Virtual Sparse Convolution for Multimodal 3D Object Detection

Hai Wu[1] Chenglu Wen[1*] Shaoshuai Shi[2] Xin Li[3] Cheng Wang[1]
[1]Xiamen University [2]Max-Planck Institute [3]Texas A&M University

Abstract

*Recently, virtual/pseudo-point-based 3D object detection that seamlessly fuses RGB images and LiDAR data by depth completion has gained great attention. However, virtual points generated from an image are very dense, introducing a huge amount of redundant computation during detection. Meanwhile, noises brought by inaccurate depth completion significantly degrade detection precision. This paper proposes a fast yet effective backbone, termed **VirConvNet**, based on a new operator **VirConv** (Virtual Sparse Convolution), for virtual-point-based 3D object detection. VirConv consists of two key designs: (1) **StVD** (Stochastic Voxel Discard) and (2) **NRConv** (Noise-Resistant Submanifold Convolution). StVD alleviates the computation problem by discarding large amounts of nearby redundant voxels. NRConv tackles the noise problem by encoding voxel features in both 2D image and 3D LiDAR space. By integrating VirConv, we first develop an efficient pipeline **VirConv-L** based on an early fusion design. Then, we build a high-precision pipeline **VirConv-T** based on a transformed refinement scheme. Finally, we develop a semi-supervised pipeline **VirConv-S** based on a pseudo-label framework. On the KITTI car 3D detection test leaderboard, our VirConv-L achieves **85% AP** with a fast running speed of **56ms**. Our VirConv-T and VirConv-S attains a high-precision of **86.3%** and **87.2% AP**, and currently **rank 2nd and 1st**[1], respectively. The code is available at https://github.com/hailanyi/VirConv.*

1. Introduction

3D object detection plays a critical role in autonomous driving [32, 45]. The LiDAR sensor measures the depth of scene [4] in the form of a point cloud and enables reliable localization of objects in various lighting environments. While LiDAR-based 3D object detection has made rapid progress in recent years [19, 23, 25, 27, 28, 42, 43, 49], its performance drops significantly on distant objects, which inevitably have sparse sampling density in the scans. Unlike

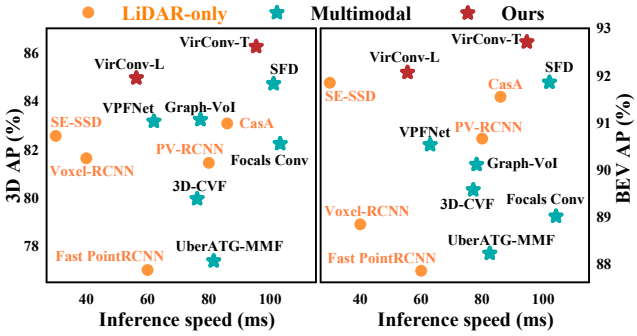

Figure 1. Our VirConv-T achieves top average precision (AP) on both 3D and BEV moderate car detection in the KITTI benchmark (more details are in Table 1). Our VirConv-L runs fast at 56ms with competitive AP.

LiDAR scans, color image sensors provide high-resolution sampling and rich context data of the scene. The RGB image and LiDAR data can complement each other and usually boost 3D detection performance [1, 6, 20, 21, 24].

Early methods [29–31] extended the features of LiDAR points with image features, such as semantic mask and 2D CNN features. They did not increase the number of points; thus, the distant points still remain sparse. In contrast, the methods based on virtual/pseudo points (for simplicity, both denoted as **virtual points** in the following) enrich the sparse points by creating additional points around the LiDAR points. For example, MVP [45] creates the virtual points by completing the depth of 2D instance points from the nearest 3D points. SFD [36] creates the virtual points based on depth completion networks [16]. The virtual points complete the geometry of distant objects, showing the great potential for high-performance 3D detection.

However, virtual points generated from an image are generally very dense. Taking the KITTI [9] dataset as an example, an 1242×375 image generates 466k virtual points (∼**27**× more than the LiDAR scan points). This brings a huge computational burden and causes a severe efficiency issue (see Fig. 2 (f)). Previous work addresses the density problem by using a larger voxel size [19, 44] or by randomly down-sampling [17] the points. However, applying such methods to virtual points will inevitably sacrifice use-

*Corresponding author
[1]On the date of CVPR deadline, *i.e.*, Nov.11, 2022

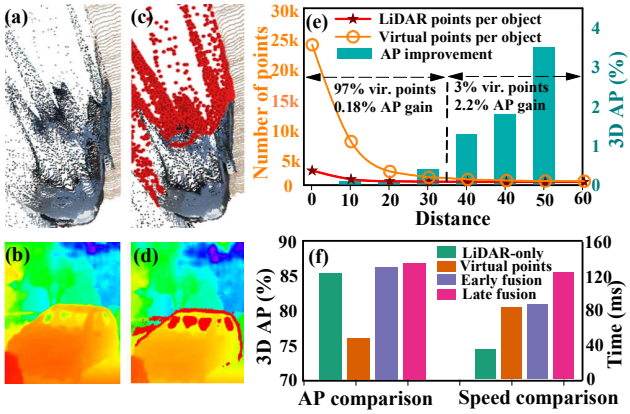

Figure 2. The noise problem and density problem of virtual points. (a) Virtual points in 3D space. (b) Virtual points in 2D space. (c) Noises (red) in 3D space. (d) Noises (red) distributed on 2D instance boundaries. (e) Virtual points number versus AP improvement along different distances by using Voxel-RCNN [7] with late fusion (details see Sec. 3.1). (f) Car 3D AP and inference time using Voxel-RCNN [7] with LiDAR-only, virtual points-only, early fusion, and late fusion (details see Sec. 3.1), respectively.

ful shape cues from faraway points and result in decreased detection accuracy.

Another issue is that the depth completion can be inaccurate, and it brings a large amount of noise in the virtual points (see Fig. 2 (c)). Since it is very difficult to distinguish the noises from the background in 3D space, the localization precision of 3D detection is greatly degraded. In addition, the noisy points are non-Gaussian distributed, and can not be filtered by conventional denoising algorithms [8, 12]. Although recent semantic segmentation network [15] show promising results, they generally require extra annotations.

To address these issues, this paper proposes a VirConvNet pipeline based on a new Virtual Sparse Convolution (VirConv) operator. Our design builds on **two main observations**. (1) First, geometries of nearby objects are often relatively complete in LiDAR scans. Hence, most virtual points of nearby objects only bring marginal performance gain (see Fig. 2 (e)(f)), but increase the computational cost significantly. (2) Second, noisy points introduced by inaccurate depth completions are mostly distributed on the instance boundaries (see Fig. 2 (d)). They can be recognized in 2D images after being projected onto the image plane.

Based on these two observations, we design a **StVD** (Stochastic Voxel Discard) scheme to retain those most important virtual points by a bin-based sampling, namely, discarding a huge number of nearby voxels while retaining faraway voxels. This can greatly speed up the network computation. We also design a **NRConv** (Noise-Resistant Submanifold Convolution) layer to encode geometry features of voxels in both 3D space and 2D image space. The extended receptive field in 2D space allows our NRConv to distinguish the noise pattern on the instance boundaries in 2D image space. Consequently, the negative impact of noise can be suppressed.

We develop three multimodal detectors to demonstrate the superiority of our VirConv: (1) a lightweight **VirConv-L** constructed from Voxel-RCNN [7]; (2) a high-precision **VirConv-T** based on multi-stage [34] and multi-transformation [35] design; (3) a semi-supervised **VirConv-S** based on a pseudo-label [33] framework. The effectiveness of our design is verified by extensive experiments on the widely used KITTI dataset [9] and nuScenes dataset [3]. Our contributions are summarized as follows:

- We propose a **VirConv** operator, which effectively encodes voxel features of virtual points by **StVD** and **NRConv**. The StVD discards a huge number of redundant voxels and substantially speeds up the 3D detection prominently. The NRConv extends the receptive field of 3D sparse convolution to the 2D image space and significantly reduces the impact of noisy points.
- Built upon VirConv, we present three new multimodal detectors: a **VirConv-L**, a **VirConv-T**, and a semi-supervised **VirConv-S** for efficient, high-precision, and semi-supervised 3D detection, respectively.
- Extensive experiments demonstrated the effectiveness of our design (see Fig. 1). On the KITTI leaderboard, our VirConv-T and VirConv-S currently **rank 2nd and 1st**, respectively. Our VirConv-L runs at **56ms** with competitive precision.

2. Related Work

LiDAR-based 3D object detection. LiDAR-based 3D object detection has been widely studied in recent years. Early methods project the point clouds into a 2D Bird's eye view (BEV) or range view images [2, 4] for 3D detection. Recently, voxel-based sparse convolution [7, 13, 19, 39] and point-based set abstraction [26,27,42,43] have become popular in designing effective detection frameworks. However, the scanning resolution of LiDAR is generally very low for distant objects. The LiDAR-only detectors usually suffer from such sparsity. This paper addresses this problem by introducing RGB image data in a form of virtual points.

Multimodal 3D object detection. The RGB image and LiDAR data can complement each other and usually boost 3D detection performance. Early methods extend the features of LiDAR points with image features [29–31]. Some works encode the feature of two modalities independently and fuse the two features in the local Region of Interest (RoI) [5, 18] or BEV plane [21]. We follow the recent work that fuses the two data via virtual points [36,45]. The virtual points explicitly complete the geometry of distant objects

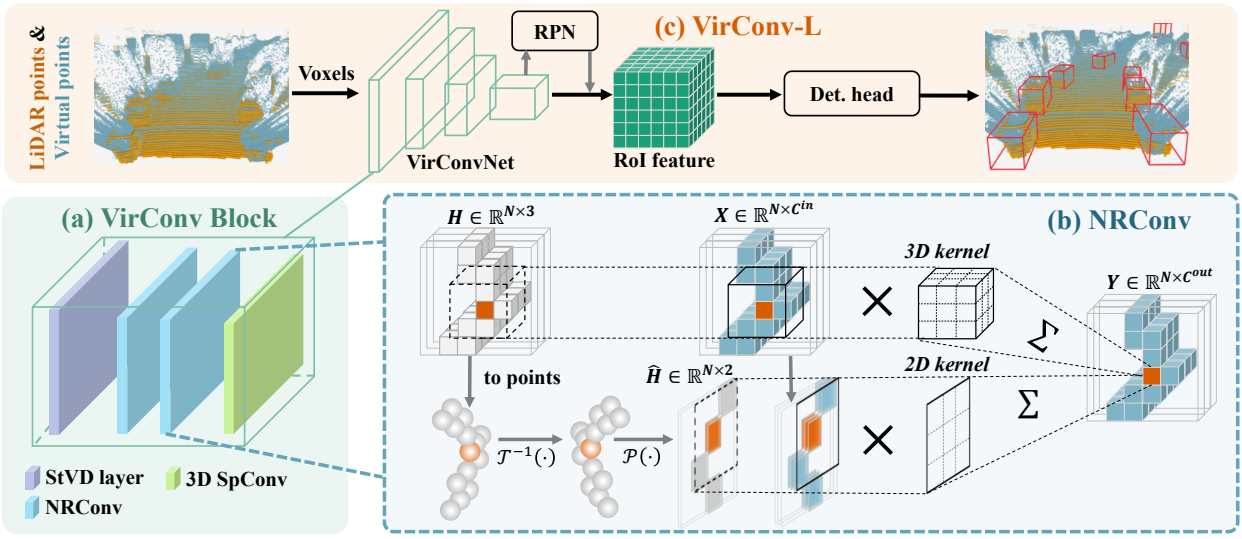

Figure 3. (a) VirConv block consists of a StVD layer, some NRConv layers and a 3D SpConv layer. (b) NRConv projects the voxels back to image space, and encodes virtual point features in both 2D and 3D space. (c) VirConv-L fuses the LiDAR points and the virtual points into a single point cloud, and encodes the multimodal features by our VirConvNet for 3D detection.

by depth estimation, showing the great potential for high-performance 3D detection. But virtual points are extremely dense and often noisy. This paper addresses these problems through two new schemes, StVD and NRConv, respectively.

3D object detection with re-sampled point clouds. The points captured by LiDAR are generally dense and unevenly distributed. Previous work speeds up the network by using a larger voxel size [19, 44] or by randomly downsampling [17] the point clouds. However, applying these methods to the virtual points will significantly decrease the useful geometry cues, especially for the faraway objects. Different from that, our StVD retains all the useful faraway voxels and speeds up the network by discarding nearby redundant voxels.

Noise handling in 3D vision. Traditional methods handle the noises by filtering algorithm [8, 11, 12]. Recently, score-based [22] and semantic segmentation networks [15] are developed for point cloud noise removal. Different from the traditional noises that are randomly distributed in 3D space, the noises brought by inaccurate depth completion are mostly distributed on 2D instance boundaries. Although the noise can be roughly removed by some 2D edge detection method [14], this will sacrifice the useful boundary points of object. We design a new scheme, NRConv, that extends the receptive field of 3D sparse convolution to 2D image space, distinguishing the noise pattern without the loss of useful boundary points.

Semi-supervised 3D object detection. Recent semi-supervised methods boost 3D object detection by a large amount of unlabeled data. Inspired by the pseudo-label-based framework [33, 37, 47], we also constructed a VirConv-S pipeline to perform semi-supervised multimodal 3D object detection.

3. VirConv for Multimodal 3D Detection

This paper proposes VirConvNet, based on a new VirConv operator, for virtual-point-based multimodal 3D object detection. As shown in Fig. 3, VirConvNet first converts points into voxels, and gradually encodes voxels into feature volumes by a series of VirConv block with $1\times$, $2\times$, $4\times$ and $8\times$ downsampling strides. The VirConv block consists of three parts (see Fig. 3 (a)): (1) an StVD layer for speeding up the network and improving density robustness; (2) multiple NRConv layers for encoding features and decreasing the impact of noise; (3) a 3D SpConv layer for down-sampling the feature map. Based on the VirConv operator, we build three detectors for efficient, accurate, and semi-supervised multimodal 3D detection, respectively.

3.1. Virtual Points for Data Fusion

Many recent 3D detectors use virtual points [45] (pseudo points [36]) generated from an image by depth completion algorithms to fuse RGB and LiDAR data. We denote the LiDAR points and virtual points as $\mathbf{P}$ and $\mathbf{V}$, respectively. Recently, two popular fusion schemes have been applied for 3D object detection: (1) early fusion [45], which fuses $\mathbf{P}$ and $\mathbf{V}$ into a single point cloud $\mathbf{P}^*$ and performs 3D object detection using existing detectors, and (2) late fusion [36], which encodes the features of $\mathbf{P}$ and $\mathbf{V}$ by different backbone networks and fuses the two types of features in BEV plane or local RoI. However, both fusion methods suffer

from the dense and noisy nature of virtual points.

(1) Density problem. As motioned in Section 1, the virtual points are usually very dense. They introduce a huge computational burden, which significantly decreases the detection speed (e.g., more than **2**$\times$ in Fig. 2 (f)). Existing work tackles the density issue by using a larger voxel size [19] or by randomly down-sampling [17] the points. But these methods will inevitably sacrifice the shape cues from the virtual points, especially for the faraway object. Based on a pilot experiment on the KITTI dataset [9] using the Voxel-RCNN [7] with a late fusion, we observed that a huge number of virtual points introduced for nearby objects are redundant. Specifically, **97%** of virtual points from the nearby objects bring only a **0.18%** performance improvement, while **3%** of virtual points for the faraway objects bring a **2.2%** performance improvement. The reason is that the geometry of nearby objects is relatively complete for LiDAR points. Such virtual points generally bring marginal performance gain but increase unnecessary computation. Motivated by this observation, we design an StVD (Stochastic Voxel Discard) scheme, which alleviates the computation problem by discarding nearby redundant voxels. In addition, the points of the distant object are much sparser than the nearby objects (see Fig. 2 (e)). The StVD can simulate sparser training samples to improve detection robustness.

(2) Noise problem. The virtual points generated by the depth completion network are usually noisy. An example is shown in Fig. 2 (c). The noise is mostly introduced by the inaccurate depth completion, and is hardly distinguishable in 3D space. By using only virtual points, the detection performance drops $\sim$9% AP compared with the LiDAR-only detector (see Fig. 2 (f)). In addition, the noisy points are non-Gaussian distributed, and cannot be filtered by traditional denoising algorithms [8, 12]. We observed that noise is mainly distributed on the instance boundaries (see Fig. 2 (d)) and can be more easily recognized in 2D images. Although the edge detection [14] could be applied here to roughly remove the noise, this will sacrifice the useful boundary points which are beneficial to the object's shape and position estimation. Our idea is to extend the receptive field of sparse convolution to the 2D image space, and distinguish the noise without the loss of shape cues.

3.2. Stochastic Voxel Discard

To alleviate the computation problem and improve the density robustness for the virtual-point-based detector, we develop the StVD. It consists of two parts: (1) input StVD, which speeds up the network by discarding input voxels of virtual points during both the training and inference process; (2) layer StVD, which improves the density robustness by discarding voxels of virtual points at every VirConv block during only the training process.

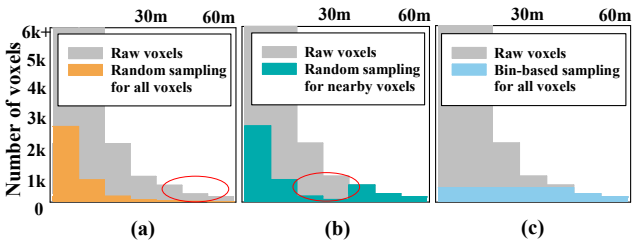

Figure 4. (a)(b) show the voxel distributions after random sampling for all and nearby voxels, respectively. (c) shows the voxel distribution after bin-based sampling for all voxels.

Input StVD. Two naive methods can keep less input voxels: (1) random sampling or (2) farthest point sampling (FPS). However, the random sampling usually keeps unbalanced voxels at different distances and inevitably sacrifices some useful shape cues (in the red region at Fig. 4 (a) (b)). In addition, FPS needs huge extra computation when down-sampling the huge number of virtual points due to the high computational complexity ($O(n^2)$). To tackle this problem, we introduce a bin-based sampling strategy to perform efficient and balanced sampling (see Fig. 4 (c)). Specifically, We first divide the input voxels into N^b bins (we adopt $N^b = 10$ in this paper) according to different distances. For the nearby bins ($\leq$30m based on the statistics in Fig. 2 (e)), we randomly keep a fixed number ($\sim$ 1K) of voxels. For distant bins, we keep all of the inside voxels. After the bin-based sampling, we discard about **90%** (which achieves the best precision-efficiency trade-off, see Fig. 6) of redundant voxels and it speeds up the network by about **2 times**.

Layer StVD. To improve the robustness of detection from sparse points, we also develop a layer StVD which is applied to the training process. Specifically, we discard voxels at each VirConv block to simulate sparser training samples. We adopt a discarding rate of 15% in this paper (the layer StVD rate is discussed in Fig. 6). The layer StVD serves as a data augmentation strategy to help enhance the 3D detector's training.

3.3. Noise-Resistant Submanifold Convolution

As analyzed in Section 3.1, the noise introduced by the inaccurate depth completion can hardly be recognized from 3D space but can be easily recognized from 2D images. We develop an NRConv (see Fig. 3 (b)) from the widely used submanifold sparse convolution [10] to address the noise problem. Specifically, given N input voxels formulated by a 3D indices vector $\mathbf{H} \in \mathbb{R}^{N\times 3}$ and a features vector $\mathbf{X} \in \mathbb{R}^{N\times C^{in}}$, we encode the noise-resistant geometry features $\mathbf{Y} \in \mathbb{R}^{N\times C^{out}}$ in both 3D and 2D image space, where C^{in} and C^{out} denote the number of input and output feature channels respectively.

Encoding geometry features in 3D space. For each

voxel feature X_i in $\mathbf{X}$, we first encode the geometry features by the 3D submanifold convolution kernel $\mathcal{K}^{3D}(\cdot)$. Specifically, the geometry features $\hat{X}_i \in \mathbb{R}^{C^{out}/2}$ are calculated from the non-empty voxels within $3 \times 3 \times 3$ neighborhood based on the corresponding 3D indices as

$$\hat{X}_i = \mathcal{R}\left\{\mathcal{K}^{3D}\left(X_i, X_i^{(f_1)}, ..., X_i^{(f_j)}\right)\right\}, \quad (1)$$

where $X_i^{(f_1)}, ..., X_i^{(f_j)}$ denote neighbor features generated by $\mathbf{H}$, and $\mathcal{R}$ denotes the nonlinear activation function.

Encoding noise-aware features in 2D image space. The noise brought by the inaccurate depth completion significantly degrade the detection performance. Since the noise is mostly distributed on the 2D instance boundaries, we extend the convolution receptive field to the 2D image space and encode the noise-aware features using the 2D neighbor voxels. Specifically, we first convert the 3D indices to a set of grid points based on the voxelization parameters (the conversion denoted as $\mathcal{G}(\cdot)$). Since state-of-the-art detectors [7, 36] also adopt the transformation augmentations (the augmentation denoted as $\mathcal{T}(\cdot)$) such as rotation and scaling, the grid points are generally misaligned with the corresponding image. Therefore, we transform the grid points backward into the original coordinate system based on the data augmentation parameters. Then we project the grid points into the 2D image plane based on the LiDAR-Camera calibration parameters (with the projection denoted as $\mathcal{P}(\cdot)$). The overall projection can be summarized by

$$\hat{\mathbf{H}} = \mathcal{P}\left(\mathcal{T}^{-1}\left(\mathcal{G}\left(\mathbf{H}\right)\right)\right), \quad (2)$$

where $\hat{\mathbf{H}} \in \mathbb{R}^{N \times 2}$ denotes the 2D indices vector. For each voxel feature $X_i \in \mathbb{R}^{C^{in}}$, we then calculate the noise-aware features $\tilde{X}_i \in \mathbb{R}^{C^{out}/2}$ from the non-empty voxels within a 3×3 neighborhood based on the corresponding 2D indices.

$$\tilde{X}_i = \mathcal{R}\left\{\mathcal{K}^{2D}\left(X_i, \tilde{X}_i^{(f_1)}, ..., \tilde{X}_i^{(f_k)}\right)\right\}, \quad (3)$$

where $\tilde{X}_i^{(f_1)}, ..., \tilde{X}_i^{(f_k)}$ denote the neighbor voxel features generated by $\hat{\mathbf{H}}$, and $\mathcal{K}^{2D}(\cdot)$ denote the 2D submanifold convolution kernel. If there are multiple features in a single 2D neighbor voxel, we perform max-pooling and keep one feature in each voxel to perform the 2D convolution.

After the 3D and 2D features encoding, we adopt a simple concatenation to implicitly learn a noise-resistant feature. Specifically, we finally concatenate $\hat{X}_i$ and $\tilde{X}_i$ to obtain the noise-resistant feature vector $\mathbf{Y} \in \mathbb{R}^{N \times C^{out}}$ as

$$\mathbf{Y} = \left[\left[\hat{X}_i, \tilde{X}_i\right]^T, ..., \left[\hat{X}_N, \tilde{X}_N\right]^T\right]^T. \quad (4)$$

Different from related noise segmentation and removal [15] methods, our NRConv implicitly distinguishes the noise pattern by extending the receptive field to 2D image space. Consequently, the impact of noise is suppressed without lose of shape cues.

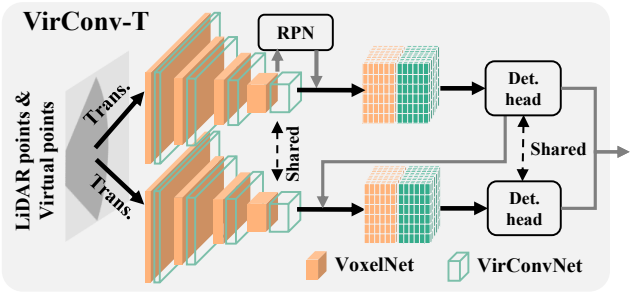

Figure 5. Transformed refinement scheme. The inputs are first transformed with different rotations and reflections. Then, VoxelNet and VirConvNet encode the LiDAR and virtual points features, respectively. Next, RoIs are generated and refined by the backbone features under different transformations. At last, the refined RoIs from different stages are fused by boxes voting [34].

3.4. Detection Frameworks with VirConv

To demonstrate the superiority of our VirConv, we constructed VirConv-L, VirConv-T and VirConv-S from the widely used Voxel-RCNN [7] for fast, accurate and semi-supervised 3D object detection, respectively.

VirConv-L. We first construct the lightweight VirConv-L (Fig. 3 (c)) for fast multimodal 3D detection. VirConv-L adopts an early fusion scheme and replaces the backbone of Voxel-RCNN with our VirConvNet. Specifically, we denote the LiDAR points as $\mathbf{P} = \{p\}, p = [x, y, z, \alpha]$, where x, y, z denotes the coordinates and α refers intensity. We denote the virtual points as $\mathbf{V} = \{v\}, v = [x, y, z]$. We fuse them into a single point cloud $\mathbf{P}^* = \{p^*\}, p^* = [x, y, z, \alpha, \beta]$, where β is an indicator denoting where the point came from. The intensity of virtual points is padded by zero. The fused points are encoded into feature volumes by our VirConvNet for 3D detection.

VirConv-T. We then construct a high-precision VirConv-T based on a Transformed Refinement Scheme (TRS) and a late fusion scheme (see Fig. 5). CasA [34] and TED [35] achieve high detection performance based on three-stage refinement and multiple transformation design, respectively. However, both of them require heavy computations. We fuse the two high computation detectors into a single efficient pipeline. Specifically, we first transform $\mathbf{P}$ and $\mathbf{V}$ with different rotations and reflections. Then we adopt the VoxelNet [7] and VirConvNet to encode the features of $\mathbf{P}$ and $\mathbf{V}$, respectively. Similar to TED [35], the convolutional weights between different transformations are shared. After that, the RoIs are generated by a Region Proposal Network (RPN) [7] and refined by the backbone features (the RoI features of $\mathbf{P}$ and $\mathbf{V}$ fused by simple concatenation) under the first transformation. The refined RoIs are further refined by the backbone features under other transformations. Next, the

Method	Reference	Modality	Car 3D AP (R40)			Car BEV AP (R40)			Time (ms)
			Easy	Mod.	Hard	Easy	Mod.	Hard	
PV-RCNN [26]	CVPR 2020	LiDAR	90.25	81.43	76.82	94.98	90.65	86.14	80*
Voxel-RCNN [7]	AAAI 2021	LiDAR	90.90	81.62	77.06	94.85	88.83	86.13	40
CT3D [25]	ICCV 2021	LiDAR	87.83	81.77	77.16	92.36	88.83	84.07	70*
SE-SSD [48]	CVPR 2021	LiDAR	91.49	82.54	77.15	95.68	91.84	86.72	**30**
BtcDet [38]	AAAI 2022	LiDAR	90.64	82.86	78.09	92.81	89.34	84.55	90
CasA [34]	TGRS 2022	LiDAR	91.58	83.06	80.08	95.19	91.54	86.82	86
Graph-Po [40]	ECCV 2022	LiDAR	91.79	83.18	77.98	95.79	92.12	87.11	60
F-PointNet [24]	CVPR 2018	LiDAR+RGB	82.19	69.79	60.59	91.17	84.67	74.77	170*
UberATG-MMF [20]	CVPR 2019	LiDAR+RGB	88.40	77.43	70.22	93.67	88.21	81.99	80
3D-CVF [46]	ECCV 2020	LiDAR+RGB	89.20	80.05	73.11	93.52	89.56	82.45	75
Focals Conv [6]	CVPR 2022	LiDAR+RGB	90.55	82.28	77.59	92.67	89.00	86.33	100*
VPFNet [50]	TMM 2022	LiDAR+RGB	91.02	83.21	78.20	93.94	90.52	86.25	62
Graph-VoI [40]	ECCV 2022	LiDAR+RGB	91.89	83.27	77.78	95.69	90.10	86.85	76
SFD [36]	CVPR 2022	LiDAR+RGB	91.73	84.76	77.92	95.64	91.85	86.83	98
VirConv-L (Ours)	-	LiDAR+RGB	91.41	85.05	80.22	95.53	91.95	87.07	56
VirConv-T (Ours)	-	LiDAR+RGB	**92.54**	**86.25**	**81.24**	**96.11**	**92.65**	**89.69**	92
VirConv-S (Our semi-supervised)	-	LiDAR+RGB	92.48	87.20	82.45	95.99	93.52	90.38	92

Table 1. Car 3D detection results on the KITTI test set, where the best fully supervised methods are in bold and $*$ denotes that the result is from the KITTI leaderboard. Our VirConv-T outperforms all the other methods in both 3D AP and BEV AP metrics. Besides, our VirConv-L runs fast at 56ms with 85.05 AP, and our VirConv-S attains a high detection performance of 87.20 AP.

refined RoIs from different refinement stages are fused by boxes voting, as is done by CasA [34]. We finally perform a non-maximum-suppression (NMS) on the fused RoIs to obtain detection results.

VirConv-S. We also design a semi-supervised pipeline, VirConv-S, using the widely used pseudo-label method [33, 41]. Specifically, first, a model is pre-trained using the labeled training data. Then, pseudo labels are generated on a larger-scale unannotated dataset using this pre-trained model. A high-score threshold (empirically, 0.9) is adopted to filter out low-quality labels. Finally, the VirConv-T model is trained using both real and pseudo labels.

4. Experiments

4.1. KITTI Datasets and Evaluation Metrics

The KITTI 3D object detection dataset [9] contains 7,481 and 7,518 LiDAR and image frames for training and testing, respectively. We divided the training data into a train split of 3712 frames and a validation split of 3769 frames following recent works [7, 34]. We also adopted the widely used evaluation metric: 3D Average Precision (AP) under 40 recall thresholds (R40). The IoU thresholds in this metric are 0.7, 0.5, and 0.5 for car, pedestrian, and cyclist, respectively. We used the KITTI odometry dataset [9] as the large-scale unlabeled dataset. The KITTI odometry dataset contains 43,552 LiDAR and image frames. We uniformly sampled 10,888 frames (denoted as the *semi* dataset) and used them to train our VirConv-S. There is no overlap found between the KITTI 3D detection dataset and the KITTI odometry dataset after checking the mapping files released by KITTI.

4.2. Setup Details

Network details. Similar to SFD, our method uses the virtual points generated by PENet [16]. VirConvNet adopts an architecture similar to the Voxel-RCNN backbone [7]. VirConvNet includes four levels of VirConv blocks with feature dimensions 16, 32, 64, and 64, respectively. The input StVD rate and layer StVD rate are set to 90% and 15% by default. On the KITTI dataset, all the detectors use the same detection range and voxel size as CasA [34].

Losses and data augmentation. VirConv-L uses the same training loss as in [7]. VirConv-T and VirConv-S use the same training loss as CasA [34]. In all these three pipelines, we adopted the widely used local and global data augmentation [27, 34, 36], including ground-truth sampling, local transformation (rotation and translation), and global transformation (rotation and flipping).

Training and inference details. All three detectors were trained on 8 Tesla V100 GPUs with the ADAM optimizer. We used a learning rate of 0.01 with a one-cycle learning rate strategy. We trained the VirConv-L and VirConv-T for 60 epochs. The weights of VirConv-S are initialized by the trained VirConv-T. We further trained the VirConv-S on the labeled and unlabeled dataset for 10 epochs. We used an NMS threshold of 0.8 to generate 160 object proposals with 1:1 positive and negative samples during training. During testing, we used an NMS threshold of 0.1 to remove redundant boxes after proposal refinement.

4.3. Main Results

Results on KITTI validation set. We report the car detection results on the KITTI validation set in Table 2. Compared with the baseline detector Voxel-RCNN [7], our VirConv-L, VirConv-T and VirConv-S show 3.42%, 5% and 5.68% 3D AP(R40) improvement in the moderate car class, respectively. We also reported the performance based on the 3D AP under 11 recall thresholds (R11). Our VirConv-L, VirConv-T and VirConv-S show 2.38%, 3.33% and 3.54% 3D AP(R11) improvement in the moderate car class, respectively. The performance gains are mostly derived from the VirConv design, which effectively addressed the density problem and noise problem brought by virtual points. Note that our VirConv-L also runs much faster than other multimodal detectors, thanks to our efficient StVD design.

Method	Car 3D AP (R40)			Mod.
	Easy	Mod.	Hard	AP(R11)
Voxel-RCNN	92.38	85.29	82.86	84.52
Voxel-RCNN(EF)	92.42	85.78	83.10	84.94
Voxel-RCNN(LF)	92.91	86.32	83.97	85.23
VirConv-L	93.36	88.71	85.83	86.70
VirConv-T	**95.81**	**90.29**	**88.10**	**87.82**
VirConv-S (semi)	95.76	90.97	89.14	88.06

Table 2. 3D car detection results on the KITTI validation set, where EF and LF denote early fusion and late fusion, respectively.

Results on KITTI test set. The experimental results on the KITTI test set are reported in Table 1. Our VirConv-L, VirConv-T, and VirConv-S outperform the baseline Voxel-RCNN [7] by 3.43%, 4.63% and 5.58% 3D AP (R40) in the moderate car class, respectively. The VirConv-L, VirConv-T, and VirConv-S also outperform the best previous 3D detector SFD [36] by 0.29%, 1.49%, and 2.44%, respectively. As of the date of the CVPR deadline (Nov.11, 2022), our VirConv-T and VirConv-S rank 2nd and 1st, respectively, on the KITTI 3D object detection leaderboard. The results further demonstrate the effectiveness of our method.

4.4. Ablation Study

VirConv performance with different fusion schemes. Virtual points only, early fusion, and late fusion are three potential choices for virtual points-based 3D object detection. To investigate the performance of VirConv under these three settings, we first constructed three baselines: Voxel-RCNN [7] with only virtual points, Voxel-RCNN [7] with early fusion, and Voxel-RCNN [7] with late fusion. Then we replaced the backbone of Voxel-RCNN with our VirConvNet. The experimental results on the KITTI validation set are shown in Table 3. With our VirConv, the 3D AP has significantly improved by 3.43%, 2.93%, and 2.65%, under virtual points only, early fusion, and late fusion settings, respectively. Meanwhile, the efficiency significantly improves. This is because VirConv speeds up the network with the StVD design and decreases the noise impact with the NRConv design.

Setting	VirConv	TRS	3D AP	Time (ms)
LiDAR points	No		85.29	38
Virtual points	No		76.12	84
	Yes		79.55	**52**
	Yes	✓	**80.91**	71
Early fusion	No		85.78	88
	Yes		88.71	**56**
	Yes	✓	**88.96**	76
Late fusion	No		86.32	120
	Yes		88.97	**74**
	Yes	✓	**90.29**	92

Table 3. Ablation results on the KITTI validation set by using different fusion scheme.

with StVD	with NRConv	3D AP			Time (ms)
		Easy	Mod.	Hard	
No	No	94.26	87.55	85.49	152
Yes	No	94.55	88.32	85.95	**87**
Yes	Yes	**95.81**	**90.29**	**88.10**	92

Table 4. Ablation results on the KITTI validation set by using different designed components.

Effectiveness of StVD. We next investigated the effectiveness of StVD. The results are shown in Table 4. With StVD, VirConv-T not only performs more accurate 3D detection but also runs faster by about 2×. The reason lies in that StVD discards about 90% of redundant voxels to speed up the network, and it also improves the detection robustness by simulating more sparse training samples.

Influence of StVD rate. We then conducted experiments to select the best input and layer StVD rate. The results are shown in Fig. 6. We observe that using a higher input StVD rate, the detection performance will decrease dramatically due to the geometry feature loss. On the contrary, using a lower input StVD rate, the efficiency is degraded with poor AP improvement. We found that by randomly discarding 90% of nearby voxels, we achieve the best accuracy-efficiency trad-off. Therefore, this paper adopts an input StVD rate of 90%. Similarly, by using a 15% layer StVD rate, we achieved the best detection accuracy.

Effectiveness of NRConv. We then investigated the effects of NRConv using VirConv-T. The results are shown in Table 4. With our NRConv, the car detection AP of VirConv-T improves from 88.32% to 90.29%. Since the NRConv encodes the voxel features in both 3D and 2D image space, reducing the noise impact brought by the inaccurate depth completion, the detection performance is significantly improved.

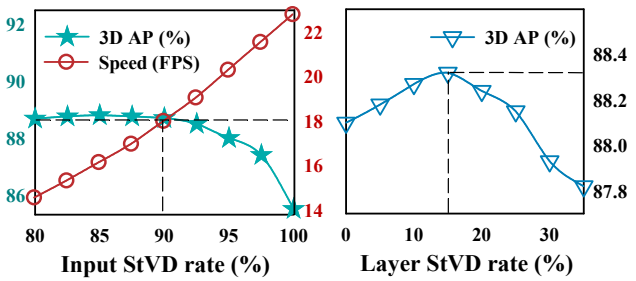

Figure 6. Left: precision and speed trade-off by using different Input StVD rate. Right: detection performance by using different layer StVD rate.

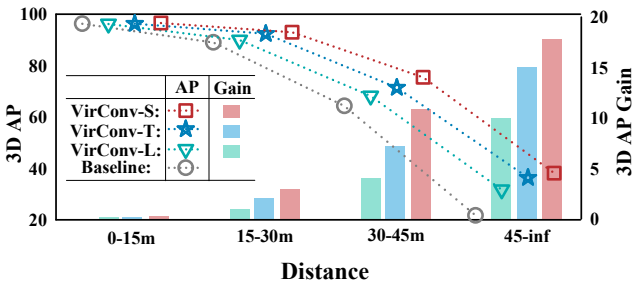

Figure 7. 3D AP and performance improvement along different detection distance (KITTI validation set).

Effectiveness of TRS. We conducted experiments to examine the effect of TRS in VirConv-T. The results are shown in Table 3. With our TRS, detectors show 1.36%, 0.25%, and 1.32% performance improvement under virtual points only, early fusion, and late fusion, respectively. The performance gain is derived from the two-transform and two-stage refinement, which improves the transformation robustness and leads to better detection performance.

Class	Method	3D AP		
		Easy	Mod.	Hard
Car	Baseline	89.39	83.83	87.73
	VirConv-T	**94.98**	**89.96**	**88.13**
Pedestrian	Baseline	70.55	62.92	57.35
	VirConv-T	**73.32**	**66.93**	**60.38**
Cyclist	Baseline	89.86	71.13	66.67
	VirConv-T	**90.04**	**73.90**	**69.06**

Table 5. 3D Detection results (3D AP (R40)) of multi-class VirConv-T (KITTI validation set).

Multi-class performance. We also trained a multi-class VirConv-T to detect car, pedestrian and cyclist class instances using a single model. We reported the multi-class 3D object detection performance in Table 5, where the baseline refers to the multi-class Voxel-RCNN [7]. Compared with the baseline, the detection performance of VirConv-T in all classes has been significantly improved. The results demonstrate that our VirConv can be easily generalized to a multi-class model and boost the detection performance.

Performance breakdown. To investigate where our model improves the baseline most, we evaluate the detection performance based on the different distances. The results are shown in Fig. 7. Our three detectors have significant improvements for faraway objects because our VirConv models better geometry features of distant sparse objects from the virtual points.

Evaluation on nuScenes test set. To demonstrate the universality of our method, we conducted an experiment on the nuScenes [3] dataset. we compared our method with CenterPoint + VP (virtual point), TransFuison-L + VP and TransFusion. We adopted the same data augmentation strategy as TransFuison-L and trained the network for 30 epochs on 8 Tesla V100 GPUs. The results on the nuScenes test set are shown in Table 6. With VirConv, the detection performance of CenterPoint + VP and TransFuison-L + VP has been significantly improved. In addition, the TransFusion-L with VirConv even surpasses the TransFusion in terms of NDS, demonstrating that our model is able to boost the virtual point-based detector significantly.

Method	mAP	NDS
CenterPoint + VP [45]	66.4	70.5
CenterPoint + VP + VirConv	**67.2**	**71.2**
TransFusion [1]	**68.9**	71.7
TransFusion-L+VP	66.7	70.8
TransFusion-L +VP + VirConv	68.7	**72.3**

Table 6. 3D detection results on the nuScenes test set.

5. Conclusion

This paper presented a new VirConv operator for virtual-point-based multimodal 3D object detection. VirConv addressed the density and noise problems of virtual points through the newly designed Stochastic Voxel Discard and Noise-Resistant Submanifold Convolution mechanisms. Built upon VirConv, we presented VirConv-L, VirConv-T, and VirConv-S for efficient, accurate, and semi-supervised 3D detection, respectively. Our VirConvNet holds the leading entry on both KITTI car 3D object detection and BEV detection leaderboards, demonstrating the effectiveness of our method.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (No.62171393), the Fundamental Research Funds for the Central Universities (No.20720220064) and the FuXiaQuan National Independent Innovation Demonstration Zone Collaborative Innovation Platform (No.3502ZCQXT2021003).

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 8

[2] Jorge Beltrán, Carlos Guindel, Francisco Miguel Moreno, Daniel Cruzado, Fernando Garcia, and Arturo De La Escalera. Birdnet: a 3d object detection framework from lidar information. In *ITSC*, pages 3517–3523. IEEE, 2018. 2

[3] 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 (CVPR)*, 2020. 2, 8

[4] 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 (CVPR)*, pages 1907–1915, 2017. 1, 2

[5] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3d: A unified sensor fusion framework for 3d detection. *ArXiv*, 2022. 2

[6] Yukang Chen, Yanwei Li, X. Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3d object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition CVPR)*, 2022. 1, 6

[7] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wen gang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. 2, 4, 5, 6, 7, 8

[8] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. Bilateral mesh denoising. *ACM SIGGRAPH 2003 Papers*, 2003. 2, 3, 4

[9] 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 (CVPR)*, pages 3354–3361, 2012. 1, 2, 4, 6

[10] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9224–9232, 2018. 4

[11] Xian-Feng Han, Jesse S. Jin, Mingjie Wang, and Wei Jiang. Guided 3d point cloud filtering. *Multimedia Tools and Applications*, 77:17397–17411, 2017. 3

[12] Xian-Feng Han, Jesse S. Jin, Mingjie Wang, Wei Jiang, Lei Gao, and Liping Xiao. A review of algorithms for filtering the 3d point cloud. *Signal Process. Image Commun.*, 57:103–112, 2017. 2, 3, 4

[13] 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 (CVPR)*, pages 11873–11882, 2020. 2

[14] Jianzhong He, Shiliang Zhang, Ming Yang, Yanhu Shan, and Tiejun Huang. Bi-directional cascade network for perceptual edge detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 4

[15] Robin Heinzler, Florian Piewak, Philipp Schindler, and Wilhelm Stork. Cnn-based lidar point cloud de-noising in adverse weather. *IEEE Robotics and Automation Letters*, 2020. 2, 3, 5

[16] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. Penet: Towards precise and efficient image guided depth completion. In *In International Conference on Robotics and Automation (ICRA)*, pages 13656–13662, 2021. 1, 6

[17] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Agathoniki Trigoni, and A. Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3, 4

[18] 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 (IROS)*, pages 1–8, 2018. 2

[19] 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 (CVPR)*, pages 12697–12705, 2019. 1, 2, 3, 4

[20] Ming Liang, Binh 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 (CVPR)*, pages 7337–7345, 2019. 1, 6

[21] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *ArXiv*, 2022. 1, 2

[22] Shitong Luo and Wei Hu. Score-based point cloud denoising. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[23] Jiageng Mao, Minzhe Niu, Haoyue Bai, Xiaodan Liang, Hang Xu, and Chunjing Xu. Pyramid r-cnn: Towards better performance and adaptability for 3d object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 1

[24] C. Qi, W. 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 (CVPR)*, pages 918–927, 2018. 1, 6

[25] Hualian Sheng, Sijia Cai, Yuan Liu, Bing Deng, Jianqiang Huang, Xiansheng Hua, and Min-Jian Zhao. Improving 3d object detection with channel-wise transformer. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 6

[26] 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 (CVPR)*, pages 10526 – 10535, 2020. 2, 6

[27] 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 (CVPR)*, pages 770–779, 2019. 1, 2, 6

[28] 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 (TPAMI)*, 43:2647–2664, 2021. 1

[29] Vishwanath A. Sindagi, Yin Zhou, and Oncel Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. In *In International Conference on Robotics and Automation (ICRA)*, 2019. 1, 2

[30] Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[31] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[32] Dequan Wang, Coline Devin, Qi-Zhi Cai, Philipp Krähenbühl, and Trevor Darrell. Monocular plan view networks for autonomous driving. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019. 1

[33] He Wang, Yezhen Cong, Or Litany, Yue Gao, and Leonidas J. Guibas. 3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition CVPR)*, pages 14610–14619, 2021. 2, 3, 6

[34] Hai Wu, Jinhao Deng, Chenglu Wen, Xin Li, and Cheng Wang. Casa: A cascade attention network for 3d object detection from lidar point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 2022. 2, 5, 6

[35] Hai Wu, Chenglu Wen, Wei Li, Ruigang Yang, and Cheng Wang. Learning transformation-equivariant features for 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. 2, 5

[36] Xiaopei Wu, Liang Peng, Honghui Yang, Liang Xie, Chenxi Huang, Chengqi Deng, Haifeng Liu, and Deng Cai. Sparse fuse dense: Towards high quality 3d detection with depth completion. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 3, 5, 6, 7

[37] Hongyi Xu, Feng Liu, Qianyu Zhou, Jinkun Hao, Zhijie Cao, Zhengyang Feng, and Lizhuang Ma. Semi-supervised 3d object detection via adaptive pseudo-labeling. *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3183–3187, 2021. 3

[38] Qiangeng Xu, Yiqi Zhong, and Ulrich Neumann. Behind the curtain: Learning occluded shapes for 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 6

[39] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18, 2018. 2

[40] Honghui Yang, Zili Liu, Xiaopei Wu, Wenxiao Wang, Wei Qian, Xiaofei He, and Deng Cai. Graph r-cnn: Towards accurate 3d object detection with semantic-decorated local graph. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 6

[41] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 6

[42] 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 (CVPR)*, pages 11040–11048, 2020. 1, 2

[43] 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 (ICCV)*, pages 1951–1960, 2019. 1, 2

[44] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 3

[45] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Multimodal virtual point 3d detection. In *NeurIPS*, volume abs/2111.06881, 2021. 1, 2, 3, 8

[46] Jin Hyeok Yoo, Yeocheol Kim, Ji Song Kim, and Jun Won Choi. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 6

[47] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Sess: Self-ensembling semi-supervised 3d object detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[48] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Se-ssd: Self-ensembling single-stage object detector from point cloud. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14494–14503, 2021. 6

[49] Dingfu Zhou, Jin Fang, Xibin Song, Chenye Guan, Junbo Yin, Yuchao Dai, and Ruigang Yang. Iou loss for 2d/3d object detection. *3DV*, pages 85–94, 2019. 1

[50] Hanqi Zhu, Jiajun Deng, Yu Zhang, Jianmin Ji, Qi-Chao Mao, Houqiang Li, and Yanyong Zhang. Vpfnet: Improving 3d object detection with virtual point based lidar and stereo data fusion. *ArXiv*, 2021. 6