a feature extractor $\phi(\cdot)$. Then, with F^j, a confidence prediction branch $\mathcal{S}(\cdot)$ and a box regression branch $\mathcal{R}(\cdot)$ output a new object confidence C^j and box B^j, respectively. This iterative refinement can be formulated as

$$F^j = \phi^j(B^{j-1}), C^j = \mathcal{S}^j(F^j), B^j = \mathcal{R}^j(F^j), \quad (1)$$

where $j = 1, 2, ..., N_r$. Such a design has been shown effective in 2D object detection.

However, directly applying this vanilla cascade structure in 3D does not bring desirable improvement. For example, on the KITTI [18] validation set, by using the state-of-the-art Voxel-RCNN [10] detector with a vanilla cascade structure, the detection performance (on the moderate car class), as shown in Table I, does not improve.

TABLE I
THE RESULTS ON THE KITTI VALIDATION SET (MODERATE CAR CLASS) BY APPLYING A VANILLA CASCADE STRUCTURE TO VOXEL-RCNN [10]. IT IS EVALUATED BY AVERAGE PRECISION (AP)(%) UNDER 40 RECALL THRESHOLDS.

Detector	Test stages		
	1	2	3
Voxel-RCNN	85.27	-	-
+Vanilla Cascade	82.96	84.88	84.82

The reasons for undesirable performance improvement are:

(1) Ignore distant objects. In multi-stage approaches, the later stages tend to be over-fitting due to the lack of negative training samples. 2D approaches establish rising IoU thresholds to re-sample balanced samples. However, in 3D point clouds, such re-sampling leads to unbalanced training between nearby and distant objects. Since point clouds are typically non-uniformly distributed. The nearby objects with dense points can produce high-quality proposals selected as positive samples, while distant objects tend to be negatives. Under such unbalanced training, a later stage predicts accurate nearby objects while ignoring distant objects. To tackle this problem, we increase more object appearances from all of the previous stages, which ensures a later stage still has enough evidence to recover the ignored distant objects.

(2) Error propagation problem. 3D detection is more challenging due to the need for object height and non-axis aligned angle estimation. Small errors can propagate along with the downstream multi-stage framework, leading to detection failure. To address this, We build more connections between

stages, and the errors from a single stage can be fixed by other stages in a complementary manner.

In short, our idea is to establish effective connections between these refiners to compose effective refinement. To this end, we designed a new feature extractor, named Cascade Attention Module, to aggregate object features across different stages.

2) Feature Aggregation through Cascade Attention: As analyzed before, we aggregate the features across stages to increase object appearance to more accurately and robustly detect distant and hard objects.

Given a region proposal B^{j-1}, the majority of the existing detectors [19] apply a region pooling module $\mathcal{P}(\cdot)$ and a feature encoding module $\mathcal{E}(\cdot)$ to extract the proposal features $\hat{F}^j \in \mathbb{R}^{1\times C}$ for box regression and confidence prediction, where C is the feature dimension. In a cascade structure, however, such a strategy can only capture the proposal feature of the current stage while ignoring previous stages. In contrast, we further aggregate the features across stages. A naive method is directly concatenating the features from different stages. However, it is hard to learn the feature importance between stages, and brings marginal performance improvement (see Table VI). Inspired by the recent attention methods [24], we develop an attention-based operation to perform proposal features aggregation from different stages.

For each encoded feature $\hat{F}^j$, we first concatenate a stage embedding P^j which is calculated by the position embedding in [35]. $\hat{F}^j = [\hat{F}^j, P^j]$. At the j-th refinement stage, we collect the encoded features from all previous stages and current stage $\mathbf{F}^j = [\hat{F}^0, \hat{F}^1, ..., \hat{F}^j]$. Then we have $\mathbf{Q}^j = \hat{F}^j\mathbf{W}_q^j$, $\mathbf{K}^j = \mathbf{F}^j\mathbf{W}_k^j$, $\mathbf{V}^j = \mathbf{F}^j\mathbf{W}_v^j$, where $\mathbf{W}_q^j, \mathbf{W}_k^j$ and $\mathbf{W}_v^j$ are linear projections. The $\mathbf{Q}^j$, $\mathbf{K}^j$ and $\mathbf{V}^j$ are query, key and value embeddings. To enhance the representational ability, we also adopt multi-head design. The embeddings from i-th head are denoted as $\mathbf{Q}_i^j$, $\mathbf{K}_i^j$ and $\mathbf{V}_i^j$. The attention value of a single head is calculated by

$$\hat{\mathbf{F}}_i^j = softmax(\frac{\mathbf{Q}_i^j(\mathbf{K}_i^j)^T}{\sqrt{C'}})\mathbf{V}_i^j, \tag{2}$$

where C' is the feature dimension in multi-head attention. Intuitively, the features from the current stage contribute more to the proposal refinement. Hence, we also concatenate the features $\hat{F}^j$ with the H multi-head attention features to formulate the feature vector F^j which is used for box regression and confidence prediction.

$$F^j = Concat(\hat{F}^j, \hat{\mathbf{F}}_1^j, \hat{\mathbf{F}}_2^j, ..., \hat{\mathbf{F}}_H^j). \tag{3}$$

For the first refinement stage, our module actually performs a self-attention operation. For other stages, we perform cross attention operations that aggregate features from different stages. By adopting such a cascade attention design, our CasA can better estimate proposal quality in all the stages, which helps improve the proposal refinement accuracy.

3) Box Regression and Part-aided Scoring: To perform a box regression, we follow the [10], [19], which regress the box size, location and orientation residuals relative to the input 3D proposal. We also designed a part-aided score α^j to enhance the confidence prediction (see Fig. 2). This is inspired by

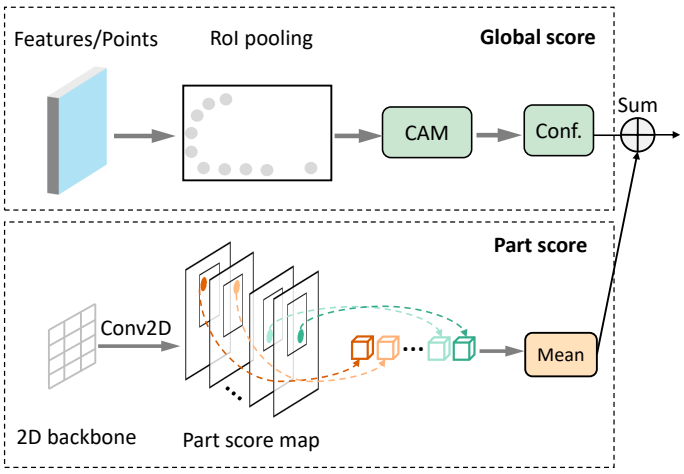


Fig. 2. Part-aided scoring. The object confidence is calculated by the sum of part scores from BEV map and global score from confidence branch.

the part-sensitive warping [15] that averages the object scores from a part score map. Such a design can help improve the confidence estimation and we incorporate it into our CasA.

In our pipeline, we compute the part-aided score from both local (part-based) and global views. Specifically, α^j is computed by the confidence prediction branch $\mathcal{S}(\cdot)$ and the part-sensitive warping $\mathcal{W}(\cdot)$ as

$$\alpha^j = \mathcal{S}^j(F^j) + \mathcal{W}(B^j, \mathbf{X}), \tag{4}$$

where $\mathbf{X}$ is the part score map predicted by RPN. With this modification, this new part-aided scoring helps the detector more accurately estimate object confidence in each stage.

In training, similar to Cascade R-CNN [12], we set the 3D IoU thresholds $u = \{u^1, u^2, ..., u^{N_r}\}$ to define the negatives and positives at different refinement stages. In testing, we average the boxes and scores from all refinement stages to generate final detection results.

4) Boxes Voting: As analyzed in Section A.1, 3D detection is more challenging due to the need for object height and non-axis aligned angle estimation. The errors tend to propagate along with the downstream multi-stage framework. To further address this problem, during testing, we propose boxes voting to build more connections between stages. This is motivated by an intuition that each stage outputs both weak and strong predictions which can be ensembled together to generate more accurate predictions. Bearing this in mind, we explore methods to merge the boxes from all refiners. A simple method is to directly perform non-maximum suppression (NMS) on all boxes, it assembles the result by selecting the boxes with the highest confidence. However, it ignores lots of boxes with low confidence, which are potential to recover missed objects. To address this, we adopt a weighted boxes voting that directly averages the detection confidence and merges the boxes weighted by detection confidence as

$$C = \frac{1}{N_r}\sum_j C^j, \tag{5}$$

$$B = \frac{1}{\sum_j C^j}\sum_j C^j \cdot B^j, \tag{6}$$

where C and B are merged confidence and box respectively. After boxes voting, we obtain a set of refined high-quality boxes. Nevertheless, there are still lots of redundant boxes as each object has many refined proposals. To remove the redundant boxes, we finally perform an NMS on the voted results to produce detection outputs. By adopting the voting mechanism, various predictions (with less confidence, and from different perspectives/scales) generated by different refiners can compose, in a complementary manner, into more accurate/reliable final predictions.

B. Backbone Network

Many recent pipelines [4], [19] use a 3D sparse convolution as backbone networks for accuracy and efficiency, we also adopt this setting. We first split the raw points $\mathbf{P}$ into small voxels. For each voxel, we calculate the raw features using the mean of raw features of all inside points. We adopt the 3D sparse convolution $\mathcal{S}(\cdot)$ to encode 3D point clouds into feature volumes. Here $\mathcal{S}(\cdot)$ consists of a series of $3\times3\times3$ 3D sparse convolution kernels, which downsample the spatial features to $1\times$, $2\times$, $4\times$, and eventually an $8\times$ downsampled tensor. The 3D features in last layer are compressed into BEV features along height dimension for object proposal generation.

C. Region Proposal Network

We follow the recent works [10], [19] that generate object proposals by applying a series of 2D convolution on BEV feature maps and generate the object proposals from the BEV maps. Specifically, we first predefined N_p object templates called anchors on the last layer of BEV maps. We generate object proposals by classifying the anchors, and regressing the residuals of object size, location and orientation angle relative to ground truth boxes. Similar to [10], [19], we assign the ground truth bounding boxes to anchors by a IoU-based matching. For ith anchor, we denote the score predict, score target, residual predict and residual target as α_i, $\hat{\alpha}_i$, δ_i and $\hat{\delta}_i^j$ respectively. The loss of proposal network is defined as

$$\begin{aligned}\mathcal{L}_{RPN} =& \frac{1}{N_p}[\sum_i \mathcal{L}_{score}(\alpha_i, \hat{\alpha}_i)+ \\ & \mathcal{I}(IoU_i > u)\sum_i \mathcal{L}_{reg}(\delta_i, \hat{\delta}_i)],\end{aligned} \tag{7}$$

where $\mathcal{I}(IoU_i > u)$ indicates that only object proposals with $IoU_i > u$ produce the regression loss, $\mathcal{L}_{reg}$ and $\mathcal{L}_{score}$ are smooth L1 and binary cross entropy loss, respectively.

D. Overall Training Loss

Our CasA can be trained end-to-end by an RPN loss L_{RPN} and a CRN loss L_{CRN}. We combine the two losses with equal weights as $L = L_{RPN} + L_{CRN}$. The RPN loss is defined in previous section. The CRN loss is the summation of multiple refinement losses in multiple stages. In each refinement stage, we adopt the box regression loss $\mathcal{L}_{reg}$ and score loss $\mathcal{L}_{score}$ like [10], [19]. For the ith proposal at the j-th refinement stage, we denote the score predict, score target, residual predict and residual target as α_i^j, $\hat{\alpha}_i^j$, δ_i^j and $\hat{\delta}_i^j$ respectively. The loss of CRN is defined as

$$\begin{aligned}\mathcal{L}_{CRN} =& \frac{1}{N_b}[\sum_i \sum_j \mathcal{L}_{score}(\alpha_i^j, \hat{\alpha}_i^j)+ \\ & \mathcal{I}(IoU_i^j > u^j)\sum_i \sum_j \mathcal{L}_{reg}(\delta_i^j, \hat{\delta}_i^j)],\end{aligned} \tag{8}$$

where $\mathcal{I}(IoU_i^j > u^j)$ indicates that only object proposals with $IoU_i^j > u^j$ produce the regression loss.

IV. Experiments

A. Datasets and Evaluation Metrics

a) KITTI dataset.: The KITTI Dataset includes 7481 and 7518 LiDAR frames for training and testing, respectively. We follow recent works [10] that divide the training data into a train split of 3712 frames and a val split of 3769 frames. The primary official evaluation metric is a 3D Average Precision (AP) under 40 recall thresholds (R40). We also follow previous works [10], [19] and report the results on the validation set by using a 3D AP under 11 recall thresholds. The IoU thresholds in this metric are 0.7, 0.5, 0.5 for cars, pedestrians, and cyclists, respectively.

b) Waymo Open Dataset.: Waymo Open Dataset [36] contains 798 and 202 sequences with 158361 and 40077 LiDAR frames for training and validation respectively. The official 3D detection evaluation metrics are mean Average Precision (mAP) (L1), mAP (L2), mAPH (L1) and mAPH (L2), where L1 and L2 denote the detection difficulty level. The mAPH metric takes into account object heading accuracy. The IoU thresholds in this metric are 0.7, 0.5, 0.5 for vehicles, pedestrians, and cyclists, respectively.

B. Setup Details

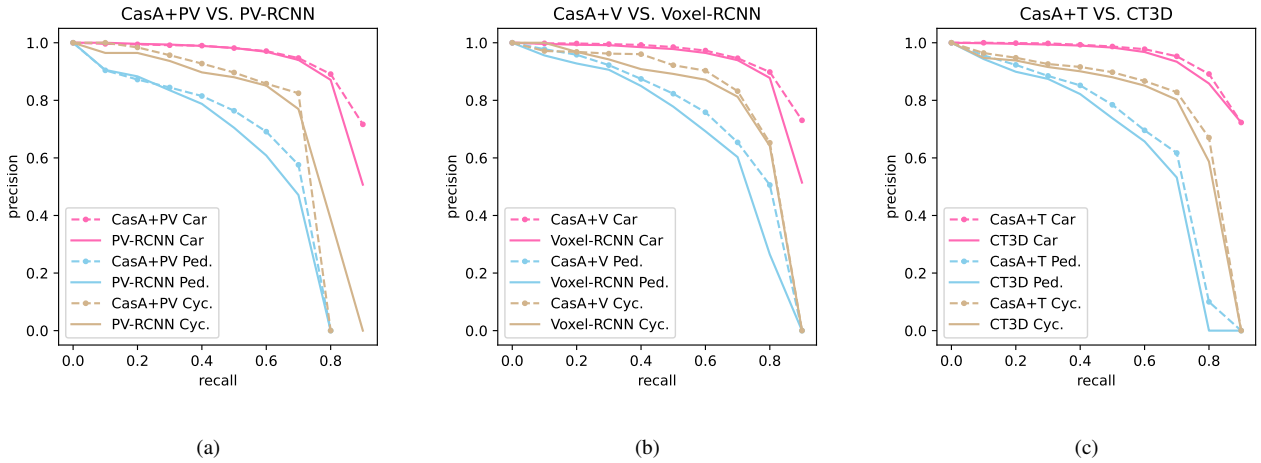
To demonstrate the universality and superiority of our CasA, we conducted our experiments on three popular baseline detectors: PV-RCNN [19], Voxel-RCNN [10] and CT3D [20]. We denoted the newly implemented detectors as CasA+PV, CasA+V and CasA+T, respectively. For each detector on each dataset, we train a *single model* for three classes. We adopted three refinement stages (the number of stages is discussed in Sec. 4.5). In training, we adopted IoU thresholds $u = 0.5, 0.55, 0.6$ for vehicles and $u = 0.4, 0.45, 0.5$ for pedestrians and cyclists. We adopted the feature channel of 256 in our Cascade Attention Module.

1) CasA+PV : Our CasA+PV is constructed from PV-RCNN [19]. Directly applied CasA to PV-RCNN is impractical, as the 2D backbone and Voxel Set Abstraction in RoI pooling is computationally intensive. It needs more GPU memory than we have. To tackle this, we simplified the 2D backbone and RoI pooling in our CasA+PV. (1) The Keypoint Weighting branch in the PV-RCNN is removed. (2) We only kept the 3D features from the last two layers and BEV features for RoI-grid pooling. (3) We set the feature dimension to half of the original feature dimension in the 2D backbone.

2) CasA+V : Our CasA+PV is constructed from Voxel-RCNN [10]. We directly applied CasA to Voxel-RCNN as detailed in the method section.

TABLE II
3D DETECTION RESULTS ON THE KITTI VALIDATION SET BY USING PV-RCNN, VOXEL-RCNN AND CT3D AS BASE DETECTORS. THE PED. AND CYC. DETECTION RESULTS OF VOXEL-RCNN ARE REPRODUCED BY THEIR OPEN-SOURCE CODE. R40 AND R11 DENOTE AVERAGE PRECISION (AP) UNDER 40 AND 11 RECALL THRESHOLDS, RESPECTIVELY. THE IMPROVED RESULTS ARE IN BOLD.

Methods	Car 3D (R40)(%)			Ped. 3D (R40)(%)			Cyc. 3D (R40)(%)			Car 3D (R11)(%)		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PV-RCNN [19]	92.57	84.83	82.69	64.26	56.67	51.91	88.88	71.95	66.78	-	83.90	-
CasA+PV	**92.73**	**85.89**	**83.57**	**68.90**	**59.86**	**53.66**	**90.95**	70.69	65.82	89.46	**86.17**	79.10
Voxel-RCNN [10]	92.38	85.29	82.86	68.22	60.97	55.63	91.28	72.54	68.46	89.41	84.52	78.93
CasA+V	**93.21**	**86.37**	**83.93**	**73.95**	**66.62**	**59.97**	**92.78**	**73.94**	**69.37**	**89.88**	**86.58**	**79.38**
CT3D [20]	92.85	85.82	83.46	65.73	58.56	53.04	91.99	71.60	67.34	89.54	86.06	78.99
CasA+T	**93.38**	**86.42**	**84.04**	**68.81**	**62.59**	**57.47**	**92.81**	**72.63**	**68.32**	**90.11**	**86.63**	**79.49**



(a) (b) (c)

Fig. 3. The recall-precision curves of our CasA on the KITTI validation set (moderate split). (a) The comparison between our CasA+PV and PV-RCNN. (b) The comparison between our CasA+V and Voxel-RCNN. (c) The comparison between our CasA+T and CT3D.

3) CasA+T : Our CasA+T is constructed from CT3D [20]. Directly applied CasA to CT3D is also computationally intensive. We made some simplifications. (1) We set the feature dimension to half of the original feature dimension in the 2D backbone. (2) We set the number of transformer encoding layers from three to two.

4) Training and Testing Details on KITTI: All of the detectors use the same standard detection range ([0, 70.4]m for the X axis, [-40, 40]m for the Y axis and [-3, 1]m for the Z axis). All the detectors are trained on two 3090 GPU cards with a batch size of four. We kept the training setup as same as baseline detectors. We used the Adam optimizer with a learning rate of 0.01 for CasA+PV and CasA+V, and 0.001 for CasA+T. Both the CasA+PV and the CasA+V are trained for 80 epochs, while the CasA+T is trained for 100 epochs (following CT3D [20]). During training, we set NMS IoU 0.8 to keep 128 proposals from RPN with 1:1 ratio of positive and negative samples. Regular data augmentation methods, such as flipping, rotation, scaling and ground-truth sampling are also adopted. During testing, we kept the top 100 proposals for cascade refinement, and finally use NMS IoU 0.1 to produce final detection results.

5) Training and Testing Details on Waymo: All the detectors use the same standard detection range of [-75.2, 75.2]m for the X and Y axes and [-2, 4]m for the Z axis. The detectors are trained on eight V100 GPU cards with a batch size of 16 for 30 epochs. Adam optimizer with a learning rate of 0.01 is adopted for CasA+PV and CasA+V, and 0.001 for CasA+T. During training, we set NMS IoU to be 0.8 to keep 128 proposals. Regular data augmentations are adopted (see the above paragraph). During testing, we kept the top 500 proposals, and use NMS IoU 0.3 to produce the final detection results.

C. Evaluation on the KITTI Dataset

a) Validation set.: We first conducted experiments on the KITTI validation set. The results are shown in Table II. Our CasA outperforms the baseline Voxel-RCNN, PV-RCNN and CT3D with 2.27%, 2.06% and 0.57% AP (R11) respectively in the moderate car class. Notably, our CasA improves the pedestrian detection performance with 3.19%, 5.65% and 4.03%, respectively. The performance gains are mostly derived from the cascade attention design that aggregates proposal features from multiple stages, leading to a more effective and comprehensive object refinement. Note that the cyclist detection performance of CasA+PV slightly drops, mostly due to that we simplified the 2D backbone of the baseline detector.

b) Test set.: To further demonstrate the advance of our cascade attention design, we trained our CasA+V using all training data of the KITTI training set. The results on the KITTI test set are summarized in Table III. Our CasA+V outperforms all published methods and achieves state-of-the-art performance. We note that our CasA+V outperforms all previous methods in all three classes (Car, Pedestrian and

TABLE III
3D DETECTION RESULTS ON THE KITTI TEST SET, THE BEST METHODS ARE IN BOLD. ONLY PUBLISHED METHODS ARE REPORTED. OUR CASA+V OUTPERFORMS ALL METHODS ON ALL OF THREE CLASSES (CAR, PEDESTRIAN AND CYCLIST).

Method	Car 3D (R40)(%)			Ped. 3D (R40)(%)			Cyc. 3D (R40)(%)		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
LiDAR+RGB									
MV3D [8]	74.97	63.63	54.00	-	-	-	-	-	-
F-PointNet [37]	82.19	69.79	60.59	50.53	42.15	38.08	72.27	56.12	49.01
F-ConvNet [37]	87.36	76.39	66.69	52.16	43.38	38.8	81.98	65.07	56.54
UberATG-MMF [38]	88.40	77.43	70.22	-	-	-	-	-	-
EPNet [39]	89.81	79.28	74.59	52.79	44.38	41.29	-	-	-
3D-CVF [40]	89.20	80.05	73.11	-	-	-	-	-	-
CLOCs [41]	88.94	80.67	77.15	-	-	-	-	-	-
LiDAR									
PointRCNN [9]	86.96	75.64	70.70	47.98	39.37	36.01	74.96	58.82	52.53
3D Iou Loss [42]	86.16	75.64	70.70	-	-	-	-	-	-
STD [43]	87.95	79.71	75.09	53.29	42.47	38.35	78.69	61.59	55.30
HotSpotNet [44]	87.60	78.31	73.34	53.10	45.37	41.47	82.59	65.95	59.00
Part A^2 [23]	87.81	78.49	73.51	53.10	43.35	40.06	79.17	63.52	56.93
Point-GNN [45]	88.33	79.47	72.29	51.92	43.77	40.14	78.60	63.48	57.08
3DSSD [5]	88.36	79.57	74.55	50.64	43.09	39.65	82.48	64.10	56.90
SA-SSD [15]	88.75	79.79	74.16	-	-	-	-	-	-
PV-RCNN [19]	90.25	81.43	76.82	52.17	43.29	40.29	78.60	63.71	57.65
CIA-SSD [46]	89.59	80.28	72.87	-	-	-	-	-	-
CT3D [20]	87.83	81.77	77.16	-	-	-	-	-	-
VoTr [24]	89.90	82.09	79.14	-	-	-	-	-	-
Pyramid-PV [47]	88.39	82.08	77.49	-	-	-	-	-	-
SPG [48]	90.50	82.13	78.90	-	-	-	-	-	-
SE-SSD [21]	91.49	82.54	77.15	-	-	-	-	-	-
BtcDet [49]	90.64	82.86	78.09	47.80	41.63	39.30	82.81	68.68	61.81
Voxel-RCNN [10]	90.90	81.62	77.06	-	-	-	-	-	-
CasA+V	**91.58**	**83.06**	**80.08**	**54.04**	**47.09**	**44.56**	**87.91**	**73.47**	**66.17**

TABLE IV
3D DETECTION RESULTS ON THE WAYMO VALIDATION SET BY USING PV-RCNN, VOXEL-RCNN, AND CT3D AS BASE DETECTORS. † : RE-IMPLEMENTED RESULTS OF PV-RCNN REPORTED IN [50]. ‡: RE-IMPLEMENTED BY OURSELVES USING THEIR CODES. THE IMPROVED RESULTS ARE IN BOLD.

Methods	Veh.(L1)(%)		Veh.(L2)(%)		Ped.(L1)(%)		Ped.(L2)(%)		Cyc.(L1)(%)		Cyc.(L2)(%)	
	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
†PV-RCNN [19]	77.51	76.89	68.98	68.41	75.01	65.65	66.04	57.61	67.81	66.35	65.39	63.98
CasA+PV	**78.55**	**78.06**	**69.67**	**69.23**	**77.22**	**70.60**	**68.06**	**62.08**	**68.19**	**66.76**	**65.73**	**64.33**
‡Voxel-RCNN [10]	77.43	76.71	68.73	68.24	76.37	68.21	67.92	60.40	68.74	67.56	66.46	65.35
CasA+V	**78.54**	**78.00**	**69.91**	**69.42**	**80.88**	**73.10**	**71.87**	**64.78**	**69.66**	**68.38**	**67.07**	**66.83**
‡CT3D [20]	77.75	77.22	68.92	68.44	75.38	68.08	66.17	59.60	69.28	67.88	66.84	65.48
CasA+T	**78.26**	**77.72**	**69.57**	**69.12**	**76.83**	**70.51**	**68.36**	**62.51**	**71.40**	**70.04**	**68.86**	**67.55**

Cyclist). We show some qualitative results of CasA+V on the KITTI test set in Fig. 5.

D. Evaluation on the Waymo Open Dataset

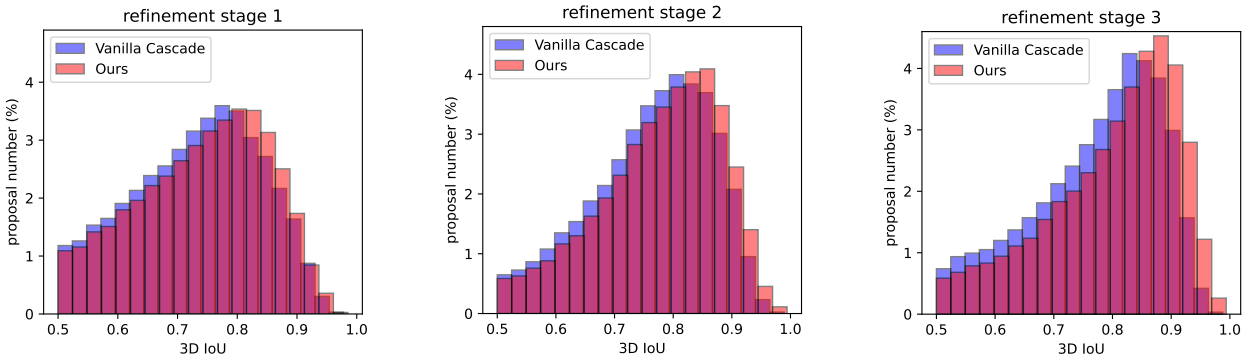
The results on the Waymo validation set are shown in Table IV. Our CasA improved all baseline detectors on all metrics by a large margin. Specifically, compared with the PV-RCNN [19], Voxel-RCNN [10] and CT3D [20], our CasA+PV, CasA+V and CasA+T improve the vehicle detection mAPH with 0.82%, 1.18% and 0.68%, respectively. Our cascade design significantly improves the pedestrian detection performance with 4.47%, 4.38% and 2.91% mAPH, respectively. The results further demonstrate the effectiveness of our method.

The results on the Waymo validation set are shown in Table IV. Our CasA improved all baseline detectors on all metrics by a large margin. Specifically, compared with the PV-RCNN [19], Voxel-RCNN [10] and CT3D [20], our CasA+PV, CasA+V and CasA+T improve the vehicle detection mAPH with 0.82%, 1.18% and 0.68%, respectively. Our cascade design significantly improves the pedestrian detection performance with 4.47%, 4.38% and 2.91% mAPH, respectively. The results further demonstrate the effectiveness of our method.

E. Ablation Study

We conducted experiments on the KITTI validation set (moderate car class), and used CasA+V to examine the hyper-parameters and each component/design of the proposed method.

1) Cascade Stages: We first tested the cascade stages. The results are shown in Table V. CasA+V achieves the best performance on moderate and hard car class by using three stages while achieving the best performance on easy car class by using four stages. We observe that the detection performance of the three and four stages is close to each other. For computational efficiency, we adopt three stages. It can also



(a) (b) (c)

Fig. 4. Proposal quality in refine stage 1 (a), refine stage 2 (b), refine stage 3 (c). With the increase of stages, the 3D IoUs between the proposals and ground truths become closer to 1. Our cascade attention design also shows better IoUs compared with the vanilla cascade structure. It demonstrates the effectiveness of our design.

achieve real-time performance with a running speed of 86ms (single 3090 GPU).

TABLE V
ABLATION RESULTS ON THE KITTI VALIDATION SET BY USING DIFFERENT CASCADE STAGES IN THE CASA+V.

Cascade stages	Car 3D (R40)(%)			Run time (ms)
	Easy	Moderate	Hard	
1	92.84	85.49	83.04	64
2	93.06	85.96	83.58	77
3	93.21	**86.37**	**83.93**	86
4	**93.30**	86.27	83.86	101
5	93.18	86.24	83.87	114

TABLE VI
ABLATION RESULTS ON KITTI VALIDATION SET USING DIFFERENT FEATURE AGGREGATION MODULES IN THE CASA+V.

Aggregation Module	Car 3D (R40)(%)		
	Easy	Moderate	Hard
Concatenation	92.79	85.32	83.17
GRU	92.93	86.01	83.64
Self-attention	93.15	86.22	83.72
Cross-attention	93.01	86.29	83.85
Self-attention & Cross-attention	**93.21**	**86.37**	**83.93**

2) Number of Attention Heads: In our cascade attention module, we use multiple attention heads. To find the best head number, we conducted an ablation experiment. The results are shown in Table VII. By using four heads, our CasA+V achieves the best performance. Therefore, we adopt four heads in this paper.

3) Effectiveness of Cascade Attention Module: To investigate the effects of the cascade attention module, we first constructed a baseline of vanilla cascade structure, which uses the Voxel-RCNN [10] as base detector and integrates a simple cascade structure with raising IoU thresholds. The comparison results are shown in Table VIII. A simple cascade structure can not bring performance improvement. By adding our proposed Cascade Attention Module, the result is further improved to 85.72%. Our Cascade Attention Module aggregates object features from different detection stages, producing more robust detections from scenes with sparse points. Especially for the distant objects (more than 40m), our module improves the baseline by around 5% AP (See Fig. 6). We also tested alternative aggregation methods such as concatenation, GRU, only self-attention, and only cross-attention. The results are shown in Table VI. The module using both self-attention and cross-attention achieves the best performance on easy (93.21%), moderate (86.37%), and hard (83.93%) car classes among the above aggregation modules. Therefore, our method uses the self-attention and cross-attention design as the aggregation strategy.

TABLE VII
ABLATION RESULTS ON THE KITTI VALIDATION SET BY USING DIFFERENT NUMBER OF ATTENTION HEADS IN THE CASA+V.

Number of attention heads	Car 3D (R40)(%)		
	Easy	Moderate	Hard
1	92.89	86.11	81.77
4	**93.21**	**86.37**	**83.93**
8	93.07	86.27	83.74
16	93.04	86.17	83.64

TABLE VIII
ABLATION STUDY RESULTS OF PART-AIDED SCORING, CASCADE ATTENTION MODULE AND BOXES VOTING ON THE KITTI VALIDATION SET (MODERATE CAR CLASS).

Structure	Car 3D (R40)(%)		
	Stage 1	Stage 2	Stage 3
Voxel-RCNN	85.27	-	-
+Vanilla Cascade	82.96	84.88	84.82
+Cascade Attention	85.48	85.73	85.72
+Part-aided Scoring	85.67	85.89	85.85
+Boxes Voting	85.67	86.24	**86.37**

4) Effectiveness of Part-aided Scoring: We also conducted an ablation study on the part-aided scoring, the results are also shown in Table VIII. By adding this design, our method achieves the results of 85.85 %, which indicates the effectiveness of the part-aided scoring design.

5) Effectiveness of Boxes Voting: As shown in Table VIII, the boxes voting further improved the 3D detection perfor-

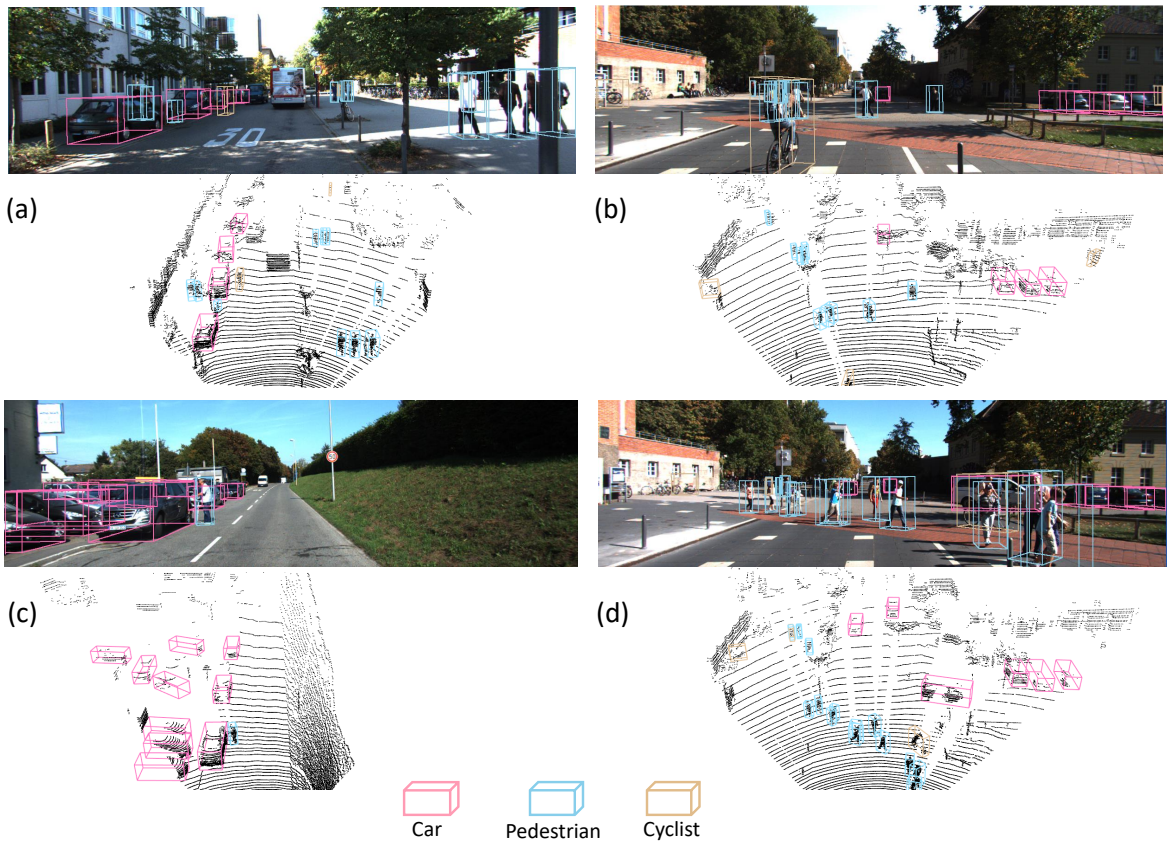


Fig. 5. Example qualitative results of CasA+V on the KITTI test set. We show our detected results of four different scenes in (a)(b)(c)(d), in which the Car, Pedestrian and Cyclist are in hotpink, skyblue and tan, respectively.

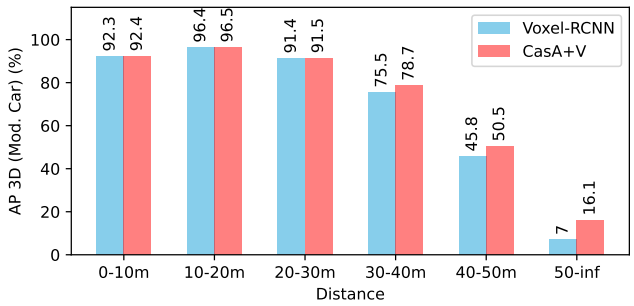


Fig. 6. Ablation results on the KITTI validation set for different distances. Compared with the Voxel-RCNN baseline, our CasA+V has a better detection performance of distant objects.

mance of moderate car class from 85.85% to 86.37%. The reason is that our method can now progressively boost and complement weak predictions from each cascade stage, and thus, generate more accurate predictions.

F. Proposal Quality in Cascade Attention Design

In Cascade RCNN [12], the proposal quality becomes better and better with the increase of stages. This leads to a better performance in 2D space. In 3D space, here, we also demonstrate the same property of CasA. With the increase of stages, the 3D IoUs between the proposals and ground truths become closer to 1. Besides, our design obtains higher IoUs than the vanilla cascade structure. It demonstrates the effectiveness of our cascade attention design. We show the results in Fig. 4.

TABLE IX
RESULTS ON THE ONCE VALIDATION SET.

Method (multi-class model)	AP 3D (%)		
	Vehicle	Pedestrian	Cyclist
Voxel-RCNN [10]	77.12	41.90	62.31
CasA+V	**78.34**	**50.34**	**70.01**

G. Results on dataset with sparser LiDAR scanning

The Waymo and KITTI datasets are collected by 64-beam LiDAR sensor, and the acquired points are relatively dense. To validate our method on various resolutions, we also conducted an experiment on the ONCE dataset [51], which is collected by a 40-beam LiDAR sensor. The points of ONCE are also sparser than Waymo and KITTI datasets. We used the metrics, the official training and validation data split provided by ONCE. The results of the ONCE validation set are shown in Table IX. Compared with the Voxel-RCNN baseline, our CasA+V has 1.2%, 8.4%, and 7.7% AP improvement on Vehicle, Pedestrian, and Cyclist, respectively. We can observe from the results that our cascade attention design performs better than the Voxel-RCNN baseline on the low-resolution dataset. These results further demonstrate the effectiveness of our method.

V. Conclusion

This paper presented a multi-stage 3D object detector, CasA, that progressively refines region proposals by a cascade attention structure. CasA addressed the problems of ignoring distant objects and error propagation in the multi-stage 3D object detection by aggregating object features from multiple stages. CasA can significantly improve the performance of various state-of-the-art 3D object detectors. We verified this design on two datasets with three different popular detectors. On the widely used KITTI validation set (moderate car class), CasA improves PV-RCNN, Voxel-RCNN and CT3D with 2.27%, 2.06% and 0.57% AP(R11), respectively. This fully demonstrated the effectiveness and generality of our method. We believe this design can be of interest to many 3D downstream tasks such as object tracking and motion prediction. With a cascade attention design, our CasA method achieves promising object detection performance on multiple datasets. Compared with the previous approaches, the performance is significantly better in the case of three stage refinement. However, there is a trade-off between computational cost and performance improvement. Future work will focus on developing approaches that can refine proposals more efficiently. Besides, it is more difficult to detect the Pedestrians than other classes (see Table III, IV and IX) due to its small size and hardly distinguishable orientation. Future work will also focus on detecting small objects more accurately.

References

[1] L. Wang, X. Fan, J. Chen, J. Cheng, J. Tan, and X. Ma, "3d object detection based on sparse convolution neural network and feature fusion for autonomous driving in smart cities," *Sustainable Cities and Society*, vol. 54, p. 102002, 2020.

[2] Y. Wang, Q. Chen, Q. Zhu, L. Liu, C. Li, and D. Zheng, "A survey of mobile laser scanning applications and key techniques over urban areas," *Remote. Sens.*, vol. 11, p. 1540, 2019.

[3] C. Yu, J. Lei, B. Peng, H. Shen, and Q. Huang, "Siev-net: A structure-information enhanced voxel network for 3d object detection from lidar point clouds," *IEEE Trans. Geosci. Remote Sens.*, 2022.

[4] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[5] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3D single stage object detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 11 040–11 048.

[6] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 12 697–12 705.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2016.

[8] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1907–1915.

[9] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 770–779.

[10] J. Deng, S. Shi, P. Li, W. gang Zhou, Y. Zhang, and H. Li, "Voxel R-CNN: Towards high performance voxel-based 3D object detection," in *AAAI Conf. Artif. Intell.*, 2021.

[11] Z. Liu, J. Du, F. Tian, and J. Wen, "MR-CNN: A multi-scale region-based convolutional neural network for small traffic sign recognition," *IEEE Access*, vol. 7, pp. 57 120–57 128, 2019.

[12] C. Zhaowei and V. N., "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 6154–6162.

[13] Z. Cai and N. Vasconcelos, "Cascade R-CNN: high quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1483–1498, 2019.

[14] Q. Meng, W. Wang, T. Zhou, J. Shen, Y. Jia, and L. V. Gool, "Towards a weakly supervised framework for 3D point cloud object detection and annotation." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, 2021.

[15] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure aware single-stage 3D object detection from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 11 873–11 882.

[16] X. Wu, L. Peng, H. Yang, L. Xie, C. Huang, C. Deng, H. Liu, and D. Cai, "Sparse fuse dense: Towards high quality 3d detection with depth completion," in *CVPR*, 2022.

[17] H. Zhu, J. Deng, Y. Zhang, J. Ji, Q.-C. Mao, H. Li, and Y. Zhang, "VPFNet: Improving 3D object detection with virtual point based lidar and stereo data fusion," *ArXiv*, vol. abs/2111.14382, 2021.

[18] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 3354–3361.

[19] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 10 526 – 10 535.

[20] H. Sheng, S. Cai, Y. Liu, B. Deng, J. Huang, X. Hua, and M.-J. Zhao, "Improving 3D object detection with channel-wise transformer," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 2743–2752.

[21] W. Zheng, W. Tang, L. Jiang, and C.-W. Fu, "SE-SSD: Self-ensembling single-stage object detector from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 14 494–14 503.

[22] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. Garcia, and A. De La Escalera, "Birdnet: a 3D object detection framework from lidar information," in *Proc. Int. Conf. on Int. Trans. Sys.*, 2018, pp. 3517–3523.

[23] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, pp. 2647–2664, 2021.

[24] J. Mao, Y. Xue, M. Niu, H. Bai, J. Feng, X. Liang, H. Xu, and C. Xu, "Voxel transformer for 3D object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 3164–3173.

[25] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware u model," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1134 – 1142.

[26] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. H. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár, "A multipath network for object detection," *ArXiv*, vol. abs/1604.02135, 2016.

[27] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 784–799.

[28] J. Gong, Z. Zhao, and N. Li, "Improving multi-stage object detection via iterative proposal refinement." in *The British Machine Vision Conference*, 2019, p. 223.

[29] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ArXiv*, vol. abs/2010.11929, 2021.

[30] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[31] M. Zhu, L. Jiao, F. Liu, S. Yang, and J. Wang, "Residual spectral-spatial attention network for hyperspectral image classification," *IEEE Trans. Geosci. Remote. Sens.*, vol. 59, pp. 449–462, 2021.

[32] J. Zhang, J. Wu, H. Wang, Y. Wang, and Y. song Li, "Cloud detection method using cnn based on cascaded feature attention and channel attention," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022.

[33] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 16 259–16 268.

[34] H. Fan, Y. Yang, and M. Kankanhalli, "Point 4D transformer networks for spatio-temporal modeling in point cloud videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 14 204–14 213.

[35] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *ArXiv*, vol. abs/1706.03762, 2017.

[36] P. Sun, H. Kretzschmar, X. Dotiwalla, and a. et, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 2443 – 2451.

[37] C. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3D object detection from rgb-d data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 918–927.

[38] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 7337–7345.
[39] T. Huang, Z. Liu, X. Chen, and X. Bai, "EPNet: Enhancing point features with image semantics for 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 35–52.
[40] J. H. Yoo, Y. Kim, J. S. Kim, and J. W. Choi, "3D-CVF: Generating joint camera and lidar features using cross-view spatial feature fusion for 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 720–736.
[41] S. Pang, D. D. Morris, and H. Radha, "CLOCs: Camera-lidar object candidates fusion for 3D object detection," *Proc. IEEE Int. Conf. Intell. Rob. Syst.*, pp. 10 386–10 393, 2020.
[42] D. Zhou, J. Fang, X. Song, C. Guan, J. Yin, Y. Dai, and R. Yang, "Iou loss for 2d/3D object detection," *International Conference on 3D Vision*, pp. 85–94, 2019.
[43] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "STD: Sparse-to-dense 3D object detector for point cloud," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1951–1960.
[44] Q. Chen, L. Sun, Z. Wang, K. Jia, and A. L. Yuille, "Object as Hotspots: An anchor-free 3D object detection approach via firing of hotspots," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 68–84.
[45] W. Shi and R. R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 1708–1716.
[46] W. Zheng, W. Tang, S. Chen, L. Jiang, and C.-W. Fu, "CIA-SSD: Confident iou-aware single-stage object detector from point cloud," *ArXiv*, vol. abs/2012.03015, 2020.
[47] J. Mao, M. Niu, H. Bai, X. Liang, H. Xu, and C. Xu, "Pyramid R-CNN: Towards better performance and adaptability for 3D object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 2723–2732.
[48] Q. Xu, Y. Zhou, W. Wang, C. Qi, and D. Anguelov, "SPG: Unsupervised domain adaptation for 3D object detection via semantic point generation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 15 446–15 456.
[49] Q. Xu, Y. Zhong, and U. Neumann, "Behind the Curtain: Learning occluded shapes for 3D object detection," *ArXiv*, vol. abs/2112.02205, 2021.
[50] S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang, and H. Li, "PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3D object detection," *ArXiv*, vol. abs/2102.00463, 2021.
[51] J. Mao, M. Niu, C. Jiang, H. Liang, X. Liang, Y. Li, C. Ye, W. Zhang, Z. Li, J. Yu, H. Xu, and C. Xu, "One million scenes for autonomous driving: ONCE dataset," *ArXiv*, vol. abs/2106.11037, 2021.