Explaining Deep Neural Networks for Point Clouds using Gradient-based Visualisations

Jawad Tayyub[⋆1], Muhammad Sarmad[⋆2], and Nicolas Schönborn[⋆1]

[1] Endress + Hauser, Maulburg, Germany
{jawad.tayyub,nicolas.schoenborn}@endress.com
[2] Norwegian University of Science and Technology, Trondheim, Norway
muhammad.sarmad@ntnu.no

SUPPLEMENTARY WORK

This supplementary material provides ablation studies, additional qualitative and quantitative results, parameter sensitivity analysis, and further technical details of the presented approach. We first present an ablation study that demonstrates the 'human' interpretability of our method (Section A). We then describe the architectural details of the networks used in this work (Section B). We then present qualitative results of our methods of explanations on both fixed and variable networks compared to existing approaches (Section C). Finally, we present a sensitivity analysis of the λ parameter from our APE algorithm (Section D).

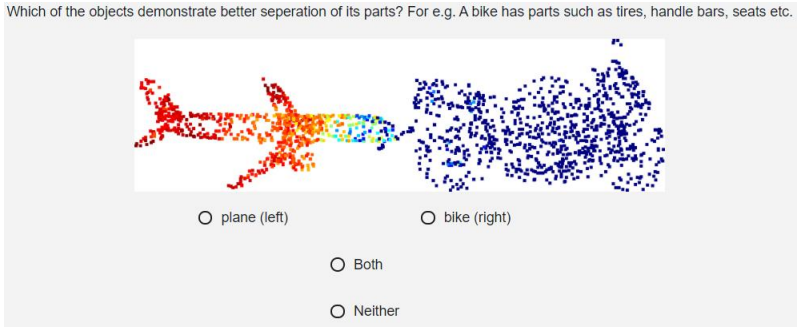
Section A

We conduct an ablation study to affirm that our method produces heatmaps that reflect the human understanding of objects, thus making them more humanly interpretable. Grad-Cam [3] utilizes the *Pointing Game* experiment and human worker feedback to evaluate the capability of various saliency methods to discriminate for localizing target objects in scenes. We also design an experiment for our approach on similar lines.

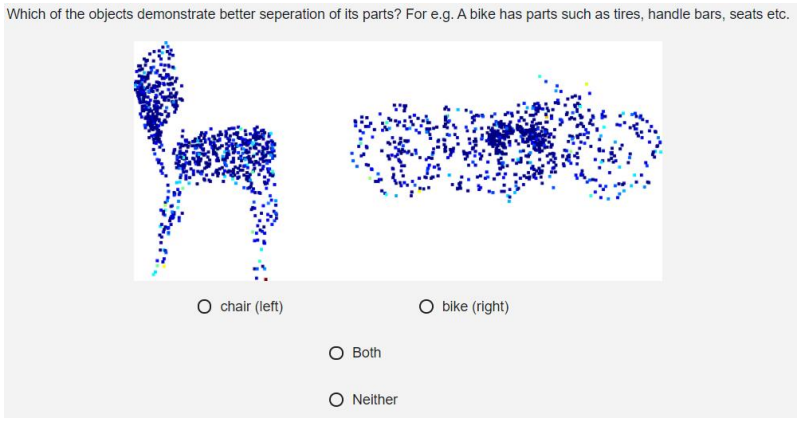
Since point cloud data often comprises only one object in a scene, an analogous experiment is designed by creating scenarios where two point clouds are concatenated from random classes and classified. Various explanation methods are then used to generate heatmaps. Users are then asked to select the point cloud, which presents a better separation of semantic parts of objects. This allows us to confirm our assumptions that our method is more human interpretable than comparative approaches.

As shown in Fig. 1, we ask these three questions for various combinations of randomly selected point clouds from a total of 270 human workers. Majority voting was then used to account for crowd sourcing anomalies. In each of the questions, we depict two point clouds. We then evaluate if the explanation heatmap then help the human in selecting the correct option by simply looking at the heatmap. For these question, we obtain an accuracy of 72.2%, 51.1%

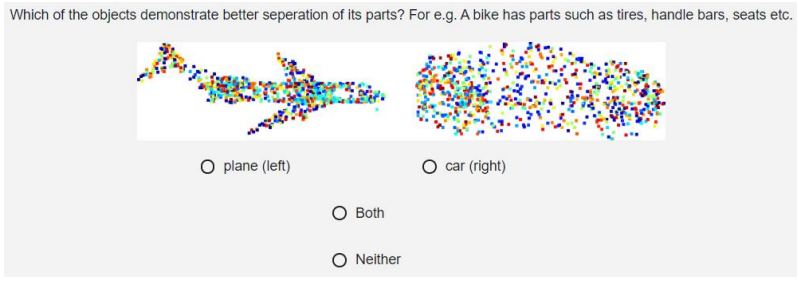
⋆ These authors contributed equally.



(a) **APE (ours)**



(b) **Gradients [4]**



(c) **PcSN [6]**

Fig. 1: **Ablation Study**

and 44.4% for our APE method, gradients [4] and PcSN [6] respectively. This reinforces our assumption that our method is more human interpretable then comparative approaches.

Section B

Section 4 presents an algorithm that captures gradients' flow through back-propagation to the final feature layer, where explanation values are computed and mapped onto the input point cloud. We supplement this by providing an illustration in Fig. 2 demonstrating the gradient flow-back and corresponding feature heatmap generation for the remaining three network architectures used: DGCNN, PointNet++, and VoteNet.

Fig. 2 a) presents the DGCNN network. This network is fixed-size, i.e., the resolution of the feature maps is maintained according to the input point cloud. Therefore feature heatmaps are a direct explanation of the input point cloud. Gradients for a target class y^c are logged back from the output layer to the last feature map, as highlighted in red. The feature heatmap generated here exhibits the same size as the input point cloud resulting in a direct transfer to create the point cloud heatmap, which explains the complete point cloud.

Fig. 2 b) presents the our approach overlayed on the classification part of the PointNet++ architecture. We flow back gradients to the final feature layer before the fully-connected layers. The feature heatmap generated at this layer is of dimension $M \times 1024$, which is smaller than the input point cloud. The APE algorithm then maps the feature heatmap to the input point cloud iteratively, resulting in the point cloud heatmap fully explaining every point of the input point cloud.

Fig. 2 c) presents the VoteNet architecture. Recall that VoteNet is designed for scene point clouds and, similar to PointNet++, exhibits loss of resolution of feature maps from the input point cloud. This figure shows that gradients are flown back to the layer right before 'propose and classify,' which are MLP layers performing down-stream tasks such as classification, segmentation, etc. The feature maps generated at this layer are of dimensions M x 1024 where M is less than N. Like PointNet++, the feature heatmap is scaled up to the input point cloud using the APE algorithm's iterative mechanism.

Note that the inner loop in the APE algorithm vastly differs from the outer loop denoted as IHU. IHU outer loop requires full heatmaps to be computed at each iteration and is designed to refine the point cloud heatmap to gain additional explanatory power. The inner loop operates on variable networks and handles dimensionality disagreement between input point cloud $\mathcal{P}$ and final feature maps $\mathcal{A}$. Moreover, point dropping in the outer loop is guided by low-contribution values, whilst in the inner loop, points are dropped because they have acquired *some* explanation value. Finally, the generated point cloud heatmaps over all iterations are combined by weighted maximum selection in the outer loop and concatenating feature heatmaps in the inner loop.

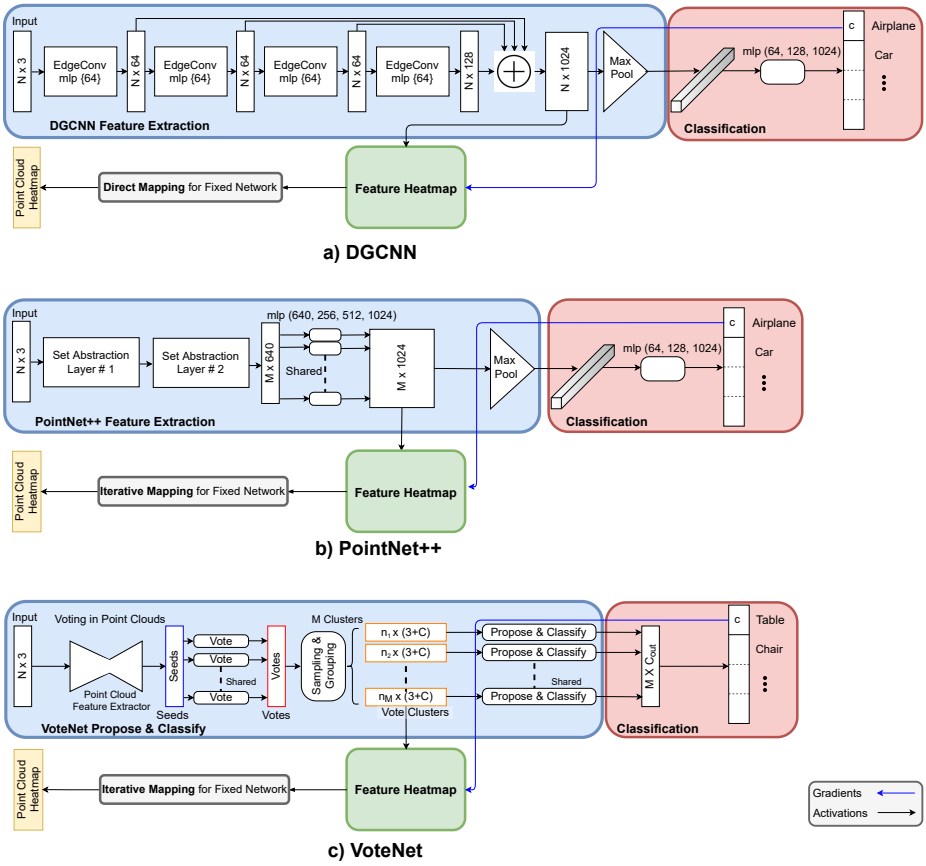


Fig. 2: **APE on Various Architectures.** These figure represent the architectures of a) DGCNN [5], b) PointNet++ [2] and c) VoteNet [1]. We have separated the architecture into the feature extraction segment and the classification segment of the networks. For all the above networks, the blue arrow represents the gradients flowing back given a target class y^c. Feature heatmaps L are extracted using the gradients from the last feature maps. The grey boxes show our proposed method for transforming the feature heatmaps to the point cloud heatmaps.

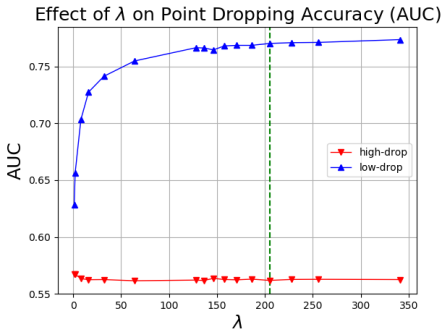


Fig. 3: Sensitivity analysis of λ parameter. An optimum value of 205 for λ is shown.

Section C

In this section, we present an extensive set of qualitative visualizations of point cloud heatmaps generated from our proposed APE algorithm. These explanations indicate most to least contributing points (red to blue) responsible for the network's classification decision.

Fig. 4 presents multiple examples of objects and scenes from the datasets used for each network architecture. Overall, we note that point cloud heatmaps correctly highlight corners, edges, and conceptually discriminating features of the objects presented. For example, the *plane* tends to have wingtips as the highly discriminating feature whereas the *bicycle* exhibits large discriminative features around the paddle or handlebar area. Similar conceptually meaningful features are seen to be identified as highly contributing across other objects as well.

Examples of fixed networks, namely PointNet and DGCNN, are presented in Fig. 4 a) and b) where for each network, we notice a consistent explanation pattern over the point clouds across these two networks. However, it is apparent that when comparing fixed networks with variable network (PointNet++), different segments of objects are detected as highly contributing. This result shows the variability of the network architecture and highlights the different weights learned by different networks resulting in explanations that emphasise different segments of objects. However, we do note some level of consistency for some objects, particularly the *plane*. A consistent pattern is noticed in explanations for the *plane* whereby the wings are identified as a highly discriminating feature across all network architectures. In contrast, a *chair* presents contributing features that switch between the back, the seat or the corners of the chair.

In scene point clouds, Fig. 4 d), we notice a highly accurate detection of contributing points that form objects that are being classified, as labeled atop the input diagrams. Furthermore, the APE algorithm continually generates explanations that posses sharp spatial boundaries. Finally we note a significant

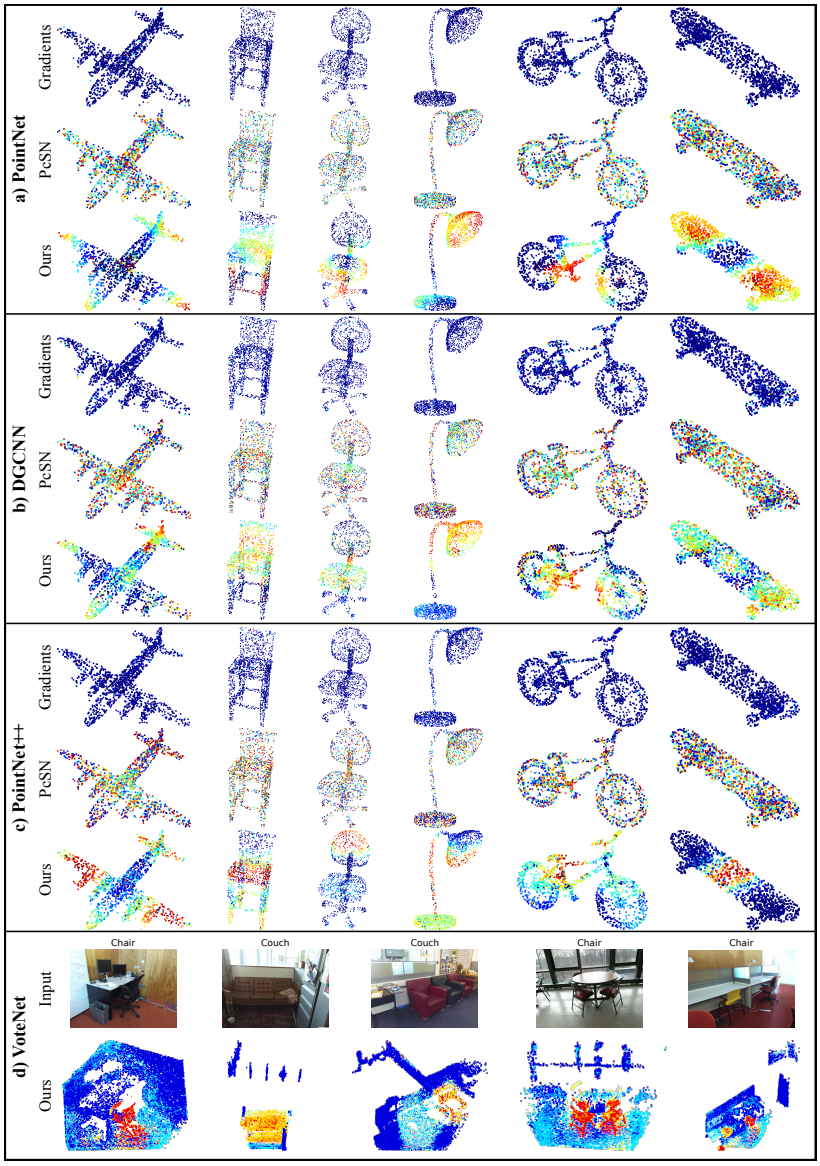


Fig. 4: **Qualitative overview of all networks and corresponding explanation methods.** Sample examples are taken from a single-object dataset as well as a scene point cloud dataset. Given a target class, the generated explanation point cloud heatmaps are presented. Ours are compared with Gradients [4] and Point Cloud Saliency Map (PcSN) [6] approaches.

improvement in interpretability of heatmaps generated from our method in comparison to the trivial Gradients method and the SOTA point cloud saliency maps approach across all examples.

Section D

Here we present the effect of the λ hyperparameter on the accuracy of the explanation point cloud heatmaps. Recall that, in the APE algorithm, λ parameter specifies the number of iterations of the outer loop (IHU). The IHU iterations control the level of refinement of the final point cloud heatmap. This is done by iteratively dropping the lowest relevance points and recomputing the heatmap. Heatmaps from all iterations are merged to create a highly descriptive point cloud heatmap. Figure 3 demonstrates the effect of λ on the accuracy (presented as the area under the curve (AUC) from point dropping experiments) as λ changes. The low-drop AUC increases gradually with a higher λ value as more iterations result in a superior point cloud. The high-drop AUC drops slightly, but the overall trend remains stable. These two observations are testament to the improvement obtained by employing an outer loop in the APE algorithm. We report a value of 205 as a good trade-off between point cloud heatmap quality obtained and the number of iterations needed.

References

1. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. CoRR **abs/1904.09664** (2019), http://arxiv.org/abs/1904.09664
2. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. CoRR **abs/1706.02413** (2017), http://arxiv.org/abs/1706.02413
3. Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., Batra, D.: Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. CoRR **abs/1610.02391** (2016), http://arxiv.org/abs/1610.02391
4. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps (2013)
5. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. CoRR **abs/1801.07829** (2018), http://arxiv.org/abs/1801.07829
6. Zheng, T., Chen, C., Yuan, J., Li, B., Ren, K.: Pointcloud saliency maps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1598–1606 (2019)