AGO-Net: Association-Guided 3D Point Cloud Object Detection Network

Liang Du, Xiaoqing Ye, Xiao Tan, Edward Johns, Bo Chen, Errui Ding, Xiangyang Xue, Jianfeng Feng

Abstract—The human brain can effortlessly recognize and localize objects, whereas current 3D object detection methods based on LiDAR point clouds still report inferior performance for detecting occluded and distant objects: the point cloud appearance varies greatly due to occlusion, and has inherent variance in point densities along the distance to sensors. Therefore, designing feature representations robust to such point clouds is critical. Inspired by human associative recognition, we propose a novel 3D detection framework that associates intact features for objects via domain adaptation. We bridge the gap between the perceptual domain, where features are derived from real scenes with sub-optimal representations, and the conceptual domain, where features are extracted from augmented scenes that consist of non-occlusion objects with rich detailed information. A feasible method is investigated to construct conceptual scenes without external datasets. We further introduce an attention-based re-weighting module that adaptively strengthens the feature adaptation of more informative regions. The network's feature enhancement ability is exploited without introducing extra cost during inference, which is plug-and-play in various 3D detection frameworks. We achieve new state-of-the-art performance on the KITTI 3D detection benchmark in both accuracy and speed. Experiments on nuScenes and Waymo datasets also validate the versatility of our method.

Index Terms—3D object detection, domain adaptation, associative recognition, point cloud, neural network, autonomous driving

1 Introduction

Object detection [1], [2], [3], [4] is drawing increasing attention from both academia and industry due to its critical role in robotics and autonomous driving [5]. Despite the recent success of image-based 2D object detection [6], [7], 3D object detection from LiDAR-based point clouds remains an under-developed research area, and a highly challenging problem due to object occlusion and variance in point distributions. The density of a point cloud decreases as the distance to a LiDAR camera increases, which causes an apparent density discrepancy. Moreover, some parts of objects are invisible due to occlusion or low point cloud occupancy. In this situation, 3D detection is unreliable and error-prone because the network fails to extract sufficient features.

Recently, many cutting-edge 3D object detection approaches [8], [9] have attempted to solve this problem. In STD [8], PointsPool is introduced to convert sparse intermediate point features to a more compact voxel representation. Nevertheless, for far-range objects, the voxelization operation does not facilitate the network in generating sufficient and robust features because most voxels are empty. A generative adversarial network (GAN) is leveraged by [9] to force the network to generate consistent features between far-range and near-range objects. However, this method aims at range domain adaptation and ignores object-wise constraints, such as shapes and the viewing perspective of the objects. PV-RCNN [10] adopts both voxel-based features and point-based features of point clouds and obtains more discriminative representations at the cost of feature extraction.

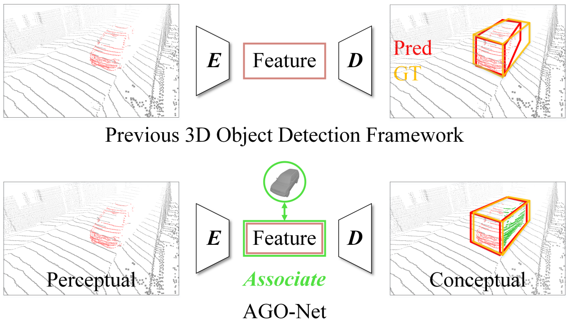


Fig. 1. Illustration of the standard 3D object detection framework and the proposed associative-recognition-inspired AGO-Net. The encoder and the decoder are denoted by E and D, respectively. Compared with previous approaches that solely utilized the encoded features of real-world point clouds for 3D object detection, our AGO-Net builds up the association between incomplete perceptual features of real-world objects and more complete features of corresponding class-wise conceptual models via domain adaptation. AGO-Net mimics the conceptual association of the human brain when perceiving objects, and fundamentally exploits the network capability of the feature enhancement.

Previous works like [11], [12] have proven that exploiting the biological and psychological plausibility of the human brain to build up intelligent systems brings significant performance gains to various tasks. For the human psychological model, perceiving 3D objects is carried out in the form of a conceptual association

- *L. Du and J. Feng are with the Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University, Shanghai, China, Key Laboratory of Computational Neuroscience and Brain-Inspired Intelligence (Fudan University), Ministry of Education, China, MOE Frontiers Center for Brain Science, Fudan University, Shanghai, China, Zhangjiang Fudan International Innovation Center. J. Feng is also with the Fudan ISTBI-ZJNU Algorithm Centre for Brain-inspired Intelligence, Zhejiang Normal University, Jinhua, China.*
- *X. Ye, X. Tan, and E. Ding are with Baidu Inc., China.*
- *E. Johns is with the Robot Learning Lab, Imperial College London, UK.*
- *B. Chen is with FAW (Nanjing) Technology Development Co., Ltd, China.*
- *X. Xue is with the School of Computer Science, Fudan University, China.*
- *Email: duliang@mail.ustc.edu.cn, jffeng@fudan.edu.cn*

process, which includes two vital stages [13]: (I) a "viewer-centered" feature representation stage, where the features of perceptual objects are presented from the viewer's perspective, and the visible features might lack structure and detail information due to occlusion and large distance, and (II) an "object-centered" stage, where the object's features are enhanced by the conceptual model of the same category retained in the memory. This association process helps map from an object to a view-invariant 3D conceptual model, which is termed as associative recognition.

Considering the previously mentioned issues, we propose a brain-inspired 3D object detection framework that associates weak perceptual features of occluded and distant objects with complete conceptual models' features that are more favorable for detection. As depicted in Figure 1, we select structurally-complete 3D models as conceptual guidance and adapt object-wise features from the perceptual domain to the conceptual domain during training following a transfer learning paradigm. Comparing with the conference version [14], there are four main differences. (1) For network architecture, instead of directly performing domain adaptation on the encoded feature, we decouple the features and only adapt the regression feature according to the dynamic attention map generated by the classification feature (SC-reweight), which achieves superior performance. (2) For the conceptual domain construction, an object completion method is proposed to maintain the original structure of each instance. (3) For the performance and experiments, our AGO-Net ranked 1^{st} among all existing one-stage detectors in the KITTI test benchmark at the time of submission. In addition to the KITTI experiments mainly on the "Car" category in [14], we further perform multi-class experiments on more challenging datasets such as nuScenes and Waymo. (4) For portability, we have also conducted experiments integrating our method into other state-of-the-art (SoTA) 3D detection frameworks, which validates its generalizability.

The main contributions are summarized as follows:

- We propose a 3D detection framework that associates real-world perceptual features with more complete features from conceptual models, which strengthens the feature robustness, especially for occluded and distant objects.
- We investigate a novel attention-based adaptation module that adapts more critical features from the real-world perceptual domain to the constructed conceptual domain.
- We present a feasible approach to generate the proposed conceptual scenes without requiring external datasets.
- Our method achieves new SoTA 3D object detection performance on the KITTI dataset in both accuracy and speed (25 FPS). More evaluations on the nuScenes and Waymo datasets further validate the effectiveness.

2 Related work

3D point cloud object detection. On the one hand, recent 3D object detection approaches usually include four types: multi-view-based, voxel-based, point-based, and graph-based methods [15]. On the other hand, if divided by whether proposals and post-refinement are required, 3D detection approaches can be divided into one-stage and two-stage methods.

The multi-view approaches such as [16], [17], [18], [19] project LiDAR point clouds to the image plane [20] or the bird's eye view (BEV) to encode features. Zhou et al. [21] proposed a multi-view fusion framework that utilizes the complementary information from BEV and perspective view.

The voxel-based methods utilize voxel representations for point cloud encoding. Vote3Deep [22] and 3DFCN [20] evenly discretize point clouds on grids and apply 3D convolutional networks to address point clouds for object detection. Zhou et al. proposed VoxelNet to learn discriminative features from sparse point clouds [48]. The following works [23], [24] improve VoxelNet by adopting 3D sparse convolutional layers to save computational resources. Lang et al. introduced PointPillars [25], which utilizes PointNets [27] to learn a representation of point clouds that are organized in vertical columns (pillars), which boosts the inference speed. Vora et al. introduced PointPainting [28], which projects predicted 2D pixel-wise semantic labels to point clouds as additional prior knowledge for the network. SA-SSD [30] employs an auxiliary network that converts the convolutional features back to point-level representations. The point-level supervision guides the convolutional features to be aware of the structure. Cheng et al. proposed PPBA [31], which learns to optimize data augmentation strategies by narrowing the search space and adopting the best parameters discovered in previous iterations.

The point-based method F-PointNet [32] is the first to exploit raw point clouds for 3D object detection. Frustum proposals generated by off-the-shelf 2D object detectors are utilized as candidate boxes, and final detection outputs are regressed based on interior points [33]. Shi et al. presented PointRCNN [34], which directly encodes the raw point cloud to predict 3D proposal regions in a bottom-up manner based on the segmentation labels from 3D bounding box annotations. 3DSSD [35] is the first one-stage point-based method that removes the refinement module in all existing point-based methods and introduces a fusion sampling strategy for retaining sufficient interior points of different objects for detection.

In summary, the voxel-based methods such as [9], [24], [30], that parse the 3D scene into a compact representation and adopt straight-through CNN networks, rather than a coarse-to-refine method with proposals and refinement, can be regarded as one-stage methods. One-stage approaches gain efficiency while losing some precision due to the progressively downscaled feature maps. The point-based methods like [8], [10], [34] exploit spatial features of the raw points based on the region proposal generated by the first stage, but they sacrifice inference speed for accuracy.

Deep neural network-based transfer learning. Transfer learning [36] is a machine learning paradigm that captures the notion of knowledge transfer between related task domains in computer vision [37], [38] and natural language processing (NLP) [39]. Traditional domain adaptation in transfer learning alleviates the influence of the distribution mismatch between the source domain and the target domain (e.g., training and testing data, synthetic and real-world data, respectively) so that the generalizability of the network can be boosted for the target domain data. Maximum Mean Discrepancy (MMD) introduced by [40] is a common domain distance metric. In addition to handcrafted distribution distance metrics, deep neural networks are leveraged to perform transfer learning. The learning-based methods can extract high-level representations that disentangle different explanatory factors of variations in the data [41]. Invariant factors that underlie different populations are also manifested, which transfer well from source domain tasks to target domain tasks [42]. [43] utilized feature distillation to bridge the domain gap between the simulated scene and real-world scene for semantic segmentation. For image classification, [44] presented Attention Transfer (AT) that squeezes intermediate feature maps into a single channel via an attention mask for feature distillation.

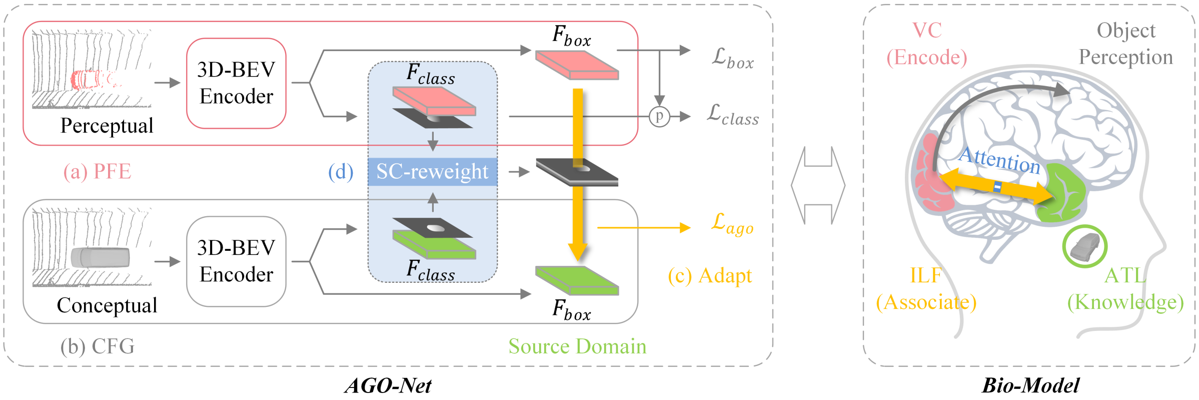


Fig. 2. Overview of the AGO-Net framework and its underlying biological model. AGO-Net consists of four parts: (a) The perceptual feature encoder (PFE), which encodes perceived information in a manner similar to that employed by the visual cortex (VC) in the human brain for object perception, as shown in the right figure. Various 3D detection methods can be utilized as alternatives to the 3D-BEV encoder, which encodes a 3D point cloud scene to a compact BEV representation. (b) The conceptual feature generator (CFG) generates the features based on the constructed conceptual scene, which provides conceptual association guidance for the PFE. This detachable auxiliary network is directly discarded during the inference time. (c) The feature adaptation from the real-world perceptual domain to the constructed conceptual domain mimics the knowledge association and retrieval process in the human brain. The large yellow arrow shows the adaptation direction. (d) The **S**patial and **C**hannel-wise attention-aware reweighting module (SC-reweight) aids the siamese network in adapting more crucial features based on the difference feature map of the conceptual and perceptual classification features. The loss terms of AGO-Net include the localization (XYZ), scale (WHL), classification, and rotation losses as well as our proposed association loss $\mathcal{L}_{ago}$. A detailed explanation of the Bio-model is provided in Sec.3.1.

3 Methodology

In this section, we introduce our association-guided 3D object detection network. As a result of the inevitable occlusion or large distance, many foreground objects contain only a few points, which makes it difficult for standard networks to perform classification and localization. Therefore, the perceptual domain corresponds to the real-world scene results that are not satisfactory due to the variance in the point cloud appearance. Alternatively, we can create a conceptual domain that learns from 3D conceptual models with more complete structures to guide feature learning in challenging cases. The conceptual features are discriminative enough for object classification and localization since they are extracted from point clouds with the finest details. If the domain gap between the perceptual and conceptual features is diminished, the network will have the capability to associate stronger features for objects adaptively.

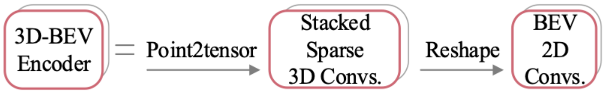


Fig. 3. One instantiated network of the PFE (red) / CFG (grey).

Figure 2 depicts the overall framework of AGO-Net, which includes four parts: (a) the perceptual feature encoder (PFE) to encode real-world target domain features for 3D object detection; (b) the conceptual feature generator (CFG) to generate the source domain features from corresponding scenes reconstructed by the conceptual models; (c) the perceptual-to-conceptual feature domain adaptation; and (d) the **S**patial and **C**hannel-wise loss reweighting module (SC-reweight) based on the classification features from both perceptual and conceptual scenes, which forces the network to strengthen the adaptation for regions that have much more critical information. Our approach to constructing conceptual scenes without involving external data is described in Sec.3.1. The network architecture and its underlying biological model that supports our motivation are also detailed in Sec.3.1. In Sec.3.2, we further explain the training process of AGO-Net.

3.1 Network of AGO-Net

Our AGO-Net can employ various 3D point cloud detection approaches, such as [24], [25], [30], etc., as its sub-networks PFE and CFG. We adopt a one-stage architecture based on sparse convolutional layers to instantiate one PFE and one CFG.

The PFE to encode perceptual features. The PFE serves as the standard feature encoder of the real scenes, which is shown in Figure 2 and colored in red. Following [30], our AGO-Net first obtains the input point features by discretizing the point coordinates to input tensor indices and then adopts 3D sparse convolution [23] for feature extraction. Specifically, for the point cloud quantization, we denote the coordinates of the i_{th} point in each scene as (C_x^i, C_y^i, C_z^i). To discretize the coordinates into integer indices for the input tensor, we define (S_x, S_y, S_z) as the quantization step along the X-, Y-, and Z-axes. Then, the index of each point is given by:

$$P(I_x^i, I_y^i, I_z^i) = P(\lfloor \frac{C_x^i}{S_x} \rfloor, \lfloor \frac{C_y^i}{S_y} \rfloor, \lfloor \frac{C_z^i}{S_z} \rfloor), \tag{1}$$

where $\lfloor \cdot \rfloor$ denotes the floor operation. Based on the obtained index, each point can be assigned to an entry of the network input tensor. If multiple points share the same index, then the entry is simply overwritten with the latest point.

As the red part depicted in Figure 2, the network architecture of the PFE can be formulated as follows:

$$\begin{cases} F_{class}^{\mathcal{P}}, F_{box}^{\mathcal{P}} = \xi^{\mathcal{P}}(I^{\mathcal{P}}) \\ O_{class}^{\mathcal{P}}, O_{box}^{\mathcal{P}} = \mathcal{H}_{class}^{\mathcal{P}}(F_{class}^{\mathcal{P}}), \mathcal{H}_{box}^{\mathcal{P}}(F_{box}^{\mathcal{P}}), \end{cases} \tag{2}$$

where $\xi^{\mathcal{P}}$ and $I^{\mathcal{P}}$ denote the 3D-BEV encoder and the perceptual input tensor, respectively. $F_{class}^{\mathcal{P}}$ and $F_{box}^{\mathcal{P}}$ are the output features of the 3D-BEV encoder for object classification and 3D bounding box regression. $\mathcal{H}_{class}^{\mathcal{P}}$ and $\mathcal{H}_{box}^{\mathcal{P}}$ represent sibling convolutions to decode the features for the output predictions $O_{class}^{\mathcal{P}}$ and $O_{box}^{\mathcal{P}}$,

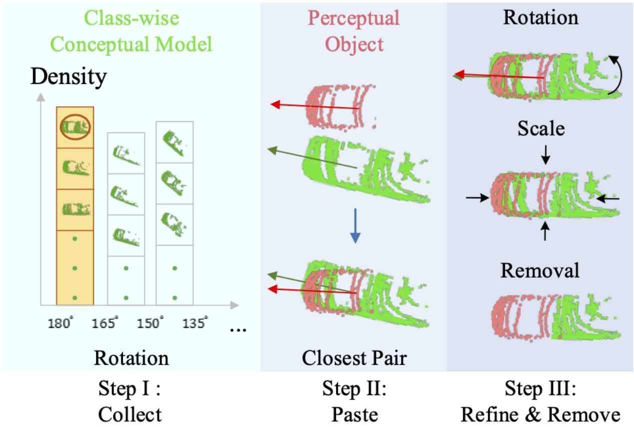


Fig. 4. The generation process of the conceptual model and scene.

respectively. Note that ξ can be instantiated by a variety of 3D object detectors.

As shown in Figure 3, the "Sparse 3D Convs." in the first red box encodes the point cloud feature into a high-dimensional feature. The feature is then reshaped and converted to a BEV representation via concatenating the feature along the height dimension of the scene into one channel. After that, "BEV 2D Convs." in the third box encodes the compacted features into $F^{\mathcal{P}}_{class}$ and $F^{\mathcal{P}}_{box}$. Finally, the network uses two heads H_{box} and H_{class} to decode two features for box regression and classification, respectively.

To alleviate the common issue of misalignment between the classification confidence maps and the corresponding predicted bounding boxes for 3D object detection, we apply the part-sensitive warping (PSWarp) method following [30] to align the classification confidences with the bounding boxes via a spatial transformation of the feature map, as shown in Figure 2 (the grey circled "p" at the end of the network). In terms of efficiency, we choose PSWarp instead of the alternative method PSRoIAlign [45], since [30] has proven that the performance of PSWarp is comparable to the performance of PSRoIAlign, while PSWarp takes only 1/10 the runtime of PSRoIAlign.

The CFG to generate conceptual features. With the same architecture of the PFE, the siamese network CFG generates the conceptual feature that served as the source domain feature. As the grey part depicted in Figure 2, the CFG is formulated as follows:

$$\begin{cases} F^{\mathcal{C}}_{class}, F^{\mathcal{C}}_{box} = \xi^{\mathcal{C}}(I^{\mathcal{C}}) \\ O^{\mathcal{C}}_{class}, O^{\mathcal{C}}_{box} = \mathcal{H}^{\mathcal{C}}_{class}(F^{\mathcal{C}}_{class}), \mathcal{H}^{\mathcal{C}}_{box}(F^{\mathcal{C}}_{box}). \end{cases} \tag{3}$$

Eq. 3 is similar to Eq. 2, except that subscript $\mathcal{C}$ in Eq. 3 denotes the conceptual domain, whereas subscript $\mathcal{P}$ in Eq. 2 represents the perceptual domain. We first train the CFG alone. The conceptual dataset is derived from the real-world dataset and is integrated with the conceptual models. Specifically, conceptual scenes denote object cloud instances that have complete appearance to some degree. Original point clouds with enough points and an intact structure are simply included. The conceptual model is a more complete point cloud for each object, such as a CAD model from external resources, or a surrogate model with more informative knowledge originating from the same dataset, i.e., self-contained. The format of the conceptual model is not specifically restrained.

After the CFG is well trained, the parameters of the CFG are frozen to provide stable feature guidance for further domain adaptation, and the decoding part of the CFG is removed, as shown in Figure 2. During the training of the entire AGO-Net, the perceptual scene and its corresponding conceptual scene are simultaneously fed into the PFE and CFG, respectively. The PFE is encouraged to generate more complete features based on the perceptual feature of the sparse and partial-visible point clouds under the guidance of the CFG. During the inference time, we simply remove the CFG from our framework.

Self-contained method to build conceptual scenes. The conceptual models that are utilized to construct the conceptual scenes can be fabricated via various methods, such as 3D CAD models and render-based techniques. In this paper, instead of appealing to costly external resources, we propose a self-constrained scene-constructing method to adopt surrogate models with more structural information that originated from the same dataset. Since 3D point cloud objects of the same category usually have a similar shape and scale, we collect the free-of-charge ground-truth objects with more complete structures as the candidate conceptual models.

As shown in Figure 4, the generation process of the conceptual scene contains three steps: (1) Objects are divided into M groups according to their rotation angles. In each group, we choose the top $K\%$ objects with the most points as our conceptual models. (2) For each object that is not chosen as the conceptual model, we choose the conceptual model with the minimum average closest point distance from the object as its correspondence. (3) The scale and rotation are further refined, and the added points within the small neighborhood of original points (distance $<= \vartheta$) are removed to maintain the original structural information of the perceptual object. The conceptual scene is composed by pasting the conceptual models into the incomplete original point cloud.

Domain adaptation-based feature association. During the training process, the perceptual feature encoded by the PFE is learned to map to the conceptual feature generated by the CFG via domain adaptation, which establishes feature association. AGO-Net optimizes an additional loss term of domain distance to guide the network to generate domain-invariant source features based on the target features. In other words, we apply a much better feature representation extracted from the conceptual scene with a more complete point cloud structure as high-dimensional supervision of the 3D detection network.

Since the locations of both the real-world scanned objects and the conceptual models are spatially aligned, the smooth-$L1$ distance between these two features is directly optimized for domain adaptation. The parameters of the PFE are tuned to force perceptual objects to generate features that resemble more informative conceptual features. We further constrain the domain adaptation regions to encourage the network to focus on the feature association of the foreground objects, which means that the adaptation is restricted to foreground pixels. As illustrated in Figure 5, the foreground object mask $\mathcal{M}_{FG}$ is obtained by reprojecting the 3D object ground truth bounding boxes to the bird's-eye view (BEV) map, which will be further explained in our supplemental material. We then downsample the mask map to match the scale of the perceptual and conceptual features.

The SC-reweight module for adaptation. The attention mechanism is widely employed in many vision tasks [46], [47]. During adaptation, some parts of an object may substantially contribute to the 3D detection task and should be given greater attention, while other parts may have little contribution. We exploit a spatial and channel-wise loss reweighting module, which enables the framework to differentiate between distinct parts.

For spatial-wise attention, as shown in Figures 5 and 7, the

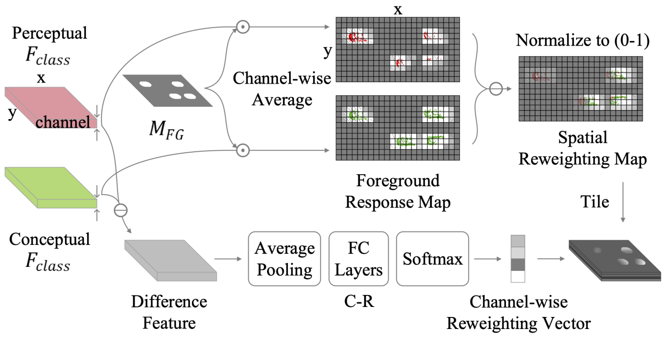


Fig. 5. Illustration of the attention-based reweighting module SC-reweight. We average the classification features along the channel of both perceptual scenes and conceptual scenes to obtain the response maps. The difference map of the response maps is calculated and multiplied by the foreground mask to obtain the spatial reweighting map. The regions with added point clouds, which contain more critical structural information and have a higher difference response, are given more attention during domain adaptation. To obtain the channel-wise reweighting vector, we utilize the difference feature map between perceptual features and conceptual features for classification as the input of a sub-network "C-R" in the SC-reweight. The channel-wise reweighting vector is calculated by the C-R and multiplied in a channel-wise manner with the tiled spatial reweighting map for adaptation.

classification response heatmap is calculated by averaging the feature along the channel dimension. We can see that with a more complete point cloud structure, the object regions of the conceptual heatmap have higher values (warmer color) than the regions of the perceptual heatmap, which means that the higher response of the feature map reflects on the regions with more complete structure information. Based on this observation, we calculate the difference map between the two response maps to obtain an attention map for regions with added structural features. Consequently, the adaptation is strengthened in these regions and forces the network to be aware of invisible object regions. For channel-wise attention, the difference map between the conceptual and perceptual feature map is fed into a sub-module, named C-R, to obtain a **C**hannel-wise **R**eweighting vector. The C-R consists of an average pooling operation, several fully-connected layers, and a softmax activation. Next, the spatial reweighting map is tiled along the channel dimension to match the size of the original feature map. The reweighting vector is then multiplied by the spatial reweighting map to obtain the final reweighting map.

This weighting approach shares the idea of focal loss [50]. We regard those parts of cars with fewer points and incomplete structure information as the hard-detected regions, which will be paid more attention during training and have much higher weights in loss computation. The SC-reweight is formulated as follows:

$$\begin{aligned}\mathcal{M}_{reweight} = \eta\{\phi[(\frac{1}{J}\sum_{j=1}^{J} {F_{class}^{\mathcal{P}}}^{j} - \frac{1}{J}\sum_{j=1}^{J} {F_{class}^{\mathcal{C}}}^{j})^2 \\ \cdot\mathcal{M}_{FG}], J\} \cdot [1 + \beta(F_{class}^{\mathcal{P}} - F_{class}^{\mathcal{C}})],\end{aligned} \tag{4}$$

where $\mathcal{M}_{reweight}$ is the output reweighting map of the SC-reweight, $F_{class}^{\mathcal{C}}$ and $F_{class}^{\mathcal{P}}$ are the inputs. J is the channels of F_{class}, ϕ represents the operation that normalizes the map to $0\sim1$. $\eta\{F, J\}$ denotes performing channel-wise tiling operation (J times) on the feature F, and $\mathcal{M}_{FG}$ is the foreground mask. β represents the feature mapping of the C-R. $F_{class}^{\mathcal{P}}$ and $F_{class}^{\mathcal{C}}$ in Eq. 4, which are calculated for $\mathcal{M}_{reweight}$, are detached from the gradient descent.

Biological model underlying the framework. The biological model of associative recognition, which underlies AGO-Net, includes three principal factors, as illustrated in Figure 2: visual cortex (VC), anterior temporal lobes (ATL), and inferior longitudinal fasciculus (ILF). The VC encodes the perceived primary information from real-world scenes, while the ATL have been implicated as a vital repository of conceptual knowledge in 3D object perception [49]. Different class-wise conceptual models stored in ATL are coded in an invariant manner with complete 3D structural features. ILF connects ATL to VC and provides the associative path between conceptual models and real-world objects. Weak features from occluded and distant objects are enhanced by the conceptual memory stored in the ATL. This conceptual knowledge retrieval and association process between the VC and the ATL is mimicked via domain adaptation.

3.2 Training of AGO-Net

Training of the CFG. There are two losses for the CFG, including binary cross entropy loss for classification and smooth-$L1$ loss for proposal generation. We denote the total loss of the CFG as:

$$\mathcal{L}_{CFG} = \mathcal{L}_{box} + \mathcal{L}_{class}. \tag{5}$$

Same regression targets as those in [24], [48] are set up, and smooth-$L1$ loss $\mathcal{L}_{box}$ is adopted to regress the normalized box parameters as:

$$\begin{aligned}&\Delta x = \frac{x_a - x_g}{d_a}, \Delta y = \frac{y_a - y_g}{h_a}, \Delta z = \frac{z_a - z_g}{d_a}, \\ &\Delta l = log(\frac{l_g}{l_a}), \Delta h = log(\frac{h_g}{h_a}), \Delta w = log(\frac{w_g}{w_a}), \\ &\Delta\theta = \theta_g - \theta_a,\end{aligned} \tag{6}$$

where x, y, and z are the center coordinates; w, l, and h are the width, length, and height, respectively; θ is the yaw rotation angle; the subscripts a, and g indicate the anchor and the ground truth, respectively; and $d_a = \sqrt{(l_a)^2 + (w_a)^2}$ is the diagonal of the base of the anchor box. $(x_a, y_a, z_a, h_a, w_a, l_a, \theta_a)$ are the parameters of 3D anchors and $(x_g, y_g, z_g, h_g, w_g, l_g, \theta_g)$ represent the corresponding ground truth box.

We use the focal loss introduced by [50] for classification, to alleviate the sample imbalance during the anchor-based training, and the classification loss $\mathcal{L}_{class}$ is formulated as follows:

$$\mathcal{L}_{class} = \alpha_t(1 - p_t)^{\gamma} log(p_t), \tag{7}$$

where p_t is the predicted classification probability and α and γ are the parameters of the focal loss.

Training of the entire AGO-Net. In addition to the classification and regression loss, there is an extra loss function for the feature adaptation. We denote the total loss of AGO-Net as $\mathcal{L}_{total}$:

$$\mathcal{L}_{total} = \mathcal{L}_{box} + \mathcal{L}_{class} + \sigma\mathcal{L}_{ago}, \tag{8}$$

where σ is a hyperparameter to balance loss terms. The association loss $\mathcal{L}_{ago}$ for feature adaptation is formulated as follows:

$$\mathcal{L}_{ago} = \frac{1}{N}\sum_{n=1}^{N}[\mathcal{S}_1({F_{box}^{\mathcal{P}}}^{n}, {F_{box}^{\mathcal{C}}}^{n}) \cdot (1 + {\mathcal{M}_{reweight}}^{n})], \tag{9}$$

where $F_{box}^{\mathcal{P}}$ and $F_{box}^{\mathcal{C}}$ are the two feature maps for box regression from the perceptual (target) and conceptual (source) domains. N denotes the number of nonzero pixels of $\mathcal{M}_{reweight}$ in Eq. 4, respectively. $\mathcal{S}_1$ denotes smooth-$L1$ distance.

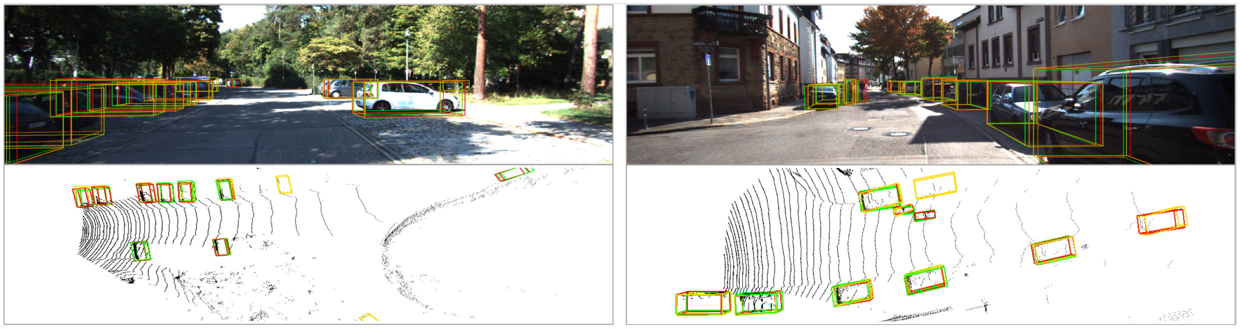


Fig. 6. Visualization of the results on the KITTI validation split set. The ground-truth 3D boxes, predicted 3D boxes of the baseline method, and predicted 3D boxes of our method are shown in green, yellow, and red, respectively, in the LiDAR phase. The first row and second row show the RGB images and the BEV images, respectively.

4 Experiments

4.1 Dataset and experimental setup

We evaluate our method on three widely acknowledged 3D object detection datasets: KITTI [5], nuScenes [51], and Waymo Open Dataset (WOD) [52].

KITTI dataset. The KITTI dataset contains 7,481 training and 7,518 test images as well as the relevant point clouds. We split the training images equally into train and validation set following previous works such as [10], [24]. Average precision (AP) metrics measured in 3D and BEV are utilized to evaluate the performance. Three levels of difficulty are defined according to the 2D bounding box height, occlusion, and truncation degree as follows: “Easy”, “Mod.”, and “Hard”.

nuScenes dataset. The nuScenes dataset contains 28,130 training samples, 6,019 validation samples, and 6,008 test samples. Each annotated frame corresponds to one point cloud captured by a 32-beam LiDAR. We follow the official evaluation protocol and report the AP (13 classes) and mean AP (mAP) results.

Waymo dataset. Waymo Open Dataset includes RGB images and point clouds from five cameras and LiDAR sensors, respectively. It consists of 1,000 scenes for training and validation and 150 scenes for testing. The dataset has 12M labeled 3D objects which are annotated in a 360-degree field. For evaluation, AP and APH [52] are used as the metric.

4.2 Implementation details

We introduce the implementation details for the KITTI dataset in this section. More details of nuScenes and Waymo datasets can be found in the released code.

Training details. For training, followed by [24], we select the LiDAR points of interest that lie in the range (0m, 70.4m), (-40m, 40m), (-3m, 1m) along X-, Y-, Z-axes, respectively, and discard the points that are invisible in the image view. During the training time for both CFG and PFE, we set the matching thresholds for the positive and negative anchors to 0.6 and 0.45, respectively. The matching IoU between the 3D bounding boxes and anchors is calculated by their nearest horizontal rectangles in BEV. The anchor for detecting the car has a size of 1.6m in width, 3.9m in length, and 1.56m in height. We discard all the anchors without any points. The construction of conceptual scenes is based only on the training set.

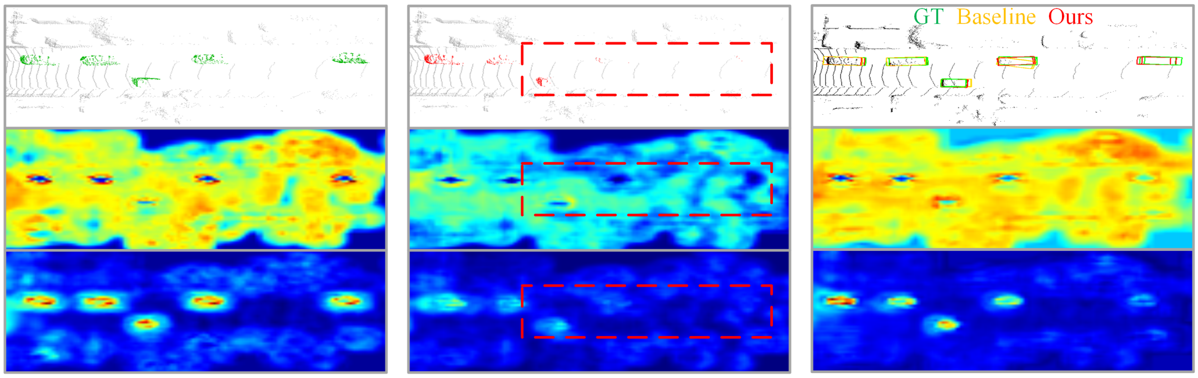
Data augmentation. To prevent AGO-Net from over-fitting, we utilize the general “cut-and-paste” strategy following [24]. Note that for each training sample, we perform the same data augmentation for both conceptual and perceptual scenes so that the objects pasted in both scenes are spatially aligned for domain adaptation.

Network architecture. Following [30], for the backbone network PFE, the point clouds of the scene are converted to the input tensor by quantization. After feeding into stacked sparse and sub-manifold convolutions for feature extraction and dimensionality reduction, the shape of the encoded tensor is 128×2×200×176, where 2 corresponds to the height dimension. We squeeze the height dimension by reshaping it into the feature map channels. The box and classification head are then separately applied to decode the features for 3D object detection. Since the PFE and CFG are siamese structures with different input data, the architecture of CFG is identical to that of the PFE. We train the CFG for conceptual models end-to-end and then keep it fixed for further AGO-Net network training. In experiments, we utilize the SGD optimizer to train AGO-Net for 50 epochs on a single NVIDIA TITAN RTX GPU for about 14 hours. We set the learning rate, weight decay, and batch size of AGO-Net to 0.01, 0.001, and 2, respectively, and apply a cosine annealing strategy to decay the learning rate. For inference, the low-confidence bounding boxes are filtered out by a threshold of 0.3. We set the IoU threshold for non-maximum suppression (NMS) to 0.1.

Training parameters. We set $\alpha = 0.25$ and $\gamma = 2$ in focal loss, $M = 24$ and $K = 20$ to create the conceptual scenes, $\sigma = 1.0$ to balance the loss terms, and $\vartheta = 0.25m$ to remove the added points that are close to the original points. For the point cloud quantization, we set (S_x, S_y, S_z) as (0.05m, 0.05m, 0.1m).

4.3 Quantitative results

KITTI dataset. Table 1 shows the results on the KITTI 3D object detection test server. On 08.10.2019, KITTI changed its evaluation setting by using 40 recall positions. We compare the proposed AGO-Net with other SoTA approaches by submitting the detection results to the KITTI server for evaluation. At the time of submission (22 Jun. 2020), our AGO-Net ranked 1^{st} among all existing one-stage detectors, 2^{nd} among all existing detectors, and 1^{st} on the “Easy” entry among all competitors on the leaderboard in the most important “Car” category, with a real-time inference speed of 25 FPS (faster than almost all other 3D object detection methods). Our method is three times faster than the two-stage top-ranked method PV-RCNN [10], while our “Easy” entry is even better than PV-RCNN, and the “Mod.” and “Hard” entries are close to it. Superior performance can be observed from Table



Conceptual Response Map (F_{box} / F_{class})

Perceptual Map before Adaptation (F_{box} / F_{class})

Perceptual Map after Adaptation (F_{box} / F_{class})

Fig. 7. Visualization of the feature maps and results on the KITTI validation set. The first row shows the BEV of the constructed conceptual scene and the perceptual scene, respectively. The ground-truth 3D boxes and the predicted 3D boxes of the baseline method and our method are drawn in green, yellow, and red, respectively, in the LiDAR phase. The second and third rows show the response maps of F_{box} and F_{class}, respectively.

TABLE 1
Results on the KITTI 3D object detection test server. The 3D object detection and BEV detection are evaluated by average precision at IoU = 0.7 for cars. According to the new policy of the KITTI dataset, all results are evaluated by the mean average precision with 40 recall positions (R40).

Stage	Method	Time(s)	Modality	3D Detection (%)			BEV Detection (%)			GPU
				Easy	Mod.	Hard	Easy	Mod.	Hard	
Two-stage	MV3D [16]	0.36	LiDAR + RGB	74.97	63.63	54.00	86.49	78.98	72.23	TITAN X
	F-PointNet [32]	0.17	LiDAR + RGB	82.19	69.79	60.59	91.17	84.67	74.77	GTX 1080
	AVOD [17]	0.10	LiDAR + RGB	76.39	66.47	60.23	89.75	84.95	78.32	TITAN XP
	PointRCNN [34]	0.10	LiDAR only	86.96	75.64	70.70	92.13	87.39	82.72	TITAN XP
	F-ConvNet [53]	0.48	LiDAR + RGB	87.36	76.39	66.69	91.51	85.84	76.11	-
	Fast Point R-CNN [54]	0.07	LiDAR only	85.29	77.40	70.24	90.87	87.84	80.52	Tesla P40
	MMF [55]	0.08	LiDAR + RGB	88.40	77.43	70.22	93.67	88.21	81.99	TITAN XP
	STD [8]	0.10	LiDAR only	87.95	79.71	75.09	94.74	89.19	86.42	TITAN V
	Patches [56]	0.15	LiDAR only	88.67	77.20	71.82	92.72	88.39	83.19	GTX 1080Ti
	Part-A^2 [57]	0.07	LiDAR only	87.81	78.49	73.51	91.70	87.79	84.61	Tesla V100
	PV-RCNN [10]	0.15	LiDAR only	90.25	**81.43**	**76.82**	94.98	90.65	**86.14**	GTX 2080Ti
One-stage	VoxelNet [48]	0.23	LiDAR only	77.82	64.17	57.51	87.95	78.39	71.29	TITAN X
	ContFuse [3]	0.06	LiDAR + RGB	83.68	68.78	61.67	94.07	85.35	75.88	-
	SECOND [24]	0.05	LiDAR only	83.34	72.55	65.82	89.39	83.77	78.59	GTX 1080Ti
	PointPillars [25]	0.02	LiDAR only	82.58	74.31	68.99	90.07	86.56	82.81	GTX 1080Ti
	Point-GNN [15]	0.64	LiDAR only	88.33	79.47	72.29	93.11	89.17	83.90	GTX 1070
	3DSSD [35]	0.04	LiDAR only	88.36	79.57	74.55	92.66	89.02	85.86	TITAN V
	SA-SSD [30]	0.04	LiDAR only	88.75	79.79	74.16	95.03	**91.03**	85.96	GTX 2080Ti
	AGO-Net (ours)	0.04	LiDAR only	**91.53**	80.77	75.23	**95.55**	90.00	84.72	GTX 2080Ti

1 when comparing AGO-Net with other cutting-edge one-stage 3D detection approaches. Our AGO-Net leads both SA-SSD [30] and 3DSSD [35] by (2.8%; 1.0%; 1.1%) and (3.2%; 1.2%; 0.7%) in 3D detection, respectively. The performance boost is mainly contributed by the domain adaptation-based feature association, which exploits the feature enhancement capability of the network.

To further validate the superior performance of AGO-Net, we evaluate AGO-Net on the KITTI validation split set, as shown in Table 2. Note that we only report AP with 11 recall positions in Table 2 and Table 12 to compare with the results from previous frameworks. For 3D object detection, by only utilizing LiDAR point clouds, our AGO-Net outperforms all existing methods on the most critical "Mod." difficulty level. Thanks to the self-contained conceptual model and feature association mechanism, our straightforward yet effective network can achieve superior performance compared with complicated networks in both accuracy and speed. AGO-Net leads the 1^{st} ranked one-stage method SA-SSD and current 1^{st} ranked two-stage method PV-RCNN by 4.0% and 0.2%, respectively.

TABLE 2
"Mod." results on 3D object detection of the KITTI validation split set at IoU = 0.7 for cars. The AP is calculated with 11 recall positions (R11) to compare our method with the previous methods.

Stage	Method	Reference	AP_{3D}
Two-stage	STD [8]	ICCV 2019	79.80
	Patches [56]	arXiv 2019	79.04
	Part-A^2 [57]	TPAMI 2020	79.47
	PV-RCNN [10]	CVPR 2020	83.90
One-stage	Point-GNN [15]	CVPR 2020	78.34
	3DSSD [35]	CVPR 2020	79.45
	SA-SSD [30]	CVPR 2020	79.91
	AGO-Net (ours)	-	**83.92**

TABLE 3
3D object detection $\mathrm{AP}_{\mathrm{3D}}$ (R40) performance for "Pedestrian" and "Cyclist" on KITTI val split set at IoU = 0.5.

Method	Pedestrian			Cyclist		
	Easy	Mod.	Hard	Easy	Mod.	Hard
conceptual	61.79	58.77	59.97	84.99	80.87	80.56
SECOND† [24]	55.94	51.14	46.17	82.96	66.74	62.78
baseline	57.47	52.02	47.44	83.31	67.55	63.94
ours	**60.39**	**54.81**	**50.59**	**87.57**	**69.24**	**64.74**

TABLE 4
3D object detection $\mathrm{AP}_{\mathrm{3D}}$ (R40) performance for "Pedestrian" and "Cyclist" on KITTI test set at IoU = 0.5.

Method	Pedestrian			Cyclist		
	Easy	Mod.	Hard	Easy	Mod.	Hard
SECOND† [24]	43.04	35.92	33.55	71.05	55.64	49.83
ours	**45.18**	**37.22**	**34.62**	**72.82**	**57.60**	**51.53**

The "Cyclist" and "Pedestrian" categories are more challenging than the "Car" for voxel-only-based 3D object detection due to their non-rigid structures and small scale. To fully validate the effectiveness of the proposed method, we still report $\mathrm{AP}_{\mathrm{3D}}$ results (IoU = 0.5) of these two categories with respect to the baseline method and SECOND [24] in Tables 3 and 4. The SECOND† is implemented in OpenPCDet [58]. Note that in the following sections, the "baseline" denotes our instantiated PFE (without the CFG) introduced in Sec. 3.1, and the "conceptual" denotes training and testing on the conceptual dataset. Despite the number and quality limitation of conceptual models, our method still outperforms the baseline method with considerable margins.

As shown in Table 8, we also report results for the near-range experiment at 0-30 m, median-range at 30-50 m, and far-range at 50-80 m with and without adaptation. Regarding both near-, median-, and far-range performances, our method outperforms the baseline method by (0.5%; 3.1%; 3.0%) on "Mod." entry, respectively, especially for the median- and far-range.

nuScenes and Waymo datasets. To demonstrate the versatility of our approach, we evaluate on nuScenes and Waymo datasets. Following previous works [10], [25], we adopt the same range and ring-view scenes for processing point clouds and official evaluation metrics of each dataset. Tables 5 and 6 demonstrate the superiority of our method on validation and test set of nuScenes, and Table 7 depicts the prevalence on the Waymo dataset compared to the baseline and the cutting-edge one-stage method [24]. For Waymo dataset, only 20% subset of data is utilized for training due to limited computation resources, which is consistent with OpenPCDet [58]. Note that the backbone of our method (derived from [30]) is different from SECOND†, therefore the performances of SECOND† and the baseline are slightly different.

4.4 Qualitative results

We present some representative comparison results of the baseline and AGO-Net on the KITTI validation set in Figure 6. 3D bounding boxes detected from LiDAR are projected onto RGB images for better visualization. The ground truth, and the predicted 3D boxes of the baseline and our method, are colored in green, yellow, and red, respectively. The first to third rows show the RGB image, the front view, and the bird's-eye view, respectively. As observed, our method can produce high-quality 3D bounding boxes in different kinds of scenes. The distant and occluded objects that are difficult for the baseline method to detect, are accurately predicted by AGO-Net due to the association capability of the network.

To further observe how the perceptual-to-conceptual domain adaptation fundamentally improves 3D object detection, we visualize the conceptual and perceptual features before and after adaptation as well as their corresponding detection results in Figure 7. The first row shows the bird's-eye view of perceptual and conceptual scenes, respectively. The second and third rows show the feature decoded for 3D bounding box and classification, respectively. We average the feature map along the channel dimension to obtain the response heatmap. Warmer color indicates a higher response. As depicted in the first and second columns, the pixel-wise difference between two feature maps is prone to be large in regions containing insufficient and incomplete structural information due to occlusion or distance. The conceptual response of distant and occluded objects is much more obvious than that of the perceptual objects (highlighted by the red boxes), as the added point clouds for each incomplete perceptual object provide adequate structural information to extract discriminative features. After the perceptual domain feature is well adapted, the response of the distant and occluded objects becomes more similar to that of conceptual objects. The qualitative comparison (Row 1 and Column 3) of the results based on the original perceptual feature (baseline) and the feature after adapted (ours) shows that the two distant objects are well detected, and the rotation angle of the object is refined due to the associated critical structural information. The feature visualization demonstrates that the proposed approach adaptively generates robust features for predicting more accurate 3D bounding boxes.

4.5 Ablation study

Ablation experiments are conducted on the KITTI dataset.

The upper bound performance of the conceptual domain. To validate the upper-bound performance of the CFG module, we conduct experiments on conceptual scenes only in both training and testing, rather than the real dataset. High performance can be observed in the first row of Table 9, which can serve to instruct our associated-guided network. As shown in the first row, the high precision (over 96%) on all entries indicates the possibility of adopting conceptual models to guide PFE for enhanced feature learning, while the second row shows the domain gap between the conceptual and perceptual dataset; therefore we cannot just use the conceptual scenes for training and test the model on the real scenes. The third row shows the performance of the baseline model. The fourth row "real + conceptual" indicates merely training baseline model with KITTI train split set and the generated conceptual data without using the siamese network for adaptation. The result validates that simply mixing more training data using the same network will not improve the detector performance.

Different settings of AGO-Net. We investigated the performance with and without the domain adaptation and the SC-reweight module, as shown in Table 10. The first row is the baseline results trained with only KITTI train split set. Owing to the object-wise domain adaptation ("CFG"), an improvement is observed in Row 2. The third row shows the results with both the CFG and the SC-reweight module ("SC"). The improvements on all difficulty levels indicate that the full approach learns more discriminative features for 3D detection.

TABLE 5
Per class performance on the nuScenes validation set. Evaluation of detections as measured by average precision (AP) or mean AP (mAP). Abbreviations: construction vehicle (C. V.), pedestrian (Ped.), motorcycle (Motor.), and traffic cone (T. C.).

Methods	mAP	Car	Truck	Bus	Trailer	C. V.	Ped.	Motor.	Bicycle	T. C.	Barrier	Reference
SECOND [24]	27.1	75.5	21.9	29.0	13.0	0.4	59.9	16.9	0.0	22.5	32.2	Sensors 2018
PointPillars [25]	29.5	70.5	25.0	34.4	20.0	4.5	59.9	16.7	1.6	29.6	33.2	CVPR 2019
WYSIWYG [29]	35.4	80.0	35.8	54.1	28.5	7.5	66.9	18.5	0.0	27.9	34.5	CVPR 2020
InfoFocus [26]	36.4	77.6	35.4	50.5	25.6	8.3	61.7	25.2	2.5	33.4	43.4	ECCV 2020
3DSSD [35]	42.7	81.2	47.2	61.4	30.5	12.6	70.2	**36.0**	**8.6**	31.1	47.9	CVPR 2020
baseline	42.9	79.8	45.9	62.0	32.5	12.7	71.1	29.6	4.5	44.1	47.2	-
ours	**45.1**	**81.5**	**50.1**	**62.2**	**34.0**	**13.3**	**72.2**	32.5	5.9	**48.1**	**51.2**	-

TABLE 6
Per class performance on the nuScenes test set. Evaluation of detections as measured by average precision (AP) or mean AP (mAP). Abbreviations: construction vehicle (C. V.), pedestrian (Ped.), motorcycle (Motor.), and traffic cone (T. C.).

Methods	mAP	Car	Truck	Bus	Trailer	C. V.	Ped.	Motor.	Bicycle	T. C.	Barrier	Reference
PointPillars [25]	30.5	68.4	23.0	28.2	23.4	4.1	59.7	27.4	1.1	30.8	38.9	CVPR 2019
CenterNet [7]	33.8	53.6	27.0	24.8	25.1	8.6	37.5	29.1	**20.7**	**58.3**	53.3	arXiv 2019
WYSIWYG [29]	35.0	79.1	30.4	46.6	40.1	7.1	65.0	18.2	0.1	28.8	34.7	CVPR 2020
InfoFocus [26]	39.5	77.9	31.4	44.8	37.3	10.7	63.4	29.0	6.1	46.5	47.8	ECCV 2020
ours	**46.2**	**80.7**	**45.0**	**57.5**	**48.4**	**12.0**	**71.4**	**30.7**	5.9	57.2	**53.6**	-

TABLE 7
Multi-class 3D detection results on the validation sequence of the Waymo dataset.

Difficulty	Method	Vehicle (IoU=0.7)		Pedestrian (IoU=0.5)		Cyclist (IoU=0.5)	
		3D AP	3D APH	3D AP	3D APH	3D AP	3D APH
LEVEL_1	SECOND† [24]	67.9	67.3	57.8	47.6	54.0	52.7
	baseline	67.4	66.8	57.4	47.8	53.5	52.3
	ours	**69.2**	**68.7**	**59.3**	**48.7**	**55.3**	**54.2**
LEVEL_2	SECOND† [24]	59.4	58.9	49.8	41.0	52.3	51.0
	baseline	58.9	58.3	49.4	41.1	51.8	50.6
	ours	**60.6**	**60.1**	**51.8**	**42.4**	**53.5**	**52.5**

TABLE 8
Near-range (0∼30m), median-range (30∼50m), and far-range (50∼80m) comparisons on $\mathrm{AP}_{\mathrm{3D}}$ (R40). Note that for the easy level in the far-range area, the AP is always 0 since there is no easy object.

Method	Near-range			Median-range			Far-range	
	Easy	Mod.	Hard	Easy	Mod.	Hard	Mod.	Hard
baseline	95.37	95.33	92.34	48.05	63.89	61.73	7.49	9.57
ours	**95.91**	**95.85**	**92.90**	**52.14**	**67.02**	**64.79**	**10.44**	**11.36**

TABLE 9
The upper bound performance of $\mathrm{AP}_{\mathrm{3D}}$ (R40) training with the conceptual data and testing on the conceptual.

training data	validation data	$\mathrm{AP}_{\mathrm{3D}}$ (IoU=0.7)		
		Easy	Mod.	Hard
conceptual	conceptual	98.74	98.72	96.56
conceptual	real	92.39	72.55	70.04
real	real	92.21	**83.03**	**79.98**
real + conceptual	real	**92.42**	82.97	79.86

TABLE 10
AGO-Net with different settings on $\mathrm{AP}_{\mathrm{3D}}$ at IoU=0.7 (R40).

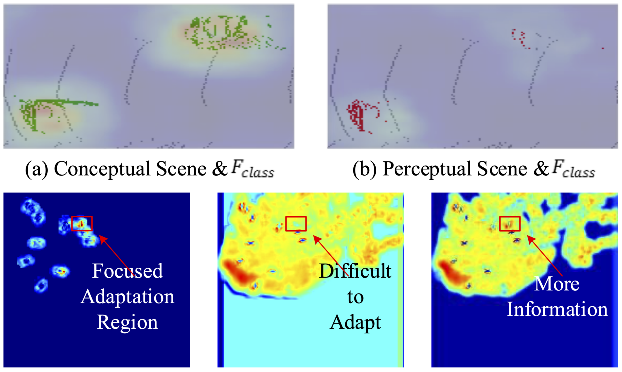
Method	$\mathrm{AP}_{\mathrm{3D}}$ (IoU=0.7)		
	Easy	Mod.	Hard
baseline	92.21	83.03	79.98
CFG	93.45	84.21	81.86
CFG + SC	**94.01**	**84.84**	**82.38**

TABLE 11
Effects of spatial and channel-wise attention in SC-reweight (R40).

Attention setting	$\mathrm{AP}_{\mathrm{3D}}$ (IoU=0.7)		
	Easy	Mod.	Hard
none	93.45	84.21	81.86
spatial	93.90	84.75	82.32
channel	93.59	84.32	81.97
spatial + channel	**94.01**	**84.84**	**82.38**

Spatial and channel-wise reweighting (SC-reweight). To validate the effectiveness of the SC-reweight, we conduct experiments with different settings of the SC-reweight, as shown in Table 11. "spatial" shows the performance based on the reweighting map that only utilizes spatial attention. "channel" reports the result with channel-wise attention. "spatial + channel" shows the performance with the full setting. The progressive improvement demonstrates that the spatial and channel-wise loss reweighting is vital to the domain adaptation process and forces the network to adapt to more critical structural features during the adaptation.

The comparison of (a) and (b) in Figure 8 shows that the regions with added point clouds in the conceptual scene have a much higher response than the regions in the perceptual scene. As shown in (c), the high-response regions of the spatial reweighting map are spatially aligned with those less informative regions that are difficult to adapt in the perceptual feature in (d). Consequently, during the adaptation process, the regions are given more attention to adapt more spatial and structural information for regressing 3D object bounding boxes from the conceptual feature in (e).



(a) Conceptual Scene & F_{class} (b) Perceptual Scene & F_{class}

(c) Spatial Reweighting Map (d) Perceptual F_{box} during Adaptation (e) Conceptual F_{box} Response Map

Fig. 8. Visualization of the feature maps and spatial reweighting maps. The first row shows the comparison of (a) conceptual scene and (b) perceptual scene with their classification response maps, respectively. The second row depicts (c) the spatial reweighting map, (d) the perceptual feature response map during adaptation, and (e) the conceptual feature response map, respectively.

TABLE 12
Applying our association-guided architecture on different frameworks on $\mathrm{AP}_{3\mathrm{D}}$. The AP is calculated with 11 recall positions (R11) to compare our method with the previous methods

Method	$\mathrm{AP}_{3\mathrm{D}}$ (IoU=0.7)		
	Easy	Mod.	Hard
PointPillars [25]	85.14	76.44	70.02
PointPillars + AGO	**86.16**	**78.53**	**75.21**
SECOND [24]	88.07	77.12	75.27
SECOND + AGO	**89.17**	**78.82**	**77.51**

Generalizability. We adopt other 3D detection networks [24], [25] to instantiate the PFE and CFG to validate the expansion capability and versatility of AGO-Net. As shown in Table 12, our association-guided framework (with "AGO") boosts these methods by a large margin, especially for "Mod." and "Hard" entries.

TABLE 13
Comparison on different construction strategies (R40).

Construction strategy	$\mathrm{AP}_{3\mathrm{D}}$ (IoU=0.7)		
	Easy	Mod.	Hard
replace	93.74	84.38	82.11
add (with removal)	**94.01**	**84.84**	**82.38**

Different conceptual scene construction strategies. As illustrated in Figure 4, different from the previous work [14], we do not simply use the conceptual model to replace the perceptual object. Instead, we paste the conceptual model onto the original perceptual object and remove those conceptual points that are close to the original perceptual points. Such a simple removal operation keeps the original structure of the object point cloud. Table 13 shows the superior performance of our improved strategy.

TABLE 14
Comparison of performing adaptation on different features (R40).

Feature for adaptation	$\mathrm{AP}_{3\mathrm{D}}$ (IoU=0.7)		
	Easy	Mod.	Hard
$F_{box} + F_{class}$	93.40	84.10	81.73
F_{class}	92.51	83.16	80.07
F_{box}	**93.45**	**84.21**	**81.86**

Comparison of performing adaptation on different features. We attempt to adapt only F_{class} or both F_{box} and F_{class} to ascertain which feature adaptation is more effective. As shown in Table 14, "$F_{box} + F_{class}$" adaptation and "F_{class}" adaptation do not perform better than "F_{box}" adaptation. The association-based mechanism digs deep into the invisible structures of occluded and distant objects for object perception, with the premise that the categories of the objects are correctly recognized.

Comparison on different attention modules. We also experiment to compare the SC-reweight module with the Incompletion-aware module (IAM) proposed in [14]. Table 15 demonstrates the superior performance of the SC-reweight module. Note that the SC-reweight does not depend on additional time-consuming deformable convolutional layers [59].

TABLE 15
Comparison on different attention modules with and without the deformable convolutional layer (R40).

Attention module	Deform layer	$\mathrm{AP}_{3\mathrm{d}}$ (IoU=0.7)			FPS
		Easy	Mod.	Hard	
none	×	93.45	84.21	81.86	25
none	✓	93.71	84.39	82.02	17
IAM [14]	✓	94.08	84.69	82.37	17
SC-reweight	×	94.01	84.84	82.38	25
SC-reweight	✓	**94.26**	**84.98**	**82.52**	17

5 Conclusion

In this paper, we proposed a brain-inspired 3D point cloud object detection framework, named AGO-Net. This learns to associate features of perceived objects from the real scene with discriminative features from their conceptual models via domain adaptation, which fundamentally enhances the feature robustness against appearance changes in point clouds. We further leverage the classification feature maps to obtain a reweighting map for the adaptation loss, which encourages the feature adaptation to focus on more informative regions, and boosts the final detection performance. A practical method for generating conceptual scenes without any external dataset is also introduced. Moreover, our simple yet effective approach can be easily integrated into many existing 3D object detection frameworks. The experimental results on the KITTI, nuScenes and Waymo datasets demonstrate the effectiveness and efficiency of the proposed framework.

Acknowledgments

This work was supported by National Key R&D Program of China (No.2019YFA0709502, 2018YFC1312904), Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01), ZJ Lab, and Shanghai Center for Brain Science and Brain-Inspired Technology, the 111 Project (No.B18015).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3104172, IEEE Transactions on Pattern Analysis and Machine Intelligence

REFERENCES

[1] Sanja Fidler, Sven Dickinson, and Raquel Urtasun. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *Advances in Neural Information Processing Systems*, pp. 611–619, 2012.

[2] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3569–3577, 2018.

[3] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision*, pp. 641–656, 2018.

[4] Xiaoqing Ye, Liang Du, Yifeng Shi, Yingying Li, Xiao Tan, Jianfeng Feng, Errui Ding, and Shilei Wen. Monocular 3D object detection via feature domain adaptation. In *Proceedings of the European Conference on Computer Vision*, pp. 17–34, 2020.

[5] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[7] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[8] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1951–1960, 2019.

[9] Ze Wang, Sihao Ding, Ying Li, Minming Zhao, Sohini Roychowdhury, Andreas Wallin, Guillermo Sapiro, and Qiang Qiu. Range adaptation for 3d object detection in lidar. In *Proceedings of the IEEE International Conference on Computer Vision Workshop*, 2019.

[10] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10529–10538, 2020.

[11] Demis Hassabis, Dharshan Kumaran, Christopher Summerfield, and Matthew Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

[12] Liang Du, Jingang Tan, Xiangyang Xue, Lili Chen, Hongkai Wen, Jianfeng Feng, Jiamao Li, and Xiaolin Zhang. 3dcfs: Fast and robust joint 3d semantic-instance segmentation via coupled feature selection. In *IEEE International Conference on Robotics and Automation*, pp. 6868–6875, 2020.

[13] Giovanni A Carlesimo, Paola Casadio, Maurizio Sabbadini, and Carlo Caltagirone. Associative visual agnosia resulting from a disconnection between intact visual memory and semantic systems. *Cortex*, 34(4):563–576, 1998.

[14] Liang Du, Xiaoqing Ye, Xiao Tan, Jianfeng Feng, Zhenbo Xu, Errui Ding, and Shilei Wen. Associate-3ddet: Perceptual-to-conceptual association for 3d point cloud object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13329–13338, 2020.

[15] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1711–1719, 2020.

[16] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, 2017.

[17] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1–8, 2018.

[18] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7652–7660, 2018.

[19] Bin Yang, Ming Liang, and Raquel Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *Conference on Robot Learning*, pp. 146–155, 2018.

[20] Bo Li. 3d fully convolutional network for vehicle detection in point cloud. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1513–1518, 2017.

[21] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-end multi-View fusion for 3D object detection in lidar point clouds. In *Conference on Robot Learning*, pp. 923–932, 2019.

[22] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *IEEE International Conference on Robotics and Automation*, pp. 1355–1361, 2017.

[23] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017.

[24] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[25] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12697–12705, 2019.

[26] Jun Wang, Shiyi Lan, Mingfei Gao, and Larry S. Davis. Infofocus: 3d object detection for autonomous driving with dynamic information modeling. In *Proceedings of the European Conference on Computer Vision*, 2020.

[27] R. Qi Charles, Hao Su, Kaichun Mo, and Leonidas J. Guibas, Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017.

[28] Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4604–4612, 2020.

[29] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What you see is what you get: Exploiting visibility for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11001–11009, 2020.

[30] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11873–11882, 2020.

[31] Shuyang Cheng, Zhaoqi Leng, Ekin Dogus Cubuk, Barret Zoph, Chunyan Bai, Jiquan Ngiam, Yang Song, Benjamin Caine, Vijay Vasudevan, Congcong Li, Quoc V. Le, Jonathon Shlens, and Dragomir Anguelov. Improving 3D object detection through progressive population based augmentation. In *Proceedings of the European Conference on Computer Vision*, pp. 279–294, 2020.

[32] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 918–927, 2018.

[33] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 244–253, 2018.

[34] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–779, 2019.

[35] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11040–11048, 2020.

[36] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[37] Judy Hoffman, Sergio Guadarrama, Eric S Tzeng, Ronghang Hu, Jeff Donahue, Ross Girshick, Trevor Darrell, and Kate Saenko. Lsda: Large scale detection through adaptation. In *Advances in Neural Information Processing Systems*, pp. 3536–3544, 2014.

[38] Liang Du, Jingang Tan, Hongye Yang, Jianfeng Feng, Xiangyang Xue, Qibao Zheng, Xiaoqing Ye, and Xiaolin Zhang. Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 982–991, 2019.

[39] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537, 2011.

[40] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Scholkopf, and Alexander J Smola. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems*, pp. 513–520, 2006.

[41] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[42] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pp. 3320–3328, 2014.

[43] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured Knowledge Distillation for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2604—2613, 2019.

[44] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *Proceedings of the International Conference on Learning Representations*, 2017.

[45] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 379—387, 2016.

[46] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794-–7803, 2018.

[47] Zhe Liu, Xin Zhao, Tengteng Huang, Ruolan Hu, Yu Zhou, and Xiang Bai. Tanet: Robust 3d object detection from point clouds with triple attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 11677–11684, 2020.

[48] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, 2018.

[49] Paul Hoffman and Matthew A Lambon Ralph. From percept to concept in the ventral temporal lobes: Graded hemispheric specialisation based on stimulus and task. *Cortex*, 101:107–118, 2018.

[50] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017.

[51] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11621–11631, 2020.

[52] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2446–2454, 2020.

[53] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1742-–1749, 2019.

[54] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9775–9784, 2019.

[55] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7345–7353, 2019.

[56] Johannes Lehner, Andreas Mitterecker, Thomas Adler, Markus Hofmarcher, Bernhard Nessler, and Sepp Hochreiter. Patch refinement–localized 3d object detection. *arXiv preprint arXiv:1910.04093*, 2019.

[57] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[58] OpenPCDet Development Team, Openpcdet: An open-source toolbox for 3D object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020.

[59] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 764–773, 2017.

Liang Du received the B.S. degree from Harbin Engineering University, China, in 2016. He received the M.S. degree from the University of Chinese Academy of Sciences, China, in 2019. He is currently pursuing the Ph.D. degree at Fudan University, China. His current research interests include 3D computer vision, transfer learning, and autonomous driving. He has contributed several papers in top conferences, such as CVPR, ICCV, and ECCV.

Xiaoqing Ye received the Ph.D. degree in information and communication engineering from the University of Chinese Academy of Sciences, China, in 2019. She is currently with the department of computer vision technology, Baidu as a Senior Engineer. Her current research interests include 3D computer vision and autonomous driving. She has contributed several papers in top conferences, such as CVPR, ECCV, ICCV and AAAI.

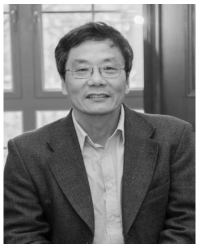
Xiao Tan received the Ph.D. degree in computer vision from the University of New South Wales, Sydney, in 2014. His research interests include computer vision, pattern recognition, and image processing. He is currently with the department of computer vision technology, Baidu as a Senior Engineer, leading the tech-team to develop visual systems for AI-city and autonomous driving. He has contributed 10+ papers in top conferences and journals, such as CVPR, ECCV, ICCV, and TIP.

Edward Johns received the B.A. and M.Eng. degrees from the University of Cambridge in 2006 and 2007, and the PhD degree from Imperial College London in 2014. He is currently Director of the Robot Learning Lab at Imperial College London, where his research focusses on the intersection between robotics, computer vision, and machine learning. He has contributed 30+ papers in top conferences and journals, such as NeurIPS, CVPR, ICCV, ECCV, and IJCV.

Bo Chen received the B.S., and M.S. degrees from Jilin University, China. He is the general manager of FAW (Nanjing) Technology Development Co., Ltd, and experts in artificial intelligence. He is currently in charge of the development of autonomous driving based on AI algorithm.

Errui Ding received Ph.D. degree from Xidian University in 2008 and currently is the director of Computer Vision Technology Department (VIS) of Baidu Inc. In recent years, he has published tens of papers on top-tier conferences and was awarded Best Paper Runner-up at ICDAR 2019. He co-organized several competitions and workshops at recent ICDAR and CVPR. He is also a member of CSIG, CCF and CAAI.

Xiangyang Xue received the B.S., M.S., and Ph.D. degrees in communication engineering from Xidian University, Xian, China, in 1989, 1992, and 1995, respectively. He is currently a professor of computer science with Fudan University, Shanghai, China. His research interests include computer vision, multimedia information processing, and machine learning.



Jianfeng Feng received the B.S., M.S., and Ph.D. degrees from the Department of Probability and Statistics, Peking University, China. He is the chair professor of Shanghai National Centre for Mathematic Sciences and the Dean of Brain-inspired AI Institute in Fudan University. He has been developing new mathematical, statistical, and computational theories and methods to meet the challenges raised in neuroscience and mental health researches.