Visualizing Global Explanations of Point Cloud DNNs

Hanxiao Tan
AI Group, TU Dortmund
hanxiao.tan@tu-dortmund.de

Abstract

So far, few researchers have targeted the explainability of point cloud neural networks. Part of the explainability methods are not directly applicable to those networks due to the structural specifics. In this work, we show that Activation Maximization (AM) with traditional pixel-wise regularizations fails to generate human-perceptible global explanations for point cloud networks. We propose new generative model-based AM approaches to clearly outline the global explanations and enhance their comprehensibility. Additionally, we propose a composite evaluation metric to address the limitations of existing evaluating methods, which simultaneously takes into account activation value, diversity and perceptibility. Extensive experiments demonstrate that our generative-based AM approaches outperform regularization-based ones both qualitatively and quantitatively. To the best of our knowledge, this is the first work investigating global explainability of point cloud networks. Our code is available at: `https://github.com/Explain3D/PointCloudAM`.

1. Introduction

Point clouds are one of the most widely used data forms for 3D representation. Due to the irregularity, traditional CNNs are not directly applicable to point cloud data. Recently, several studies have proposed multifarious deep neural networks (DNN)s for point clouds [26, 27, 40] that achieved state-of-the-art accuracies in existing benchmark datasets. So far, however, very little attention has been paid to the trustworthiness of point cloud networks. In fields where human lives are at stake, such as autonomous driving, models without trustworthiness will pose potential risks. The research of explainability plays an important role in addressing the issue of trustworthy AI. Previous studies proposed a considerable number of explainability approaches including gradient-based [34, 4, 32, 36, 35, 38] and local surrogate model-based [28, 16, 29], which generate post-hoc local explanations to a specific input instance. Global explainability approaches are another solution that allow for an inclusive explanation of the entire black-box model, such as surrogate model simplification [14] and Activation Maximization (AM) [25]. Although the aforementioned approaches facilitate the faithfulness of models dealing with tabular and image data, there has been little discussion about the explainability of point cloud networks. Due to the specific architecture, point cloud networks possess distinctive properties from traditional multi-width convolutional neural networks (for instance, [10] found the features learned by point cloud networks are extremely sparse), suggesting that explainability studies on point cloud networks may lead to novel discoveries.

On the other hand, it is difficult to quantitatively evaluate the accuracies of the generated explanations due to the lack of ground truth, and human assessments are highly subjective and therefore lack persuasiveness and reproducibility. For AM, several previous studies have used quantitative metrics to evaluate the quality of the synthesized images [21, 18, 46]. However, we argue that the performance assessed by these traditional metrics is neither comprehensive nor precise for point cloud networks.

This work strives to investigate the global explanations of the popular point cloud networks with AM, which visualizes what point cloud models learn from the distribution of the entire dataset. We also show that non-generative network-based AM approaches for images are not applicable to point clouds (see figure 1), and propose generative AM methods for the global explainability of point cloud networks. Additionally, we propose a more persuasive and comprehensive evaluation metric for point cloud AM, and demonstrate that our point cloud AM methods outperform all other methods both at the human cognitive level and in quantitative assessment. Our contributions are primarily summarized as follows:

- As the first work investigating global explainability of point cloud networks, we exhibit that non-generative AM methods are unable to generate human-comprehensible explanations. Addressing the challenge, we propose generative model-based AM approaches that depict the global peculiarities of point cloud networks.

- We propose a convincing evaluation metric for point cloud AM: Point Cloud-Activation Maximization Score (PC-AMs), which simultaneously captures the activation value, diversity, human perception-level and physical-level authenticity of generated AM examples.

The rest of this paper is organized as follows: Section 2 introduces explainability methods for point clouds, especially AM and corresponding evaluation methods. Section 3 provides the proposed generative AM approaches for point clouds as well as a more persuasive evaluation metric. Section 4 demonstrates our experimental results and we summarize our work in Section 5.

2. Related Work

In this section, we introduce popular explainability methods, review the proposed AM approaches, and state the current progress of explainability research on point cloud neural networks.

Explainability methods: In contrast to interpretability approaches that render the decision process understandable, explainability methods aim to elucidate the operating principles of black-box models with mechanisms that are asynchronous with the decision-making periods. Explainability methods are categorized into two groups according to their objects: local and global explainers.

Local explainers typically generate explanations corresponding to individual inputs by tracing gradients [34, 4, 32, 36, 35, 38] or employing surrogate models and perturbations [28, 16, 29]. Nonetheless, gradient-based explainability methods are considered noisy, and in recent sanity studies, part of the methods were found to be model-independent [2]. Surrogate model-based approaches require extensive perturbation instances as training datasets and are therefore computationally intensive. Another common drawback of local explainability methods is the lack of holistic views of the overall datasets, compounding the cost of intrinsically understanding the decision process.

Global explainers provide explanations in regard to entire datasets rather than individual input instances by demonstrating its inherent characteristics. The global explanation may not be precise for each classification case, however, it provides a more intuitive representation of how the model works. Global explanations are typically presented in the following forms: [14] extracts decision rules from the original model that is comprehensible for users, [5, 8] rank the aggregated feature importance according to the whole datasets. For computer vision tasks, listing the feature importance is challenging because of the extensive number of unaligned features. As an alternative, AM is thus proposed to exhibit intuitive global explanations by generating highly representative examples of specific classes.

Activation Maximization (AM): AM is a high-level feature visualization technique that was first proposed by [9]. AM chooses a target activation unit and maximizes it by optimizing the input vector while freezing all other neurons in the DNN. However, without incorporating any prior or constraints, AM will synthesize mosaic images that are incomprehensible to humans and are not explainable [23]. Optimization constrains, such as L2-norm [33], Gaussian blur [44], Total Variation [17] or priors, such as average image initialization [24] and patch dataset [41, 20], successfully synthesize object images with clear outlines, and therefore facilitate the explainability. Another solution for enhancing the comprehensibility of AM images is to learn the distribution of real objects with generative models. [22, 46, 18, 15] utilized auto-encoders and GANs to produce high quality AM images. [21] proposed Plug & Play embedding generative networks that simultaneously address the high-resolution and diversity of synthesized AM images. Additionally, [43] proposed a black-box AM approach based on evolutionary algorithms. Nevertheless, point clouds are structurally different from traditional image DNNs so that the aforementioned AM methods are not directly applicable to point cloud networks.

Moreover, evaluating the quality of AM images is challenging and so far, most previous work relies on subjective human intuition as the evaluation criterion. [21] accessed the definition and diversity of AM images via Inception Score (IS) [30]. [18] incorporated Fréchet inception distance (FID) [13] to estimate the similarity between generated AM examples and real instances in latent spaces. AM score, another evaluation metric proposed by [46], is ameliorated from IS and addresses the uneven distribution of data categories.

Explainability research on point clouds: There are relatively few explainability studies in the area of point clouds. [45] traced the critical points to generate saliency maps of the point cloud network by dropping points. [10] was the first work to incorporate explainability methods, who started an observation of the intrinsic feature of point cloud networks via Integrated Gradients (IG). A follow-on study was conducted by [39], which proposed a local surrogate model-based approach for explaining point cloud networks. However, one limitation of the approaches mentioned above is that local explainability methods are only concerned with specific inputs that can hardly present the intrinsic properties of the whole point cloud network.

3. Methods

In this section, we demonstrate our AM approach for point clouds (Section 3.1) as well as the proposed evaluation metric for point cloud AM (Section 3.2).

3.1. Global explanation and AM

Global explainability can be considered as a summarization of the data distribution and the model behavior. In contrast to local explainability, it focuses more on the intrinsic properties of the whole model and data rather than on the attribution of individual decisions. Global explainability can be achieved by various techniques, e.g., by generalizing the model decision rules [12] or by training a global surrogate model [19]. For point clouds, the above methodologies are challenging due to the high dimensionality and structural complexity in 3D space. Since the structures of point cloud models are opaque, it is difficult to generalize their decision conditions. Explainable global surrogate models often suffer from significant performance degradation due to their inability to emulate complex architectures [19]. AM is a more intuitive global explanation for point clouds, which visualizes instances that maximize a certain activation and presents a globally representative input for a specific class to humans [31]. To visualize such an activation in DNNs, [9] proposed the AM, which is formulated as:

$$x^* = \underset{x}{\operatorname{argmax}} \left(a_i^l(\theta, x)\right) \tag{1}$$

where x and θ denote the input instance and the parameters in the DNN respectively, and $a_i^l(\theta, x)$ denotes the i^{th} neuron at l^{th} layer. The selected layer is typically the last layer (logits), since the output of this layer can be considered as the predicted probability of the corresponding class, while the neurons in the intermediate layers possess no semantics. However, 2D AM without any prior suffers from generating examples with high-frequency mosaics that are unrecognizable [23]. Several studies have investigated regularizing AM examples with non-generative priors, such as L2 Norm, Gaussian blur and Total variation [33, 44, 17]. While the above mentioned enhancements have made progress in human interpretability for 2D images, their effectiveness is severely compromised while processing point clouds (see figure 1). We believe that on the one hand, the features of point cloud networks are comparatively sparse and the global structure information of instances is seriously impaired [10], and on the other hand, the adjacency-based regularizations fail due to the disorderliness of point clouds.

To address the scarcity of structural information, we attempt to search for outputs which subject to two obligatory restrictions: they highly activate a neuron at a high level of the networks (equation 1) and are under the similar distribution as the dataset that is recognizable for humans. The former is a straightforward task and only requires maximizing an activation of the point cloud network by back propagation. For the latter, we choose generative models to constrain the distribution of generated point clouds to be as realistic as possible. An outline of our approach is shown in Fig. 2. In the following contents we present the details of the proposed module.

AM for point cloud NNs: Typically, an instance as an input to a point cloud model f_c can be represented as $P = \{p_i \mid i = 1, ..., N\} \in \mathbb{R}^{N \times D}$, where N is the number of points and D is the dimensions ($D = 6$ if color information is attached, otherwise $D = 3$). The model outputs a *logits* vector $f_c(P) \in \mathbb{R}^{N_c \times 1}$, where N_c denotes the number of classes. Our goal is to build a module which outputs a P_g that $\operatorname{argmax} f_c^i(P_g)$, and $O_g \sim P_x$, where $f_c^i(P_g)$ denotes the i^{th} activation of the logits and P_x denotes the real instances from dataset X. Our approach starts by training a module that searches the $\mathbb{R}^{N \times D}$ space for examples with similar distribution to P_x, and then filters out those that maximize a target activation.

Point cloud AutoEncoder (AE): Existing study [1] has demonstrated that the Autoencoder can reconstruct point cloud instances with a high level of restoration. They utilize point-wise convolutions followed by a symmetric pooling layer to encode point clouds into 1-dimensional latent representations, and build a simple multi-layer, fully connected network to decode the latent vector. We follow their design and exploit an AutoEncoder (AE) to learn the distribution from real data. The point cloud AE consists of two components, the encoder h_{AE} and the generator (decoder) g_{AE}. The input of the encoder h_{AE} is a point cloud instance $P_x \in \mathbb{R}^{N \times D}$ and the output is a latent encoded vector $V \in \mathbb{R}^{1 \times k}$, where k is an adjustable dimensionality parameter. The generator g_{AE} takes V as input and generates a point cloud P_g with the same dimensions as P_x and $P_g \sim P_x$. The detailed structure of the AE is shown in Fig. S4. When training, we measure the gap between the generated examples and the original data with Chamfer Distance (CD) loss $\mathcal{L}_C$, which is formulated as:

$$\mathcal{L}_{g_{AE}} = \mathcal{L}_C = \frac{1}{|P_g|} \sum_{p_m \in P_x} \min_{p_n \in P_g} \|p_m - p_n\|_2 \tag{2}$$

AutoEncoder with Discriminator (AED): Although AE is capable of reconstructing point cloud instances at a high level, it is not sufficient as global explanations, since diversity is an important property for explainability [7]. Adding Gaussian noise during *AM optimization phase* is a potential solution. However, unrestricted noise inclines to downgrade the quality of explanations rather than enhance their diversity. Therefore, we propose AutoEncoder with Discriminator (AED), which is based on AE with two enhancements: a discriminator D_c and a latent distance loss $\mathcal{L}_F$. D_c acts similarly in GANs: while the generator of AED (g_{AED}) tries to fool D_c by generating fake examples that mislead D_c to classify them as real instances, and D_c attempts to correctly identify both. The input of D_c is also a point cloud of $N \times D$, and the output is a probability $pb \in [0, 1]$ for each input ($pb \to 1$ for real instances and $pb \to 0$ for fake examples). We build a discriminant loss

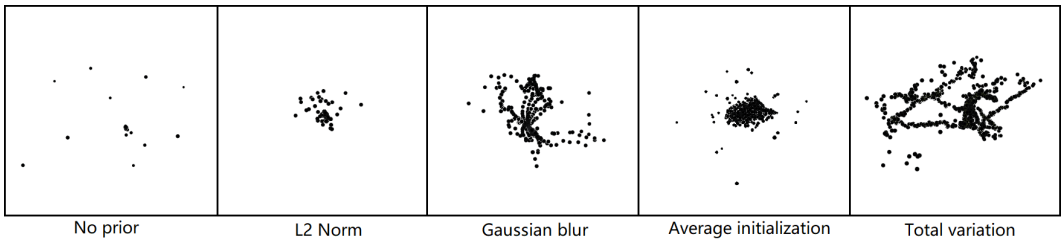

Figure 1. AM for point clouds without generative priors (class "car"). Due to the specific architecture of the point cloud network, traditional regularization priors (for 2D images) are incapable of generating human-perceivable global explanations. More details can be found in Sec. S1.1.

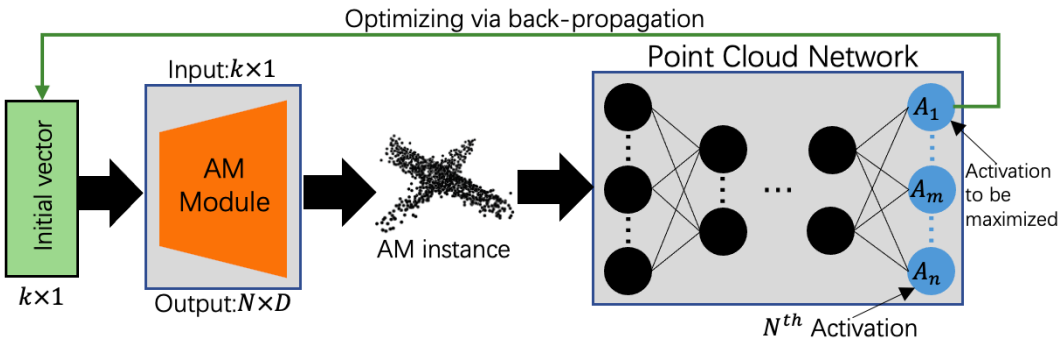

Figure 2. General overview of the architecture for point cloud AM. The green and gray bars represent vectors and networks, respectively. In the point cloud network, the black and blue circles represent the neurons in the middle layer and the last layer (the activations), respectively. The thick black arrows and thin green arrows represent forward inference and backward propagation, respectively.

$\mathcal{L}^d_{D_{AED}}$ with D_c for the discriminator, which is formulated as:

$$\mathcal{L}^d_{D_{AED}} = D_c(P_g) - D_c(P_x) \tag{3}$$

Note that the value domain of $\mathcal{L}^d_{D_{AED}}$ is $[-1, 1]$. We observe that since the performance of D_c easily outperforms g_{AED} ($\mathcal{L}^d_{D_{AED}} \to -1$) during training, the latter struggles to be further optimized [3]. We therefore train only one of them alternately for each batch: If $\mathcal{L}^d_{D_{AED}} < 0$, we train the g_{AED} only and vice versa. Furthermore, if the discriminator is overperforming ($\mathcal{L}^d_{D_{AED}} < -0.75$), we add Gaussian noise to its parameters to disrupt the performance.

The latent distance loss $\mathcal{L}_F$ measures the feature distinction between two inputs. We choose the output of the second convolutional layer for measurement, which is a hidden vector of dimension $N \times 128$. The latent distance loss is computed as:

$$\mathcal{L}_F = \frac{1}{|V^c_g|} \sum_{v_m \in V^c_x} \min_{V_n \in V^c_g} \|v_m - v_n\| \tag{4}$$

where V^c_x and V^c_g represent the output of the second convolutional layer in the encoder, computed with real instances and generated examples, respectively. $\mathcal{L}_F$ can be regarded as the CD computed on the latent space.

The final generative loss of AED is formulated as:

$$\mathcal{L}_{g_{AED}} = \mathcal{L}_C + w_F \mathcal{L}_F - w_D \mathcal{L}^g_{D_{AED}} \tag{5}$$

where w_F, w_D are the corresponding weights and $\mathcal{L}^g_{D_{AED}}$ denotes the loss for the generator to deceive the discriminator, which is $-D_c(P_g)$. The detailed architecture of AED is presented in S5.

Noisy AutoEncoder with Discriminator (NAED): Despite the enhancement in diversity, practice shows that the samples generated by AED suffer from instability. To address this issue, we continue to refine the structure on the basis of AED. There are two main improvements: a) Gaussian noises are added to the encoder and b) another global latent distance regularization is introduced. The former is straightforward to implement, requiring only the insertion of Gaussian noise to the output of each layer in the encoder. However, experiments demonstrate that it is significant (see Sec. S1.5). For the latter, recall the latent distance regularization used in AED, whose latent vectors are extracted from the second convolutional layer of the encoder. However, due to the irregularity, the convolutional layers of point cloud networks typically extract local features only and lack global information. Therefore, in NAED, we append an additional loss $\mathcal{L}_{F2}$, which is obtained by computing the latent distance of the output from the max-pooling layer. The distance measurement is identical to Eq. 3, with the only difference that the local vector V^c is replaced by a global one $V \in \mathbb{R}^k$. The final generative loss of NAED is formulated as:

$$\mathcal{L}_{g_{NAED}} = (\mathcal{L}_C + w_F \mathcal{L}_F + w_{F2} \mathcal{L}_{F2}) - w_D \mathcal{L}^g_{D_{NAED}} \tag{6}$$

where w_{F2} is the weights of $\mathcal{L}_{F2}$ and $\mathcal{L}^g_{D_{NAED}}$ is the discriminative loss of NAED and is calculated identically as $\mathcal{L}^g_{D_{AED}}$. The elaborated architecture of NAED is shown in Fig. S6.

AM optimization: After the aforementioned modules are well-trained, we concatenate them with the point cloud model. The final optimization process is that we initialize a latent vector $V_{ini} \in \mathbb{R}^{1\times k}$ and decode it with the generator ($g_{(N)AE(D)}$). Here an initialized point cloud example $P_{ini} \in \mathbb{R}^{N\times D}$ is generated. Subsequently P_{ini} is fed into the whole encoder-decoder system and we extract the output P'_{ini} and the discriminator loss $\mathcal{L}_{D_{(N)AED}}$, which forces the generated examples to be close to real ones (No $\mathcal{L}_{D_{AE}}$ exists for AE, but for fairness, we repeat this encoding and decoding process as well). We then obtain the target activation value $f^t_c(P'_{ini})$ and optimize V_{ini} via back-propagation. The general term of the AM optimization loss is:

$$-(f^t_c(P'_{ini}) + \mathcal{L}_{D_{(N))AED}}) \tag{7}$$

Moreover, inspired by [24], we calculate the average of the dataset and encode it as V_{ini} so that the initial distribution does not deviate significantly from the real data. When the optimization process is stuck, we introduce Gaussian noises to V_{ini} to escape from the local optimum. Finally, the optimization stops after reaching a certain number of iterations.

3.2. Evaluation Metrics for Point Clouds AM

Most previous research evaluates explainability methods by showing examples to humans. However, this approach is costly and relatively subjective. Our goal is to find a quantitative measurement that is both consistent with human perception and computationally assessable in a quantitative way. Since there is no proposed metric for point clouds AM, we list three types of metrics that measure activation values or prototype similarity:

Activation-targeted metrics, represented by IS [30] or AM Score [46], aim to assess the maximization of a certain neuron in logits. However, this series of approaches only evaluates the generation quality by calculating the entropy of the logits, while the disparity in human perception levels is absent. For point clouds, they fail to distinguish between AM methods without priors and those based on generative models, although the latter are apparently more comprehensible to humans.

Pixel-wise metircs, represented by L_p (2D), Chamfer and Hausdorff distances (3D), address forcing the generated instances to be pixel-wisely approximated to the real objects. Nevertheless, instances that comply with these metrics may lose the ability to be "global explainable" as it does not require the instances to be globally representative. Suppose a generator that perfectly reconstructs the original instance, even though the distance loss can be minimized to 0, but it does not facilitate human understanding of the model peculiarities.

Latent feature metrics, represented by FID, measure the distinction on the feature level, which are theoretically promising and widely applied in 2D generative models. We follow the FID from [37] which compared the global features from the PointNet architecture. Nonetheless, we observe that the metric is vulnerable for AM (see table 1: randomly initialized instances achieve FID scores as high as those from generative models, though they are not perceived well by humans). We believe that the FID is affected to some extent by the sparsity of the point clouds due to the scarcity of adjacent relations in the point cloud networks.

PC-AMs: Targeting the limitations of the aforementioned methods, we propose a composite AM evaluation metric: PC-AMs. Our PC-AMs is formulated as:

$$PCAMS = IS_m - \frac{(log(FID_{PN}) + log(CD))}{2} \tag{8}$$

IS_m denotes the modified Inception Score (M-IS) [11], which is formulated as:

$$IS_m = e^{\mathbb{E}_{x_i}[\mathbb{E}_{x_j}[(\mathbb{KL}(p(y|x_i)||p(y|x_j))]]} \tag{9}$$

where x_i and x_j denotes different instances with the same label. In addition to the values of the corresponding activations, M-IS concentrates more on the diversity of the generated examples within classes than the variety of inter-class labels. Therefore we utilize the M-IS which employs the cross-entropy of the predictions within intra-class examples. The value range of M-IS is $[1, N_c]$.

FID_{PN} denotes the PointNet-based FID and is formulated as:

$$FID_{PN} = \|\mu_r - \mu_g\|^2 + Tr(\sigma_r + \sigma_g - 2(\sigma_r\sigma_g)^{\frac{1}{2}}) \tag{10}$$

where $A_r \sim \mathcal{N}(\mu_r, \sigma_r)$ and $A_g \sim \mathcal{N}(\mu_g, \sigma_g)$ are the activations from the reference network, which are approximately considered as Gaussian distributions. FID measures the distance between the two distributions, lower FID scores imply closer proximity of the generated examples to the real instances, and therefore higher perceptibility. Nevertheless, the standard reference network *Inception-v3* is no longer applicable to FID_{PN} since the multi-width convolutional kernel for images fails to extract adjacent features from unordered point clouds. Following [37], we substitute the backbone of PointNet for Inception-v3 and choose from the layers above the max-pooling (global features) as the activation. The value range of FID is $[0, +\infty]$.

Due to the fragility of FID_{PN}, we introduce an additional perceptibility measure: CD, formulated as:

$$CD(x_g, x_i) = \frac{1}{|x_g|}\sum_{p_m \in x_g} \min_{p_n \in x_i} \|p_m - p_n\|_2 \tag{11}$$

whose value range is also $[0, +\infty]$. Although CD estimates the similarity between examples more precisely, it lacks generality as a scoring criterion for AM. To alleviate this deficiency, we randomly draw several instances from the dataset with the same labels as the generated examples and calculate the average of the CDs. We finalize the aforementioned three metrics by logarithmically scaling FID and CD to the same order of magnitude with M-IS, such that the final score does not collapse due to the numerical explosion of any single term. The final value field of PC-AMs is $[-\infty, N_c]$. In addition, we introduce another point-wise distance used for comparison in Sec 4: Earth Mover's Distance (EMD), which is formulated as:

$$EMD(x_g, x_i) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} Pr_{i,j} d_{i,j}}{\sum_{i=1}^{m}\sum_{j=1}^{n} Pr_{i,j}} \qquad (12)$$

where $Pr_{i,j}$ denotes the pair-wise combination of points in x_g and x_i, and $d_{i,j}$ denotes their spatial distance.

In summary, PC-AMs simultaneously considers activation values (M-IS), diversity (M-IS), point-wise distances (CD), and latent distances (FID) when evaluating AM explanations of point clouds.

4. Experiments

In this section, we qualitatively demonstrate the generated examples of our proposed point cloud-applicable AM (section 4.1), and show the quantitative evaluations of existing point cloud AM approaches (section 4.2). Additionally, we also provide an example of application scenes of proposed methods for prediction examination in section 4.3. In our experiments, we choose ModelNet40 [42] as test dataset, which contains 12311 CAD models in 40 common classes and is currently the most widely-used point cloud dataset. Besides, we also test our approaches on the classification set of ShapeNet [6], which is composed of 45969 point cloud instances (35708 for training and 10261 for testing) in 55 classes. We select PointNet as our primary experimental model, which is the pioneer of deep learning for raw point clouds. We also validate our result in the most popular point cloud models i.e., PointNet++ [27] and DGCNN [40]. During AM generation, we heuristically set the latent dimensions of the AE as 128 and the learning rate as $5e-6$. The AM optimization stops after 2×10^4 iterations. All introduced Gaussian noises are $\mathcal{N}(0, 1e-5)$. All loss weights are 1 (e.g. w_F, w_{F2} and w_D) when training the generation module. For quantitative evaluation, we generate 10 AM examples for each class, and we randomly select 5 real instances from the dataset as the baseline for calculating FIDs and CDs and average the corresponding results.

4.1. Point Cloud AM Visualization

Perceptibility: Figure 3 shows the point cloud AM examples of common classes generated by multifarious approaches on ModelNet40. Zero and random initialization, while highly activating the selected neurons, results in only the expansion of individual points due to the lack of a prior and therefore fails to yield human understandable global explanations. Initialization with the average of the test data performs better in 2D images. However in point clouds, explainability is not significantly enhanced compared to the no-prior methods since the point cluster in the center struggles to render the distribution of common objects. Initialized from a specific instance though outlines the objects best, nevertheless, the information of the "global" is absent, i.e., the general distribution of the whole dataset. The contours of the objects are derived from the input instances themselves rather than the global activation-optimization process. The former tends to expose more local information about particular inputs and is therefore more generally utilized in adversarial attacks. In addition, due to the irregularity of point clouds, incorporating traditional regularizations (L2, Gaussian blur and Total variation) also fail to yield globally perceivable explanations. In comparison, our generators with latent priors dominate in terms of both shape consistency and human perceptibly.

Among the generative methods, AM examples provided by AE are intuitively more stable, especially compared to those from AED. We believe this is due to the absence of noise mechanisms and the singularity of the loss term. In AE, no noise is incorporated except for the neuron maximization module that prevents the optimization process from sticking in local optimums, and the generator is trained via an one-fold CD loss which only forces the output to be point-wise approximated to real objects. These mechanisms regularize the profile of the generated examples to be reconstructed precisely as the real instances from the dataset while the outputs suffer from a scarcity of diversity. On the other hand, in AED and NAED, the multi-fold loss functions balance the constraints of approximating the dataset in both point-wise and latent feature levels. Compared to AE, this module causes a few collapses of the output geometries, but by introducing adversarial learning with a discriminator, the generator is still able to reconstruct the contours of real objects and enrich their diversity simultaneously. Moreover, we surprisingly find that incorporating cascaded Gaussian noise to the encoder during training further enhances the quality and diversity of the AM outputs. We present the generation diversity in the next subsection.

Diversity: Another key factor of AM quality is the diversity. In figure 4, we visualize 5 examples for each generative AM methods which are randomly selected from the generation repository. We also demonstrate the five examples in the dataset that most highly activate the neuron "table", as well as five stochastically selected examples respectively for references. As can be seen from the figure, AE is more stable than the others, while lacks diversity. In comparison,

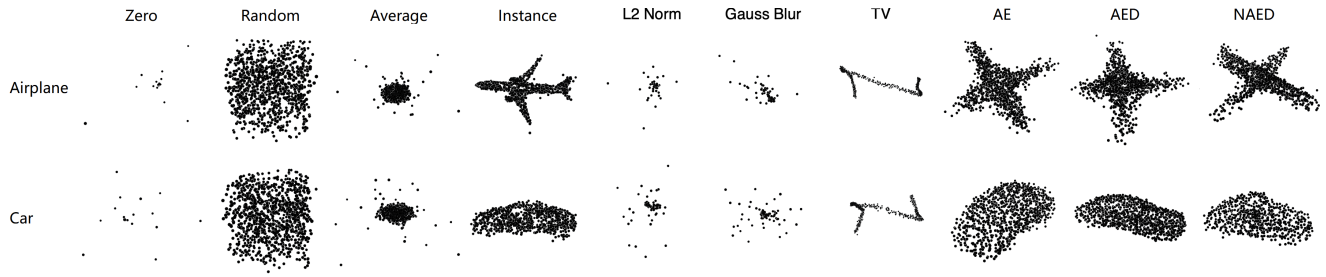

Figure 3. AM results of different approaches. From left to right: Zero initialization, random initialization, initialized with the average of the test data per class, initialized from a specific instance, regularization with L2 Norm, Gaussian Blur and Total Variation, and our proposed AE, AED and noisy NAED. Apparently, except for the instance initialization, the non-generative model-based approaches suffer from serious flaws in perceivability of AM examples. Moreover, the AM example initialized from a certain instance lacks the "global" property, and the generated examples are unrepresentative. More qualitative comparisons can be found in Fig. S2.

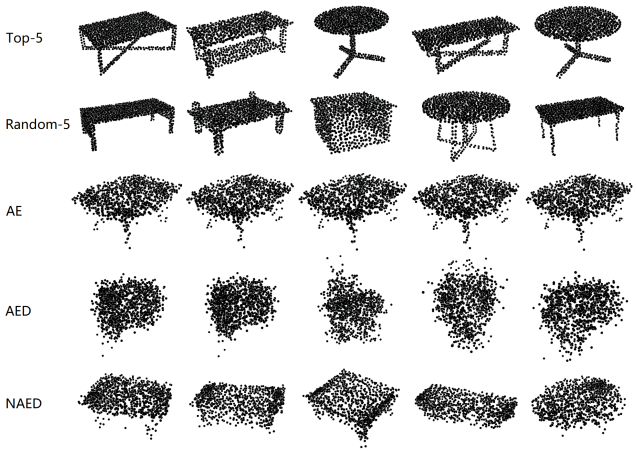

Figure 4. Diversity of AM generations. We choose 5 examples from instances that (from top to bottom) 1) most hightly activate the neuron 2) are selected randomly 3) are from the generations of AE 4) of AED 5) of NAED. It can be seen that although the examples generated by AE are stable, they are severely deficient in diversity. AED enhances diversity but suffers from instability, where part of the generated examples are imperceptible. NAED outperforms in both diversity and stability.

both AED and NAED depict the multiplicity of the objects while AED is somewhat deficient in terms of stability.

We conduct ablation studies for each module and demonstrate the results in Sec. S1.5.

Experiments on ShapeNet: We also present the AM results of the class "airplane" generated by the proposed methods employing ShapeNet as the experimental dataset in figure S3. Similar to ModelNet40, the global explanations presented by AE also exhibit only minimal spatial offsets, while AED and NAED outperform AE in terms of the diversity of object outlines. Subjectively, the examples generated by NAED are more stable due to the noise introduction in the training process.

4.2. Evaluation Metric of Point Cloud AM

Visually assessing the AM global explanation is highly subjective, and therefore we quantitatively evaluate the results via the proposed methods in table 1. Since there is no existing AM study for point clouds, we consider the *no prior* and *point-wise* prior approaches as our baseline. Note that in terms of FID, AMs with random initialization also achieve a satisfactory loss while the examples are almost indistinguishable by humans, which results in the inability to accurately capture the perceptual distance between examples. Therefore, we introduce CD as another regularization. We also incorporate EMD to validate the approximation of the examples. According to the comparisons, our generative AM approaches (latent prior) dominate the rest regarding the PC-AMs. Though AE possesses the minimum distance loss, it suffers from a significant drawback of diversity, which leads to the M-IS being lower than the other approaches (which is consistent with the demonstrations in figure 4). In addition, figure 3 reports the corresponding evaluations on ShapeNet, where it can be seen that our proposed approaches consistently achieve similar performance on different datasets.

We also evaluate the performance of the proposed methods on different point cloud networks with PC-AMs, and present the results in table 2. As a reference, we show an example of the corresponding visualization in figure 5. We notice that AED performs unstably, especially when explaining PointNet++, which occasionally fails to generate perceptible structures (middle plot of the second row). This is also revealed in PC-AMs: in table 2, the lowest score is obtained by explaining PointNet++ with AED.

Another interesting observation we noticed is that the global feature-based FID proposed by [37], to some extent, measures only the "**diffusion degree**" rather than the "similarity" to real objects. For verification, we synthesize instances that are randomly distributed and therefore completely "dissimilar". We yield examples that are uniformly distributed $P_u \sim U(-r, r)$, and normally distributed

		m-IS	FID	CD	EMD	PC-AMs
Initializations	Zero	1.113	0.119	0.266	364.35	2.84
	Random	1.081	0.016	0.245	413.52	3.85
	Average	1.001	0.097	0.230	377.20	2.90
	Instance	1.015	0.071	0.085	228.87	3.57
Regularizations	L2 Norm	1.001	0.256	0.139	375.93	2.66
	Gaus blur	1.000	0.420	0.148	372.88	2.38
	TV	1.000	0.092	0.376	490.842	2.67
Generative Model	AE	1.085	0.016	**0.044**	**143.13**	4.71
	AED	1.124	0.018	0.086	241.35	4.37
	NAED	**1.461**	**0.014**	0.074	207.65	**4.89**

Table 1. PC-AMs evaluation metric for point cloud AMs. EMD is also introduced for point-wise distance validation. Note that since there is no comparable **global** explainability method for point clouds, we consider the traditional AMs as baselines. Detailed descriptions of the baselines can be found in Sec. S1.1.

		m-IS	FID	CD	EMD	PC-AMs
AE	PN	1.085	0.016	0.044	143.13	4.71
	PN++	1.103	**0.008**	**0.041**	**134.16**	5.12
	DGCNN	1.020	0.010	0.105	252.82	4.43
AED	PN	1.124	0.018	0.086	241.35	4.37
	PN++	1.107	0.020	0.122	255.46	4.12
	DGCNN	1.358	0.013	0.109	343.15	4.63
NAED	PN	1.578	0.018	0.071	353.10	4.92
	PN++	**1.866**	0.011	0.072	236.42	**5.43**
	DGCNN	1.316	0.015	0.109	335.51	4.52

Table 2. PC-AMs evaluations for different point cloud models, where PN and PN++ denotes PointNet and PointNet++.

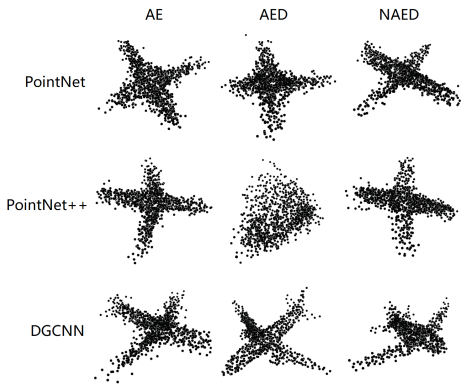

Figure 5. AM visualization for the most popular point cloud networks: PointNet, PointNet++ and DGCNN. The proposed method is applicable to all point cloud networks.

	m-IS	FID	CD	EMD	PC-AMs
AE	1.012	0.017	**0.047**	**147.87**	4.57
AED	1.146	0.012	0.076	208.02	4.65
NAED	**1.157**	**0.011**	0.067	203.74	**4.75**

Table 3. Quantitative evaluations on ShapeNet.

$P_n \sim \mathcal{N}(0, \sigma^2)$, where r increase from 0 to 1 and σ grows from 0 to 0.1 in 10 steps respectively, in order to represent inputs with different "diffusion degrees". For comparison, we stochastically choose real objects from the dataset, and calculate their FID with objects of the same class. Theoretically, FID performs consistently with human judgment. Randomly distributed artificial examples should exhibit significantly large FID with real objects, as they possess no recognizable geometric structures. However, as figure S7 demonstrates, FID (the brighter blue line) dramatically decreases with the point expansion of the instances ($r = 0.1$ and $\sigma = 0.02$). After the diffusion reaches the threshold ($r \approx 0.2$ and $\sigma \approx 0.05$), FID fails to distinguish the meaningless point clouds from the real objects (the darker blue line), though we can still observe the discrepancies between them through CD and EMD. A better point cloud-applicable perceptibility metric for generating examples in terms of latent distance is a promising research direction.

4.3. AM for data reviewing

Explanations can facilitate human understanding of the operating behavior of DNNs. As a global explainability method, AM depicts the ideal input learned by the model. When the performance of the model is sufficiently promising, one considers that the result of AM should be a generalization of an outline of the objects from the corresponding class. Therefore, we can review those misclassified input instances utilizing this characteristic. An example is shown in figure S8. Several instances in the dataset with the "plant" label are misclassified as "vase", whereas a comparison exhibits that a single "plant" label is ambiguous since the composite instance also contains the "vase" fraction. Observing the second and third columns, AM correctly describes the object outlines of the corresponding neurons in the model without any confusion. For validation, we also generate explanations for these instances employing the point cloud-applicable LIME [39] (the last column). The conclusions of the two explanations are approximately analogous, and the explanation given by the model is consistent with its predicted label in human perception.

5. Conclusion

This work proposes three generative model-based AM approaches which significantly enhance the perceptibility of the generated examples while also maintaining their diversity. A composite evaluation metric, balancing activation value, diversity and perceptibility is proposed. The results show that our generative AM methods outperform the regularization-based ones in both qualitative and quantitative aspects. For future work, we look forward to more efficient AM generation methods as well as visualizations of low-level neurons to further explore the working mechanism of point cloud neural networks.

Acknowledgements. This research has been funded by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr-Institute for Machine Learning and Artificial Intelligence, LAMARR22B.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018.

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[4] Sebastian Bach et al. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):1–46, 2015.

[5] Giuseppe Casalicchio, Christoph Molnar, and Bernd Bischl. Visualizing the feature importance for black box models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 655–670. Springer, 2018.

[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[7] Roberto Confalonieri, Ludovik Coba, Benedikt Wagner, and Tarek R Besold. A historical perspective of explainable artificial intelligence. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 11(1):e1391, 2021.

[8] Ian Covert, Scott Lundberg, and Su-In Lee. Understanding global feature contributions with additive importance measures. *arXiv preprint arXiv:2004.00668*, 2020.

[9] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

[10] Ananya Gupta, Simon Watson, and Hujun Yin. 3d point cloud feature explanations using gradient-based methods. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[11] Swaminathan Gurumurthy, Ravi Kiran Sarvadevabhatla, and R Venkatesh Babu. Deligan: Generative adversarial networks for diverse and limited data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 166–174, 2017.

[12] Tameru Hailesilassie. Rule extraction algorithm for deep neural networks: A review. *arXiv preprint arXiv:1610.05267*, 2016.

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[14] SM Kamruzzaman, MD Islam, et al. An algorithm to extract rules from artificial neural networks for medical diagnosis problems. *arXiv preprint arXiv:1009.4566*, 2010.

[15] Alexander Katzmann, Oliver Taubmann, Stephen Ahmad, Alexander Mühlberg, Michael Sühling, and Horst-Michael Groß. Explaining clinical decision support systems in medical imaging using cycle-consistent activation maximization. *Neurocomputing*, 458:141–156, 2021.

[16] Scott M. Lundberg and Su In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 2017-Dec.(Section 2):4766–4775, 2017. arXiv preprint, arXiv:1705.07874.

[17] Aravindh Mahendran and Andrea Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120(3):233–255, 2016.

[18] Saumitra Mishra, Daniel Stoller, Emmanouil Benetos, Bob L Sturm, and Simon Dixon. Gan-based generation and automatic selection of explanations for neural networks. *arXiv preprint arXiv:1904.09533*, 2019.

[19] Christoph Molnar. *Interpretable Machine Learning*. 2 edition, 2022.

[20] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. 2015.

[21] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4467–4477, 2017.

[22] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems*, 29:3387–3395, 2016.

[23] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[24] Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*, 2016.

[25] Anh Nguyen, Jason Yosinski, and Jeff Clune. *Understanding Neural Networks via Feature Visualization: A Survey*, pages 55–76. Springer International Publishing, Cham, 2019.

[26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[27] 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.

[28] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" Explaining the predictions of any classifier. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-Augu:1135–1144, 2016. arXiv preprint, arXiv:1602.04938v3.

[29] 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.

[30] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29:2234–2242, 2016.

[31] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. *Explainable AI: interpreting, explaining and visualizing deep learning*, volume 11700. Springer Nature, 2019.

[32] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences, 2019. arXiv preprint, arXiv:1704.02685.

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

[34] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014. arXiv preprint, arXiv:1312.6034.

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

[36] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net, 2015. arXiv preprint, arXiv:1412.6806.

[37] Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 61–70, 2020.

[38] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *34th International Conference on Machine Learning, ICML 2017*, 7:5109–5118, 2017. arXiv preprint, arXiv:1703.01365.

[39] Hanxiao Tan and Helena Kotthaus. Surrogate model-based explainability methods for point cloud nns. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2239–2248, January 2022.

[40] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[41] Donglai Wei, Bolei Zhou, Antonio Torrabla, and William Freeman. Understanding intra-class knowledge inside cnn. *arXiv preprint arXiv:1507.02379*, 2015.

[42] Zhirong Wu et al. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[43] Will Xiao and Gabriel Kreiman. Gradient-free activation maximization for identifying effective stimuli. *arXiv preprint arXiv:1905.00378*, 2019.

[44] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[45] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. Pointcloud saliency maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[46] Zhiming Zhou, Han Cai, Shu Rong, Yuxuan Song, Kan Ren, Weinan Zhang, Yong Yu, and Jun Wang. Activation maximization generative adversarial nets. *arXiv preprint arXiv:1703.02000*, 2017.