FFAM: Feature Factorization Activation Map for Explanation of 3D Detectors

Shuai Liu, Boyang Li, Zhiyu Fang, Mingyue Cui, Kai Huang[*]
School of Computer Science and Engineering, Sun Yat-sen University
{liush376@mail2, liby83@mail, fangzhy9@mail2, cuimy@mail2, huangk36@mail}.sysu.edu.cn

Abstract

LiDAR-based 3D object detection has made impressive progress recently, yet most existing models are black-box, lacking interpretability. Previous explanation approaches primarily focus on analyzing image-based models and are not readily applicable to LiDAR-based 3D detectors. In this paper, we propose a *feature factorization activation map* (FFAM) to generate high-quality visual explanations for 3D detectors. FFAM employs non-negative matrix factorization to generate concept activation maps and subsequently aggregates these maps to obtain a global visual explanation. To achieve object-specific visual explanations, we refine the global visual explanation using the feature gradient of a target object. Additionally, we introduce a voxel upsampling strategy to align the scale between the activation map and input point cloud. We qualitatively and quantitatively analyze FFAM with multiple detectors on several datasets. Experimental results validate the high-quality visual explanations produced by FFAM. The code is available at `https://github.com/Say2L/FFAM.git`.

1 Introduction

In recent years, there has been rapid development in LiDAR-based 3D object detection [36, 32, 34, 38, 12], making it widely utilized in autonomous driving, industrial automation, and robot navigation. However, existing detection methods predominantly rely on deep neural networks with highly nonlinear and complex structures. Essentially, these models can be considered as "black box" systems. Such opaque modeling techniques hinder users from fully trusting the detection models, particularly in sensitive and high-risk domains. Consequently, understanding the decision-making process of these inherently opaque models is urgently needed.

Visual explanation methods [20, 1, 33, 15, 16] have gained widespread adoption for analyzing models based on deep neural networks. These methods generate saliency maps that highlight the crucial elements influencing the model's decision within the input map. Perturbation-based [15, 16], class activation map (CAM)-based [41, 20, 1], and gradient-based [28, 24, 25] methods are the three main categories of visual explanation methods. However, these methods primarily focus on image-based models and are not directly applicable to point cloud-based models. The pioneering work in analyzing 3D detectors is OccAM [19], which extends D-RISE [16] to perturb point clouds. As a perturbation-based approach, OccAM first randomly samples numerous sub-point clouds and measures the change in model predictions. However, the large number of inference calculations makes OccAM computationally intensive, and the sampling number easily impacts the quality of generated saliency maps.

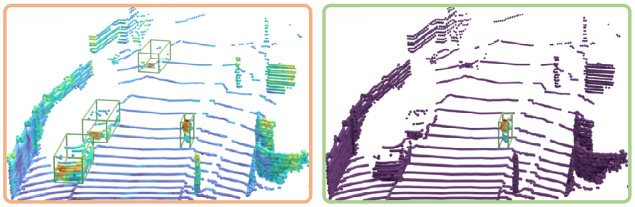
Interpreting 3D detectors presents three key challenges. First, point clouds are inherently three-dimensional (3D). It is essential to generate corresponding 3D saliency maps for accurate interpreta-

[*] Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

tion. However, existing methods, such as popular CAM-based techniques, primarily utilize activation maps from the network's last layer to generate 2D saliency maps. Second, the explanation method for 3D detectors should provide detailed explanations for individual objects of interest. Yet, most existing methods yield class-specific saliency maps, which means they cannot focus on explaining a specific detection object. Lastly, point clouds are sparsely distributed in 3D space, rendering linear interpolation employed by many image-based explanation methods ineffective.

To address the aforementioned challenges, this paper introduces a feature factorization activation map (FFAM) to obtain visual explanations for 3D detectors. Specifically, to solve the first challenge, FFAM leverages the 3D feature maps within the 3D backbone [36] of detectors, rather than relying on the bird's eye view (BEV) feature maps from the last layer. Drawing inspiration from DFF [3], we employ non-negative matrix factorization (NMF) [11] to uncover latent semantic concepts within these 3D feature maps. Typically, point features with effective detection clues in 3D detectors contain richer semantic concepts. Thus, we aggregate concept activation maps generated by NMF to obtain a global concept activation map that highlights important points, as shown in Figure 1(a). To address the second challenge of obtaining object-specific saliency maps, we utilize the gradients of the 3D feature map, generated by an object-specific loss, to refine the global concept activation map. This process is illustrated in Figure 1(b), showcasing the desired effect. To tackle the final challenge, we introduce a voxel upsampling strategy to sample values from sparse neighbors, ensuring accurate saliency map generation.



(a) (b)

Figure 1: Visualization of FFAM outputs. (a) global concept activation map and (b) object-specific activation map.

We compare our FFAM with the current state-of-the-art method OccAM [19], as well as other image-based explanation methods including Grad-CAM [20] and ODAM [39]. We conduct experiments on the KITTI [7] and Waymo Open [26] datasets, employing detectors such as SECOND [36] and CenterPoint [37]. The qualitative and quantitative results demonstrate that our FFAM significantly outperforms the previous methods. The contributions of this work can be summarized as follows:

- We propose a feature factorization activation map (FFAM) method to obtain high-quality visual explanations for 3D detectors.
- We first introduce NMF in explaining point cloud detectors. By aggregating different concept activation maps, we obtain a global concept activation map that highlights points with significant detection clues.
- We utilize feature gradients of an object-specific loss to refine the global concept activation map, enabling the generation of object-specific saliency maps.
- A voxel upsampling strategy is proposed to upsample sparse voxels, thus aligning the scale between the activation map and input point cloud.

2 Related Work

Explanation Methods for Image-based Models. Existing explanation methods primarily focus on image classification models. Perturbation-based methods [27, 15, 4, 31] are widely used for interpreting image classification models. The core idea is to assign importance scores to perturbed feature components by disturbing the model's input and observing the output changes. CAM-based methods [41, 20, 1, 9] generate saliency maps by linearly combining activation maps from intermediate layers, weighted by their respective contributions. Some approaches (e.g. Score-CAM [33] and Ablation-CAM [18]) combine perturbation- and CAM-based ideas to eliminate dependence on backpropagation gradients. Additionally, gradient-based explanation methods [23, 28, 24, 25] use gradients to quantify input impact on network predictions. Higher gradient values indicate greater importance of the corresponding input elements. Moreover, feature factorization techniques like principal component analysis (PCA) and non-negative matrix factorization (NMF) can uncover latent patterns in deep features. DFF [3] employs NMF to localize semantic concepts within images.

Compared to explanation methods for classifiers, only a limited number of approaches investigate explanations for object detection models. The aforementioned methods generate class-specific explanations, which are not feasible for object detection models. D-RISE [16] employs a perturbation strategy to generate instance-specific explanations by defining a detection similarity metric. In [35], a directed acyclic AND-OR Graph (AOG) is utilized to uncover latent structures in object detectors. G-CAME [14] combines activation maps with a Gaussian kernel of gradients to generate a saliency map for a predicted bounding box. ODAM [39] employs pixel-wise gradients of a target object to weigh the activation maps, thereby producing an instance-specific saliency map.

Explanation Methods for Point Cloud-based Models. In contrast to explanation methods for image-based models, the field of explanation for point cloud-based models is relatively underdeveloped. Existing methods primarily focus on point cloud classification models. For instance, [40] utilizes the loss gradient to measure the contribution of each point in the classifier. Similarly, [8] applies a gradient-based strategy to analyze the intermediate features of the network. Another approach [29] combines a generative model with the activation maximization method [6] to obtain a global explanation for point cloud networks.

Research on the explanation of 3D detectors is still quite limited. One perturbation-based method, OccAM [19], estimates the importance of individual points by testing the model with randomly generated subsets of the input point cloud. However, the scale of points in 3D space is considerably large, and the distribution of points acquired through LiDAR varies with distance. These aforementioned issues result in the following challenges for perturbation-based methods: (1) It is difficult to exhaustively perturb the point cloud, limiting the quality of visual explanations; (2) Generating ample random subsets of points requires multiple iterations, thereby reducing efficiency. Taking inspiration from feature factorization techniques [3] and gradient-based approaches [28, 14, 39], we propose an explanation method called FFAM. It aims to efficiently generate high-quality saliency maps for 3D detectors.

LiDAR-based 3D Object Detection. These methods can be categorized into two main groups: one-stage and two-stage detectors. **One-stage detectors** typically employ simple network architectures to achieve high speeds. For instance, SECOND [36] efficiently encodes sparse voxel features using a proposed 3D sparse convolution technique. PointPillars [10] divides a point cloud into pillar voxels, eliminating the need for 3D convolution layers and achieving fast inference speed. VoxelNeXt [2] introduces a fully sparse convolution network that eliminates the requirement for sparse-to-dense conversion. **Two-stage detectors** generally incorporate an additional stage to refine proposals generated by a one-stage network. PointRCNN [22] utilizes PointNet++ [17] to generate proposals from raw points and then refines the bounding boxes in the second stage. PV-RCNN [21] combines a voxel-based proposal network with a point-based refinement network. CenterPoint [37] extracts point features from the surface centers of proposal bounding boxes for refinement. Voxel R-CNN [5] utilizes voxel features from the 3D backbone to refine the proposals. Our explanation method FFAM is adaptable to both one- and two-stage detectors without being limited by the detector type. We primarily conduct experiments on widely used detectors, including the one-stage detector SECOND and the two-stage detector CenterPoint.

3 Method

The goal of visual explanation for a 3D detector f is to produce a saliency map for each detection. Given a point cloud $P \in \mathbb{R}^{N\times 4}$, the saliency map consists of N values presenting the importance of each point in P for a detection d which consists of a bounding box, a confidence score and a category label. We denote a detection d as follows:

$$d = [x, y, z, l, w, h, s, c] \tag{1}$$

where (x, y, z) denotes the center location, (l, w, h) represents the object size (i.e., length, width and height), s and c indicate the confidence score and category label, respectively.

We propose FFAM to produce saliency maps in point cloud format for 3D detectors. The overview of our method is illustrated in Figure 2. It can be divided into three phases as follows: (1) Feature factorization (Sec. 3.1); (2) Gradient weighting (Sec. 3.2); (3) Voxel upsampling (Sec. 3.3).

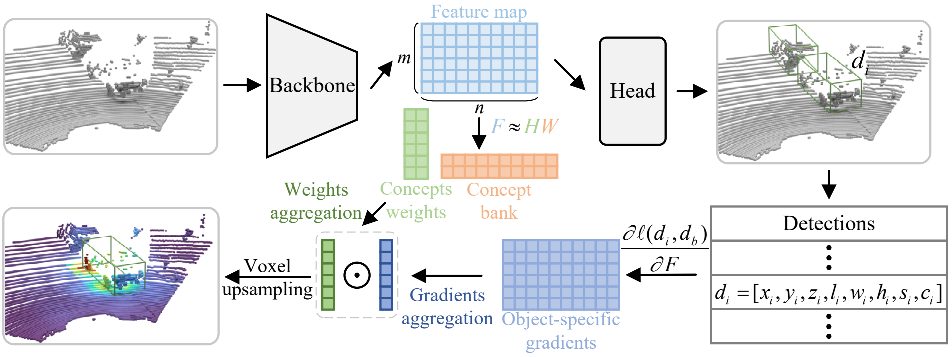


Figure 2: Overall framework of our FFAM which can generate an object-specific saliency map for a detection d_i.

3.1 Feature Factorization Activation Map

Matrix factorization is widely used in fields such as recommendation systems, image processing, and natural language processing to extract potential features and reduce dimensionality. Non-negative matrix factorization (NMF) as a classical matrix factorization algorithm approximates a non-negative matrix by decomposing it into the product of two non-negative matrices. With this decomposition, NMF can discover potential patterns and conceptions in the raw matrix and extract the most important features. Given a non-negative matrix $A \in \mathbb{R}^{m \times n}$, NMF retrieves an approximation $\hat{A} \in \mathbb{R}^{m \times n}$ as follows:

$$\begin{aligned} \mathrm{NMF}(A) = \underset{\hat{A}}{\operatorname{argmin}} \left\| A - \hat{A} \right\|_F^2, \\ \text{s.t. } \hat{A} = HW, \forall ij, H_{ij}, W_{ij} \geq 0, \end{aligned} \tag{2}$$

where $H \in \mathbb{R}^{m \times r}$ and $W \in \mathbb{R}^{r \times n}$ denote two non-negative matrics. r is a predefined parameter indicating the number of latent concepts in matrix A. Each row $W_j \in \mathbb{R}^n (1 \leq j \leq r)$ of W represents a concept vector. These concept vectors are typically well-interpreted and associated with object-part features, such as wheels, car doors, car roofs, and so on, following the non-negative additivity property of W_j. Furthermore, each row $H_i \in \mathbb{R}^r$ (where $1 \leq i \leq m$) of matrix H represents the combination weights of different concept vectors in W. Combining these concept vectors using the weights H_i, we obtain the i-th row feature of matrix $\hat{A}$.

In this paper, we employ non-negative matrix factorization to handle the voxel feature map within the 3D backbone of detectors. Typically, voxel features that contain crucial detection clues tend to activate more concepts (e.g., license plates, car fronts, car edges) in detectors. As a result, aggregating all weights in H_i indicates the significance of the i-th voxel feature in the voxel feature map, as demonstrated in Figure 1(a).

Specifically, given a voxel feature map $F \in \mathbb{R}^{M \times d}$ where M represents the voxel number and d denotes the channel number, a voxel feature $F_i \in \mathbb{R}^d$ in F can be factorized as follows:

$$F_i = \sum_{j=1}^{r} H_{ij} W_j. \tag{3}$$

Further, we obtain the global concept activation map V by aggregating concept weight matrix H as follows:

$$V = \sum_{j=1}^{r} H_{\cdot j}, \tag{4}$$

where $H_{\cdot j}$ denotes j-th column of H. The resulting V emphasizes points with multiple activated concepts from a global perspective. And due to the downsampling operation in the detection network, the granularity of V is typically coarse. Therefore, further processing is required to obtain an object-specific and fine-grained activation map, as described in Sec. 3.2 and Sec. 3.3.

3.2 Object-Specific Gradient Weighting

In a 3D detector, the output contains a large number of detections. To obtain an object-specific activation map, we establish a loss function for a specific detection. Specifically, given a detection d, we create a baseline detection d_b to calculate the loss ℓ:

$$\ell = \|d - d_b\|_1 . \tag{5}$$

For simplicity, we use the L1 loss function and set all values in d_b equal to 0. Then we obtain the gradient map $G \in \mathbb{R}^{M \times d}$ of the feature map F:

$$G = \frac{\partial \ell}{\partial F}. \tag{6}$$

Considering an optimization process from d to d_b, the matrix G denotes the optimal direction for reducing the loss. If we iteratively update the feature map F based on the gradient map G, the information related to the detection d will be diminished. Alternatively, by utilizing G, we can identify the locations in the feature map F^l that contain clues about d. Consequently, an object-specific activation map M for d can be obtained as follows:

$$\begin{gathered} \omega = \sum_{k=1}^{d} \left| G_{\cdot k} \right| , \\ M = \Phi(\omega) \odot \Phi(V), \end{gathered} \tag{7}$$

where $G_{\cdot k} \in \mathbb{R}^M$ refers to the k-th column of G, while Φ represents the normalization operation, and $\odot$ denotes element-wise multiplication. By modifying the loss function to a specific attribute p in detection d, we can examine the specific points on which the detector concentrates when predicting attribute p.

3.3 Voxel Upsampling

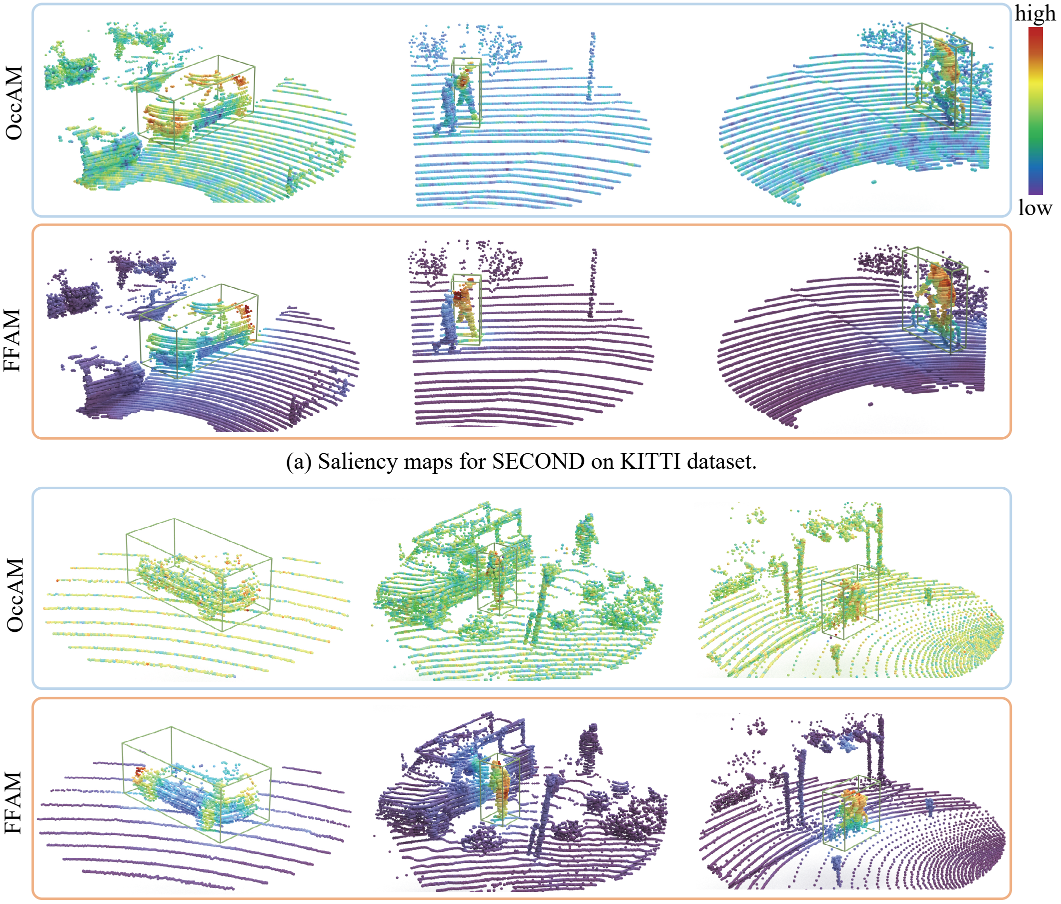
Due to downsampling operations in 3D detection networks, the scale of the activation map is typically smaller than that of the input point cloud. Consequently, upsampling the activation map M becomes necessary. However, unlike 2D images, linear interpolation for upsampling 3D sparse voxels presents challenges. To address this, we draw inspiration from the voxel query technique proposed by [5] and introduce a voxel upsampling strategy for 3D sparse voxels. Specifically, we define the voxel size as s, and the ranges of the point cloud for three axes as $[x_l, x_r]$, $[y_l, y_r]$, and $[z_l, z_r]$ respectively. Given a point $p = (x, y, z)$, we calculate the coordinate (x_p, y_p, z_p) of voxel v_p to which p belongs as follows:

$$x_p = \left\lfloor \frac{x - x_l}{s} \right\rfloor , \; y_p = \left\lfloor \frac{y - y_l}{s} \right\rfloor , \; z_p = \left\lfloor \frac{z - z_l}{s} \right\rfloor . \tag{8}$$

Then we query neighbor voxels on activation map M for p, using Manhattan distance to control the query range:

$$d(v_p, v) = |x_n - x_p| + |y_n - y_p| + |z_n - z_p|, \tag{9}$$

where (x_n, y_n, z_n) is the coordinate of an neighbor voxel v, $d(\cdot, \cdot)$ is the Manhattan distance between two voxels. We sample up to k neighbor voxels within a distance threshold. Finally, the salience score s_p of point p is calculated as follows:



(a) Saliency maps for SECOND on KITTI dataset.

(b) Saliency maps for CenterPoint on Waymo Open dataset.

Figure 3: Saliency maps for SECOND [36] and CenterPoint [37]. The green bounding boxes indicate the detected objects, while warmer colors (using the turbo colormap) represent higher point contributions to these detections. The crops are provided for visualization purposes only.

$$s_p = \sum_{v \in \aleph} \frac{\Psi(d(v_p, v))}{\sum_{v \in \aleph} \Psi(d(v_p, v))} M_v, \tag{10}$$

where $\aleph$ is the set of neighbor voxels, Ψ denotes a Gaussian kernel with a standard normal distribution, M_v represents the value of voxel v on activation map M.

4 Experiments

In this section, we compare our FFAM with existing explanation methods, including Grad-CAM [20] and ODAM [39], for image-based models, as well as with OccAM [19], the state-of-the-art explanation method for point cloud-based models. We adopt two datasets for evaluation: KITTI [7], a widely used autonomous driving dataset, and Waymo Open [26], containing complex multi-object scenes. For KITTI, experiments are conducted on SECOND [36]. For Waymo Open, we mainly evaluate on CenterPoint [37]. The experiments are run using PyTorch and an RTX 3090 GPU. The hyperparameters of detectors and OccAM remain consistent with their official implementations. The parameter r used in NMF is set to 64. The Manhattan distance threshold and parameter k in voxel upsampling are set to 2 and 16, respectively. We use the 3D feature map from the third block of the 3D backbone as FFAM input. Hyperparameters analysis and ablation study are in App. A.1 and App. A.4, respectively.

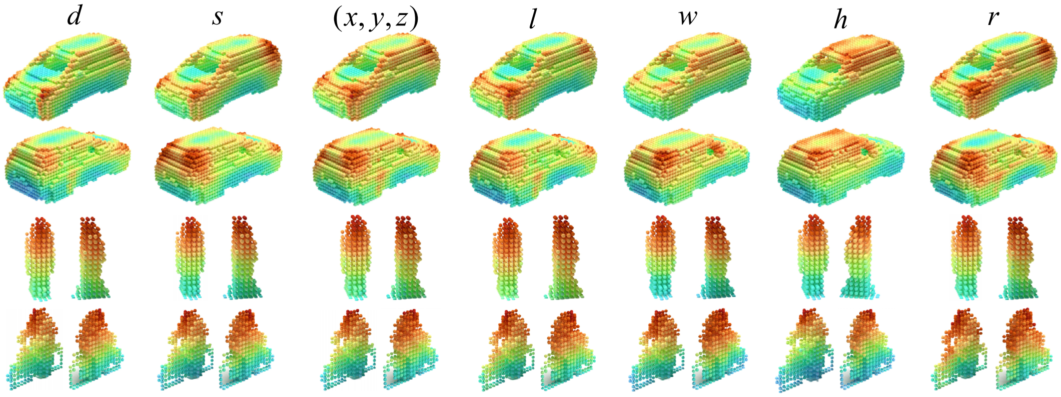


Figure 4: Average saliency maps for different object attributes. (x, y, z) denotes the center of predicted object. l, w, h, r and s represent the length, width, height, rotation angle and classification score of predicted object, respectively. d indicates the combination of all attributes.

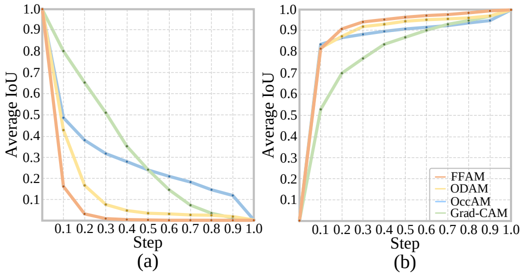
4.1 Qualitative Results

To verify the interpretability of our FFAM, we visualize explanations for some objects. We also visualize the average saliency maps of different categories for specific object attributes to study the latent pattern of 3D detectors.

Visualization of Saliency Map. We compare the visual explanations generated by FFAM and OccAM [19] for cars, pedestrians, and cyclists in Figure 3(a). These detection results are obtained by SECOND [36] detector trained on KITTI [7]. OccAM exhibits significant background noise due to its random masking mechanism. In contrast, our FFAM demonstrates a strong ability to generate clear, distinct object-specific saliency maps. We observe the detector also captures relevant clues from the background and neighboring objects. Furthermore, we compare saliency maps generated by FFAM and OccAM on Waymo Open [26] using the CenterPoint [37] detector, as shown in Figure 3(b). The saliency maps produced by OccAM struggle to focus on the intended object for interpretation. They have more highly salient points distributed on the background compared to KITTI. We attribute this discrepancy to the larger number of points in Waymo Open samples, challenging the random masking mechanism to sample diverse point masks effectively. Conversely, our FFAM consistently generates high-quality saliency maps on Waymo Open.

Average Saliency Map. To further explore the detection mode of detectors and verify the interpretability of FFAM, we average the saliency maps of specific classes, including cars, pedestrians and cyclists. We use SECOND trained on KITTI [7] as the detector. To accomplish this, we first scale all boxes and associated points to a uniform size and then align them with respect to their center and rotation angle. Next, we voxelize the resulting point cloud and calculate the average saliency values of individual points within each voxel. The resulting saliency maps for different object attributes are presented in Figure 4.

As depicted in the first two rows of Figure 4, the detector primarily identifies and localizes car objects based on the points located at the four corners of the car. By analyzing features from these points, the detector infers various attributes of a car, such as its center location, length, width, rotation angle, and classification score. This can be attributed to the fact that car objects are often incomplete in outdoor point clouds, and their corners are frequently scanned by LiDAR and used as key features. However, there is a special case, as shown in the first two rows of the penultimate column of Figure 4, where the height attribute is predicted primarily based on the points at the top of the car. As illustrated in the third row of Figure 4, the detector predicts pedestrian objects mostly based on the points distributed on the head and shoulder regions. Additionally, the detector recognizes cyclist objects mainly based on the points distributed on the head and back of the human body, as shown in the last row of Figure 4. Furthermore, we observe that the prediction of cyclist height heavily relies on the points distributed on the head, similar to the prediction of car height. Additional average saliency maps of other detectors are in App. A.2.



(a)

(b)

Figure 5: AUC diagrams for Deletion and Insertion. Average IoU vs. (a) Deletion steps and (b) Insertion steps.

Method	VEA ↑			
	All	Car	Ped.	Cyc.
Grad-CAM	0.015	0.015	0.018	0.013
ODAM	0.179	0.163	0.280	0.233
OccAM	0.064	0.063	0.080	0.042
FFAM (Ours)	**0.391**	**0.363**	**0.543**	**0.515**

Table 1: Comparison of visual explanation accuracy metric for different categories. 'all' denotes the three categories are included.

Method	Deletion ↓				Insertion ↑			
	All	Car	Ped.	Cyc.	All	Car	Ped.	Cyc.
Grad-CAM	0.335	0.373	0.137	0.129	0.797	0.821	0.688	0.725
ODAM	0.134	0.138	0.122	0.098	0.885	0.902	0.785	0.828
OccAM	0.286	0.311	0.146	0.167	0.863	0.880	0.761	0.790
FFAM (Ours)	**0.071**	**0.068**	**0.098**	**0.078**	**0.907**	**0.923**	**0.806**	**0.854**

Table 2: AUC for Deletion and Insertion curves. The results of different categories are reported. 'all' means the combination of the three categories.

4.2 Quantitative Results

We adopt Deletion, Insertion [15, 16, 39], visual explanation accuracy (VEA) [13] and Pointing games (PG) to evaluate our FFAM. SECOND trained on KITTI is used as the baseline detector. Following previous work [39], we use the well-detected objects in the evaluation dataset as the subjects to be explained. In particular, a predicted object is considered well-detected if the IoU between it and its ground truth is greater than [0,7, 0.5, 0.5] for car, pedestrian and cyclist classes, respectively. See App. A.3 for results on Waymo Open.

Deletion and Insertion. Deletion and Insertion are widely used to evaluate explanation methods for image-based detection models [16, 39]. Deletion involves sequentially removing highly salient elements from a scene, measuring the rate model predictions diverge from the original. Insertion progressively adds salient elements to an empty scene, measuring how quickly predictions approach the original. Considering the similarity between pixels in an image and points in a point cloud, we employ Deletion and Insertion to evaluate FFAM. In outdoor point cloud scenes, objects are relatively small compared to global scenes, so we only operate on points within twice the diagonal length of an object's bounding box from its center. We use IoU between a prediction and ground truth as the measure score. Average IoU curves are presented in Figure 5(a-b), and Table 2 reports the area under the curve (AUC) for different categories. A lower Deletion AUC indicates a steeper drop in the IoU score, reflecting a more pronounced impact of removed salient points. Conversely, a higher Insertion AUC suggests a larger increase in the IoU score per step, indicating the significance of added salient points. Our methods have the fastest performance drop and largest increase for Deletion and Insertion, showing points highlighted in our saliency maps have a greater effect on detector predictions than the other methods.

Visual Explanation Accuracy. VEA calculates the point-level intersection over union (IoU) between the ground truth masks and saliency maps, which are thresholded at various values. The results of VEA for different object categories can be found in Table 1. Notably, our FFAM achieves the highest VEA scores across all categories, indicating the compactness of the visual explanations generated by FFAM. On the other hand, OccAM and Grad-CAM exhibit lower performance on this metric. OccAM tends to mark a significant number of background points, while Grad-CAM is a class-specific visual explanation method, which may explain their comparatively weak performance.

Pointing Game. To further assess the localization capability of FFAM, we present the results of the Pointing game (PG). In this evaluation, a hit is recorded if the point with the highest saliency value falls within the ground truth bounding box, while a miss is counted otherwise. The PG metric

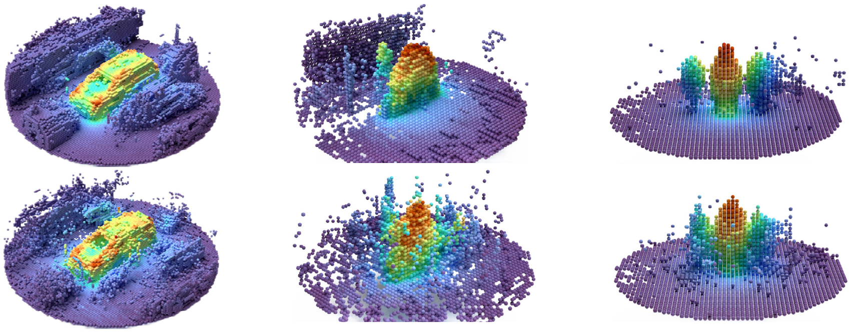


Figure 6: Average saliency maps for true and false positives. The 1^{st} and 2^{nd} rows represent cases of true and false positives, respectively.

Method	PG ↑				enPG ↑			
	All	Car	Ped.	Cyc.	All	Car	Ped.	Cyc.
Grad-CAM	0.093	0.080	0.166	0.163	0.021	0.022	0.014	0.011
ODAM	0.901	0.895	0.939	0.926	0.633	0.639	0.577	0.654
OccAM	0.946	0.957	0.898	0.860	0.023	0.024	0.019	0.013
FFAM (Ours)	**0.991**	**0.989**	**0.999**	**0.998**	**0.664**	**0.671**	**0.591**	**0.719**

Table 3: Comparison of Pointing game (PG) and energy-based Pointing game (enPG) metrics.

measures the accuracy of saliency maps by calculating the ratio of hits to the total number of hits and misses. Furthermore, we report the energy-based PG metric (enPG) proposed in [33], which considers the energy within the ground truth region compared to the global scene. As shown in Table 3, our FFAM surpasses previous methods on all metrics, indicating its superior ability to focus on the explained object. Notably, Grad-CAM performs poorly on both PG and enPG, which aligns with the VEA results presented in Table 1. This suggests that classification-based explanation methods alone are insufficient for generating meaningful explanations for detectors.

4.3 Modes of False Positive

FFAM can be used to identify false positive modes of a detector. A detection is considered a true positive if correctly classified and the Intersection over Union (IoU) between the prediction box and ground truth exceeds a threshold. Otherwise, it is a false positive. The IoU thresholds are 0.7, 0.5, and 0.5 for car, pedestrian, and cyclist objects, aligning with the KITTI official metric [7]. To reveal detection modes, we compute average saliency maps separately for true positives and false positives. Results are shown in Figure 6. First, we observe the average saliency maps of false positives exhibit similarities to those of true positives. The detector predicts a false positive because it detects a similar pattern to that of a true positive. Second, false positives tend to be surrounded by more noise points, with a point density of approximately one-third of true positives. We believe noises and sparse density may be significant factors contributing to the occurrence of false positives. Lastly, the ratio of car, pedestrian, and cyclist objects in true positives is approximately 36:5:2, while in false positives, it is 13:8:2. This suggests car objects are less prone to false positives compared to pedestrian and cyclist objects.

5 Conclusion

In this paper, we propose a visual explanation method, FFAM, that efficiently generates high-quality explanations for 3D detectors. FFAM utilizes non-maximum matrix factorization to obtain a global concept activation map, which is then refined using object-specific gradients. To align the granularity of the input point cloud and intermediate features, we introduce a voxel upsampling strategy. Qualitative and quantitative experiments demonstrate that our FFAM provides more interpretable and compact visual explanations than previous methods. The limitation of FFAM is that it needs to access the feature maps within 3D detectors. In future work, we will explore using visual explanations to enhance the accuracy and efficiency of 3D detectors.

Acknowledgments

This work was supported by the Project of National Natural Science Foundation of China (No. 62232008) and Guangxi Key R & D Program (No. GuikeAB24010324).

References

[1] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*, 2018.

[2] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *CVPR*, 2023.

[3] Edo Collins, Radhakrishna Achanta, and Sabine Susstrunk. Deep feature factorization for concept discovery. In *ECCV*, 2018.

[4] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *NeurIPS*, 2017.

[5] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *AAAI*, 2021.

[6] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 2009.

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[8] Ananya Gupta, Simon Watson, and Hujun Yin. 3d point cloud feature explanations using gradient-based methods. In *IJCNN*, 2020.

[9] Peng-Tao Jiang, Chang-Bin Zhang, Qibin Hou, Ming-Ming Cheng, and Yunchao Wei. Layercam: Exploring hierarchical class activation maps for localization. *TIP*, 2021.

[10] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019.

[11] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *nature*, 1999.

[12] Shuai Liu, Boyang Li, Zhiyu Fang, and Kai Huang. Dcdet: Dynamic cross-based 3d object detector. In *IJCAI*, 2024.

[13] José Oramas M., Kaili Wang, and Tinne Tuytelaars. Visual explanation by interpretation: Improving visual feedback capabilities of deep neural networks. In *ICLR*, 2019.

[14] Quoc Khanh Nguyen, Truong Thanh Hung Nguyen, Vo Thanh Khang Nguyen, Van Binh Truong, and Quoc Hung Cao. G-came: Gaussian-class activation mapping explainer for object detectors. *arXiv preprint arXiv:2306.03400*, 2023.

[15] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[16] Vitali Petsiuk, Rajiv Jain, Varun Manjunatha, Vlad I Morariu, Ashutosh Mehra, Vicente Ordonez, and Kate Saenko. Black-box explanation of object detectors via saliency maps. In *CVPR*, 2021.

[17] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017.

[18] Harish Guruprasad Ramaswamy et al. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *WACV*, 2020.

[19] David Schinagl, Georg Krispel, Horst Possegger, Peter M Roth, and Horst Bischof. Occam's laser: Occlusion-based attribution maps for 3d object detectors on lidar data. In *CVPR*, 2022.

[20] 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 *ICCV*, 2017.

[21] 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 *CVPR*, 2020.

[22] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019.

[23] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[24] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[25] Suraj Srinivas and François Fleuret. Full-gradient representation for neural network visualization. In *NeurIPS*, 2019.

[26] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

[27] Youcheng Sun, Hana Chockler, Xiaowei Huang, and Daniel Kroening. Explaining image classifiers using statistical fault localization. In *ECCV*, 2020.

[28] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, 2017.

[29] Hanxiao Tan. Visualizing global explanations of point cloud dnns. In *WACV*, 2023.

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

[31] Jorg Wagner, Jan Mathias Kohler, Tobias Gindele, Leon Hetzel, Jakob Thaddaus Wiedemer, and Sven Behnke. Interpretable and fine-grained visual explanations for convolutional neural networks. In *CVPR*, 2019.

[32] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. Dsvt: Dynamic sparse voxel transformer with rotated sets. In *CVPR*, 2023.

[33] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *CVPRW*, 2020.

[34] Yingjie Wang, Jiajun Deng, Yuenan Hou, Yao Li, Yu Zhang, Jianmin Ji, Wanli Ouyang, and Yanyong Zhang. Club: Cluster meets bev for lidar-based 3d object detection. In *NeurIPS*, 2024.

[35] Tianfu Wu and Xi Song. Towards interpretable object detection by unfolding latent structures. In *ICCV*, pages 6033–6043, 2019.

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

[37] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *CVPR*, 2021.

[38] Gang Zhang, Chen Junnan, Guohuan Gao, Jianmin Li, and Xiaolin Hu. Hednet: A hierarchical encoder-decoder network for 3d object detection in point clouds. In *NeurIPS*, 2023.

[39] Chenyang Zhao and Antoni B. Chan. ODAM: gradient-based instance-specific visual explanations for object detection. In *ICLR*, 2023.

[40] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. Pointcloud saliency maps. In *ICCV*, 2019.

[41] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

A Appendix

Layer	Del. ↓	Ins. ↑	VEA ↑	PG ↑	enPG ↑
conv1	0.167	0.911	0.084	0.885	0.562
conv2	0.102	**0.912**	0.116	0.905	0.615
conv3	**0.091**	0.909	0.161	**0.955**	**0.654**
conv4	0.093	0.905	**0.208**	0.946	0.644

Table 4: Results of different layer settings. 'conv1', 'conv2', 'conv3' and 'conv4' represent the 1^{st}, 2^{nd}, 3^{rd} and 4^{th} blocks in the 3D backbone.

(Range, k)	Del. ↓	Ins. ↑	VEA ↑	PG ↑	enPG ↑
(0, 1)	0.091	0.909	0.161	0.955	0.654
(1, 4)	0.076	**0.910**	0.218	0.965	0.635
(2, 16)	**0.069**	0.909	**0.313**	**0.981**	**0.644**
(3, 64)	0.072	0.907	0.309	0.980	0.642

Table 5: Results of different (Range, k) settings. 'Range' means the Manhattan distance threshold and k denotes the upper bound of neighbor number in the voxel upsampling strategy.

r	Del. ↓	Ins. ↑	VEA ↑	PG ↑	enPG ↑
8	**0.067**	**0.909**	0.315	0.980	0.639
16	0.069	**0.909**	0.313	0.981	0.644
32	0.069	**0.909**	0.313	0.981	0.647
64	0.071	0.907	**0.391**	**0.991**	**0.664**
128	0.069	**0.909**	0.306	0.980	0.644

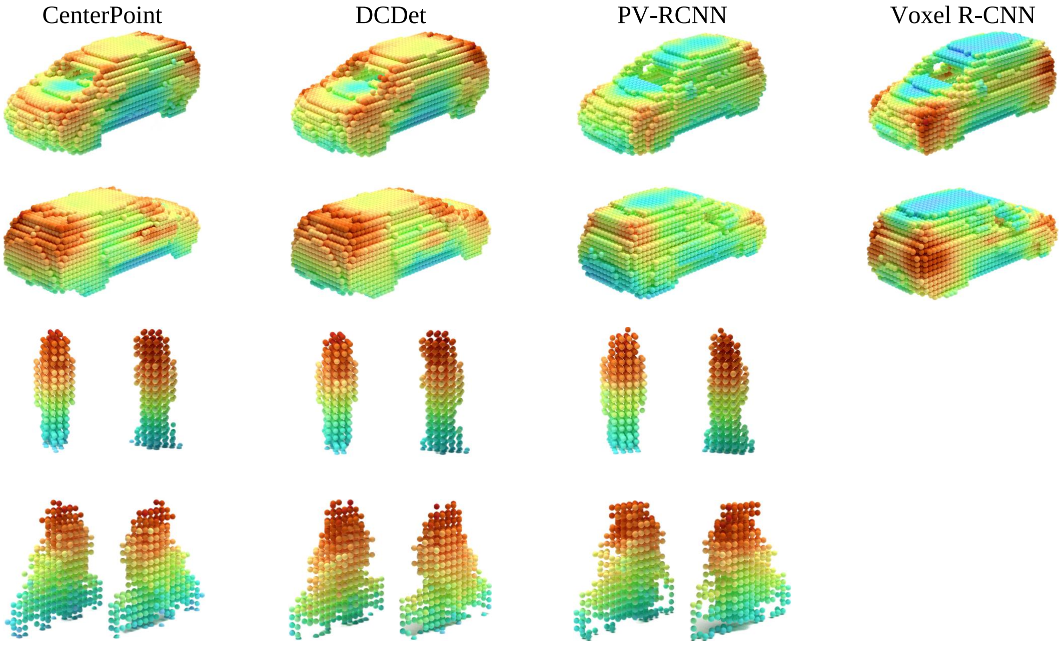
Table 6: Results of different concept number settings. r denotes the concept number.

A.1 Hyperparameters Analysis

In this section, we determine suitable hyperparameters including the feature map position, concept number r and sampling range for FFAM. Specifically, we select feature maps from the 1^{st}, 2^{nd}, 3^{rd} and 4^{th} blocks of 3D backbone of SECOND, and then generate visual explanations utilizing these feature maps. As shown in Table 4, selecting the feature maps from the 3^{rd} block obtains the best results on most metrics. Therefore, in other experiments, we utilize the feature maps from the 3^{rd} block. Then, to determine the setting of the sampling range, we set different combinations of the Manhattan distance threshold and neighbor number k. As illustrated in Table 5, (2, 16) is an appropriate setting for FFAM. Finally, we set r equal to 8, 16, 32, 64 and 128, respectively. The experimental results are shown in Table 6. As we can see, $r = 64$ greatly outperforms other settings on VEA, PG and enPG metrics, and performs slightly worse on Deletion and Insertion metrics. Consequently, we select $r = 64$ as the default setting.

A.2 Average Saliency Maps for Other Detectors

To reveal the detection modes of additional detectors, we present average saliency maps from various 3D detectors, namely CenterPoint [37], DCDet [12], PV-RCNN [21], and Voxel R-CNN [5]. CenterPoint and DCDet are trained and evaluated using Waymo Open [26], while PV-RCNN and Voxel R-CNN employ KITTI [7] for training and evaluation. The results are displayed in Figure 7. Notably, for pedestrian and cyclist categories, different detectors trained on distinct datasets exhibit similar areas of focus, such as the head and shoulder regions for pedestrians and the head and back regions for cyclists. However, in the car/vehicle category, detectors trained on diverse datasets reveal distinct patterns. Detectors trained on Waymo Open tend to concentrate on the front, back, and A-pillars of the vehicle category, whereas detectors trained on KITTI tend to emphasize the four corners of the car category.



CenterPoint DCDet PV-RCNN Voxel R-CNN

Figure 7: Average saliency maps for different detectors. Voxel R-CNN is only available for the car category (parameter weights are downloaded from OpenPCDet [30]).

Method	Deletion ↓				Insertion ↑			
	All	Vehicle	Ped.	Cyc.	All	Vehicle	Ped.	Cyc.
Grad-CAM	0.332	0.380	0.239	0.133	0.805	0.833	0.777	0.804
ODAM	0.252	0.279	0.200	0.151	0.853	0.875	0.809	0.824
OccAM	0.562	0.594	0.505	0.357	0.871	0.893	0.826	0.832
FFAM (Ours)	**0.095**	**0.120**	**0.085**	**0.104**	**0.897**	**0.922**	**0.845**	**0.853**

Table 7: AUC for Deletion and Insertion curves. The results of different categories are reported. 'all' means the combination of the three categories.

Method	PG ↑				enPG ↑			
	All	Car	Ped.	Cyc.	All	Car	Ped.	Cyc.
Grad-CAM	0.028	0.016	0.045	0.247	0.006	0.008	0.002	0.004
ODAM	0.917	0.941	0.865	0.970	0.476	0.568	0.285	0.575
OccAM	0.691	0.681	0.712	0.634	0.004	0.005	0.001	0.002
FFAM (Ours)	**0.975**	**0.980**	**0.963**	**0.980**	**0.517**	**0.597**	**0.349**	**0.650**

Table 8: Comparison of Pointing Game and energy-based Pointing Game metrics.

Method	VEA ↑			
	All	Car	Ped.	Cyc.
Grad-CAM	0.009	009	0.010	0.027
ODAM	0.234	0.220	0.261	0.260
OccAM	0.012	0.010	0.018	0.012
FFAM (Ours)	**0.388**	**0.358**	**0.448**	**0.458**

Table 9: Comparison of visual explanation accuracy metric for different categories. 'all' denotes the three categories are included.

OG	*VU*	*FF*	Del. ↓	Ins. ↑	VEA ↑	PG ↑	enPG ↑
✓			0.082	0.909	0.252	0.933	0.567
✓	✓		0.076	0.906	0.335	0.957	0.563
✓	✓	✓	0.071	0.907	0.391	0.991	0.664

Table 10: Effect of different components of FFAM. *OG*, *VU* and *FF* denote object-specific gradient, voxel upsampling and feature factorization, respectively.

A.3 Quantitative Results on Waymo Open Dataset

To further assess the visual explanation quality produced by our FFAM, we conducted experiments on Waymo Open [26], utilizing CenterPoint [37] as the detector for interpretation. We also employ the Deletion, Insertion, PG, enPG and VEA as the evaluation metrics. The results, presented in Table 7-9, showcase the superior performance of our FFAM across all metrics, mirroring the outcomes observed on KITTI. These findings demonstrate FFAM's remarkable adaptability to diverse detectors trained on different datasets.

A.4 Ablation Study

To investigate the impact of each component of FFAM, we conduct an ablation analysis using the SECOND detector on KITTI. Initially, we utilize the object-specific gradient alone to generate saliency maps, which yield relatively satisfactory results, as depicted in the first row of Table 10. Subsequently, we introduce the voxel upsampling strategy into the flow, resulting in significant improvements across most metrics, notably VEA and PG, as indicated in the second row of Table 10. Lastly, we incorporate the complete FFAM components. The third row of Table 10 demonstrates that feature factorization greatly enhances the quality of saliency maps, with the PG metric approaching a value close to 1, signifying the precise localization achieved by FFAM. We also observe that voxel upsampling and feature factorization do not yield improvements for the Insertion metric. We believe this is due to the models relying, to some extent, on the neighbor context for detection, whereas voxel upsampling and feature factorization result in a more compact saliency map (i.e., focusing on the object itself).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of this work in the conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide assumptions for each theoretical result.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide sufficient information to reproduce the main experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We release the implementation code of our method.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not reported because of high computational cost.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information about the computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper confirms the NeurIPS code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our research work mainly focuses on the improvement and understanding of algorithms and models, rather than areas directly related to social or ethical issues

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper doesn't release high-risk models and datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All resources used in our paper, including code, data, and models, strictly adhere to appropriate licenses and terms of use, and have been appropriately authored by the original owner.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our paper introduces new assets that are well-documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.