This CVPR 2021 workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Sparse Activation Maps for Interpreting 3D Object Detection

Qiuxiao Chen
Utah State University
qiuxiao.chen@aggiemail.usu.edu

Pengfei Li
University of California, Riverside
pli081@ucr.edu

Meng Xu
Utah State University
meng.xu0201@gmail.com

Xiaojun Qi
Utah State University
Xiaojun.Qi@usu.edu

Abstract

We propose a technique to generate "visual explanations" for interpretability of volumetric-based 3D object detection networks. Specifically, we use the average pooling of weights to produce a Sparse Activation Map (SAM) which highlights the important regions of the 3D point cloud data. The SAMs is applicable to any volumetric-based models (model agnostic) to provide intuitive intermediate results at different layers to understand the complex network structures. The SAMs at the 3D feature map layer and the 2D feature map layer help to understand the effectiveness of neurons to capture the object information. The SAMs at the classification layer for each object class helps to understand the true positives and false positives of each network. The experimental results on the KITTI dataset demonstrate the visual observations of the SAM match the detection results of three volumetric-based models.

1. Introduction

3D object detection has become an important research topic since LiDAR techniques have been widely used in a variety of applications ranging from autonomous driving to robotic vision to capture 3D point cloud data. Recently, Deep Neural Network (DNN) based methods have been used to achieve superior performance on 3D object detection. Due to the black-box nature of DNNs, active research has been exploring the explainability of DNNs to provide intuitive intermediate results to understand the complexity of network structures. This understanding will help researchers identify the strength and weakness of DNN structures and therefore come up with a viable solution to improve DNN structures to achieve better object detection. Furthermore, it will also help researchers to build trustworthy DNNs for object detection.

A few representative works to study the explainability of DNNs is briefly reviewed here. Zintgraf et al. [23] present a prediction difference analysis method to explain 2D image classification decisions made by DNNs. Kim et al. [4] introduce Concept Activation Vectors (CAVs) to provide an interpretation of a DNN's internal state in terms of human-friendly concepts. Mahendran and Vedaldi [6] conduct an inverting representation technique to analyze the visual information contained in DNN representations. However, all these methods focus on providing interpretability of 2D object detection networks.

To the best of our knowledge, there is no prior work that handles the explanation of 3D object detection. In this paper, we propose an effective interpretability method to explain DNN-based 3D objection detection models so the efficacy of the DNNs in detecting any objects can be directly visualized at any intermediate layers. Specifically, we build the interpretability method on top of volumetric-based models to study the strength and weakness of any layers of its associated DNNs since volumetric-based models are more computational efficient to detect 3D objects than point-based models. To this end, we first customize the design of average pooling to generate one average value for each kernel (i.e., weights of convolutional layers) to maintain the sparsity pattern of the point cloud. We then employ a weighted linear combination on feature maps to generate a Sparse Activation Map (SAM). This SAM is capable of explaining the effect of any intermediate layers of DNNs in volumetric models. For example, the SAM can be used at the last classification convolution layer to visualize the important features corresponding to each class (e.g., cars, pedestrians, and cyclists). It can also be easily deployed at any intermediate convolution layers to visualize the weighted feature map. Therefore, the proposed interpretability method can serve as a diagnostic tool for 3D object detection models, which provides insights on intermediate layers. It can also be used to qualitatively compare the performance of different models in addition to the evalua-

tion metric. Figure 1 presents an overview of volumetric-based DNN models for 3D object detection and demonstrates a SAM at the last classification convolution layer.

Our contributions are four-fold.

1. Introducing SAMs to provide visual explanations for 3D object detection networks on trained DNNs without a need for architecture change or re-training.

2. Proposing the customized average pooling of convolutional layer weights to maintain the sparsity pattern of point clouds.

3. Providing insights of not only the last classification layer but also middle network layers, i.e., 3D and 2D convolutional layers, to gain understanding of DNN structures and therefore increase the confidence of deploying the strong DNN in 3D object detection.

4. Applying SAMs to compare the effectiveness of different volumetric-based 3D object detection models via intuitive visualizations at different layers and provide insights regarding the effectiveness of the neurons and the true positives and false positives which directly correlate with the 3D object detection performance.

2. Related Work

In this section, we first present two categories of DNN-based 3D object detection. We then introduce three kinds of interpretability of DNNs and the interpretability of the instance-wise explanation method on a few representative 3D DNNs.

2.1. DNN-based 3D Object Detection Models

DNN-based 3D object detection models can be divided into two categories including volumetric-based and point-based models [13]. Volumetric-based models generally transform the irregular point clouds to regular representations such as voxels by 3D or 2D Convolutional Neural Network (CNN) to learn point features for 3D detection. Point-based models directly extract discriminative features from raw point clouds for 3D detection. Volumetric-based models are more computational effective than point-based models.

Here, we briefly review several influential works in volumetric-based 3D object detection since our proposed interpretability method handles the 3D object detection in this category. VoxelNet [22] is a pioneer work in volumetric-based 3D object detection. To improve the detection accuracy of LiDAR datasets, it divides a raw point cloud into equal 3D voxels and applies the voxel feature encoding layer to transform a group of points within each voxel to a feature representation. SECOND [20] improves VoxelNet by employing an improved sparse convolution method to increase both training and inference speed and significantly reduce the detection time. It also introduces a new form of angle loss regression and a new data augmentation approach. The former is to improve the orientation estimation performance and the latter is to enhance the convergence speed and performance. Part-A^2 [14] further improves SECOND by considering a part-aware stage to utilize free-of-charge part supervisions and a part-aggregation stage to explore the spatial relationship of the pooled intra-object part locations. PVRCNN [13] deeply integrates both 3D voxel CNN and PointNet-based set abstraction for accurate 3D object detection from point clouds. All these methods mainly focus on improving 3D object detection accuracy, but systematic interpretations of their proposed models are not provided.

2.2. Interpretability for DNNs

Huang and Kroening [3] divide DNN interpretability methods into three major categories: instance-wise explanation [6, 16, 21, 11], model explanation [5, 9, 19, 7], and information-flow explanation [17, 15, 1, 10].

Instance-wise explanation aims to understand the representation learned by DNNs through visualization over another form generated from the current input. Three common forms are synthesized input (optimizing over a hidden neural or an approximation inverse of an image representation), a ranking of a set of features computed by different methods (e.g., Local Interpretable Model-agnostic Explanations (LIME), integrated gradients, Layer-wise Relevance Propagation (LRP)), and saliency maps. Model explanation aims to use a simpler model or a set of simpler models to approximate neural network. Representative simpler models include rule extraction, decision tree extraction, linear classifiers, and automata extraction. Information-flow explanation aims to use information theoretical methods to explain the training procedure. Commonly used methods include information bottleneck method, information plane method, and stochastic DNN-based method.

Here, we briefly review a few representative work to employ an instance-wise explanation technique on 3D DNN-based models since they are closely related to our proposed work. Wang et al. [18] propose to convert images with depth maps to pseudo-LiDAR as the input to a volumetric-based DNN model to detect 3D objects. Hu et al. [2] propose to add a grid visibility map as an additional input to a volumetric-based DNN model to provide more information for accurately detecting 3D objects. For indoor 3D object classification, Qi et al. [8] use a point-based DNN model to convert the high-density grid cells back to 3D point clouds to visualize the first-level kernel features, which show the structures of planes, double planes, lines, corners and so on. Shen et al. [12] use a point-based DNN model to correlate various kernels with original features to represent

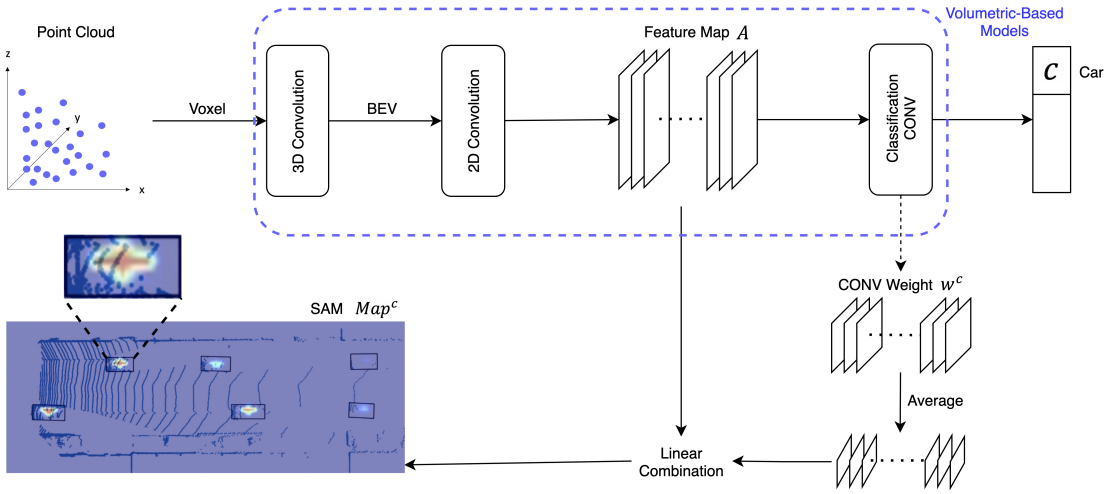

Figure 1. Overview of volumetric-based DNN models for 3D object detection and demonstration of Sparse Activation Map (SAM) at the last classification convolution layer overlaid on top of the point cloud: Given a 3D point cloud containing any object (e.g., cars, pedestrians, and cyclists) as input, we forward propagate the image through the volumetric-based networks to obtain the global feature maps and classification convolutional layer weights. The weights are then averaged to obtain the average weight vector, which is linearly combined with the global feature maps to generate the SAM (blue heatmap overlaid on the projected 2D point image). In SAM, red represents high activation (high possibility to contain an object) and blue represents low activation (low probability to contain an object). Each black rectangle contains a ground truth. One blow-up ground truth is shown in an enlarged rectangle connected by two dashed lines to its corresponding ground-truth bounding box in the projected 2D point image.

various structures in terms of plane, edge, corner, concave and convex surfaces for 3D object classification. In general, instance-wise explanation techniques on volumetric-based DNNs increase the interpretability of the input data. Instance-wise explanation techniques on point-based DNNs consider the representation in first several layers containing some low-level features corresponding to characteristics of the raw point clouds. However, they neglect to consider last several feature layers that contain more high-level semantic information. As a result, their interpretability is limited.

3. The Proposed Method

We propose an instance-wise explanation technique on any volumetric-based DNNs to visualize activations of middle and last layers of DNNs to understand the efficacy of 3D object detection. To the best of our knowledge, our proposed interpretability model is the first work to visualize the middle layers and the last layer of volumetric-based 3D object detection DNNs. To this end, we aim to explain three state-of-the-art volumetric-based models, namely, SECOND, Part-A^2, and PVRCNN, via visual interpretability of both middle layers and the last layer. This technique can be easily used to explain activations in any layer of a deep network. A number of previous research asserts that deeper representations of CNN capture higher-level visual constructs for easy understanding of the semantic information [33]. As a result, we choose the deeper feature maps at both 3D level and 2D level to visualize the activation (likelihood of the objects). Specifically, we explain activations at two intermediate convolution layers, namely, the second to the last layer of 2D convolution block and the last layer of the 3D convolution block, to respectively visualize the weighted feature map corresponding to the important features of 2D and 3D objects. We also explain activations at the classification convolution layer to visualize the important features corresponding to each class (e.g., cars, pedestrians, and cyclists).

3.1. Overview of Sparse Activation Maps

Volumetric-based models apply several layers of 3D convolutional operations on the voxel data to extract 3D feature maps. We present a Sparse Activation Map (SAM) of 3D features to demonstrate the important features corresponding to 3D objects. The 3D feature maps are warped into Bird Eye View (BEV) and then passed to several layers of 2D convolutional operations to extract 2D feature maps. We present a SAM of 2D features generated from the second to the last 2D convolution operation to demonstrate the important features corresponding to height compressed 2D objects. Based on 2D feature maps from the last 2D convolution operation, most volumetric-based models use convolution layers instead of fully connected layers to generate bounding box proposals and calculate classification scores for each proposal. We present a classification SAM of 2D feature maps generated from the last 2D convolution operation to demonstrate likelihood of each object class. Figure 1 illustrates the SAM at the classification convolution layer to

demonstrate the weighted features learned by volumetric-based models, where red represents high activation (high likelihood to contain an object class) and blue represents low activation (low likelihood to contain an object class). To generate each of the three aforementioned SAMs, average pooling is performed on the weights of convolutional layers (i.e., kernel) to generate one average value for each kernel. This pooling prevents the loss of data sparsity. A weighted linear combination is then applied on feature maps to generate the corresponding activation map.

3.2. Classification Sparse Activation Map

In volumetric-based models, final classification scores are computed by applying sigmoid operations on the prediction scores, which are computed from the classification convolution operation. In the proposed instance-wise explanation method, we use the global feature maps A generated from the 2D convolution and the convolution kernels w to generate a classification SAM, which keeps the data sparsity characteristics of point cloud. Similar operation can be applied to other intermediate convolution layers to generate their corresponding SAMs. In the following, we explain the detailed steps to generate the classification SAM.

Let A_m represent feature maps of channel m generated from the last convolutional layer of 2D convolution, w_m^c represent the convolutional kernel that connects the m-th channel and the c-th class, and $w_m^c(i,j)$ represent the convolutional kernel weights (coefficients) at location (i,j). The prediction score of the c-th class is computed by:

$$S_{pred}^c = \sum_m w_m^c \star A_m$$

where $\star$ is the conventional convolution operation.

We observe this conventional convolution operation cannot work well when the input data are sparse. Figure 2 illustrates this shortcoming in the context of the sparse point clouds, where a data point does not have any neighbors (refer to the example in the second column of the top row) or a data point has few neighbors (refer to the example in the second column of the bottom row). The third column shows that the filtered result obtained from the conventional convolution operation exhibits a blurring and ringing effect around the sparse data point, which does not preserve the characteristics of the point cloud.

To maintain the same data sparsity pattern, we propose a customized average pooling to reduce the kernel to a value (e.g., average of the kernel coefficients). This value is multiplied with the feature map to generate the weighted feature map. A linear combination is then applied to combine all channels of the weighted feature map to yield a SAM. These operations to obtain a classification SAM Map^c for

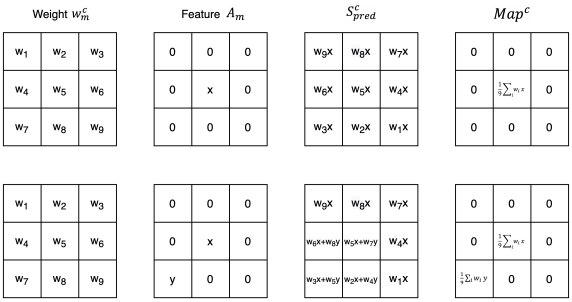

Figure 2. Comparison of filtering results obtained from the conventional convolution operation and the customized average pooling operation. First column: Convolution kernel with nine coefficients; Second column: Sparse input features; Third column: Conventional convolution results; Fourth column: Customized average pooling results.

the c-th class can be written as follows:

$$Map^c(x,y) = \frac{1}{M}\sum_m [A_m(x,y) \cdot \frac{1}{N}\sum_{i,j} w_m^c(i,j)]$$

where M is the total number of channels of feature maps, N is the total number of pixels in the kernel, and $\cdot$ is a scalar multiplication. Finally, we align the classification SAM with the original 3D point cloud data by upsampling it to the size of the BEV of the point cloud, which is transformed by compressing the height of raw point cloud.

Figure 2 also presents the filtering results obtained by performing the customized average pooling operation on the sparse input features. It clearly shows that both input features (second column) and SAM (fourth column) generated from the customized average pooling share the same sparsity pattern. We can conclude that the customized average pooling not only preserves the same sparsity pattern possessed in the original point cloud but also incorporates the kenel weight information in the SAM. Similarly, we can generate a SAM for any intermediate layers of DNNs to visualize important features. We aim to generate SAM for 3D feature maps and 2D feature maps, which have been shown to contain high-level semantic information, to visualize the semantic information corresponding to each object class and understand the effect of different neurons to capture important features of different objects.

4. Experiments

We choose three state-of-the-art volumetric-based models to illustrate the visualization results at three layers, namely, the 3D Convolution layer, the 2D Convolution layer, and the final classification layer. Specifically, we choose SECOND[22], Part-A^2[29], and PVRCNN[23] to illustrate their visualization results due to their superior performance in 3D object detection shown in Table 1 and the use of the same backbone as their network architecture. All

three models use different anchors (sliding windows of different sizes and orientations) to locate 3D objects of different sizes at different directions. These visualization results can effectively compare the subtle differences resulted from different anchors and the additional features and distinct loss functions incorporated into each model.

In subsection 4.1, we first present the car's Classification SAM of the three volumetric-based models to visually compare their performance in capturing car's semantic information by considering multiple anchors and individual anchors. We then present the Classification SAM of the three models for three objects (cars, pedestrians, and cyclists) to visually compare their performance in capturing the semantic information of three objects by considering multiple anchors. In subsection 4.2, we present the car's SAM of the best volumetric-based model, PVRCNN, generated from 3D feature maps and 2D feature maps to compare the effect of different neurons and provide more insights on their efficacy to capture cars' important features.

4.1. Visualizing the Classification SAM of 3D Point Clouds

The top row in Figure 3 shows the visualization of an object class (e.g., a car) in the final SAM generated from the last classification convolution layer of three volumetric-based models by combining six anchors with different orientations. It clearly shows that all three models generate a Classification SAM containing six high activations (highest likelihood to contain a car) shown in red within six ground-truth bounding boxes of cars. The highest activities also occur at the center of the ground-truth bounding box. However, all three models also show some moderate activations (moderate likelihood to contain a car) along road areas. For instance, there is one wrong high likelihood detection result as shown in red perpendicular to the road direction for all three models. There are three wrong moderate likelihood detection results as shown in light blue along the road direction for all three models. Part-A^2 has one more wrong detection result with low activations shown in dark blue that has around 45 degree intersection with the road direction. The comparison of these visualization results shows that the high activation areas identified in Classification SAM perfectly match with the correct car detection results for all three models. However, moderate activation areas identified in Classification SAM may correspond to the false positive detection. The more moderate activation areas, the lower the detection accuracy. As a result, we can use the Classification SAM to quickly understand which model tends to achieve better detection results in terms of true positives and false positives based on the activation areas.

In addition, Figure 3 also shows the visualization of cars in the SAM at the last classification convolution layer for two anchors of a similar size as the car and different orientations. The middle row illustrates the Classification SAM of three models for a proper anchor with a similar size and direction as the size and direction of the car. It clearly shows that all three models generate a Classification SAM containing high activation shown in red within the ground-truth bounding box of the car. However, SECOND also shows a lot of wrong moderate activities as shown in light blue along the road and outside the point clouds. Part-A^2 shows a few wrong moderate activities as shown in light blue along the road and few wrong moderate activities outside the point clouds. The bottom row in Figure 3 illustrates the Classification SAM of three methods for another anchor with a similar size of the car and a different direction. It clearly shows that SECOND generates a Classification SAM containing a lot of wrong high activations shown in red outside the ground-truth bounding box of the car. Part-A^2 generates some wrong high activities outside the ground-truth bounding box of the car. PVRCNN generates a small wrong moderate activity shown in light blue outside the ground-truth bounding box of the car.

	car	pedestrian	cyclist
SECOND	78.62	52.98	67.15
Part-A^2	79.40	60.05	69.90
PVRCCN	83.61	57.90	70.47

Table 1. Performance comparison of 3D object detection results of three models on the moderate level of KITTI validation data set containing cars, pedestrians, and cyclists. Mean Average Precision (mAP) is calculated by 11 recall positions.

For both scenarios of different anchors, SECOND tends to have the highest false positives, Part-A^2 tends to have moderate false positives, and PVRCNN tends to have no false positives for this point cloud data. This visualization provides more explanation about the semantics of the 3D objects in the classification convolution layer. In other words, SECOND seems to provide a lot of false positives due to a lot of high activities areas in the Classification SAM. PVRCNN seems to provide the least amount of false positives due to high activities areas at the location of the car. This observation matches with the car detection results reported in Table 1, where PVRCNN achieves the highest detection precision of 83.61% and SECOND achieves the lowest detection precision of 78.62%. It is interesting to observe that the wrong detection result for Part-A^2, that has around 45 degree intersection with the road direction, does not bring up moderate activations in Classification SAM for the two anchors. We think that there may be some issues in Part-A^2 when combining the detection results from multiple anchors.

Figure 4 shows the visualization of three object classes (car, pedestrian, and cyclist) in the final SAM at the last classification convolution layer of three volumetric-based models. The ground truth includes three cars in the hori-

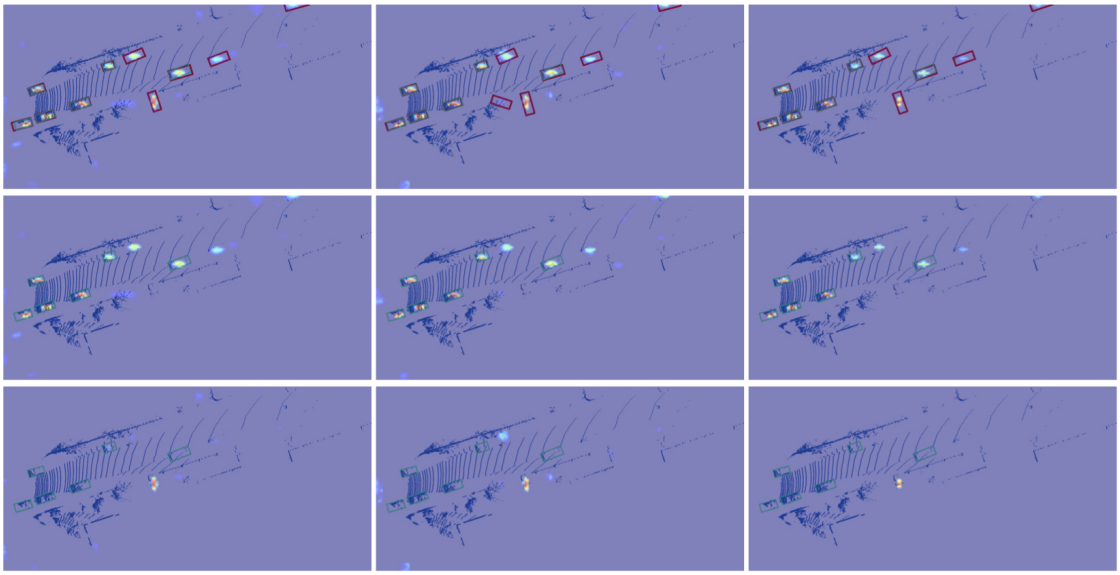

Figure 3. Illustration of the car's final Classification SAM of three state-of-the-art models (top row) and the car's Classification SAM obtained from two kinds of anchors overlaid on top of the point cloud (middle row: an anchor of approximately same size and same orientation of the car; bottom row: an anchor of approximately same size and different orientation of the car). The ground-truth bounding boxes are shown in green and the detection bounding boxes are shown in red.

zontal direction along the road, one pedestrian in the lower section of the point cloud data, and one cyclist in the upper left section of the point cloud data with an orientation of around 135 degree. The visualization results for three kinds of 3D objects show that all three models have high activation areas, which correctly correspond to the 3D objects in the point cloud data. However, there are moderate activation areas in Classification SAM for all three models, which do not correspond to the objects and may lead to false positive detection results. For example, for the car point cloud data, SECOND and Part-A^2 tend to have high false positives due to many moderate activation areas shown in Classification SAM along the road and outside point clouds. PVRCNN tends to have the lowest false positives due to three wrong moderate non-horizontal activation areas shown in Classification SAM along the road. For the pedestrian point cloud data, SECOND tends to have high false positives since it has many high activity areas shown in Classification SAM along the road and outside point clouds. Part-A^2 tends to have the lowest false positives since it has two high activity areas in Classification SAM. For the cyclist point cloud data, SECOND has the highest false positives and PVRCNN has no false positives. All these observations based on the activation areas in the Classification SAM are in align with the detection performance reported in Table 1.

To further demonstrate the details of Classification SAM, we show the blow-up results of SECOND, Part-A^2, and PVRCNN within the ground-truth bounding box of car, pedestrian, and cyclist in Figure 5. It is clear that all three models present high activation shown in red near the center of each ground-truth bounding box. So we can compare high activation areas in Classification SAM for different models to estimate their effectiveness to represent each object and their level of false positives.

4.2. Visualizing the SAM of PVRCNN at Intermediate Layers

In the following, we will only present the SAM of PVRCNN to illustrate its effectiveness from another perspective since it outperforms both SECOND and Part-A^2 to detect 3D objects. Figure 6 presents the 2D scatter image containing six car ground-truth bounding boxes and the SAM of the 3D feature map generated from the last layer of 3D convolution operations of PVRCNN. This 2D scatter image is projected from the original 3D point cloud along the vertical axis (z-value). In this example, we display five sections of the 3D feature map based on its depth. Each section contains individual parts of the whole point cloud. For example, the first section does not seem to contain any information since the SAM does not contain any noticeable activities. The second section mostly contains road information with little activities in four ground-truth bounding boxes, which indicates that some objects may be captured in this section. The third section mostly contains car information since there are significant activities within each of six ground-truth bounding boxes, which indicates that cars

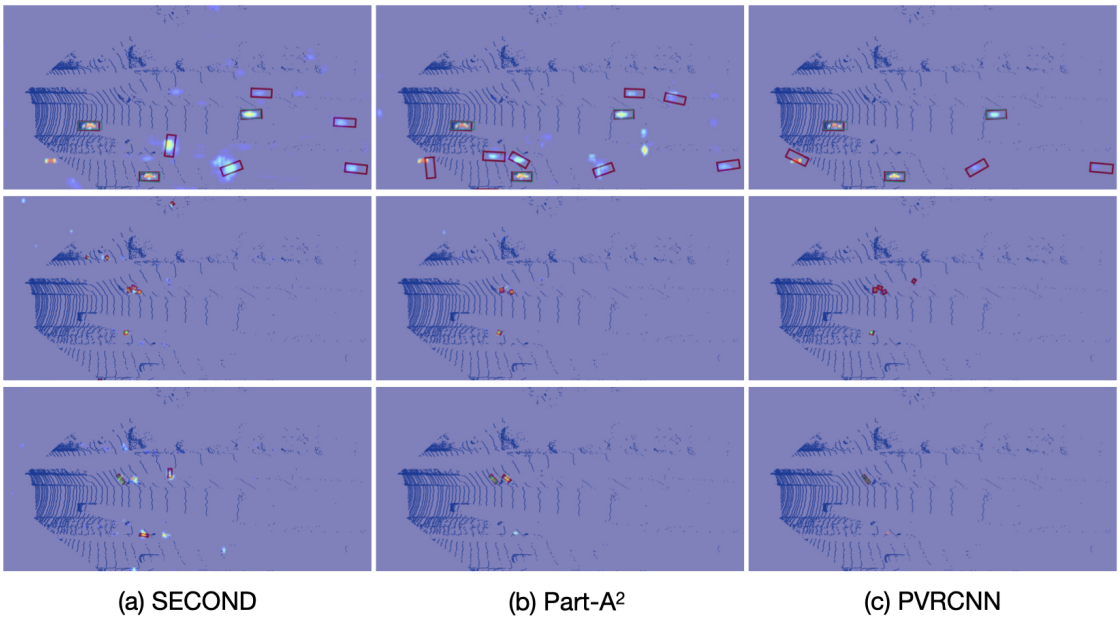

Figure 4. Illustration of the Classification SAM of three state-of-the-art models for three objects (Top row: car; middle row: pedestrian; bottom row: cyclist) overlaid on top of the point cloud. The ground-truth bounding boxes are shown in green and the detection bounding boxes are shown in red.

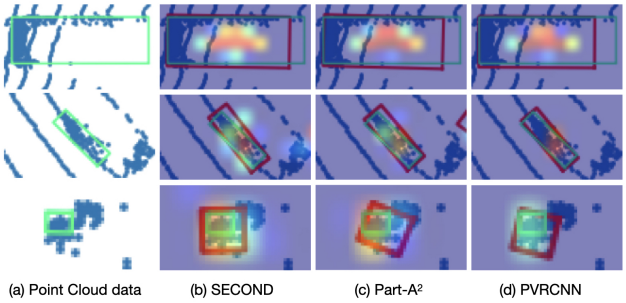

Figure 5. Illustration of the details of Classification SAM for three state-of-the-art models within ground-truth bounding boxes for three objects (Top row: car; middle row: pedestrian; bottom row: cyclist). The ground-truth bounding boxes are shown in green and the detection bounding boxes are shown in red.

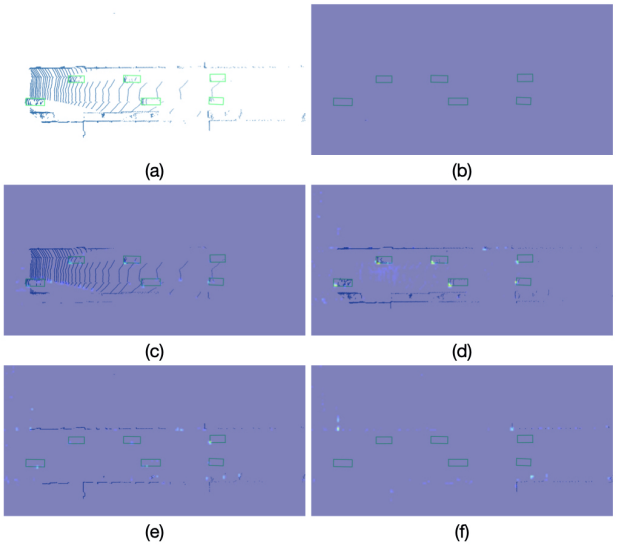

Figure 6. Illustration of the SAM of the 3D feature maps of PVRCNN overlaid on top of the point cloud (a) original 2D scatter image with six cars and their ground truth locations; SAM overlaid on top of each layer of the 3D feature maps: (b) first layer, (c) second layer, (d) third layer, (e) fourth layer, and (f) fifth layer

are captured in this section. Little activities in SAM are captured in the fourth and fifth sections. These visualization results show that SAM of the third section is more prone to capture cars at their precise locations. In other words, the third section is more effective than other sections for car classification.

Figure 7 presents the SAM of the 2D feature map generated from the second to the last layer of 2D convolution operations of PVRCNN. In this example, we randomly choose two 2D feature maps to show each map provides different activities in its SAM. For example, there are strong activities around the car shown in the SAM on the left and there are strong activities around the road regions and car shown in the SAM on the right. It shows that different SAMs are able to show the effect of different neurons to capture different objects. This will provide some insights regarding to the effectiveness of each neuron so a more powerful DNN can be constructed by eliminating non-effective neurons.

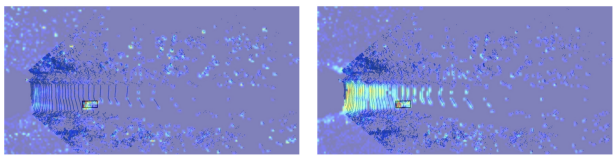

Figure 7. Illustration of SAM of two 2D feature maps of PVRCNN overlaid on top of the point cloud (left) capturing car information (right) capturing road and car information. The ground-truth bounding boxes are shown in black

5. Conclusions

In this work, we propose an instance-wise explanation technique on any volumetric-based DNNs to generate a Sparse Activation Map (SAM) to visually explain the 3D object detection models without a need for architecture change or re-training. The SAM has the ability to highlight the important regions of the 3D point cloud data and simultaneously maintain the sparsity of point clouds owing to the proposed customized average pooling of convolutional layer weights. The SAM is also applicable to comparing any volumetric-based networks via intuitive intermediate visualizations at different layers and insights of neuron effectiveness. Specifically, the SAM helps to gain understanding of the complex DNN structures at the 3D and 2D convolutional layers. It helps to understand the true positives and false positives for the detection results of each object class, which directly correlate with the model performance, at the classification layer. Our SAMs on KITTI dataset demonstrate the visual explanations match detection results of three state-of-the-art volumetric-based models. For wider applications, the SAM can easily be generalized to any sparse 3D object detection or classification tasks. The classification SAMs maintain the data sparsity pattern and incorporate the kernel weights to understand the high-level semantic of each object. As a result, they can aid researchers in understanding the DNN's network structures and efficiency in capturing important features of the objects and increase researchers' confidence in deploying DNNs in 3D object detection.

References

[1] Ziv Goldfeld, Ewout van den Berg, Kristjan Greenewald, Igor Melnyk, Nam Nguyen, Brian Kingsbury, and Yury Polyanskiy. Estimating information flow in deep neural networks. *arXiv preprint arXiv:1810.05728*, 2018. 2

[2] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What you see is what you get: Exploiting visibility for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11001–11009, 2020. 2

[3] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review*, 37:100270, 2020. 2

[4] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2673–2682. PMLR, 2018. 1

[5] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017. 2

[6] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015. 1, 2

[7] Christian W Omlin and C Lee Giles. Extraction of rules from discrete-time recurrent neural networks. *Neural networks*, 9(1):41–52, 1996. 2

[8] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 2

[9] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 2

[10] Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019. 2

[11] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 2

[12] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4548–4557, 2018. 2

[13] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 2

[14] 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*, 2020. 2

[15] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017. 2

[16] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017. 2

[17] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 2

[18] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019. 2

[19] Gail Weiss, Yoav Goldberg, and Eran Yahav. Extracting automata from recurrent neural networks using queries and counterexamples. In *International Conference on Machine Learning*, pages 5247–5256. PMLR, 2018. 2

[20] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2

[21] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 2

[22] 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

[23] Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 1