Progress Evaluation

Enhancing PV-RCNN++ for 3D Object Detection

University of Moratuwa

Department of Computer Science and Engineering

August 2025

Dulmith M.W.P. - 210151B

1 Introduction

1.1 Background

3D object detection in LiDAR point clouds has become a turning point of autonomous driving and robotics, where accurately localizing vehicles, pedestrians, and other objects is critical [1]. Unlike 2D images, raw point clouds are sparse and irregular, making direct application of standard 2D convolutional detectors ineffective [2].

1.1.1 Voxel and Pillar based Methods

Early methods tackle this by converting point clouds into regular grids. Voxel based approaches, such as VoxelNet [2] and SECOND [3], encode points in 3D voxels processed by 3D/2D CNNs. Pillar-based encodings like PointPillars [4], collapse vertical columns into "pillars" and apply 2D convolutions. These grid-based methods improve speed and exploit mature CNN pipelines but incur quantization errors and lose fine-grained localization accuracy.

1.1.2 Point-based Methods

In contrast, point-based methods such as PointNet and PointNet++ operate directly on raw points to preserve precise geometry, at the cost of high computational load. Hybrid approaches attempt to combine both representations: for example, F-PointNet and related methods fuse image based proposals with point processing but often only stack stages without deeply intertwining the voxel and point representations [5, 6].

1.1.3 PV-RCNN and PV-RCNN++

PV-RCNN [6] addresses this by deeply integrating voxel and point features. In its first stage, a sparse 3D CNN backbone (using sparse convolutions [3]) generates multi-scale volumetric features and high-quality 3D proposals. Voxel features are abstracted into a set of keypoints sampled from the raw cloud via farthest point sampling (FPS) [7], ensuring representative coverage. In stage two, each proposal's local neighborhood is abstracted back onto a regular grid (via RoI-grid pooling) to refine box regression and classification [6]. PV-RCNN thus exploits the global context of sparse CNNs and the local precision of point-based networks simultaneously [6, 8], achieving state-of-the-art 3D detection accuracy on benchmarks like KITTI and Waymo, albeit with substantial computational cost.

Recent work, PV-RCNN++ [1, 6], builds on this hybrid framework to improve efficiency and effectiveness. Its two key innovations are:

1. Sectorized proposal-centric keypoint sampling, which accelerates FPS by performing sampling in parallel radial sectors centered on each proposal, concentrating keypoints near objects without redundancy [1].

2. VectorPool local aggregation, which splits each point's local neighborhood into compact sub-voxels and encodes their features with separate weights. This vector representation reduces computation compared to the original set abstraction pooling [9, 10].

Together, these reduce runtime by approximately 3× while slightly boosting accuracy [1, 10]. PV-RCNN++ achieves real-time (∼10 FPS) 3D detection on large scenes (150 m range) on the Waymo Open Dataset with state-of-the-art performance [1].

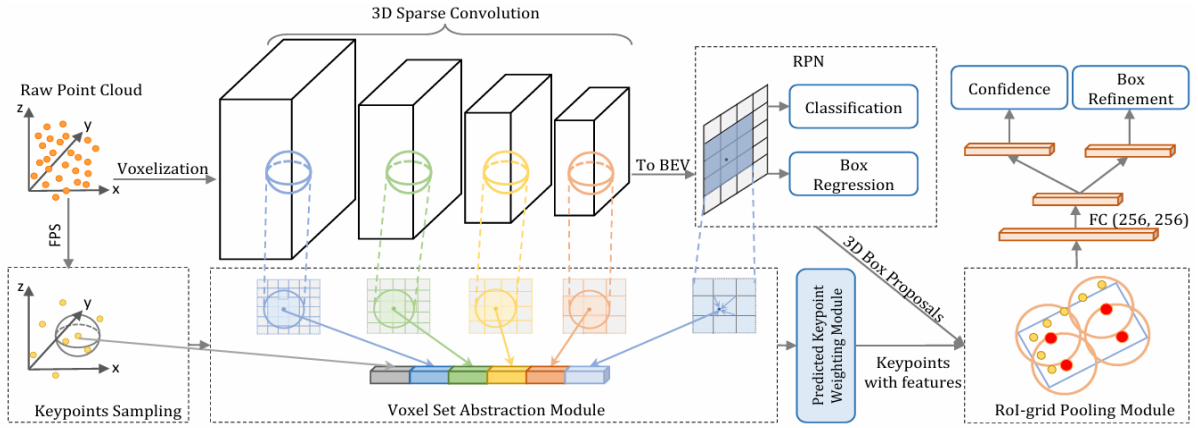


Figure 1: The overall architecture of the PV-RCNN. The raw point clouds are rst voxelized to feed into the 3D sparse convolution based encoder to learn multi-scale semantic features and generate 3D object proposals. Then the learned voxel-wise feature volumes at multiple neural layers are summarized into a small set of key points via the novel voxel set abstraction module. Finally the keypoint features are aggregated to the RoI-grid points to learn proposal speci c features for ne-grained proposal re nement and condence prediction.[1]

2 Research Objectives

1. Reproduce the PV-RCNN++ baseline on standard 3D detection datasets (KITTI, Waymo) to establish a performance benchmark.
2. Enhancing model performance through:
   (a) Adaptive keypoint sampling and representation
   (b) Attention or graph based local feature aggregation
   (c) Improved bounding box regression and classification losses (IoU-based, focal loss)
   (d) Architectural enhancements, including lightweight transformer blocks
3. Evaluate training strategies and augmentations, including ground-truth sampling, scaling/rotation, and cutout, to maximize generalization.
4. Provide a well-documented, reproducible implementation with ablation analysis and an accessible GitHub repository.

3 Literature Review

Research in 3D object detection has rapidly evolved along several axes: data representation (voxels, points, images), architecture (single stage vs. two stage, anchor vs. anchor

free), and learning methods (CNNs, attention, graph networks). We briefly review leading approaches relevant to PV-RCNN++ enhancements.

3.1 Voxel-based Detectors

VoxelNet [2] pioneered using a 3D CNN by partitioning space into equally-spaced voxels and encoding each with a Voxel Feature Encoding (VFE) layer, making LiDAR detection end to end trainable. SECOND [3] improved on VoxelNet by using sparse 3D convolutions, introducing efficient sparse convolutional modules and GPU rule-generation to accelerate both training and inference. It also developed LiDAR-specific augmentations, such as ground-truth sampling from a database [3] and angle-regression techniques to better estimate object orientation [8]. Voxel R-CNN (2020) and other works further refine the proposal pipeline but share voxel grid representations. Voxel Transformer (VoTr) [7] replaces parts of the sparse CNN backbone with self attention: "sparse" and "submanifold" modules process non empty voxels with local and dilated attention. VoTr captures longer range context and shows consistent improvement over pure CNN backbones on KITTI and Waymo [7].

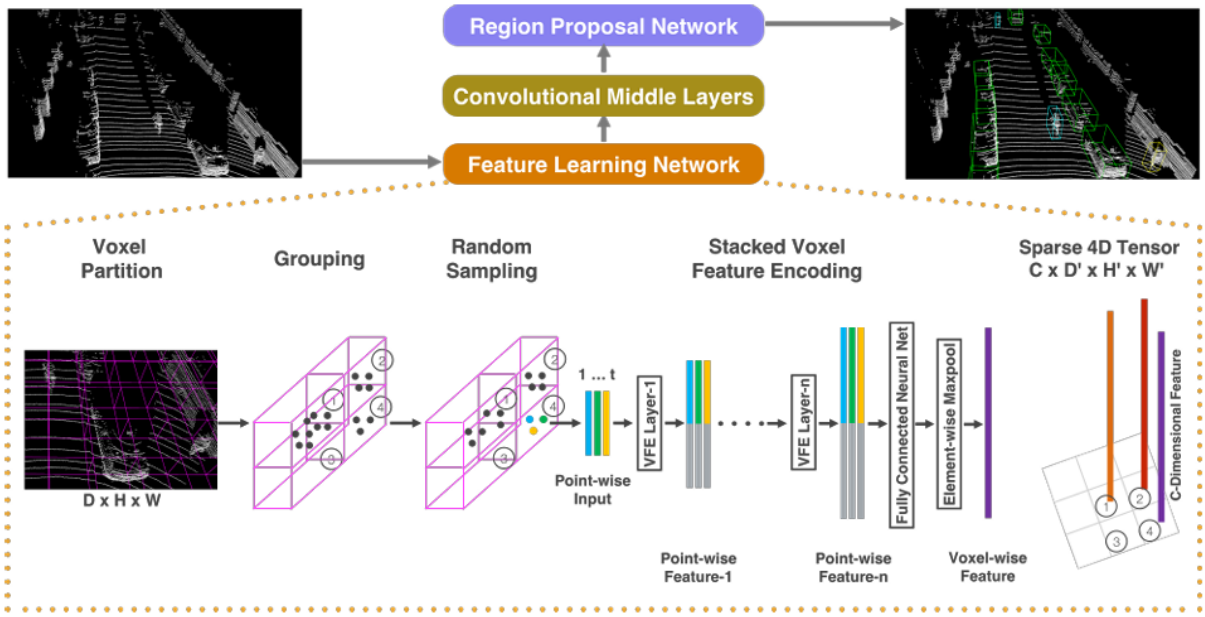


Figure 2: VoxelNet architecture. The feature learning network takes a raw point cloud as input, partitions the space into voxels, and transforms points within each voxel to a vector representation characterizing the shape information. The space is represented as a sparse 4Dtensor. The convolutional middle layers processes the 4D tensor to aggregate spatial context. Finally, a RPN generates the 3D detection.[2]

3.2 Point-based Detectors

PointNet and PointNet++ introduced set abstraction (grouping of points) for tasks like classification and segmentation. PointRCNN [5] was the first two stage 3D detector to operate purely on raw points. Its stage 1 segments foreground points to generate bottom-up 3D proposals; stage 2 then pools points within each proposal (canonicalized) to refine boxes. Crucially, PointRCNN introduced a bin based bounding box loss for

stage-1 regression, encoding offsets and orientation into bins to avoid ambiguous angle regression. Shi et al. show this bin-based loss converges faster and yields higher recall than residual or corner losses [11]. VoteNet (2019) applies Hough voting for indoor scenes, but is less common in outdoor LiDAR. Other point-based approaches like PointRCNN rely on heavy multi-layer perceptrons per point, which do not scale well to millions of points without subsampling.

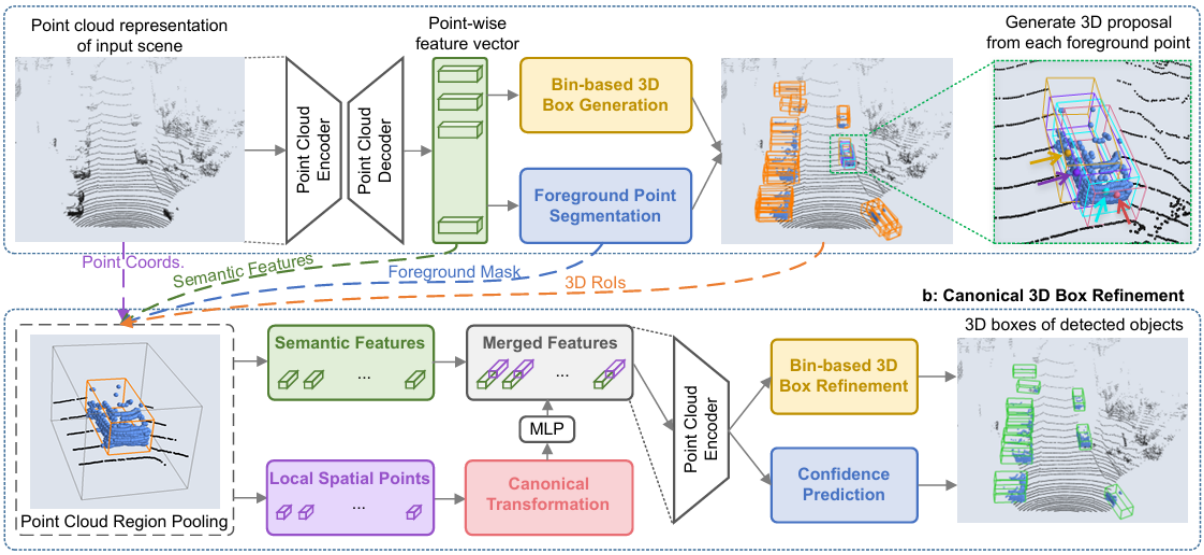


Figure 3: The PointRCNN architecture for 3D object detection from point cloud. The whole network consists of two parts: (a) for generating 3D proposals from raw point cloud in a bottom-up manner. (b) for refining the 3D proposals in canonical coordinate.[5]

3.3 Pillar and Projection Methods

Projecting LiDAR to 2D views is another category. PointPillars [4] voxelizes only the horizontal plane into vertical "pillars" and applies PointNet within each pillar to encode features. These are fed into a 2D CNN, achieving remarkable speed and competitive accuracy. Image-based methods (Mono3D, Stereo-RCNN) typically suffer from coarse depth estimation, but recent trends show promise; however, they often complement rather than replace LiDAR detection in safety-critical applications.

3.4 Hybrid and Two-stage Detectors

The trend towards two-stage architectures for 3D detection has been strong. For example, Part-A$\hat{2}$ Net follows a coarse-to-fine paradigm: a "part-aware" first stage predicts coarse proposals and intra-object part locations, and a RoI-aware pooling aggregates these parts to refine boxes. This improves on purely point based or voxel based features by explicitly modeling object part structure. PV-RCNN [6] uniquely combines sparse 3D convolution for proposal generation with point set abstraction for refinement. It uses farthest point sampling (FPS) [7] to pick keypoints with spatial coverage, and its RoI grid pooling encodes rich local context for each proposal [6]. PV-RCNN significantly outperforms prior state-of-the-art by deeply fusing both representations.

3.5 Anchor-free and Transformer-based Detectors

Recently, anchor-free methods have gained traction. CenterPoint [9] reformulates objects as 3D center points. A heatmap of object centers is predicted (using a focal-style keypoint loss), and each center is regressed to size, orientation, and velocity [9]. A second stage pools local points to refine each detection. CenterPoint achieved state-of-art on nuScenes and Waymo, illustrating the power of anchor-free paradigms. In parallel, transformer models are entering 3D vision. 3DETR [10] applies a standard vision Transformer (with learned object queries and positional embeddings) directly to point clouds. Without many hand-designed 3D modules, it outperforms strong VoteNet baselines on ScanNet by 9.5% AP [10]. These works suggest that flexible, self-attention-based modules can capture geometry without voxelization or laborious feature design.

3.6 Loss Functions and Training Strategies

Across these models, loss design is critical. Classification typically uses cross-entropy or focal loss (to handle imbalance), while box regression often uses Smooth L1 or IoU-based losses. For example, SECOND and others use Smooth L1 for localization and focal for classification. PointRCNN's bin-based loss [11] and CenterPoint's heatmap regression are specialized to 3D. Data augmentation is also widely used: flipping, scaling, rotation, random dropout of points, and GT-sampling as in SECOND [3]. Normalization layers and staging are common strategies to stabilize learning.

3.7 Summary

PV-RCNN++ stands on the shoulders of voxel CNNs, point-set learning, and key advances in sampling and pooling. By reviewing these leading methods—VoxelNet [2], SECOND [3], PointPillars [4], PointRCNN [5], Part-A$\hat{2}$, CenterPoint [9], and Transformers like VoTr [7]/3DETR [10]—we identify a rich set of ideas to incorporate. Our methodology will leverage sparse convolution backbones, adaptive sampling, advanced local aggregation , and enhanced loss functions, aiming to push PV-RCNN++ further in both speed and accuracy.

4 Proposed Methodology

Building on the PV-RCNN++ framework [1], our methodology introduces several enhancements:

4.1 Adaptive Keypoint Sampling

In PV-RCNN++, keypoints are drawn near proposals using sectorized sampling. We will extend this by making the sampling adaptive to proposal size and shape. For example, larger proposals may need more keypoints to capture detail, while smaller ones need fewer to save computation. We propose a criterion that allocates a budget of keypoints per proposal based on its estimated size or uncertainty. Additionally, we will explore learnable sampling weights that prioritize points likely to be informative over background. One approach is to apply a small point-wise neural network (or MLP) to score candidate

points before sampling, similar to attention scoring [12]. This could further focus the limited keypoints on object regions, complementing the geometric FPS criterion [7].

4.2 Enhanced Local Aggregation (VectorPool + Attention)

PV-RCNN++'s VectorPool splits each center's local neighborhood into sub-voxels and concatenates their feature vectors [1]. To enhance this, we propose adding cross-subvoxel attention. After obtaining the sub-voxel features, a lightweight self-attention module can learn to weigh different sub-voxels, effectively allowing the network to focus on the most discriminative parts of the neighborhood. This is inspired by VoTr [7], which captures long-range voxel relations, and by transformer DETR modules [12]. Concretely, for each center point, we treat the sub-voxel features as a set of tokens and apply a multi-head attention layer. This yields a fused local feature that encodes interactions between different spatial bins. We will compare this against the baseline (simple concatenation followed by MLP) to quantify gains. We will also consider graph-based aggregation: for instance, building a k-NN graph of points within the local region and applying graph convolutions to capture point relationships.

4.3 Multi-scale Feature Fusion

PV-RCNN++ currently pools features from two voxel backbone scales. We will investigate adding an extra scale or employing a feature pyramid network (FPN) style fusion of multi-resolution features for proposals, as is done in some state of the art 2D detectors. This could help detect objects of varying sizes more robustly. The strategy will be to interpolate or pool voxel features from multiple depths of the CNN backbone into each keypoint, similarly to PV-RCNN's multi-scale set abstraction [6].

4.4 Loss Function Improvements

For box regression, we will experiment with a 3D IoU-based loss (such as GIoU or DIoU extended to 3D). These losses directly optimize box overlap and have shown improvements in 2D/3D tasks over L1 losses [11]. We will implement a 3D GIoU loss and compare it to the default smooth L1 + bin loss. For orientation, we may adopt a discrete classification of heading bins plus a residual (as in SECOND) or a quaternion-based loss to avoid discontinuities. For classification/objectness, we will try the focal loss (Lin et al.) to handle class imbalance, following CenterPoint's approach of heatmap focal loss [9]. Optionally, incorporating an IoU branch (predicting IoU of the predicted box with ground truth) can help suppress low-quality detections, following approaches in 2D detection.

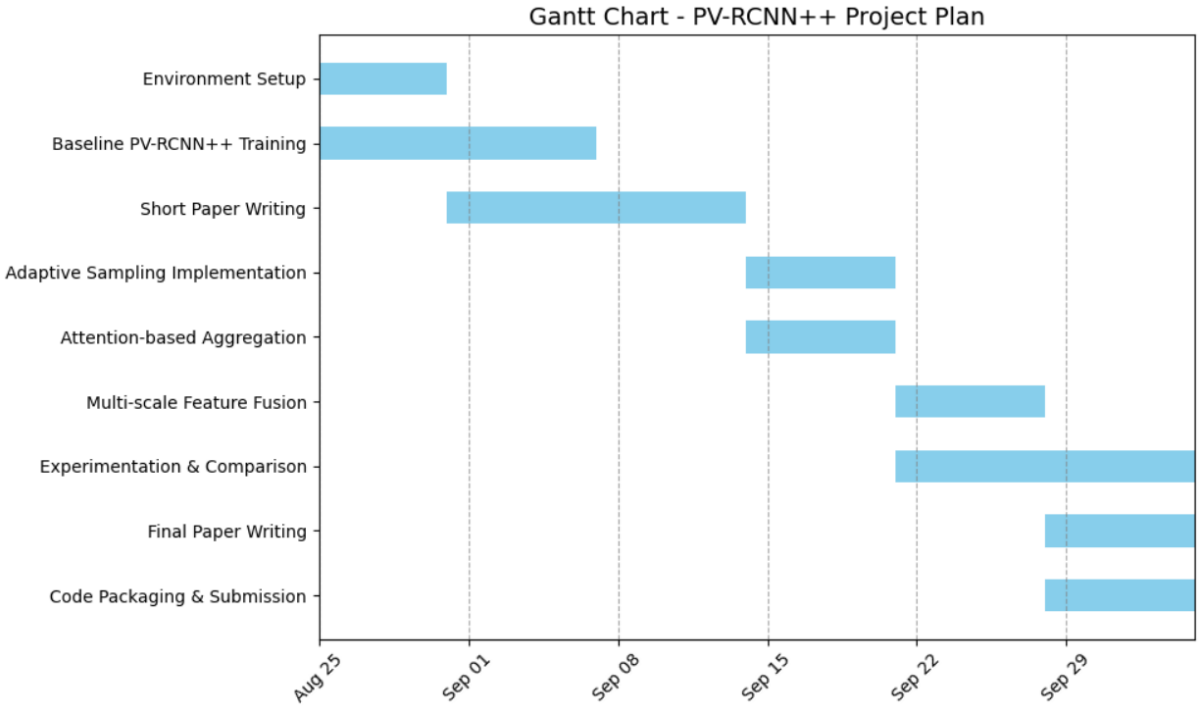
4.5 Training and Augmentation

We will adopt strong data augmentation. This includes random flipping (x-axis), scaling, rotation (about vertical axis), and ground-truth sampling from a database (as in SECOND [3]) to enrich scenes. We will also use dropout of points or cutout (blanking out cuboids) to improve robustness. For optimization, we will use a multi-stage training schedule: train the first-stage backbone and proposal head first, then fix it and train the second-stage refinement, and finally fine-tune all jointly. Batch normalization, learning

rate warm-up, and multi-GPU sync batch-norm will be used to stabilize training on large point clouds.

To implement these methods, we will build on an existing open-source codebase (OpenPCDet) that supports PV-RCNN++ and modern loss modules. Extensive ablation studies will isolate the impact of each modification. The ultimate aim is that the enhanced PV-RCNN++ achieves equal or higher accuracy on standard benchmarks (KITTI, Waymo) than the original, with comparable or lower latency.

5 Project Timeline



6 Conclusion

In this work, we propose to advance the PV-RCNN++ framework by introducing adaptive sampling, attention-enhanced local aggregation, multi-scale feature fusion, and improved loss functions. The literature indicates that combining voxel and point representations yields strong 3D detection performance [6, 13], and recent methods in sparse convolution [4], anchor-free detection [9], and transformer modules [7, 12] provide promising design cues. By leveraging these ideas, the enhanced PV-RCNN++ is expected to achieve higher accuracy and efficiency in 3D object detection. Our extensive methodology builds on established work [3, 5] and integrates cutting-edge techniques, with the goal of setting a new state-of-the-art on LiDAR detection benchmarks. We will demonstrate the effectiveness of our contributions through rigorous evaluation and aim to open-source the code for community use.

References

[1] S. Shi *et al.*, "Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection," *International Journal of Computer Vision*, 2023.

[2] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," *arXiv preprint arXiv:1711.06396*, 2017.

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

[4] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," *arXiv preprint arXiv:1812.05784*, 2018.

[5] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[6] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[7] J. Mao, Y. Chen, X. Wang, and H. Li, "Voxel transformer for 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[8] B. Graham, M. Engelcke, and L. van der Maaten, "Submanifold sparse convolutional networks," *arXiv preprint arXiv:1706.01307*, 2018.

[9] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3d object detection and tracking," *arXiv preprint arXiv:2006.11275*, 2020.

[10] I. Misra, R. Girdhar, and A. Joulin, "3detr: An end-to-end transformer model for 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[11] A. Saxena *et al.*, "Adaptive iou and regression loss for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.

[12] K. He *et al.*, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[13] C. R. Qi *et al.*, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.