quantization loss

$(\mathbf{AF})^2$-S3Net: Attentive Feature Fusion with Adaptive Feature Selection for Sparse Semantic Segmentation Network

Ran Cheng[1], Ryan Razani[1], Ehsan Taghavi[1], Enxu Li[1], and Bingbing Liu[1]

[1]Noah Ark's Lab, Huawei, Markham, ON, Canada

Abstract

Autonomous robotic systems and self driving cars rely on accurate perception of their surroundings as the safety of the passengers and pedestrians is the top priority. Semantic segmentation is one of the essential components of road scene perception that provides semantic information of the surrounding environment. Recently, several methods have been introduced for 3D LiDAR semantic segmentation. While they can lead to improved performance, they are either afflicted by high computational complexity, therefore are inefficient, or they lack fine details of smaller instances. To alleviate these problems, we propose $(\mathbf{AF})^2$-*S3Net, an end-to-end encoder-decoder CNN network for 3D LiDAR semantic segmentation. We present a novel multi-branch attentive feature fusion module in the encoder and a unique adaptive feature selection module with feature map re-weighting in the decoder. Our* $(\mathbf{AF})^2$-*S3Net fuses the voxel-based learning and point-based learning methods into a unified framework to effectively process the large 3D scene. Our experimental results show that the proposed method outperforms the state-of-the-art approaches on the large-scale SemanticKITTI benchmark, ranking* $\mathbf{1^{st}}$ *on the competitive public leaderboard competition upon publication.*

1. Introduction

Understanding of the surrounding environment has been one of the most fundamental tasks in autonomous robotic systems. With the challenges introduced with recent technologies such as self-driving cars, a detailed and accurate understanding of the road scene has become a main part of any outdoor autonomous robotic system in the past few years. To achieve an acceptable level of road scene understanding, many frameworks benefit from image semantic segmentation, where a specific class is predicted for every pixel in the input image, giving a clear perspective of the scene.

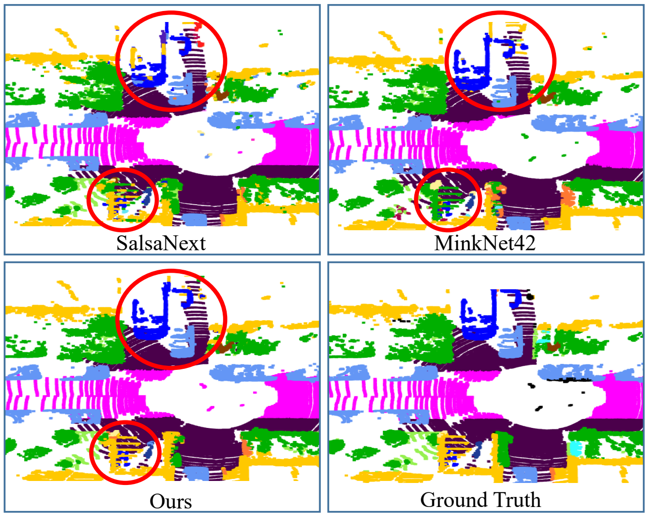

Figure 1: Comparison of our proposed method with SalsaNext [9] and MinkNet42 [8] on SemanticKITTI benchmark [3].

Although image semantic segmentation is an important step in realizing self driving cars, the limitations of a vision sensor such as inability to record data in poor lighting conditions, variable sensor sensitivity, lack of depth information and limited field-of-view (FOV) makes it difficult for vision sensors to be the sole primary source for scene understanding and semantic segmentation. In contrast, Light Detection and Ranging (LiDAR) sensors can record accurate depth information regardless of the lighting conditions with high density and frame rate, making it a reliable source of information for critical tasks such as self driving.

LiDAR sensor generates point cloud by scanning the environment and calculating time-of-flight for the emitted laser beams. In doing so, LiDARs can collect valuable information, such as range (e.g., in Cartesian coordinates) and intensity (a measure of reflection from the surface of the objects). Recent advancement in LiDAR technology makes it possible to generate high quality, low noise and dense scans from desired environments, making the task of scene understanding a possibility using LiDARs. Although rich in information, LiDAR data often comes in an unstructured format and partially sparse at far ranges. These characteristics make the task of scene understating challenging using LiDAR as primary sensor. Nevertheless, research in scene understanding and in specific, semantic segmentation using LiDARs, has seen an increase in the past few years with the availability of datasets such as semanticKITTI [3].

The unstructured nature and partial sparsity of LiDAR data brings challenges to semantic segmentation. However, a great effort has been put by researchers to address these obstacles and many successful methods have been proposed in the literature (see Section 2). From real-time methods which use projection techniques to benefit from the available 2D computer vision techniques, to fully 3D approaches which target higher accuracy, there exist a range of methods to build on. To better process LiDAR point cloud in 3D and to overcome limitations such as non-uniform point densities and loss of 颗粒状的 granular information in voxelization step, we propose **$(\mathbf{AF})^{2}$-S3Net**, which is built upon Minkowski Engine [8] to suit varying levels of sparsity in LiDAR point clouds, achieving state-of-the-art accuracy in semantic segmentation methods on SemanticKITTI [3]. Fig. 1 demonstrates qualitative results of our approach compared to SalsaNext [9] and MinkNet42 [8]. We summarize our contributions as,

- An end-to-end encoder-decoder 3D sparse CNN that achieves state-of-the-art accuracy in semanticKITTI benchmark [3];
- A multi-branch attentive feature fusion module in the encoder to learn both global contexts and local details;
- An adaptive feature selection module with feature map re-weighting in the decoder to actively emphasize the contextual information from feature fusion module to improve the generalizability;
- A comprehensive analysis on semantic segmentation and classification performance of our model as opposed to existing methods on three benchmarks, semanticKITTI [3], nuScenes-lidarseg [5], and ModelNet40 [33] through ablation studies, qualitative and quantitative results.

2. Related Work

2.1. 2D semantic Segmentation

SqueezeSeg [31] is one of the first works on LiDAR semantic segmentation using range-image, where LiDAR point cloud projected on a 2D plane using spherical transformation. SqueezeSeg [31] network is based on an encoder-decoder using Fully Connected Neural Network (FCNN) and a Conditional Random Fields (CRF) as a Recurrent Neural Network (RNN) layer. In order to reduce number of the parameters in the network, SqueezeSeg incorporates “fireModules” from [14]. In a subsequent work, SqueezeSegV2 [32] introduced Context Aggregation Module (CAM), a refined loss function and batch normalization to further improve the model. SqueezeSegV3 [34] stands on the shoulder of [31, 14], adopting a Spatially-Adaptive Convolution (SAC) to use different filters in different locations in relation to the input image. Inspired by YOLOv3 [25], RangeNet++ [21] uses a DarkNet backbone to process a range-image. In addition to a novel CNN, RangeNet++ [21] proposes an efficient way of predicting labels for the full point cloud using a fast implementation of K-nearest neighbour (KNN).

Benefiting from a new 2D projection, PolarNet [37] takes on a different approach using a polar Birds-Eye-View (BEV) instead of the standard 2D grid-based BEV projections. Moreover, PolarNet encapsulates the information regarding each polar gird using PointNet, rather than using hand crafted features, resulting in a data-driven feature extraction, a nearest-neighbor-free method and a balanced grid distribution. Finally, in a more successful attempt, SalsaNext [9], makes a series of improvements to the backbone introduced in SalsaNet [1] such as, a new global contextual block, an improved encoder-decoder and Lovász-Softmax loss [4] to achieve state-of-the-art results in 2D LiDAR semantic segmentation using range-image input.

2.2. 3D semantic Segmentation

The category of large scale 3D perception methods kicked off by early works such as [6, 20, 23, 29, 39] in which a voxel representation was adopted to capitalize vanilla 3D convolutions. In attempt to process unstructured point cloud directly, PointNet [22] proposed a Multi-Layer Perception (MLP) to extract features from input points without any voxelization. PointNet++ [24] which is an extension to the nominal work Pointnet [22], introduced sampling at different scales to extract relevant features, both local and global. Although effective for smaller point clouds, Methods rely on Pointnet [22] and its variations are slow in processing large-scale data.

Down-sampling is at the core of the method proposed in RandLA-Net [13]. As down-sampling removes features randomly, a local feature aggregation module is also introduced to progressively increase the receptive field for each 3D point. The two techniques used jointly to achieve both efficiency and accuracy in large-scale point cloud semantic segmentation. In a different approach, Cylinder3D [38] uses cylindrical grids to partition the raw point cloud. To extract features, authors in [38] introduced two new CNN blocks. An asymmetric residual block to ensure features related to cuboid 立方体 objects are being preserved and Dimensiondecomposition based Context Modeling in which multiple low-rank contexts are merged to model a high-ranked tensor suitable for 3D point cloud data.

Authors in KPConv [28] introduced a new point convolution without any intermediate steps taken in processing point clouds. In essence, KPConv is a convolution operation which takes points in the neighborhood as input and processes them with spatially located weights. Furthermore, a deformable version of this convolution operator was also introduced that learns local shifts to make them adapt to point cloud geometry. Finally, MinkowskiNet [8] introduces a novel 4D sparse convolution for spatio-temporal 3D point cloud data along with an open-source library to support auto-differentiation for sparse tensors. Overall, where we consider the accuracy and efficiency, voxel-based methods such as MinkowskiNet [8] stands above others, achieving state-of-the-art results within all sub-categories of 3D semantic segmentation.

2.3. Hybrid Methods

Hybrid methods, where a mixture of voxel-based, projection-based and/or point-wise operations are used to process the point cloud, has been less investigated in the past, but with availability of more memory efficient designs, are becoming more successful in producing competitive results. For example, FusionNet [35] uses a voxel-based MLP, called *voxel-based mini-PointNet* which directly aggregates features from all the points in the neighborhood voxels to the target voxel. This allows FusionNet [35] to search neighborhoods with low complexity, processing large scale point cloud with acceptable performance. In another approach, 3D-MiniNet [2] proposes a learning-based projection module to extract local and global information from the 3D data and then feeds it to a 2D FCNN in order to generate semantic segmentation predictions. In a slightly different approach, MVLidarNet [7] benefits form range-image LiDAR semantic segmentation to refine object instances in bird's-eye-view perspective, showcasing the applicability of LiDAR semantic segmentation in real-world applications.

Finally, SPVNAS [27] builds upon the Minkowski Engine [8] and designs a hybrid approach of using 4D sparse convolution and point-wise operations to achieve state-of-the-art results in LiDAR semantic segmentation. To do this, authors in SPVNAS [27] use a neural architecture search (NAS) [18] to efficiently design a NN, based on their novel Sparse Point-Voxel Convolution (SPVConv) operation.

3. Proposed Approach

The sparsity of outdoor-scene point clouds makes it difficult to extract spatial information compared to indoor-scene point clouds with fixed number of points or based on the dense image-based dataset. Therefore, it is difficult to leverage the indoor-scene or image-based segmentation methods to achieve good performance on a large-scale driving scene covering more than 100m with non-uniform point densities. Majority of the LiDAR segmentation methods attempt to either transform 3D LiDAR point cloud into 2D image using spherical projection (i.e., perspective, bird-eye-view) or directly process the raw point clouds. The former approach abandons valuable 3D geometric structures and suffers from information loss due to projection process. The latter approach requires heavy computations and not feasible to be deployed in constrained systems with limited resources. Recently, sparse 3D convolution became popular due to its success on outdoor LiDAR semantic segmentation task. However, out of a few methods proposed in [8, 27], no advanced feature extractors were proposed to enhance the results similar to computer vision and 2D convolutions.

To overcome this, we propose (AF)^2-S3Net for LiDAR semantic segmentation in which a baseline model of MinkNet42 [8] is transformed into an end-to-end encoder-decoder with attention blocks and achieves stat-of-the-art results. In this Section we first present the proposed network architecture along with its novel components, namely AF2M and AFSM. Then, the network optimization is introduced followed by the training details.

3.1. Problem statement

Lets consider a semantic segmentation task in which a LiDAR point cloud frame is given with a set of unordered points $(P, L) = (\{p_i, l_i\})$ with $p_i \in \mathbb{R}^{d_{in}}$ and $i = 1, ..., N$, where N denotes the number of points in an input point cloud scan. Each point p_i contains d_{in} input features, i.e., Cartesian coordinates (x, y, z), intensity of returning laser beam (i), colors (R, G, B), etc. Here, $l_i \in \mathbb{R}$ represents the ground truth labels corresponding to each point p_i. However, in object classification task, a single class label $\mathcal{L}$ is assigned to an individual scene containing P points.

Our goal is to learn a function $\mathfrak{F}_{cls}(., \Phi)$ parameterized by Φ that assigns a single class label $\mathcal{L}$ for all the points in the point cloud or in other words, $\mathfrak{F}_{seg}(., \Phi)$, that assigns a per point label $\hat{c}_i$ to each point p_i. To this end, we propose $(\mathbf{AF})^{\mathbf{2}}$-S3Net to minimize the difference between the predicted label(s), $\hat{\mathcal{L}}$ and $\hat{c}_i$, and the ground truth class label(s), $\mathcal{L}$ and l_i, for the tasks of classification and segmentation, respectively.

3.2. Network architecture

The block diagram of the proposed method, $(\mathbf{AF})^{\mathbf{2}}$-S3Net, is illustrated in Fig. 2. $(\mathbf{AF})^{\mathbf{2}}$-S3Net consists of a residual network based backbone and two novel modules, namely Attentive Feature Fusion module (AF2M) and Adaptive Feature Selection Module (AFSM). The model takes in a 3D LiDAR point cloud and transforms it into sparse tensors containing coordinates and features corresponding to each point. Then, the input sparse tensor is

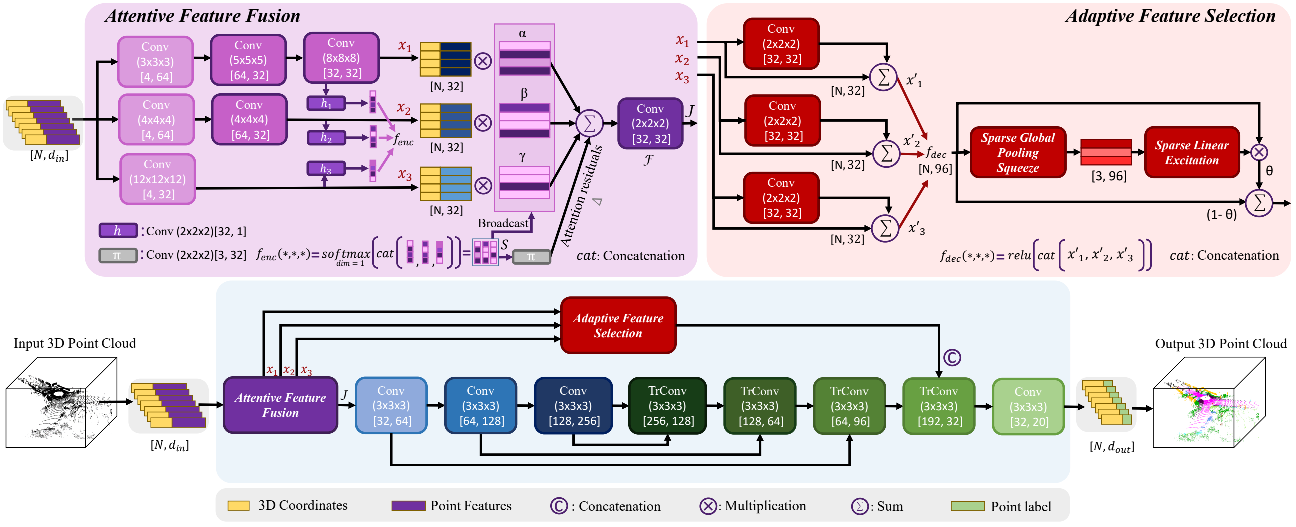

Figure 2: Overview of (AF)^2-S3Net. The top left block is Attentive Feature Fusion Module (AF2M) that aggregates Local and global context using a weighted combination of mutually exclusive learnable masks, α, β, and γ. The top right block illustrates how Adaptive Feature Selection Module (AFSM) uses shared parameters to learn inter relationship between channels across multi-scale feature maps from AF2M. (best viewed on display)

processed by $(\mathbf{AF})^{\mathbf{2}}$-S3Net which is built upon 3D sparse convolution operations which suits sparse point clouds and effectively predicts a class label for each point given a LiDAR scan.

coordinate feature

A sparse tensor can be expressed as $P_s = [C, F]$, where $C \in \mathbb{R}^{N \times M}$ represents the input coordinate matrix with M coordinates and $F \in \mathbb{R}^{N \times K}$ denotes its corresponding feature matrix with K feature dimensions. In this work, we consider 3D coordinates of points (x, y, z), as our sparse tensor coordinate C, and per point normal features (n_x, n_y, n_z) along with intensity of returning laser beam (i) as our sparse tensor feature F. Exploiting normal features helps the model to learn additional directional information, hence, the model performance can be improved by differentiating the fine details of the objects. The detailed description of the network architecture is provided below.

Attentive Feature Fusion (AF2M): To better extract the global contexts, AF2M embodies a hybrid approach, covering small, medium and large kernel sizes, which focuses on point-based, medium-scale voxel-based and large-scale voxel-based features, respectively. The block diagram of AF2M is depicted in Fig. 2 (top-left). Principally, the proposed AF2M fuses the features $\bar{x} = [x_1, x_2, x_3]$ at the corresponding branches using $g(\cdot)$ which is defined as,

$$g(x_1, x_2, x_3) \triangleq \alpha x_1 + \beta x_2 + \gamma x_3 + \Delta \tag{1}$$

where α, β and γ are the corresponding coefficients that scale the feature columns for each point in the sparse tensor, and are processed by function $f_{enc}(\cdot)$ as shown in Fig. 2. Moreover, the attention residuals, Δ, is introduced to stabilize the attention layers $h_i(\cdot), \forall i \in \{1, 2, 3\}$, by adding the residual damping factor. This damping factor is the output of residual convolution layer π. Further, function π can be formulated as

$$\pi \triangleq sigmoid(bn(conv(f_{enc}(\cdot)))) \tag{2}$$

Finally, the output of AF2M is generated by $\mathcal{F}(g(\cdot))$, where $\mathcal{F}$ is used to align the sparse tensor scale space with the next convolution block. As illustrated in Fig. 2 (top-left), for each h_i, $\forall i \in \{1, 2, 3\}$, the corresponding gradient of weight w_{h_i} can be computed as:

$$w_{h_i} = w_{h_i} - \frac{\partial J}{\partial g} \frac{\partial g}{\partial f_{enc}} \frac{\partial f_{enc}}{\partial h_i} - \frac{\partial J}{\partial g} \frac{\partial g}{\partial \pi} \frac{\partial \pi}{\partial h_i} \tag{3}$$

where J is the output of $\mathcal{F}$. Considering $g(\cdot)$ is a linear function of concatenated features $\bar{x}$ and Δ, we can rewrite Eq. 3 as follows:

$$w_{h_i} = w_{h_i} - \frac{\partial J}{\partial g} \bar{x} \frac{\partial f_{enc}}{\partial h_i} - \frac{\partial J}{\partial g} \frac{\partial \pi}{\partial h_i} \tag{4}$$

where $\frac{\partial f_{enc}}{\partial h_i} = \mathcal{S}_j(\delta_{ij} - \mathcal{S}_j)$ is the Jacobian of softmax function $S(\bar{x}) : \mathbb{R}^N \to \mathbb{R}^N$ and maps ith input feature column to jth output feature column, and δ is Kronecker delta function where $\delta_{i=j} = 1$ and $\delta_{i \neq j} = 0$. As shown in Eq. 5,

$$\mathcal{S}_j(\delta_{ij} - \mathcal{S}_j) = \begin{bmatrix} -S_1^2 & S_1(1 - S_2) & \dots \\ \vdots & \ddots & \\ S_N(1 - S_1) & \dots & -S_N^2 \end{bmatrix} \tag{5}$$

when the softmax output S is close to 0, the term $\frac{\partial f_{enc}}{\partial h_i}$ approaches to zero which prompts no gradient, and when S is close to 1, the gradient is close to identity matrix. As a result, when $S \rightarrow 1$, all values in α or β or γ get very high confidence and the update of w_{h_i} becomes:

$$w_{h_i} = w_{h_i} - \frac{\partial J}{\partial g}\frac{\partial \pi}{\partial h_i} + \frac{\partial J}{\partial g}\bar{x}I \tag{6}$$

and in the case of $S \rightarrow 0$, the update gradient will only depends on π. Fig. 3 further illustrates the capability of the proposed AF2M and highlights the effect of each branch visually.

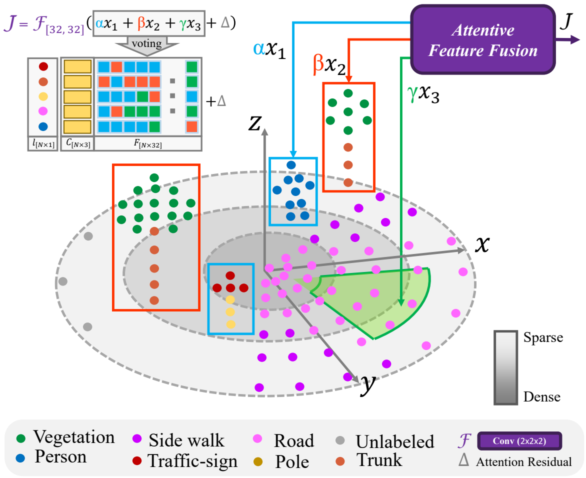

Figure 3: Illustration of Attentive Feature Fusion and spatial geometry of point cloud. The three labels αx_1, βx_2, and γx_3 represent the branches in AF2M encoder block. The first branch, αx_1, learns to capture and emphasize the fine details for smaller instances such as person, pole and traffic-sign across the driving scenes with varying point densities. The shallower branches, βx_2 and γx_3, learn different attention-maps that focus on global contexts embodied in larger instances such as vegetation, sidewalk and road surface. (best viewed on display)

Adaptive Feature Selection module (AFSM): The block diagram of AFSM is shown in Fig. 2 (top-right). In AFSM decoder block, the feature maps from multiple branches in AF2M, x_1, x_2, and x_3, are further processed by residual convolution units. The resulted output, $x_1^{'}$, $x_2^{'}$, and $x_3^{'}$, are concatenated, shown as f_{dec}, and are passed into a shared squeeze re-weighting network [12] in which different feature maps are voted. This module acts like an adaptive dropout that intentionally filters out several feature maps that are not contributing to the final results. Instead of directly passing through the weighted feature maps as output, we employed a damping factor $\theta = 0.35$, to regularize the weighting effect. It is worth noting that the skip connection connecting the attentive feature fusion module branches to the last decoder block, ensures that the error gradient propagates back to the encoder branches for better learning stability.

3.3. Network Optimization

We leveraged a linear combination of geo-aware anisotrophic [17], Exponential-log loss [30] and Lovász loss [4] to optimize our network. In particular, geo-aware anisotrophic loss is beneficial to recover the fine details in a LiDAR scene. Moreover, Exponential-log loss [30] loss is used to further improve the segmentation performance by focusing on both small and large structures given a highly unbalanced dataset.

The geo-aware anisotrophic loss can be computed by,

$$L_{geo}(y, \hat{y}) = -\frac{1}{N}\sum_{i,j,k}\sum_{c=1}^{C}\frac{M_{LGA}}{\psi}y_{ijk,c}log\hat{y}_{ijk,c} \tag{7}$$

where y and $\hat{y}$ are the ground truth label and predicted label. Parameter N is the local tensor neighborhood and in our experiment, we empirically set it as 5 (a 10 voxels size cube). Parameter C is the semantic classes, $M_{LGA} = \sum_{\psi=1}^{\Psi}(c_p \oplus c_{q_\psi})$, defined in [17]. We normalized local geometric anisotropy within the sliding window Ψ of the current voxel cell p and its neighbor voxel grid $q_\psi \in \Psi$.

Therefore, the total loss used to train the proposed network is given by,

$$L_{tot}(y, \hat{y}) = w_1 L_{exp}(y, \hat{y}) + w_2 L_{geo}(y, \hat{y}) + w_3 L_{lov}(y, \hat{y}) \tag{8}$$

where w_1, w_2, and w_3 denote the weights of Exponential-log loss [30], geo-aware anisotrophic, and Lovász loss, respectively. They are set as 1, 1.5 and 1.5 in our experiments.

4. Experimental results

We base our experimental results on three different dataset, namely, SemanticKITTI, nuScenes and ModelNet40 to show the applicability of the proposed methods in different scenes and domains. As for the SemanticKITTI and ModelNet40, $(\mathbf{AF})^{\mathbf{2}}$-S3Net is compared to the previous state-of-the-art, but due to a recently announce challenge for nuScenes-lidarseg dataset [5], we provide our own evaluation results against the baseline model.

To evaluate the performance of the proposed method and compare with others, we leverage mean Intersection over Union (**mIoU**) as our evaluation metric. **mIoU** is the most popular metric for evaluating semantic point cloud segmentation and can be formalized as $mIoU = \frac{1}{n}\sum_{c=1}^{n}\frac{TP_c}{TP_c+FP_c+FN_c}$, where TP_c is the number of true positive points for class c, FP_c is the number of false positives, and FN_c is the number of false negatives.

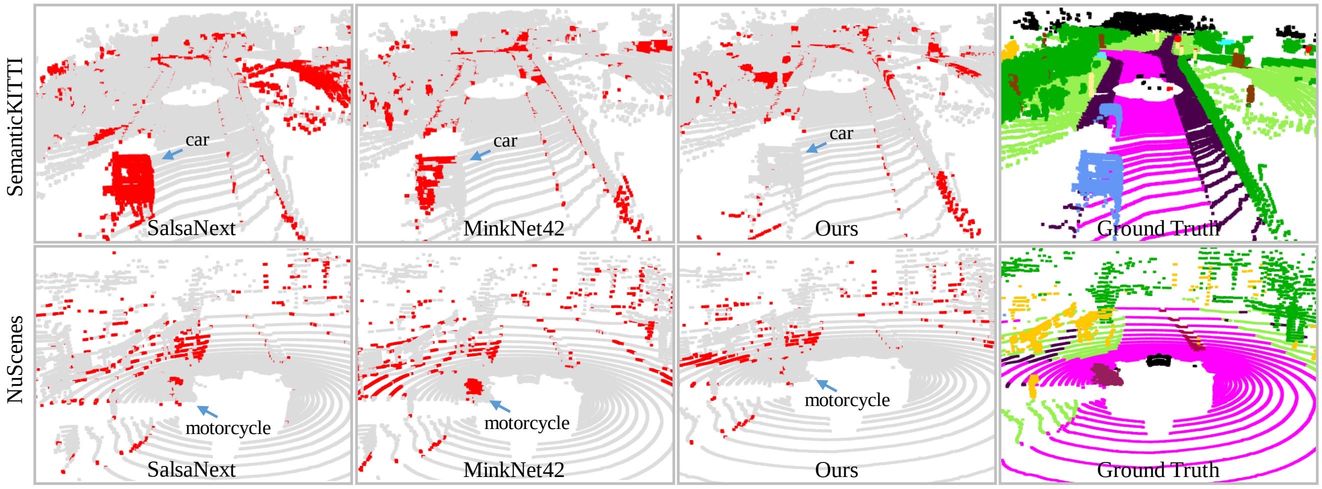

Figure 4: Compared to SalsaNext and MinkNet42, our method has a lower error (shown in red) recognizing region surface and smaller objects on nuScenes validation set, thanks to the proposed attention modules.

As for the training parameters, we trained our model with SGD optimizer with momentum of 0.9 and learning rate of 0.001, weight decay of 0.0005 for 50 epochs. The experiments are conducted using 8 Nvidia V100 GPUs.

4.1. Quantitative Evaluation

In this Section, we provide quantitative evaluation of $(\mathbf{AF})^{\mathbf{2}}$-S3Net on two outdoor large-scale public dataset: SemanticKITTI [3] and nuScenes-lidarseg dataset [5] for semantic segmentation task and on ModelNet40 [33] for classification task.

SemanticKITTI dataset: we conduct our experiments on SemanticKITTI [3] dataset, the largest dataset for autonomous vehicle LiDAR segmentation. This dataset is based on the KITTI dataset introduced in [11], containing 41000 total frames which captured in 21 sequences. We list our experiments with all other published works in Table 1. As shown in Table 1, our method achieves state-of-the-art performance in SemanticKITTI test set in terms of mean IoU. With our proposed method, $(\mathbf{AF})^{\mathbf{2}}$-S3Net, we see a 2.7% improvement from the second best method [27] and 15.4% improvement from baseline model (MinkNet42 [8]). Our method dominates greatly in classifying small objects such as bicycle, person and motorcycle, making it a reliable solution to understating complex scenes. It is worth noting that $(\mathbf{AF})^{\mathbf{2}}$-S3Net only uses the voxelized data as input, whereas the competing methods like SPVNAS [27] use both voxelized data and point-wise features.

Nuscenes dataset: to prove the generalizability of our proposed method, we trained our network with nuScenes-lidarseg dataset [5], one of the recently available large-scale datasets that provides point level labels of LiDAR point clouds. It consists of 1000 driving scenes from various locations in Boston and Singapore, providing a rich set of labeled data to advance self driving car technology. Among these 1000 scenes, 850 of them is reserved for training and validation, and the remaining 150 scenes for testing. The labels are, to some extent, similar to the semanticKITTI dataset [3], making it a new challenge to propose methods that can handle both datasets well, given the different sensor setups and environment they record the dataset. In Table 2, we compared our proposed method with MinkNet42 [8] baseline and the projection based method SalsaNext [9]. Results in Table 2 shows that our proposed method can handle the small objects in nuScenes dataset and indicates a large margin improvement from the competing methods. Considering the large difference between the two public datasets, we can prove that our work can generalize well.

ModelNet40**:** to expand and evaluate the capabilities of the proposed method in different applications, ModelNet40, a 3D object classification dataset [33] is adopted for evaluation. ModelNet40 contains $12,311$ meshed CAD models from 40 different object categories. From all the samples, $9,843$ models are used for training and $2,468$ models for testing. To evaluate our method against existing stat-of-the-art, we compare $(\mathbf{AF})^{\mathbf{2}}$-S3Net with techniques in which a single input (e.g., single view, sampled point cloud, voxel) has been used to train and evaluate the models. To make $(\mathbf{AF})^{\mathbf{2}}$-S3Net compatible for the task of classification, the decoder part of the network is removed and the output of the encoder is directly reshaped to the number of the classes in ModelNet40 dataset. Moreover, the model is trained only using cross-entropy loss. Table 3 presents the overall classification accuracy results for our proposed method and previous state-of-the-art. With the introduction of AF2M in our network, we achieved similar performance to the point-based methods which leverage fine-grain local features.

Method	Mean IoU	Car	Bicycle	Motorcycle	Truck	Other-vehicle	Person	Bicyclist	Motorcyclist	Road	Parking	Sidewalk	Other-ground	Building	Fence	Vegetation	Trunk	Terrain	Pole	Traffic-sign
S-BKI [10]	51.3	83.8	30.6	43.0	26.0	19.6	8.5	3.4	0.0	92.6	65.3	77.4	30.1	89.7	63.7	83.4	64.3	67.4	58.6	67.1
RangeNet++ [21]	52.2	91.4	25.7	34.4	25.7	23.0	38.3	38.8	4.8	91.8	65.0	75.2	27.8	87.4	58.6	80.5	55.1	64.6	47.9	55.9
LatticeNet [26]	52.9	92.9	16.6	22.2	26.6	21.4	35.6	43.0	46.0	90.0	59.4	74.1	22.0	88.2	58.8	81.7	63.6	63.1	51.9	48.4
RandLA-Net [13]	53.9	94.2	26.0	25.8	40.1	38.9	49.2	48.2	7.2	90.7	60.3	73.7	20.4	86.9	56.3	81.4	61.3	66.8	49.2	47.7
PolarNet [37]	54.3	93.8	40.3	30.1	22.9	28.5	43.2	40.2	5.6	90.8	61.7	74.4	21.7	90.0	61.3	84.0	65.5	67.8	51.8	57.5
MinkNet42 [8]	54.3	94.3	23.1	26.2	26.1	36.7	43.1	36.4	7.9	91.1	63.8	69.7	29.3	**92.7**	57.1	83.7	68.4	64.7	57.3	60.1
3D-MiniNet [2]	55.8	90.5	42.3	42.1	28.5	29.4	47.8	44.1	14.5	91.6	64.2	74.5	25.4	89.4	60.8	82.8	60.8	66.7	48.0	56.6
SqueezeSegV3 [34]	55.9	92.5	38.7	36.5	29.6	33.0	45.6	46.2	20.1	91.7	63.4	74.8	26.4	89.0	59.4	82.0	58.7	65.4	49.6	58.9
Kpconv [28]	58.8	96.0	30.2	42.5	33.4	44.3	61.5	61.6	11.8	88.8	61.3	72.7	31.6	90.5	64.2	84.8	69.2	69.1	56.4	47.4
SalsaNext [9]	59.5	91.9	48.3	38.6	38.9	31.9	60.2	59.0	19.4	91.7	63.7	75.8	29.1	90.2	64.2	81.8	63.6	66.5	54.3	62.1
FusionNet [35]	61.3	95.3	47.5	37.7	41.8	34.5	59.5	56.8	11.9	91.8	68.8	77.1	30.8	92.5	**69.4**	84.5	69.8	68.5	60.4	66.5
KPRNet [15]	63.1	95.5	54.1	47.9	23.6	42.6	65.9	65.0	16.5	**93.2**	**73.9**	**80.6**	30.2	91.7	68.4	85.7	69.8	**71.2**	58.7	64.1
SPVNAS [27]	67.0	**97.2**	50.6	50.4	**56.6**	**58.0**	67.4	67.1	50.3	90.2	67.6	75.4	21.8	91.6	66.9	**86.1**	**73.4**	71.0	**64.3**	67.3
$(\mathbf{AF})^{\mathbf{2}}$**-S3Net** [Ours]	**69.7**	94.5	**65.4**	**86.8**	39.2	41.1	**80.7**	**80.4**	**74.3**	91.3	68.8	72.5	**53.5**	87.9	63.2	70.2	68.5	53.7	61.5	**71.0**

Table 1: Segmentation IoU (%) results on the SemanticKITTI [3] test dataset.

Method	FW mIoU	**Mean IoU**	Barrier	Bicycle	Bus	Car	Construction vehicle	Motorcycle	Pedestrian	Traffic cone	Trailer	Truck	Driveable surface	Other flat ground	Sidewalk	Terrain	Manmade	Vegetation
SalsaNext [9]	82.8	58.8	56.6	4.7	77.1	**81.0**	18.4	47.5	52.8	43.5	38.3	65.7	94.2	60.0	**68.9**	**70.3**	81.2	80.5
MinkNet42 [8]	82.7	60.8	**63.1**	8.3	77.4	77.1	23.0	55.1	55.6	**50.0**	**42.5**	62.2	94.0	67.2	64.1	68.6	**83.7**	80.8
$(\mathbf{AF})^{\mathbf{2}}$**-S3Net** [Ours]	**83.0**	**62.2**	60.3	**12.6**	**82.3**	80.0	**20.1**	**62.0**	**59.0**	49.0	42.2	**67.4**	**94.2**	**68.0**	64.1	68.6	82.9	**82.4**

Table 2: Segmentation IoU (%) results on the nuScenes-lidarseg [5] validation dataset. Frequency-Weighted IoU denotes that each IoU is weighted by the point-level frequency of its class.

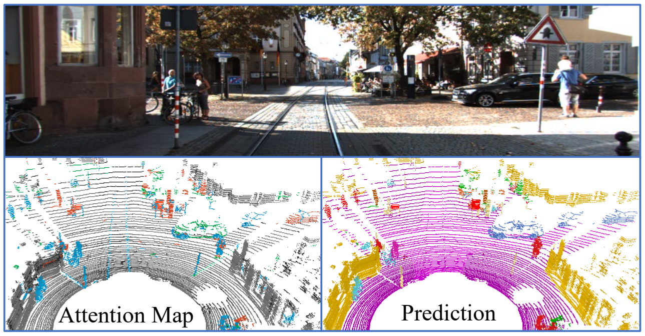

Figure 5: Reference image (top), Prediction (bottom-right), attention map (bottom-left) on SemanticKITTI test set. Color codes are: ■ road |■ side-walk |■ parking |■ car |■ bicyclist |■ pole |■ vegetation |■ terrain |■ trunk |■ building |■ other-structure |■ other-object.

4.2. Qualitative Evaluation

In this section, we visualize the attention maps in AF2M by projecting the scaled feature maps back to original point cloud. Moreover, to better present the the improvements that has been made against the baseline model MinkNet42 [8] and SalsaNext [9], we provide the error maps which highlights the superior performance of our method.

As shown in Fig. 5, our method is capable of capturing fine details in a scene. To demonstrate this, we train $(\mathbf{AF})^{\mathbf{2}}$-S3Net on SemanticKITTI as explained above and visualize a test frame. In Fig. 5 we highlight the points with top 5% feature norm from each scaled feature maps of αx_1, βx_2 and γx_3 with cyan, orange and green colors, respectively. It can be observed that our model learns to put its attention on small instances (i.e., person, pole, bicycle, etc.) as well as larger instances (i.e., car, region boundaries, etc.). Fig. 4 shows some qualitative results on SemanticKITTI (top) and nuScenes (bottom) benchmark. It can be observed that the proposed method surpasses the baseline (MinkNet42 [8]) and range-based SalsaNext [9] by a large margin, which failed to capture fine details such as cars and vegetation.

4.3. Ablation Studies

To show the effectiveness of the proposed attention mechanisms, namely, AF2M and AFSM introduced in Section 3, along with other design choices such as loss functions, this section is dedicated to a thorough ablation study starting from our baseline model introduced in [8]. The baseline is MinkNet42 which is a semantic segmentation residual NN model for 3D sparse data. To start off with a well trained baseline, we use Exponential Logarithmic Loss [30] to train the model which results in 59.8% $\mathbf{mIoU}$ accuracy for the validation set on semanticKITTI.

Method	Input	Main operator	Overall Accuracy (%)
Vox-Net [20]	voxels	3D Operation	83.00
Mink-ResNet50 [8]	voxels	Sparse 3D Operation	85.30
Pointnet [22]	point cloud	Point-wise MLP	89.20
Pointnet++ [24]	point cloud	Local feature	90.70
DGCNN[36] (1 vote)	point cloud	Local feature	91.84
GGM-Net [16]	point cloud	Local feature	92.60
RS-CNN [19]	point cloud	Local feature	**93.60**
Ours (AF2M)	voxels	Sparse 3D Operation	93.16

Table 3: Classification accuracy results on ModelNet40 dataset [33], for input size 1024×3.

Next, we add our proposed AF2M to the baseline model to help the model extract richer features from the raw data. This addition of AF2M improves the **mIoU** to 65.1%, an increase of 5.3%. In our second study and to show the effectiveness of the AFSM only, we first reduce the AF2M block to only output $\{x_1, x_2, x_3\}$ (see Fig. 2 for reference), and then add the AFSM to the model. Adding AFSM shows an increase of 3.5% in **mIoU** from the baseline. In the last step of improving the NN model, we combine AF2M and AFSM together as shown in Fig. 2, which result in **mIoU** of 68.6% and an increase of 8.8% from the baseline model.

Finally, in our last two experiments, we study the effect of our loss function by adding Lovász loss and the combination of Lovász and geo-aware anisotrophic loss, resulting in **mIoU** of 70.2% and 74.2%, respectively. The ablation studies presented, shows a series of adequate steps in the design of $(\mathbf{AF})^{\mathbf{2}}$-S3Net, proving the steps taken in the design of the proposed model are effective and can be used separately in other NN models to improve the accuracy.

Architecture	**AF2M**	**AFSM**	**Lovász**	**Lovász+Geo**	**mIoU**
Baseline					59.8
Proposed	✓				65.1
		✓			63.3
	✓	✓			68.6
	✓	✓	✓		70.2
	✓	✓	✓	✓	74.2

Table 4: Ablation study of the proposed method vs baseline evaluated on SemanticKITTI [3] validation dataset (seq 08).

4.4. Distance-based Evaluation

In this section, we investigate how segmentation is affected by distance of the points to the ego-vehicle. In order to show the improvements, we follow our ablation study and compare $(\mathbf{AF})^{\mathbf{2}}$-S3Net and the baseline (MinkNet42) on the SemanticKITTI validation set (seq 8). Fig. 6 illustrates the **mIoU** of $(\mathbf{AF})^{\mathbf{2}}$-S3Net as opposed to the baseline and SalsaNext w.r.t. the distance to the ego-vehicle's LiDAR sensors. The results of all the methods get worse by increasing the distance due to the fact that point clouds generated by LiDAR are relatively sparse, especially at large distances. However, the proposed method can produce better results at all distances, making it an effective method to be deployed on autonomous systems. It is worth noting that, while the baseline methods attempt to alleviate the sparsity problem of point clouds by using sparse convolutions in a residual style network, it lacks the necessary encapsulation of features proposed in Section 3 to robustly predict the semantics.

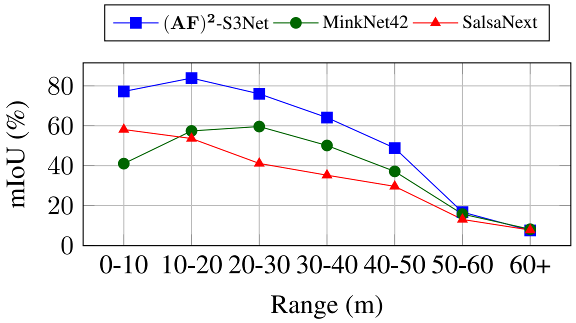

Figure 6: mIoU vs Distance for $(\mathbf{AF})^{\mathbf{2}}$-S3Net vs. baseline.

5. conclusion

In this paper, we presented an end-to-end CNN model to address the problem of semantic segmentation and classification of 3D LiDAR point cloud. We proposed $(\mathbf{AF})^{\mathbf{2}}$-S3Net, a 3D sparse convolution based network with two novel attention blocks called Attentive Feature Fusion Module (AF2M) and Adaptive Feature Selection Module (AFSM), to effectively learn local and global contexts and emphasize the fine detailed information in a given LiDAR point cloud. Extensive experiments on several benchmarks, SemanticKITTI, nuScenes-lidarseg, and ModelNet40 demonstrated the ability to capture the local details and the state-of-the-art performance of our proposed model. Future work will include the extension of our method to end-to-end 3D instance segmentation and object detection on large-scale LiDAR point cloud.

References

[1] Eren Erdal Aksoy, Saimir Baci, and Selcuk Cavdar. Salsanet: Fast road and vehicle segmentation in lidar point clouds for autonomous driving. In *IEEE Intelligent Vehicles Symposium (IV2020)*, 2020. 2

[2] Iñigo Alonso, Luis Riazuelo, Luis Montesano, and Ana C Murillo. 3d-mininet: Learning a 2d representation from point clouds for fast and efficient 3d lidar semantic segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020. 3, 7

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jürgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9296–9306. IEEE, 2019. 1, 2, 6, 7, 8

[4] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: a tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018. 2, 5

[5] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 2, 5, 6, 7

[6] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 2

[7] Ke Chen, Ryan Oldja, Nikolai Smolyanskiy, Stan Birchfield, Alexander Popov, David Wehr, Ibrahim Eden, and Joachim Pehserl. Mvlidarnet: Real-time multi-class scene understanding for autonomous driving using multiple views. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2288–2294, 2020. 3

[8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 1, 2, 3, 6, 7, 8

[9] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast semantic segmentation of lidar point clouds for autonomous driving. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 655–661, 2020. 1, 2, 6, 7

[10] Lu Gan, Ray Zhang, Jessy W Grizzle, Ryan M Eustice, and Maani Ghaffari. Bayesian spatial kernel smoothing for scalable dense semantic mapping. *IEEE Robotics and Automation Letters*, 5(2):790–797, 2020. 7

[11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. 6

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 5

[13] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020. 2, 7

[14] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 2

[15] Deyvid Kochanov, Fatemeh Karimi Nejadasl, and Olaf Booij. Kprnet: Improving projection-based lidar semantic segmentation. *ECCV Workshop*, 2020. 7

[16] Dilong Li, Xin Shen, Yongtao Yu, Haiyan Guan, Hanyun Wang, and Deren Li. Ggm-net: Graph geometric moments convolution neural network for point cloud shape classification. *IEEE Access*, 8:124989–124998, 2020. 8

[17] Jie Li, Yu Liu, Xia Yuan, Chunxia Zhao, Roland Siegwart, Ian Reid, and Cesar Cadena. Depth based semantic scene completion with position importance aware loss. *IEEE Robotics and Automation Letters*, 5(1):219–226, 2019. 5

[18] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018. 3

[19] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. 8

[20] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015. 2, 8

[21] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019. 2, 7

[22] Charles R Qi, 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*, pages 652–660, 2017. 2, 8

[23] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016. 2

[24] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on

point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 2, 8

[25] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[26] Radu Alexandru Rosu, Peer Schütt, Jan Quenzel, and Sven Behnke. Latticenet: Fast point cloud segmentation using permutohedral lattices. *Robotics: Science and Systems (RSS)*, 2020. 7

[27] Haotian* Tang, Zhijian* Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European Conference on Computer Vision*, 2020. 3, 6, 7

[28] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019. 3, 7

[29] Zongji Wang and Feng Lu. Voxsegnet: Volumetric cnns for semantic part segmentation of 3d shapes. *IEEE transactions on visualization and computer graphics*, 2019. 2

[30] Ken CL Wong, Mehdi Moradi, Hui Tang, and Tanveer Syeda-Mahmood. 3d segmentation with exponential logarithmic loss for highly unbalanced object sizes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 612–619. Springer, 2018. 5, 7

[31] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1887–1893. IEEE, 2018. 2

[32] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4376–4382. IEEE, 2019. 2

[33] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 2, 6, 8

[34] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation, 2020. 2, 7

[35] Feihu Zhang, Jin Fang, Benjamin Wah, and Philip Torr. Deep fusionnet for point cloud semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3, 7

[36] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proceedings of AAAI Conference on Artificial Inteligence*, 2018. 8

[37] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2020. 2, 7

[38] Hui Zhou, Xinge Zhu, Xiao Song, Yuexin Ma, Zhe Wang, Hongsheng Li, and Dahua Lin. Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. *arXiv preprint arXiv:2008.01550*, 2020. 2

[39] 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*, pages 4490–4499, 2018. 2