Evaluable Explainability and Applications to 3D Vision

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

Hanxiao Tan

Dortmund

2025

Tag der mündlichen Prüfung:	27. 10.2025
Dekan:	Prof. Dr. Jens Teubner
Gutachter*innen:	Prof. Dr. Emmanuel Müller, Prof. Dr. Daniel Neider

Abstract

With the breakthroughs in the performance of deep neural networks, they are applied in a wide range of fields, including several with high requirements for security. However, these black-box models suffer from potential risks, the most threatening of which is their opaque decision-making process. The recent rise of explainability studies on black-box models is a promising research direction to enhance the trustworthiness of models. Nevertheless, existing explainability studies are still limited, one being that they are difficult to be evaluated objectively due to the lack of ground truth, and the other being that the vast majority of relevant studies are constrained to a particular data format and lack extensibility.

The two main parts of this dissertation address each of these two limitations. In the first half, we aim to improve the evaluability of explainability methods. We optimize the choice of baseline involved in the explanation evaluations and part of the explainability approaches to satisfy the uninformative definition. In addition, we complement the explanation evaluation metrics from three novel perspectives, namely robustness to parameter perturbations, generalizability, and sensitivity consistency. In the latter half, we extend the applicability of the explainability approaches to 3D computer vision field so that the trustworthiness of point cloud models is enhanced. We first extend the perturbation-based approach to point clouds and provide online toolkits to facilitate practical implementation. Subsequently, we propose two activation maximization-based point cloud global explainability approaches, which visualize input instances that are representative for specific categories. Moreover, we propose a non-DNN point cloud classifier that utilizes multi-scale fractal windows to extract distributional information and makes predictions via random forests, which significantly enhances the explainability compared to DNNs. Further, we adversarially analyze the decision sensitivity of point cloud models with the help of saliency maps generated by explainability methods. Finally, we analyze how the model learns 3D geometric features by analyzing the distribution of activations in the intermediate layers. Extensive experiments demonstrate that the proposed method contributes to the explainability evaluation and its adaptability on point clouds.

Publications

This dissertation is based on the following publications. [1], [4], [5] are supervised by Prof. Katharina Morik, and [7], [9], [6], [8], [3], [10], [11] are under the supervision of Prof. Emmanuel Müller. Dr. Helena Kotthaus serves as supervisor and editor for [1], [7]. [2] is collaborated with Katharina Beckh, Sebastian Müller, Matthias Jakobs,Vanessa Toborek, Raphael Fischer, Pascal Welke, Sebastian Houben and Laura von Rueden.

[1] H. Tan and H. Kotthaus, "Surrogate model-based explainability methods for point cloud nns," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2239–2248.

[2] K. Beckh et al., "Harnessing prior knowledge for explainable machine learning: An overview," in *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, IEEE, 2023, pp. 450–463.

[3] H. Tan, "Fractal projection forest: Fast and explainable point cloud classifier," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 4240–4249.

[4] H. Tan, "Maximum entropy baseline for integrated gradients," in *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023, pp. 1–8. DOI: 10.1109/IJCNN54540.2023.10191554.

[5] H. Tan, "The generalizability of explanations," in *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023, pp. 1–8. DOI: 10.1109/IJCNN54540.2023.10191972.

[6] H. Tan, "Visualizing global explanations of point cloud dnns," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 4741–4750.

[7] H. Tan and H. Kotthaus, "Explainability-aware one point attack for point cloud neural networks," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 4581–4590.

[8] H. Tan, *Dam: Diffusion activation maximization for 3d global explanations*, 2024.

[9] H. Tan, "Evaluating explanation robustness to model pruning," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–8. DOI: 10.1109/IJCNN60899.2024.10650278.

[10] H. Tan, "Do point cloud models learn object contour features?" In *2025 International Joint Conference on Neural Networks (IJCNN)*, 2025, pp. 1–8. DOI: 10.1109/IJCNN64981.2025.11228371.

[11] H. Tan, "Evaluating sensitivity consistency of explanations," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025, pp. 182–191. DOI: 10.1109/WACV61041.2025.00028.

Contents

List of Figures

List of Tables

List of Online Resources

Part of the code for this dissertation is available in the following URLs

- The code for *Surrogate Model-Based Explainability Methods for Point Cloud NNs* (chapter 6) is available on `https://github.com/Explain3D/LIME-3D`.
- The code for *Visualizing Global Explanations of Point Cloud DNNs* (chapter 7) is available on `https://github.com/Explain3D/PointCloudAM`.
- The code for *DAM: Diffusion Activation Maximization for 3D Global Explanations* (chapter 8) is available on `https://github.com/Explain3D/DAM`.
- The code for *Explainability-Aware One Point Attack for Point Cloud Neural Networks* (chapter 10) is available on `https://github.com/Explain3D/Exp-One-Point-Atk-PC`.

Part I.

Introduction

1. Introduction

1.1. Motivation

With the rapid evolution of deep learning, artificial intelligence (AI) systems have achieved remarkable performance in various complex tasks, gradually becoming core components in safety-critical domains such as autonomous driving, medical diagnosis, and industrial control. However, the inherent black-box nature of deep learning models has become a prominent bottleneck restricting their reliable deployment. These models make decisions based on high-dimensional and abstract feature representations, and the logical chain between input data and output results is often incomprehensible to humans. This lack of explainability not only undermines users' trust in AI systems but also poses severe potential risks - when AI makes erroneous decisions in critical scenarios (e.g., autonomous driving, medical diagnosis), the inability to trace the root cause may lead to irreversible consequences such as financial losses or threats to human life.

Explainable AI (XAI), as a discipline dedicated to decoding the black-box of models, has emerged as a key solution to address this problem. In recent years, XAI research has made significant progress in the 2D vision field, proposing a series of explainability methods such as generating saliency maps and counterfactuals. However, a critical gap exists in the current XAI research landscape: the majority of methods focus on generating explanations while neglecting the evaluation of explanations. Due to the absence of ground truth, most explanation results are only verified through subjective human judgment, lacking unified, objective, and quantitative evaluation criteria. This leads to three core issues: first, the quality of explanations cannot be compared horizontally between different methods. Second, some explanations may present spurious correlations that superficially match model decisions but deviate from the true causal relationship. Third, the fidelity of explanations cannot be guaranteed, making it impossible for them to provide reliable decision support in critical scenarios. Therefore, the construction of an evaluable XAI system - realizing quantitative assessment, validity verification, and quality ranking of explanation results - has become an urgent demand in the development of XAI, which constitutes the fundamental motivation for studying evaluable explainability.

While the above evaluable XAI framework addresses core limitations in general XAI research, its application to complex 3D vision tasks—particularly those involving point cloud data—presents unique challenges due to the intrinsic properties of point clouds. This amplifies the urgency of adapting evaluable XAI to 3D modalities. As a typical representation of 3D spatial data, point clouds directly record the 3D coordinate information of objects, and thus have become the primary data form in 3D vision tasks such as 3D object

detection, point cloud segmentation, and robotic grasping. However, the unique characteristics of point clouds - including disorderliness, sparsity, and strong spatial correlation - pose unprecedented challenges to the application of XAI methods and their evaluation systems. Compared with structured 2D images, the visualization of point cloud explanations (e.g., highlighting critical points or local regions) is more abstract, and subjective assessment is more error-prone. Meanwhile, the spatial invariance and scale variability of point clouds make it difficult to directly migrate the evaluation metrics applicable to 2D vision to the 3D field, resulting in a more severe lack of effective evaluation tools for point cloud-based XAI methods.

To sum up, the general demand for breaking the black-box limitation of AI models promotes the emergence of evaluable XAI, and the unique characteristics of point cloud data and its critical application value further strengthen the urgency of applying evaluable XAI to 3D vision scenarios. This dissertation thus focuses on these dual motivations with two core objectives. For evaluable XAI, it aims to construct a scientific and comprehensive theoretical and methodological system for explanation evaluation. For point cloud applications, it intends to specifically design explanation methods and assessment mechanisms adapted to the inherent properties of point cloud data. By integrating these two core objectives, this research aims to transform explanation results from subjective description to objective quantification, thereby facilitating the reliable deployment of 3D vision AI systems in safety-critical domains.

1.2. Evaluable explainability

1.2.1. Existing metrics

Existing XAI evaluation methods are fundamentally categorized into two complementary paradigms, each with distinct strengths and inherent limitations, while algorithm-based quantitative methods have further evolved into multiple focused evaluation directions:

- **Human-based qualitative evaluation**: Centered on user perception, this paradigm primarily relies on user studies - presenting explanations to participants to collect subjective ratings or feedback [193]. Its core advantage lies in directly capturing human-centric assessment, aligning with the human-oriented essence of explanations. However, it is plagued by subjectivity (conflicting feedback between experts and non-experts), poor reproducibility, and high costs (e.g., recruiting sufficient participants), limiting its scalability as a universal evaluation standard.
- **Algorithm-based quantitative evaluation**: This paradigm quantifies explanation quality via designed metrics, eliminating human bias to ensure high reproducibility and low implementation costs, thus becoming the mainstream in academic research [179]. Nevertheless, it suffers from three key flaws: first, most metrics focus on a single property of explanations (e.g., only fidelity or sensitivity) rather than their holistic performance. Second, the absence of ground-truth explanations leads to a

lack of universally recognized unified standards. Third, results may deviate from human intuition, thereby weakening evaluation trustworthiness. Typical focused directions of quantitative evaluation include:

- **Sensitivity**: This is the most intuitive metric, verifying explanation fidelity by observing prediction changes when high-attribution features are modified (deleted/inserted) [35]. While widely used, traditional methods either disrupt input feature distributions (direct modification [95], [189], [194]) or deviate from the original model (e.g., ROAR's retraining strategy [103]), failing to balance validity and fidelity.
- **Robustness**: It evaluates whether explanations remain consistent for slightly perturbed inputs, based on the premise that similar inputs should yield similar explanations. Existing research [86], [199] mostly focuses on input-level perturbations (e.g., Gaussian noise) and overlooks changes in the model itself.
- **Sanity check**: It validates if explanations rely on meaningful model parameters by randomizing internal model structures. Explanations unaffected by randomization are deemed low-fidelity [84].
- **Pointing game**: Specific to visual tasks, it measures the probability that high-attribution features align with target objects [89], [144]. Its limitation lies in ignoring cases where models make decisions based on non-target features, reducing result reliability [107].
- **Custom metrics**: These are tailored for specific explanation methods (e.g., surrogate model fidelity for LIME/SHAP [196], [200], feasibility for counterfactual explanations), but lack universality and general applicability.

1.2.2. Challenges

While existing evaluation methods lay a foundational groundwork for explaining AI models, they still fall short in addressing practical application demands, with four core challenges standing out prominently. These challenges not only limit the reliability of evaluation results but also hinder the systematic development of XAI, as detailed below:

- **Violation of "uninformativeness"**: For sensitivity tests and some gradient-based attribution methods (e.g., Integrated Gradients), baseline selection remains highly arbitrary and lacks a universal standard. More critically, most commonly used baselines (such as zero vectors, random noise, or mean value vectors) fail to meet the theoretical "uninformative" definition, as they inherently carry prior information about input data distribution, which contaminates the attribution results and leads to bias. This not only leads to significant discrepancies in faithfulness evaluation across different baselines but also makes it impossible to accurately quantify the true contribution of input features to model decisions.

- **Input-centric robustness, ignoring parameter-level stability:** Current robustness evaluation of explanations is almost exclusively centered on input feature perturbations (e.g., adding Gaussian noise, cropping images). It largely overlooks the robustness of explanations against model-internal parameter changes - such as model pruning, quantization, or fine-tuning, which are common operations in practical deployment. Explanations that appear robust to input disturbances may undergo drastic changes when the model's parameter structure is adjusted, resulting in a loss of practical value in lightweight or updated model versions.
- **Incomplete coverage of high-impact features (necessity vs. sufficiency):** Traditional sensitivity evaluation typically adopts a feature deletion paradigm - removing high-attribution features highlighted by explanations and observing whether the model's prediction confidence declines significantly. This approach only verifies the necessity of high-attribution features, i.e., whether these features have a significant impact on the model's prediction results. However, it fails to validate the sufficiency of the explanation itself: that is, whether all features that exert a major influence on the prediction are included in the high-attribution feature set identified by the explanation. There may thus be a critical gap: some non-highlighted features (low-attribution in explanations) could actually be key to the model's decision-making, but traditional sensitivity tests fail to detect such omissions, leading to an incomplete and one-sided assessment of explanation quality.
- **Bias from disruption of feature distribution:** A majority of quantitative evaluation methods (e.g., direct deletion/insertion of high-attribution features) disrupt the inherent distribution of input features. The modified inputs (e.g., images with arbitrary regions erased, point clouds with key points removed) often fall outside the data distribution that the model was trained on, meaning the model's response to these unnatural inputs does not reflect its real decision-making logic. This distribution mismatch directly leads to biased evaluation results, thereby making it difficult to trust the authenticity of the assessment.

Figure 1.1 presents an overview of the current challenges in evaluable explainability, which almost cover the entire decision-making process and pose non-negligible threats to the credibility of evaluation of explainability methods.

1.2.3. Contributions

Aiming at the above challenges in the field of XAI evaluation, this dissertation carries out systematic exploration around four core directions, and achieves targeted breakthroughs through method innovation:

- **Uniform and uninformative baselines - Chapter 2.** To tackle the core issue of baseline inconsistency and uninformative violation, this work focuses on the baseline flaws in the sensitivity test and gradient-based attribution methods (e.g., Integrated Gradients) and proposes a unified maximum entropy baseline that conforms to the

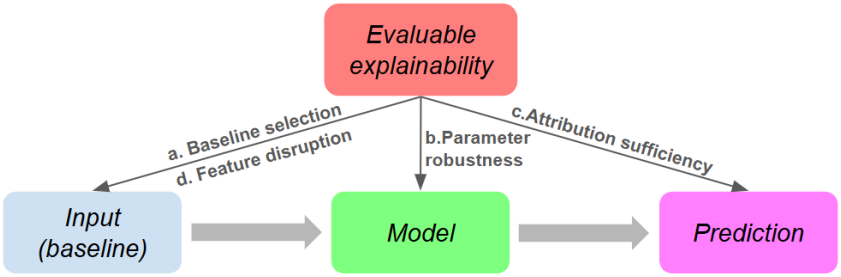

Figure 1.1.: Overview of the four challenges in evaluating explainability during the decision-making process. We propose corresponding solutions for these issues respectively in Chapters 2, 3, 4 and 5 to enhance the evaluability and plausibility of explainability methods.

uninformative definition. This baseline completely eliminates all residual input-related information, fundamentally avoiding the information interference of traditional baselines (such as zero vectors and random noise) and ensuring that feature attribution results truly reflect the model's decision-making logic. This standardized baseline can break the comparison barrier in cross-study sensitivity evaluation, providing a consistent measurement benchmark for different XAI works and facilitating the quantitative comparability of evaluation results.

- **Addressing parameter robustness - Chapter 3.** To address the limitation of input-centric robustness while ignoring parameter-level stability, this work breaks out of the traditional framework of input perturbation and establishes the first explanation robustness evaluation system tailored for model pruning scenarios. With explanation consistency against pruning as the core indicator, the system quantitatively monitors the stability of explanation results during continuous model parameter compression by integrating gradient-based pruning strategies and dynamic evaluation protocols. This method fills the gap in robustness evaluation under model structure changes and provides key technical support for the reliable application of XAI methods in lightweight models.

- **Evaluation of attribution sufficiency - Chapter 4.** To address the incomplete coverage of high-impact features in traditional sensitivity tests, this work proposes a dual-sensitivity verification framework by comparing the consistency between prediction sensitivity and explanation sensitivity. Specifically, we separately test two dimensions: 1) the sensitivity of model predictions to feature changes (to identify all high-impact features for decisions). 2) the sensitivity of explanations to the same feature changes (to confirm which features are marked as high-attribution). By quantifying the consistency between these two sensitivity profiles, we ensure bidirectional validity: high-attribution features in explanations indeed exert significant impacts on predictions, and all features with major predictive impacts are included in the high-attribution set of explanations. This approach overcomes the

one-sidedness of traditional single-direction sensitivity tests and achieves comprehensive verification of explanation necessity and sufficiency.

- **Avoiding feature distribution disruption - Chapter 5.** Targeting the feature distribution disruption issue, this work develops a generalizability-based evaluation method that eliminates the need for any artificial feature perturbation. The core idea is to verify the learnability of the association between explanations and inputs using a generative model: we train a generative model to learn the inherent correlation between input data and corresponding explanations from the original dataset (without modifying features). The generalizability of explanations is then assessed by the model's ability to accurately map new, unseen inputs to reasonable explanations. Since this process relies solely on natural data distribution and avoids manual feature manipulation, it fundamentally prevents distribution disruption, ensuring that evaluation results reflect the authentic association between inputs, explanations, and model decisions.

1.3. Applications to point clouds

1.3.1. The specificity of point clouds

Image data is typically structured as a dense, grid-like tensor where local connectivity and translational invariance are easily exploited by standard convolutional neural networks. However, 3D point cloud data presents a fundamentally different paradigm. A point cloud is an unordered set of 3D coordinates, often complemented by attributes like color or intensity. This inherent structure dictates the design of specialized point cloud neural networks and poses unique challenges:

- **Permutation invariance:** The core requirement for point cloud models is invariance to the ordering of input points, as an arbitrary permutation of the points should not change the output prediction. This is typically addressed through symmetric functions, most notably global max-pooling employed in PointNet [69].
- **Sparsity and irregularity:** Unlike the uniform density of image pixels, points are sparsely and non-uniformly sampled from the 3D surface of an object. This irregularity makes applying standard grid-based convolution operators inefficient and inappropriate.
- **Neighborhood definition:** Point cloud models must employ specialized mechanisms (such as k-Nearest Neighbors or ball querying) to define and aggregate features from local regions, often leading to complex graph-based or sampling/grouping operations (e.g., in PointNet++ [73] and DGCNN [119]).

These structural differences necessitate models that perform complex geometric reasoning while maintaining permutation invariance, thus fundamentally differentiating point cloud models from standard CNNs designed for dense image grids.

1.3.2. Research gaps and possible routes in point cloud XAI

Given the unique architectural designs and the structural complexity of point clouds, research into the explainability of point cloud models is currently in its nascent stage. As a result, this area lacks the vast body of foundational research, standardized evaluation metrics, and easily adopted baseline methodologies that are readily available in the mature 2D image domain. Almost all existing methods remain at the most basic stage of explainability analysis, for instance, the direct transplantation of attribution methods from 2D images to point clouds [139], or the simple analysis of sensitivity between outputs and inputs [126]. While these methods offer some assistance in promoting the reliability of point cloud models, a more in-depth explainability analysis needs to be conducted to meet human safety requirements in various fields where point cloud data is applied.

The potential development of point cloud XAI primarily follows two distinct paths, both of which highlight the divergence from standard image explainability:

1. **Adaptive modification from 2D methods:** Post-hoc XAI techniques, such as those based on local fidelity (like LIME [54]) or input saliency mapping derived from gradients (like Integrated Gradients [78]), hold conceptual appeal for point clouds. These methods aim to identify input features - individual points or small localized 3D clusters - that are most influential in determining a model's output. However, a direct transplantation of these techniques proves largely ineffective. The core issue lies in the fundamental difference between perturbing a dense, continuous pixel grid versus a sparse, unordered set of points. Applying perturbations (e.g., masking or occlusion) in the 3D space often requires different strategies to preserve the underlying geometry and permutation invariance. For instance, occluding a point cloud must be handled by point removal or shifting, which fundamentally alters the input domain, rather than simply replacing continuous pixel values. Therefore, achieving meaningful and faithful explanations requires significant adaptive modifications to properly model the sampling and grouping operations inherent to point cloud architectures, ensuring the resultant explanations reflect true model behavior rather than artifacts caused by unrealistic data perturbations. This adaptation is critical for translating successful 2D XAI concepts into the 3D domain.

2. **Point cloud structure-specific XAI:** The second, and arguably more crucial, strategic path involves designing XAI methods specifically tailored to exploit and analyze the geometric and structural properties unique to point cloud models. These approaches are fundamentally not transplantable from image XAI because they are intrinsically linked to point cloud model features like local feature aggregation, grouping operations, or specialized pooling functions. Key directions of investigation within this path include:

    - **Intrinsically interpretable design:** The development of interpretable and transparent classifiers that eliminate the need for post-hoc analysis by providing decision rules directly traceable to geometric or spatial features, thereby offering immediate explanation and improving efficiency.

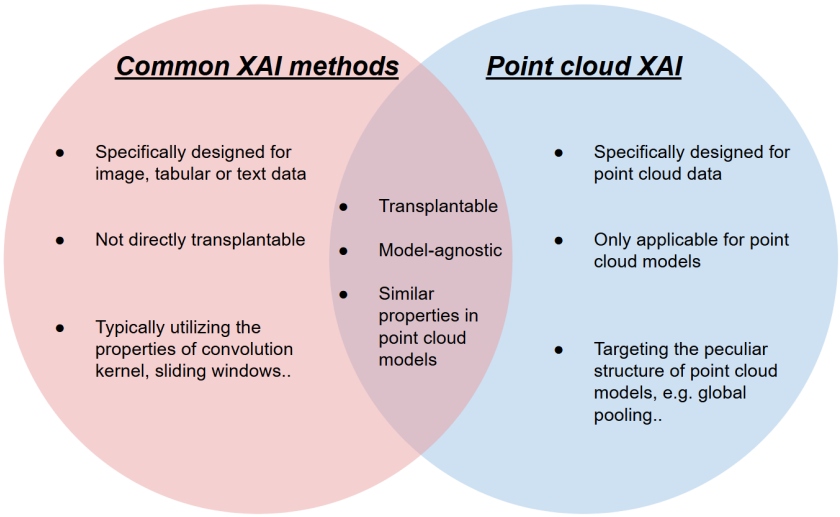

Figure 1.2.: The relationship between general and point cloud XAI methods. The overlapping area refers to methods applicable to point clouds, which are extended from general methods through adaptive modifications. This includes the work presented in Chapters 6, 7 and 8. The remaining non-overlapping area corresponds to XAI methods specifically designed based on the special structure of point cloud models, covering the work in Chapters 9, 10 and 11.

- **Critical point sensitivity analysis:** Focusing on the sparse nature of point clouds to perform rigorous sensitivity analysis aimed at identifying and quantifying the influence of minimal, critical point subsets on the model's prediction, which is crucial for assessing robustness in sparse data environments.
- **Intermediate feature representation probing:** Analyzing the hidden vector representations in the intermediate layers of the point cloud network to decipher the specific geometric and structural features extracted by the unique aggregation and pooling mechanisms, thereby illuminating the network's internal reasoning process.

In summary, the field of point cloud XAI is still emerging and lacks established baselines. Current efforts proceed along two main directions: first, adaptively modifying existing 2D XAI methods to handle the sparse, unordered nature of point clouds. And second, designing point cloud structure-specific XAI tailored to unique 3D architectural features. Figure 1.2 presents an overview of the relationship between general and point cloud XAI, and also illustrates two promising research directions within point cloud XAI.

1.3.3. Contributions

To address the challenges in developing and adapting explainability methods for point cloud analysis, this dissertation provides significant advancements along the two principal paths of point cloud XAI research. Specifically, our work simultaneously develops

robust adaptive extensions of established 2D techniques and pioneers novel structure-specific methods tailored to the unique geometry of point clouds. The key contributions are outlined below:

- **Adaptive modification of 2D explainability methods:** This path focuses on overcoming the limitations of directly porting image XAI techniques by proposing structurally-aware modifications:
  - **3D surrogate model explainer - Chapter 6:** We extend the perturbation and surrogate model-based method (LIME) to point clouds. By leveraging Farthest Point Sampling (FPS) to define uniformly segmented super-points and enabling baseline-free perturbation via selective point ablation, we circumvent the critical issue of perturbation baseline selection inherent in 3D data. This results in the first black-box explainability method demonstrably applicable to point clouds, with code released for practical deployment.
  - **3D global explanation visualization - Chapters 7 and 8:** We initiated the study of point cloud global explanation visualization by introducing a stable synthesis approach. Specifically, we first demonstrate the ineffectiveness of standard image-based parametric regularization for point cloud Activation Maximization (AM) and proposed an initial solution using an Autoencoder-based regularization approach (Chapter 7). Building upon this, we further optimized the visual quality and perceptibility by integrating Denoising Diffusion Probabilistic Models (DDPM) as a superior generative model. This optimization allows the precise focus on the trajectories of prediction-critical points, significantly enhancing the fidelity of the synthesized global explanations (Chapter 8).
- **Structure-specific XAI for point clouds:** This path involves developing methods intrinsically linked to the geometric and structural properties of point cloud architectures:
  - **Inherently interpretable classifier - Chapter 9:** We introduce Fractal Projection Forest (FPF), a non-deep learning classifier that achieves enhanced transparency and explainability. FPF leverages the multi-dimensional projection properties of point clouds to learn statistical features via multi-scale sliding windows, significantly reducing runtime and enhancing transparency compared to DNNs without substantial accuracy loss.
  - **Critical point sensitivity analysis - Chapter 10:** Through adversarial attacks guided by Integrated Gradients (IG)-derived saliency maps, we provide a rigorous sensitivity analysis showing that point cloud model predictions are highly sensitive to perturbations at critical input points. Furthermore, we demonstrate that replacing the max-pooling layer with less sensitive alternatives (e.g., average pooling) can be a viable strategy to improve decision stability.

– **Latent feature alignment for global explanation synthesis - Chapter 11:** We conduct an in-depth analysis of intermediate layer activation distributions, revealing a strong correlation between similar input contours and highly similar latent activation patterns. Based on this discovery, we propose a novel explainability approach that synthesizes well-contoured global explanations without requiring any generative models, achieved solely by aligning these latent activation distributions.

1.4. Summary and outline

This dissertation makes substantial contributions to the field of XAI by addressing two fundamental dimensions: the establishment of a more comprehensive evaluation framework to ensure explanation reliability, and the development of domain-specific methodologies tailored for point cloud analysis. Table 1.1 provides an overview of the topics, challenges, and corresponding contributions of this dissertation.

The structure of this dissertation is as follows: Part I presents an introduction and background related to the dissertation. In Parts II and III, we elaborate on the proposed methods for evaluable explainability and the application of explainability to point clouds, respectively. Finally, in Part IV, we conclude the dissertation and discuss future work.

Part	Title	Challenge/gap	Topic	Publication
I	Introduction & Backgrounds			
II	Assessable explainability	Uninformative baseline	Maximum entropy baseline	Chapter 2, [4]
		Parameter-Level stability	Robustness against pruning	Chapter 3, [9]
		Explanation sufficiency	Sensitivity consistency	Chapter 4, [11]
		Disruption of feature distribution	Generalizability	Chapter 5, [5]
III	3D explainable applications	Adaptive modification from 2D methods	Perturbation & surrogate model-based method	Chapter 6, [1]
			Global point cloud explanation	Chapter 7, [6]
				Chapter 8, [8]
		Point cloud structure-specific XAI	Interpretable 3D model	Chapter 9, [3]
			Adversarial sensitivity analysis	Chapter 10, [7]
			Latent activation analysis	Chapter 11, [10]
IV	Conclusion & Future work			

Table 1.1.: Overview of the challenges, topics, and corresponding chapters in this dissertation.

Part II.

Evaluable explainability

2. Maximum Entropy Baseline for Integrated Gradients

The choice of baseline is critical both for perturbation-based evaluations and for those explainability methods that involve the integration of information. A baseline that fulfills the definition of uninformative enhances the credibility of the evaluation as well as the performance of the explainability approaches. However, no widely acknowledged baseline that strictly satisfies the definition of uninformative has been proposed to date, leading to possible mechanistic biases in the assessment of explanations from different works. In this chapter, we address the baseline challenges identified in Sec. 1.2.2 and investigate the choice of baseline starting from Integrated Gradients, whose performance is most sensitive to the baseline selection, as well as being the most easily observed. We propose a new uniform baseline, i.e., the Maximum Entropy Baseline, which is consistent with the "uninformative" property of baselines defined in IG. In addition, we propose an improved ablating evaluation approach incorporating the new baseline, where the information conservativeness is maintained. Furthermore, we explain the linear transformation invariance of IG baselines from an information perspective. Finally, extensive experiments indicate that IG with Maximum Entropy Baseline performs superiorly and the ablation tests derived from it are more plausible.

2.1. Introduction

Integrated Gradients (IG) [78] is one of the most widely used explainability methods at present, which illustrates the attribution of each pixel in the input x to the prediction of the model F. IG is a gradient-based approach that addresses the gradient saturation problem [58] of vanilla gradients by integrating them from a chosen baseline x'. IG is formulated as:

$$IG_i = (x_i - x_i') \cdot \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x} d\alpha \tag{2.1}$$

The baseline is one of the most important parameters of IG, which strongly impacts the performance of the explanations. The existing studies, although specifically suggesting different baselines, are almost unanimous in their overall definition: missingness, i.e., the input that disables the model from capturing information. Currently the most prevalent baselines are: 1) *Zeros*: filling all input units with zeros. 2) *Black (white) vectors*: replacing

all units with the minimum (maximum) of the current input. 3) *Random initializations*. In addition, [152] complements several possible alternatives: 4) *Max-distance pixels (Xdist)*: their study reveals that the explanations generated by IG tend to be more sensitive to pixels that differ significantly from the baseline, while ignoring those that are similar, even if they are located inside the object to be explained. Thus, they propose Xdist, i.e., replacing each pixel with the point that is farthest apart in chromaticity space. 5) *Blurring*: information is eliminated by blurring the input with kernel functions. 6) *Random noises*: as a supplement of 3), random noise of different distributions (e.g. Gaussian and uniform) can be substituted or added to the input. 7) *Average over datasets*: random samples are drawn from the training dataset and the average is taken. Notably, while 1)-4) assume that the pixels are independent of each other, 5)-7) incorporate correlation between pixels. However, we argue that the existing concept "missingness" lacks quantitative metrics, which leads to the ambiguity in the choice of baselines.

Another cause of difficulty in baseline selection is the lack of ground truth, such that evaluating the performance of explanations is challenging. Ablation test [152] is a simple and intuitive solution. The core principle is that if the pixels with positive attributions are ablated, the model drops its confidence in the prediction and vice versa. Nevertheless, we believe that the existing ablation tests suffer from insufficient reliability, primarily due to 1) No uniform replacements as substitutes for the original pixels. For instance, both [35] and [56] employ ablation tests to evaluate performance of explanations. However, with respect to the substitution pixels, [35] chooses the opposite number of the current pixel while [56] samples random numbers from uniform distribution. This raises the question of which substitute is more appropriate, or if there is a better alternative. 2) No guarantee of "Neutral": after flipping selected pixels, there is no metric for whether information residues survive. Both of the above points may potentially threaten the reliability of explanation assessment.

To address the aforementioned weaknesses, this work re-examines the role of the baseline from the perspective of information and refines the ablation test to satisfy conservativeness. We observe and propose our conjecture on a synthetic toy dataset and validate it on MNIST handwriting dataset. Moreover, we also compare ours with other existing baselines on CIFAR10 and a real world dataset called Stanford Car. Our contribution is primarily summarized as follows:

- We propose a new baseline that remains "uninformative" by definition, and by constructing toy datasets with transparent ground truth explanations, we observe that our baselines closely approximates the ground truths.
- We propose an improved ablation test for evaluating explanations that specifies a uniform substitution of ablated pixels and simultaneously satisfies the conservativeness of the information quantity.
- We explain the failure of the linear invariance experiments of the IG (proposed by [105]) from the perspective of the entropy.

- We quantitatively compare all possible baselines with the proposed evaluation metrics on different data sets.

This chapter is structured as follows: Section 2.2 introduces the current studies on IG baselines and ablation tests. Sections 2.3 and 2.4 elaborate the experiments related to the proposed baseline and ablation test, respectively. Section 2.5 presents the results of the experimental evaluation of the proposed methods.

2.2. Related Work

Investigation of IG baseline. So far, there are no sufficient researches for IG baselines. Several studies [105], [114], [152] point out that the choice of baseline impacts dramatically on the explanation performance, with the appropriate baseline generating clearer attribution maps and the opposite distorting them. [105] also indicates that a number of specific baselines (e.g., zeros) do not satisfy linear transformation invariance and are hence unreliable. The proposer of IG [78] provides a definition of baselines, i.e., when the model prediction is neutral, and suggest that for most networks the zero baseline is applicable, while also specifying different applications for networks in different domains, e.g., the black background for image networks and the zero embedding vector for NLP tasks. As a complement, [114] proposes several additional baselines that are intuitively feasible, such as the Max-Distance and Blurred baselines. Although these baselines are either consistent with human intuition or address certain deficiencies of existing ones, there is no convincing argument that the proposed baselines are "uninformative".

Ablation test. In LRP [35], the ablation test is first proposed as a validation for the explainability method, which has being followed or expanded in several studies relating to feature-wise attributions [56], [113], [126], [1]. However, [103] pointed out that the method suffers from the trouble that ablating a single feature might impair the feature correlation and data distribution, and they suggest that a new model should be retrained after ablating for accuracy validation (Remove & Retrain). Again, this also leads to debates that the explanations should be faithful to the data or the model [145], [155]. Leaving aside the controversy about feature correlation, we expose another potential risk of ablation experiments from the information perspective, i.e., non-conservation, and propose a corresponding solution.

2.3. Max entropy baseline

The ideal baseline is the input that "is neutral" [78] and "contains no information" [152]. However, measuring the amount of information embedded in the inputs is challenging since most models are black boxes and only final predictions are observable. On the other hand, we argue that the referent of "neutrality" is ambiguous. Existing referents for the baseline are 3 categories: uninformative for humans, data and models. Most baseline choices are for humans, such as zero, black and random baselines: it is intuitively

assumed **by humans** that these values do not yield any information. Nevertheless, for the model to be explained, this may be problematic (e.g., a classifier that distinguishes between black and white images). Subsequently, those baselines for the dataset (e.g., Average of training data) also raise controversy: whether the explanation should be true to the data or the model [145], [152], [155]. So far, there is hardly any baseline for the model being proposed. To address the aforementioned flaws, we introduce the entropy of logits as a metric for quantifying information residual in the model. We denote the proposed baseline as B_{Xentr}, and is formulated as:

$$B_{Xentr} = \arg\max_{x} Entr(Softmax(f_l(x)) \tag{2.2}$$

where f_l denotes the logits of the model and $Entr(*)$ denotes the entropy function [13]:

$$Entr(A) = -\sum_{i=1}^{n} P(a_i) log P(a_i) \tag{2.3}$$

Our newly proposed baseline has the following advantages:

- *Input information excluded.* Several baselines incorporate (or incompletely ablate) information from input instances to attain better visualizations. We argue that this violates the definition of baselines. For example, The Maximum Distance baseline (Xdist) [152], which calculates the maximum colorimetric distance of each pixel from the input, moderates the phenomenon of "attribution disappearance" in the explanations. Nevertheless, this baseline obviously contains extensive information about the input, such as object outlines (see figure 2.1).
- *Feature correlation Incorporated.* The plausibility of generating explanations based on the assumption of feature independence remains questionable. Owing to the strong correlation between features, learning a certain distribution based on the dataset is a more convincing solution [152]. Alternatively, the baseline we proposed is derived from the optimization of a trained model, which itself possesses the distribution of the dataset.
- *Linear transformation invariant.* [105] suggests that the generated explanation should remain constant when the input and the model undergo a uniform linear transformation (all pixels transformed with the same amplitude). A portion of the baselines fails this test (e.g., zero baseline), while several others (e.g., black baseline and ours) survive. In addition, we elucidate from an information perspective why all baselines fail the non-uniform linear transformation test in section 2.3.3.
- *Computational simplicity.* Only a simple gradient ascent procedure is required once for each model to obtain B_{Xentr}, and is applicable to all predictions made by the model.

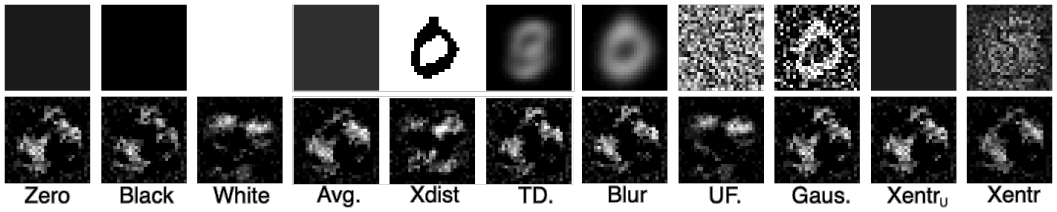

Figure 2.1.: Visualization of different baselines and corresponding explanations. From left to right: Zero, black, white, average of current instance, Max Distance, average of training data, blurred, uniform distributed, Gaussian distributed, uniform and non-uniform Maximum Entropy baselines.

Figure 2.1 shows all currently available IG baselines and their generated explanations. Ahead of assessing the validity of the proposed baseline, we show our observations with respect to the correlation between the entropy of logits and the explanations of IG. In section 2.3.1 we show experimental observations in a tabular toy dataset, and in section 2.3.2 we extend the findings to the MNIST handwriting dataset. Note that for achievable computational complexity, in this section the baseline is treated as a uniform value (B_{Xentr_u}).

2.3.1. Warm-up: tabular toy datasets

The lack of ground truth explanations is one of the obstacles to investigating baselines. Before experimenting on the real dataset, we simplified the problem by selectively creating features relevant to the labels to obtain ground truth attributions in advance. We artificially create tabular toy datasets whose ground truth explanations are available. An overview of the dataset is shown in figure 2.2. Our dataset involves 3 parameters, namely n_f, k and c, representing the total number of features, the number of features correlated with the label and the total number of label categories, respectively. For each individual dataset, there is at least one feature correlated with the label ($k \geq 1$). Hence, we can obtain a unique ground truth explanation for this dataset, i.e., an attribution of 1 for the features relevant to the labels and 0 for the opposite. Note that when $k > 1$, the labels must be computed jointly based on all relevant features, otherwise the model may learn only a part of them, leaving the ground truth explanations redundant. Each dataset consists of 10^4 instances (most of the data are duplicates due to the limited combination of features), of which 80% are used as training data and the rest are for testing. A simple two-layer fully connected network is trained, which achieves 100% accuracy on both the training and test sets. We traverse the explanations of all the baselines in the valid data value range with 100-step IG. Subsequently, we measure the gap between the IG-generated and ground truth explanations. As the exact value of the ground truth explanations is not reachable, we quantify the loss with KL-divergence, i.e., features related to the label should possess as large attributions as possible, otherwise should approximate to 0. It is notably that only the results of the instances where all features are 0 as the object being explained are shown, since according to the properties of the dataset, the ground truth

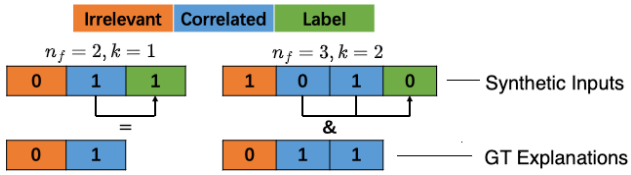

Figure 2.2.: The structure of the tabular toy data set, where n_f and k denote the total amount of features and the number of those relevant to the labels, respectively. Each label is derived by logical operations based on the selected features (middle row), i.e. it is relevant only to the features involved in the operation. The last row illustrates the ground truth explanations.

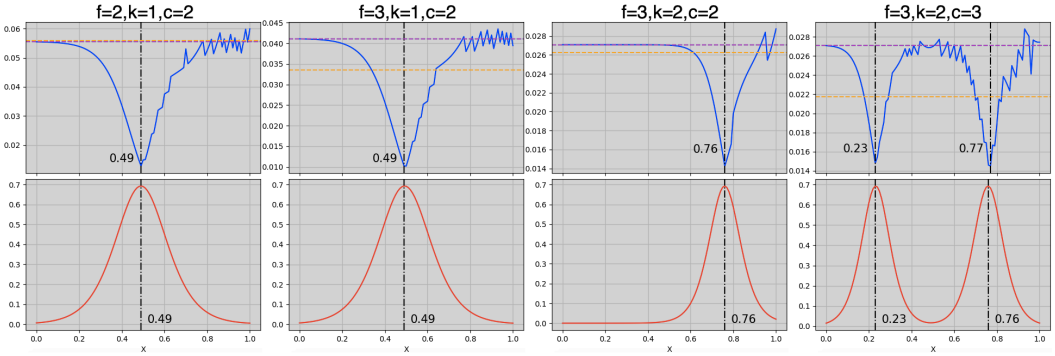

Figure 2.3.: First row: KL-losses of between the IG explanations obtained with the corresponding inputs (x-axis) as baselines and the ground truth explanations. Second row: entropy curves of the model. n_f denotes the total number of features, k denotes the number of features associated with the label, and c denotes the total number of label categories. We also marked the loss of zero (green), black (cyan), white (purple) and random (orange) baselines (partially obscured by each other, but significantly higher than the loss of B_{Xentr}).

explanations should be independent of the input instances, and we practically tested the input instances for all feature combinations and the results are consistent. On the other hand, we draw the entropy curve of the model over the valid range of data values. The prediction are fed into a Softmax function to assure that each logits neuron is positive, and then the entropy of this probability vector is calculated.

The results are reported in figure 2.3. Interestingly, the loss and entropy curves follow similar trends and reach extreme values at approximate baselines. These results are intuitively correct: the maximum predictive entropy implies the minimum information content of the input vector, which exactly coincides with the definition of the baseline in IG.

2.3.2. Observation: MNIST handwriting dataset

To extend this conclusion to a more general scenario, a similar experiment is conducted on the MNIST handwritten dataset. However, the major challenge with real datasets compared to the previous one is that no prior knowledge is available about which features (pixels) in the instance are decisive for the labels, i.e., the ground truth explanations are not accessible. Existing studies treat those pixels located inside the object as ground truth [32], [98]. Nevertheless, no guarantee can be given that the model does not utilize any information from the background when inferring [131]. Our alternative approach is to observe the hybrid explanations generated by multiple explainers (average of explanations generated by Back-Propagation-Based, Taylor Decomposition, LRP and DeepLift series). Notwithstanding the inability to precisely restore the ground truth explanations, this approximated saliency map captures the trend of the attribution distribution given by the majority of explainers. Before training, all data are transformed to the value domain $[-0.42, 2.82]$ for higher accuracy. We train two different types of networks, the FC networks consisting of only fully connected layers and the CNN networks containing convolutional layers. Similarly, we search all the baselines in the (discretized) valid value range and produce the explanations by a 100-step IG. Owing to the high dimensionality, KL-loss struggles to reveal the distributional discrepancy, we therefore adopt Spearman's rank correlation coefficient to measure the similarity between explanations. We randomly selected 100 instances from the test set for each model. The methodology of statistics is that we initialize a histogram containing m bins, each corresponds to a baseline within the (discretized) valid value range. For each instance, we try all m baselines and yield m explanations, we then compare each of them with the hybrid explanation. The baseline that generates the closest explanation to the hybrid one (with maximum Spearman's coefficient) is counted in the corresponding bin. The curve of logits entropy is also plotted.

The results are demonstrated in figure 2.4 and the maximum of both recordings are annotated with a dashed line in the corresponding color. As can be observed from FC1 and CNN1, the baseline with the lowest loss almost coincides with the input of the maximum logits entropy. To exclude the possibility that the minimum loss baseline is around 0, two additional models are trained whose input of maximum logits entropy deviate far from the origin point (FC2 and CNN2). Although the minimum loss distributions are not perfectly concentrated at the maximum entropy values, significant following offsets can be observed. Additionally, we find that the reason why zero is feasible as a baseline alternative is that the global maximum of the entropy curve for a portion of neural networks lies approximately adjacent to the origin point. Nevertheless, the opposite also exists, for instance, the FC2 and CNN2 in our experiments. FC2 is derived from performing an identical linear transformation on the dataset and the bias of the first layer of the network for FC (the experiment is first proposed by [105] and is described in detail in the next section), while CNN2 is a convolutional neural network with more sophisticated architectures. The maximum values of their entropy curves deviate from the origin, at which point the zero baseline can no longer be considered as an approximation, while

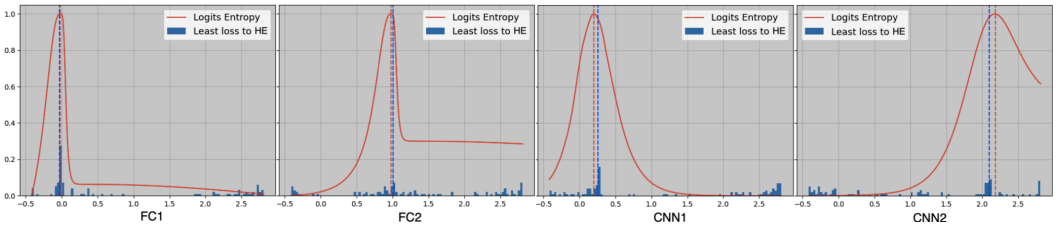

Figure 2.4.: The entropy curves of the logits and density histograms of baselines which achieve the minimum Spearman loss with the hybrid explanations (HE). The red and blue dashed lines indicate the maximum values of the logits entropies and density boxes, respectively.

the baseline closest to the hybrid explanations can be observed to be shifting towards the positive direction of the X-axis, rather than remaining at the origin.

2.3.3. Unreliable? Linear shifts on baselines

Previous study point out that a fraction of the IG baselines fail the linear transformation test [105]. They shift the input instances and the parameters from the first layer of the model with the same linear offset, and subsequently observe whether the explanations generated by the explainer are consistent before and after the transformation. In the experiment, two alternative shifts are adopted as offsets, i.e., the uniform and non-uniform shifts, where the former is equally shifted at each pixel, while the latter is not subject to this restriction. They utilize black and zero baselines for IG, while only the former maintains the same explanation before and after the uniform shift. We reproduce the experiments and present them in figure 2.5 (the shift amplitudes, including all uniform shifts and the non-zero pixels of the cross-shaped shifts, are 0.5). We argue that the black baseline is adaptive and shifts with the input, whereas the zero baseline remains constant, which lacks fairness. Therefore, we transform the zero baseline with the same offset and observe the explanation invariance. The results are shown in the first row where the zero baseline maintains the consistent explanation after the transformation. The second row exhibits the results of the non-uniform shift, and interestingly, when we perform non-uniform shift for the zero baseline, the explanation is irreducible.

This can be explained by the entropy curves. Figure 2.6 plots the curves of the three models in parts of the valid value range. Suppose the entropy curve of the original model is $Entr(x)$ (black curve), which turns into $Entr_U(x)$ after the uniform linear transformation (red curve), and the transformation amplitude is A_s. The correlation can be easily derived from the diagram:

$$Entr_U(x) = Entr(x - A_s) \tag{2.4}$$

Equation 2.4 indicates that the information content of the two models is identical except for a phase difference of A_s. However, the model with cross-shaped transformation

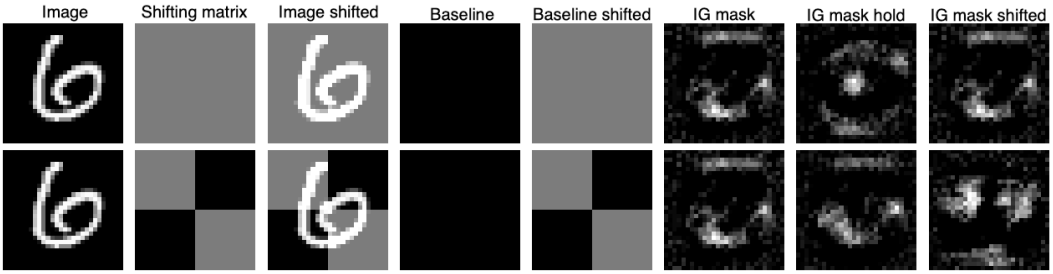

Figure 2.5.: Visualizations of the uniform and non-uniform linear transformations on the instances and baselines. The last three columns are the corresponding explanations. IG mask hold demotes the linear transformation is applied only to the instances, whose baselines remain non-transformed.

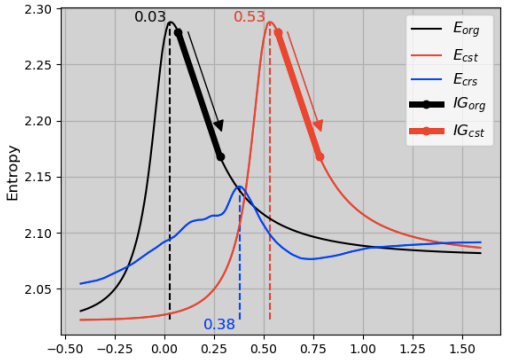

Figure 2.6.: Entropy curves of the model after different shifts, where org stands for no shift, cst stands for a uniform constant shift of the input vector, and crs stands for a cross-shaped shift of the input vector with the midline as the dividing point, IG_{org} and IG_{cst} denote the integral of information from high-entropy initiations to low-entropy destinations (the explanations).

possesses a severely distorted entropy curve (blue curve). When the original model is explained with IG, the gradient is accumulated from an "uninformative" initiation to the destination to be explained, and an identical path can be found in the uniformly transformed model, with the starting and ending points translated by A_s (e.g., bolded black and red segments), while the integrals are equivalent. In contrast, such a segment is absent in the curve of the model with the cross-shaped transformation. Consequently, *linear invariance is only adequate for transformations that do not distort the model entropy curve.*

2.4. Ablation-based evaluation methods

In this section, we elaborate on the flaw of the classical ablation-based evaluating metric for explainability methods, which monitors the prediction activations in 2.4.1, and propose a novel one, which targets the entropy as the surveillance 2.4.2.

2.4.1. Existing ablation test

The ablation approach and its variants are the most commonly used methods to evaluate explanations, which is comprehensively summarized in [152]. Let $x \in X$ denote an instance containing n features $x_o = (x_o^1, x_o^2, ..., x_o^n)$ (X is the set of all valid instances), the information quantity it contains before the ablation test is initiated as $I(x)$. After all n features are ablated, the input now is noted as $x_\phi = (\phi^1, \phi^2, ..., \phi^n)$, where ϕ_i denotes the ablated substitutions, and the information quantity it holds is marked as $I(x_\phi)$, the ablating evaluation can be formulated as:

$$S_e \propto I(X_o) - I(X_\phi^p) \tag{2.5}$$

where S_e indicates the score of the explainer and X_ϕ^p indicates the removal of p positively attributed features.

Several studies [35], [56], [113] employed ablation approaches to validate the reliability of explainability methods, while differing slightly in the details. The major distinctions are:

- *Surveillance target.* Let l, f represent the prediction label and the model to be explained. Frequently monitored objects are collectively defined as "prediction scores" and are divided into 1. the corresponding activations in logits: $a^l(x)$. 2. the corresponding cells in logits after taking Softmax, which can be considered as the probability of the class: $Softmax(a^l(x))$. In these works, they consider the reduction of prediction scores as a result of "information removal" [56], the surveillance object is regarded as a measurement of "information quantity", i.e., $I \approx \Delta a^l(x)$ or $\Delta Softmax(a^l(x))$.
- *Ablation destination.* The substitution value for ablation is also controversial. There are several competitors: zero, minimum value of the dataset, reversing the sign of the current pixel [35], blurring via Gaussian kernel [152] etc. The purpose of ablation is to conceal information about a specific pixel (or feature), the flipping destinations themselves should therefore carry no information, i.e., $I(x_\phi) \approx 0$.

However, we argue that such metrics suffer from non-conservation. According to the definition of the ablation test, the information quantity of the ablated input x_ϕ should satisfy:

$$\forall x \in X, I(x) \geqslant I(x_\phi) \tag{2.6}$$

Note that practically the information quantity is not directly available and the existing methods record the value of logits a^l (or the softmax of the logits $Softmax(a^l)$) as substitutes. Thus, Equation 2.6 can be rewritten as:

$$\forall x \in X, a^l(x) \geqslant a^l(x_\phi) \text{ or } Softmax(a^l(x)) \geqslant Softmax(a^l(x_\phi)) \tag{2.7}$$

Taking the CNN in Section 2.3.2 as an example, we employ a 1000-step gradient descent algorithm to minimize the value of the surveillance target and record the changing curve.

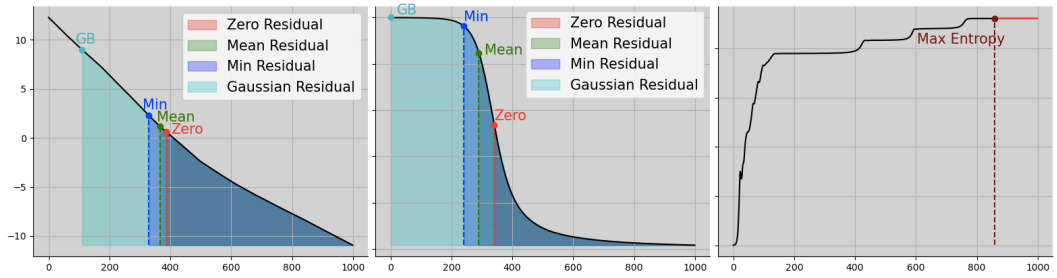

Figure 2.7.: The ablation experiments with the raw logits activations (left), softmax of logits activations (middle) and the entropy of logits as the surveillance target (right), respectively. The x-axis denotes the number of optimization steps and the y-axis represents the value of the corresponding monitored target. The shaded areas with colors represent "information residues".

Subsequently, we feed various x_ϕ into the model, derive the individual predicted values, and annotate them on the curve. Notably, the optimization process is performed in the valid range of the original dataset. According to figure 2.7, equation 2.7 is violated by the optimization results, and numerous inputs with lower monitoring target values can be identified in the valid range. If $a^l(x)$ is considered as the information quantity, x_ϕ suffers from extensive information redundancy (left plot). Compared to the former, $Softmax(a^l(x_\phi))$ mitigates this deficiency, while the information residual in x_ϕ is still visible (middle plot). Experiments show that the ablating test designed on the basis of inputs that are commonly understood by humans as "uninformative" may still contain information and may be problematic.

2.4.2. Entropy-based ablation test

To fulfill the conservativeness in Equation 2.6, we extend the definition in equation 2.2, that monitors the entropy of logits as the quantitative indicator of information, i.e.

$$I(x) \propto\!\!\!\!\sim \frac{1}{Entr(Softmax(l))} \tag{2.8}$$

The increment of information diminishes the uncertainty of events, which can be expressed by the entropy. The introduction of entropy facilitates the understanding of "information quantity". On the other hand, by redefining the unique ablated destination as $x_\phi = B_{Xentr}$, the conservativeness of the ablation assessment is assured

$$I(B_{Xentr}) \approx \frac{1}{Entr(Softmax(a^l(B_{Xentr})))} \tag{2.9}$$

Recalling equation 2.2, which yields

$$\forall x \in X, Entr(Softmax(a^l(x))) \leq Entr(Softmax(a^l(B_{Xentr}))) \tag{2.10}$$

and refers to equation 2.8, equation 2.6 holds. To experimentally verify the conservativeness, we depict the entropy curve and mark out B_{Xentr}, as shown in Figure 2.7 (right plot). Note that in this plot, the area **above** the entropy curve represents the information residual, which does not exist since B_{Xentr} is the input that maximizes the entropy.

2.5. Quantitative evaluations

We choose two artificial and one real-world datasets for evaluation experiments: MNIST, CIFAR10 and Stanford Car Dataset. For MNIST, we train a fully connected network, which achieves 98.2% accuracy on the test set. For CIFAR10, we train a ResNet18 [48] network, whose test accuracy is 95.6%. For Stanford Car Dataset, we trained a ResNet152 with 92.2% accuracy. During evaluation, for MNIST we assess all $10,000$ test data, while on the remaining two we select $1,000$ examples from the dataset for evaluation.

As illustrated in Fig. 2.8, $Xentr_u$ and $Xentr$ are on par with or superior to other baselines in the same categories (i.e. $Xentr_u$ and $Xentr$ rank first, second and second amongst the Uniform and Non-Uniform baselines, respectively). Interestingly, the Max Distance baseline consistently performs worse than the remaining ones in the non-uniform baseline. There is extensive information about the input instances in the Max Distance baseline, including object boundaries, gray value information (see Figure 2.1), which validates our view that the more information about current instances is contained in the baseline, the less is integrated by IG and thus the generated explanation is weaker in terms of credibility.

Morever, to exhibit the superiority of the improved maximum entropy ablation test, we compare the other ablation methods as well. We choose FC2 in Sec. 2.3.2 as the model for the reason that the maximum entropy baseline of FC2 deviates from the origin, enabling the results to be obvious at a glance. To exhibit the plausibility of the maximum entropy ablation, four different ablation tests are performed and shown in Fig. 2.9. The ablation tests with zero and the maximum value of the dataset as the destinations fail to reasonably evaluate the performance of explanations (randomly generated explanations outperform the IG with any baselines, which is counterintuitive). The reason that the minimum ablation also exhibits reasonable results is that, for FC2, the minimum value (1) is exactly approaching the maximum entropy baseline (see figure 2.4). Note that the evaluation experiments in this section are not absolutely precise. There are still numerous unaddressed issues in the ablation study, for example, perturbing a pixel corrupts the correlation between neighboring pixels. A potential future work is to enable ablation studies to take into account the maximum entropy theory and pixel-wise correlations simultaneously.

More importantly, such an "uninformative" baseline is not only desired in IG. Explainability methods such as KernelSHAP [65], Occlusion [34] and RISE [95] have all introduced the concept of information absence. Taking KernelSHAP as an example, the model substitutes part of the features with the baseline when sampling around the instance to be

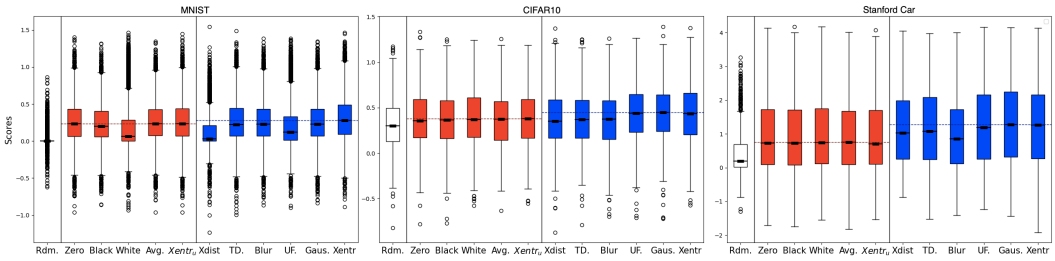

Figure 2.8.: Ablation tests on different IG baselines on MNIST, CIFAR10 and Stanford Car datasets. Bars from left to right: randomly generated saliency maps (comparison reference), zero, black, white, average of current input, max entropy with uniform values, max distance, average of training data, blurred, uniform, Gaussian noise and max entropy baselines, respectively. The red boxes represent baselines with uniform values on all pixels, while the blue boxes are free of this restriction. The bolded black bar in the box is the median, and the horizontal dashed line indicates the optimal median value of the baseline in the current category (uniform or non-uniform).

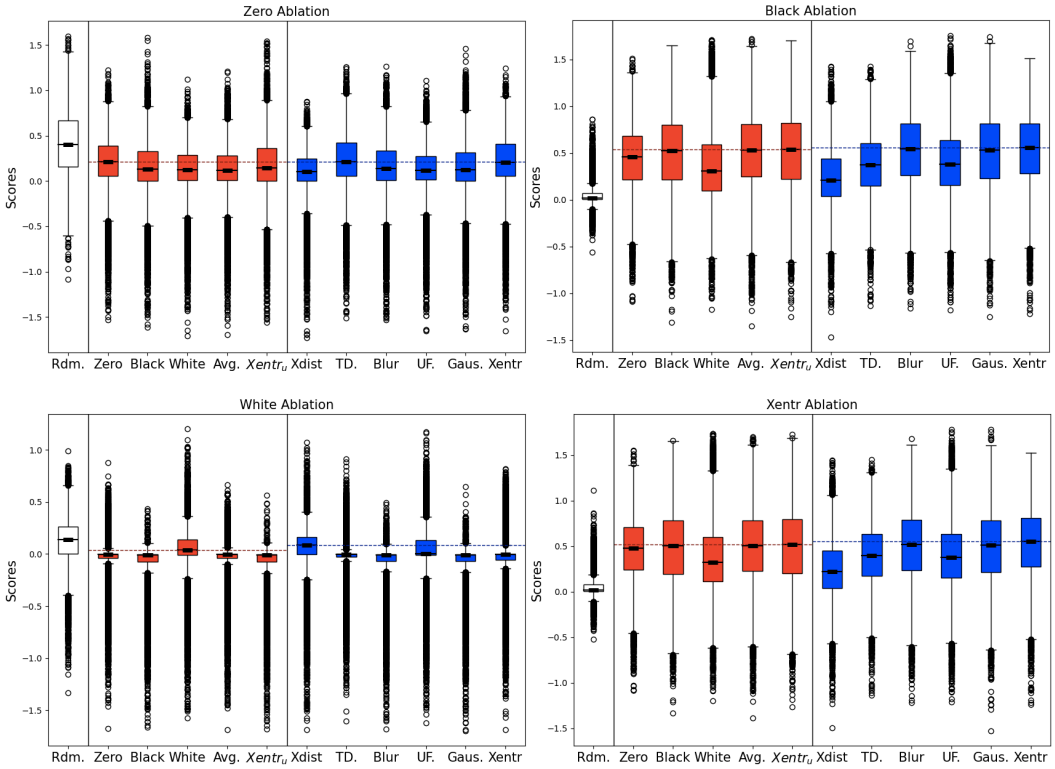

Figure 2.9.: Four different ablation tests. The ablation destinations are: zero (top left), the minimum (top right) and maximum (bottom left) values of the data set and the maximum entropy baseline (bottom right), respectively.

Scores	Random	Uniform Baselines					Non-uniform Baselines					
	Rdm.	Zero	Black	White	Avg.	X_{entr_u}	Xdist	TD.	Blur	UF.	Gaus.	X_{entr}
$\bar{S}$	0.157	0.582	0.571	0.208	0.585	0.581	0.318	0.570	0.534	0.408	0.163	**0.593**
$\tilde{S}$	0.059	0.598	0.582	0.142	0.599	0.594	0.298	0.582	0.544	0.402	0.054	**0.608**
$\sigma^2(S)$	0.044	0.074	0.079	0.049	0.076	0.075	0.059	0.073	0.070	0.068	0.054	0.075

Table 2.1.: Evaluation experiments of KernelSHAP with various baselines.

explained. Therefore, employing perturbation baselines with residual information may impair the performance of the surrogate model and thus diminish the credibility of the explanations (see Table 2.1 for the evaluation tests). Therefore, a well-accepted and effective baseline is pivotal not only for IG, but also for broader explainability research.

2.6. Conclusion

This chapter identifies the conservation deficiencies of the existing IG baselines and ablation tests from an informational perspective, and proposes a new baseline and an enhanced ablating evaluation method based on the "missingness" necessitated by the explainability approaches. However, we acknowledge that existing ablation tests are still controversial in terms of, for example, feature correlation. Therefore, in order to mitigate the aforementioned challenge, explanations should be assessed from as many different perspectives as possible with respect to various properties. In the next chapter, we will extend the sensitivity test to the parameter level and evaluate the robustness of explanations to parameter pruning.

3. Evaluating Explanation Robustness to Model Pruning

The baseline selection in the previous chapter is involved in perturbation-based ablation tests, where the basic idea is to assess whether features with high attribution in the explanation actually contribute to the model predictions. However, this merely reflects the fidelity of the explanations to the model predictions, which is not a sufficient requirement for a plausible explanation. In this chapter, addressing the challenge of the lack of parameter robustness in Sec. 1.2.2, we extend the idea of ablation test to the parameter level of the model. We examine the plausibility of explainability approaches from a novel perspective, i.e., the robustness to model pruning. We show that even when only those neurons with the lowest importance scores are eliminated and there are no noticeable fluctuations in the prediction performance, the explanations are dramatically corrupted. Extensive experiments qualitatively and quantitatively illustrate that most of the popular explainability methods are insufficiently robust to the simplest model pruning algorithms.

3.1. Introduction

As mentioned in Sec. 1.2.2, a limitation of existing sensitivity tests is that it only focuses on the impact of input features on predictions, while neglecting that model parameters are also key components involved in the decision-making process. In this chapter, we extend the sensitivity test to the parameter level of the model and analyze the impact of parameter removal on prediction results. Specifically, we incorporate a novel perspective to assess the reliability of explanations, namely model pruning. We observe that even though the pruned model achieves the same or a negligibly lower accuracy, the generated saliency maps are remarkably altered. Note that unlike the Rashomon effect mentioned in [182], [192], pruning (without fine-tuning) does not modify the majority of the model parameters, which implies that simply removing those inactive neurons may also cause noticeable interference to the generated explanations. Based on the observation, the reliability and robustness of explainability methods may confront new challenges.

In summary, our contributions are as follows:

- We investigate the reliability of explanations from a novel perspective by comparing explanations generated from the original model and the pruned ones. To the

best of our knowledge, this is the first work to investigate the impact of model pruning algorithms on explainability approaches.

- Extensive experiments have shown that even if the pruned models achieve slightly worse or identical performance, there are observable distinctions in their explanations.

3.2. Related Work

Reliability of explanations: Several works have identified deficiencies in the reliability and robustness of existing explainability methods. [84] argues that the explanations need to be relevant to the model parameters. By randomizing (part of) the network parameters, they observe that several explainability methods still maintain the same quality and are considered as failures of the sanity checks. [105] designed a linear transformation that applied equivalently to the inputs and the first layer of the model, which cause no effect on the predictions. However, they found that several explainability methods are interfered with and their saliency maps are distinctively altered. In addition, [86] argues that the robustness of the explanations should be reflected in that adjacent inputs result in analogous saliency maps, but again there are parts of the explainability approaches that violate this rule. The Rashomon effect is a recently introduced topic of explanation uncertainty, which demonstrates that explanations may vary dramatically even for models with the equivalent accuracy on the same dataset [182], [192]. The essential difference between our work and theirs is that instead of training new models with comparable accuracy, we prune the original model itself. Non-fine-tuning pruning leaves the vast majority of the parameters unaltered, and therefore the conclusion can be made more convincing by excluding the variations in explanations due to parameter discrepancies.

Model pruning: Model pruning was first proposed by [17], with the idea of removing neurons that contribute the least to the prediction for the purpose of lightweighting the network. Subsequently, [19] suggested that the product of Hessian's diagonal and the square of the weights can be regarded as a valid indicator of the neuron importance. A recent study [111] proposes to approximate the cost function with first-order Taylor expansions to derive an alternative estimate of importance, i.e., the squared product of the weights and the gradient, which maintains the model performance optimally while pruning out the same proportion of parameters.

3.3. Method

The purpose of this chapter is to observe the robustness of the explanations generated by models with approximate performances. Existing research [192] has shown that there may be discrepancies in the explanations from models trained in the same dataset with an equal accuracy, but it also recognizes the possibility that predictive multiplicity exists, where the model achieves a comparable performance through a diverse set of parameters.

To minimize this impact, we rely on model pruning technique to preserve most of the parameters invariant and observe the explanation stability.

The idea of our experiment is as follows: We train a neural network F with the training set X_{tr}. Afterwards, we input the i^{th} data from the test set $x^i_{te} \in X_{te}$ into F and compute the prediction y_i by a forward process and obtain the explanation E^{y_i} with the explainability method to be evaluated. We then prune the model according to the parameter importance provided by chosen pruning algorithms, denoted model F', and input x^i again, and employ to generate the new explanation E'^{y_i}. We adjust the hyperparameters of pruning algorithms to guarantee that the performance of F is almost identical to that of F', with the vast majority of the parameters remaining untouched. Empirically, their explanations for the same input should be highly analogous, which is in line with the intuition of explanation robustness, i.e. $E^{y_i} \cong E'^{y_i}$. To compare the proximity of explanations, we employ the following metrics:

- **Spearman's rank correlation coefficient.** There may be orders of magnitude differences in the explanation values generated by different models and explainability methods. Thus, the correlation in ranking is more reflective of the relevance between explanations than the magnitude. We chose Spearman's rank correlation coefficient to measure the ranking difference of each pixel across explanations to estimate their proximity, which is formulated as:

$$\rho_S = \frac{COV(R(E), R(E'))}{\sigma_{R(E)}\sigma_{R(E')}} \tag{3.1}$$

  where $R(*)$ is the rank function, COV and σ are the covariance and standard deviation, respectively.

- **Sign agreement**. In explanation, there are pixels with positive and negative attributions, which denote supporters and opponents of the current prediction label, respectively. Pixels with the same symbol should dominate in two analogous explanations, and vice versa. Therefore, we select the percentage of sign-agreed pixels as another similarity measurement, which is formulated as:

$$SA = \sum_{i=0}^{N_E} \frac{|\{Sign(e_i) = Sign(e'_i)\}|}{N_E} \tag{3.2}$$

  where $Sign$ is the sign of the pixel, $e \in E$ is the pixel on the explanation E and N_E is the nunber of pixels in E. Note that Sign has three types of output, positive, negative and zero.

- **Top-k agreement**. In explanation, those pixels or regions with the highest attribution are typically given more attention, e.g., the area of the target object in the image, etc. For two explanations, we consider them to be analogous as long as their attributions (of rankings) in the object region approach each other. Therefore, we consider Top-k agreement as another similarity measure, which evaluates

whether the indexes of the k pixels with the highest attribution agree between two explanations. Top-k agreement is formulated as:

$$TA = \frac{|TopK(E) \cap TopK(E')|}{|TopK(E)|} \tag{3.3}$$

Additionally, we assess the consistency of the predictions between the pruned model and the original one with three metrics: Total accuracy, average prediction loss and prediction agreement. Total accuracy and average loss are the accuracy and average prediction loss of the model on the test set, respectively, and a numerically proximation indicates that the two models are globally comparable in performance. Prediction agreement can be formulated as

$$PA = \sum_{i=0}^{N_d} \frac{|\{y_i = {y'}_i\}|}{N_d} \tag{3.4}$$

where y_i and y'_i are the outputs of the i^{th} data predicted by the original model F and the pruned one F', and N_d is the total number of data. Prediction agreement reflects the similarity of the local performance of models and mitigates the Rashomon effect that occurs when two models with comparable accuracies make mistakes on different test data.

In our experiments, we first generate the explanations for the test set E with the original model, and then iteratively prune the model F by percentage, yielding F'_1,...,F'_k, until there is a significant drop in the performance. We subsequently predict the test set with F and F'_k and generate corresponding explanations E and E'_m by utilizing the selected explainability methods, and finally compare the similarities between E and each E'_m.

3.4. Experiments

Our experiments are conducted on two different models and datasets, including a simply-constructed, ModelNetCNN consisting of only convolutional and fully-connected layers, and ResNet18 , which are well-trained on the MNIST handwritten dataset and CIFAR-10, with accuracies of 98.59% and 93.2%, respectively.

We employ two various pruning methods, Optimal Brain Damage (OBD) [19] and First-Order Taylor expansion (FOTaylor) [111], respectively. OBD is a pruning method according to the neuron magnitude, which is based on the assumption that the model converges with its second-order gradient (Hessian) being a positive definite matrix, and that removing any neuron results in a rising loss. In OBD, the importance of the i^{th} neuron can be formulated as:

$$Imp_i^{OBD} = \frac{h_{ii} m_i^2}{2} \tag{3.5}$$

where h_{ii} is the diagonal Hessian matrix, i.e., the second-order derivative of the neuron itself, and m_i is its magnitude (weight). FOTaylor is a neuron gradient-based pruning method, which calculates the first-order (or second-order) Taylor expansion of the error function to identify neurons that, even if removed, would not cause a significant impact on the prediction. FOTaylor utilizes the following formula to calculate the neuron importance:

$$Imp_i^{FOTaylor} = \left(g_i m_i - \frac{1}{2} m_i h_i M \right)^2 \tag{3.6}$$

where g_i, m_i, h_i and M are the first-order gradient, the magnitude of the neuron, the i^{th} row of the Hessian matrix and the magnitude of all neuron parameters, respectively.

We choose both gradient and perturbation-based explainability methods for explaining the models and estimating the proximity of the generated attributions. The selected methods include 1) gradient-based methods: Vanilla Gradient (VB) [28], Integrated Gradients (IG) [78], Layer-wise Relevance Propagation (LRP) [35], GradCAM [75] and 2) perturbation-based methods: LIME [54] and KernelShap (KShap) [65]. For evaluation of the explnation similarity, when calculating Top-k agreement, we focus on the top 10% of pixels with the largest attributions.

Visualizing the explanations of pruned models. We demonstrate in Fig. 3.1 an instance of one explanation on the CIFAR-10 dataset that is correctly categorized as "cats" by ResNet18 model with a confidence level of 99.96%. After pruning 40% of the parameters, the model is still able to correctly categorize the input image with only a 0.02% decrease in confidence. However, it can be seen that the model that has been pruned is significantly different from the original one in terms of the generated explanations. The vast majority of gradient-based explainability methods, including VB, IG and GradCAM, exhibit a relative robustness, with the critical points almost consistently converged on specific areas. A closer examination, nonetheless, reveals that the point with the largest attribution (the brightest pixels) alternately shifts back and forth as the model is pruned. Such inconsistencies can be more easily observed in subsequent quantitative comparisons. The remaining explainability methods, i.e., LRP, LIME, and KShap, show stochastic variation in explanation during the pruning process, which implies that they are barely robust to pruning.

Quantitative experiments on MNIST. For this low-difficulty dataset, we design a model with a simple structure called ModelNetCNN, which consists of only convolutional and fully-connected layers, and and is roughly structured as Conv1-MP2d-DR1-Conv2-MP2d-DR2-FC1-DR3-FC2, where MP2d and DR are 2D max-pooling and dropout layers. ModelNetCNN contains 147146 parameters and achieves an accuracy of 98.59% on the MNIST test set before pruning. We perform 5 incremental prunings on ModelNetCNN, eliminating 10% of the parameters each time. We found in our experiments that retaining 50% of the parameters still enables ModelNetCNN to maintain accuracy, loss, and prediction agreement at the same level.

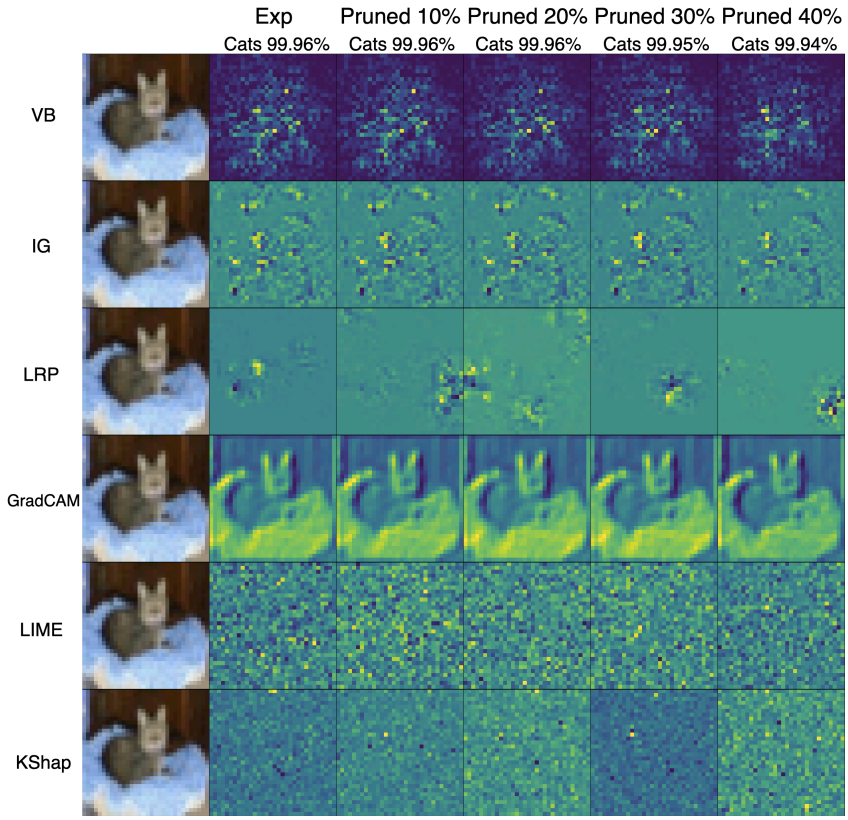

Figure 3.1.: Visualization of the explanations generated by the original model as well as the pruned models. From left to right are the input images, the explanation generated by the original model, and the explanations generated by the models with 10%, 20%, 30%, and 40% of the parameters pruned, respectively. The brighter the pixel the greater the attribution it possesses.

To begin with, we test OBD [19] pruning method and present the experimental results in Fig. 3.2 (solid lines). The three metrics in the first row represent the comparison of model performance before and after pruning. It can be seen that the pruned model is almost on par with the original one in terms of accuracy and loss. Meanwhile, the models before and after pruning also maintain an extremely high level of prediction agreement (98.88% when 50% parameters pruned), which nearly rules out the possibility of the "Rashomon effect" on prediction accuracies. As a conclusion, the pruned model can be regarded as an equivalent substitute for the original one.

Nevertheless, the differences in the explanations generated for (pruned) models with almost identical performance are significant. In terms of Spearman coefficients, almost all explainability methods exhibit a linear decline, with LIME and KernelSHAP dropping off the bottom as the pruning begins, indicating that the explanations generated before and after the pruning are barely related in attribution ranking. Additionally, sign and Top-k agreement demonstrate similar tendencies. Note that according to Equation 3.2, the baseline for sign agreement is 50%, which denotes a complete random guess for the sign. The last and most important metric is Top-k agreement, which determines whether the regions with the highest attribution (most important) are consistent across explanations. As a conclusion, given that the models are pruned 50% and the performance is approaching, most of the explanations only ensure that about half of the critical pixels are clustered in the same area, while part of the explainability methods such as LIME and KShap almost lose the consistency of attribution in the important areas (the baseline is 10%). In summary, in terms of explanation similarity before and after pruning, all selected explainability methods exhibit significant degradation during the pruning process, with the gradient-based explainability methods being more stable compared to the perturbation-based ones. Subsequently, we switch to another pruning method, FOTaylor, to investigate whether this observation is caused by a specific pruning algorithm. The results are also presented in Fig. 3.2 (dashed lines). The conclusions regarding the robustness of the explanations to pruning are almost identical to those of the OBD: as the parameters being pruned progressively increase, the difference between the generated explanation and the original one grows rapidly. In this case, the gradient-based explainability approaches collapses at a rate comparable to OBD, which is linearly correlated with the total number of pruning parameters, whereas the explanations generated by the perturbation-based approaches share little resemblance with the original ones right at the beginning of the pruning. Interestingly, for the FOTaylor pruning algorithm, VB achieves a dominating robustness in terms of sign consistency, maintaining almost the same sign agreement as the original explanations up to the point where 40% of the parameters are pruned.

Overall, the experimental results are counterintuitive. In the absence of significant variations in model performance, one possibility for the discrepancies in explanations can be explained as "Rashomon effect", where models make similar predictions based on diverse input features. Nonetheless, this does not applicable to models that are (non-fine-tuned) pruned, whose critical neurons are identical to the original model, i.e., the crucial weights for learning the features are not altered, and thus it can be approximately assumed that the critical features they learn from the images are identical. The lack of robustness in

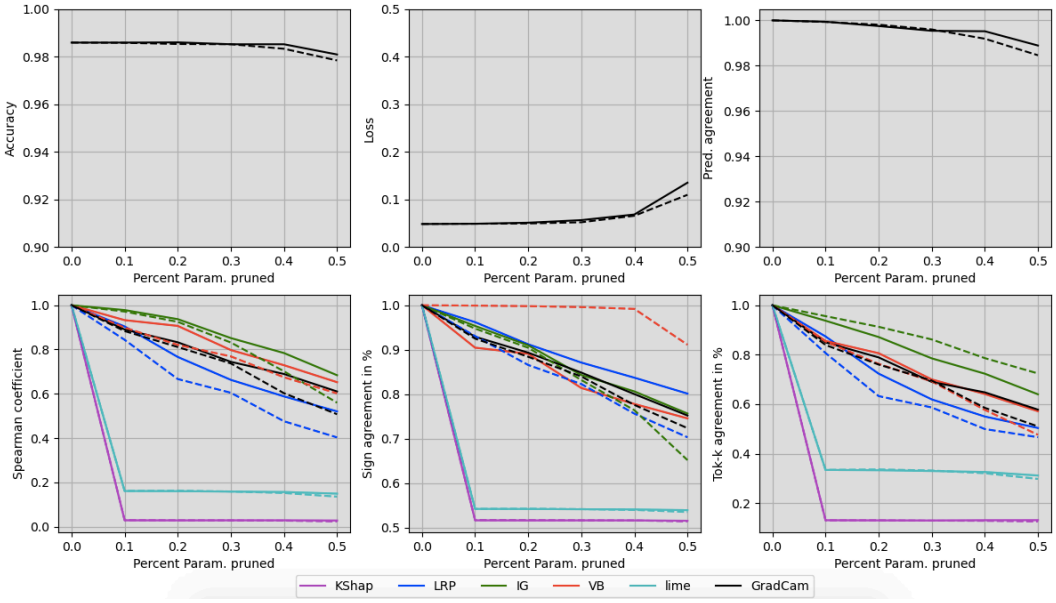

Figure 3.2.: Robustness evaluation of different explainability methods during pruning process. The result is from ModelNetCNN on the MNIST handwritten dataset and The pruning methods are OBD (solid lines) and FOTaylor (dashed lines). The first row presents the performance of the corresponding pruned models compared to the original one, respectively, from left to right: accuracy, loss and prediction agreement. The second row provides the evaluations of the explanation discrepancy, from left to right: Spearman's coefficient, sign agreement and Top-k agreement.

the explanations of pruned models implies that the explainability approaches may not be responsive to the critical features learned from the input images.

Quantitative experiments on CIFAR-10. We further test the explanation robustness of the pruning models on a more complex dataset, CIFAR-10. We train a ResNet18, which consists of more complex network structures and achieves an accuracy of 93.2% on the test set. We iteratively perform 4 round of prunings, each eliminating 10% of the total parameters. Compared to ModelNetCNN, ResNet18 contains additional special structures such as BN layers, and in pruning, we only eliminate the weights on the convolutional and fully-connected layers, which also account for the vast majority of the total number of model parameters.

As the results shown in Fig. 3.3, eliminating 40% of the weights causes no significant performance collapse for ResNet18 and the growths in losses are also nearly negligible. Furthermore, the prediction labels of the pruned model and the original one are highly consistent. However, the conclusions about the robustness of pruned explanations still hold, i.e., The discrepancy between the explanations generated by the pruned models and the original one is noticeable, on the precondition that the performance variation of these models is almost negligible. Among them, gradient-based methods, such as VB, IG and GradCAM, remain more robust in general than perturbation-based methods.

We find that the explanation robustness of a few methods is improved as compared to the MNIST dataset, such as IG and GradCAM. The perturbation-based methods LIME and KShap, on the other hand, perform stably, their explanations being completely different from the original model at the very beginning of the pruning, with almost no consistency. LRP joins this category as well, although in the MNIST dataset it still exhibits explanation proximity that are negatively linear-correlated with the total number of pruned parameters, on ResNet18 the explanation collapses entirely at the very beginning of the pruning. Overall, the experiments on CIFAR-10 lead to conclusions consistent with MNIST, i.e., even though the performance of the pruned models is almost identical to the original one, the explanations they generate are significantly varied.

Explanations for fine-tuning pruning. In addition to the Non-fine-tuning pruning methods, we further investigated the interference that fine-tuning pruning methods cause to the explanations. Fine-tuning pruning methods, after summarizing the attribution and pruning the neurons based on a certain amount of data, continue to train the remaining neurons according to the gradients from back-propagation, which leads to a certain degree of weight alteration on the remaining neurons as compared to non-fine-tuning methods. We still choose OBD and FOTaylor as the methods for computing neuron attribution. In contrast, pruning is conducted during training: the data is fed into the model as training batches. We consider 10 training batches as a minibatch, and after a minibatch has been trained, the global average attributions of each neurons are calculated for model pruning. The global average attribution is derived by Exponential Moving Average, which can be formulated as:

$$\overline{Imp_m} = \beta \times \overline{Imp_{m-1}} + (1 - \beta) \times Imp_m \tag{3.7}$$

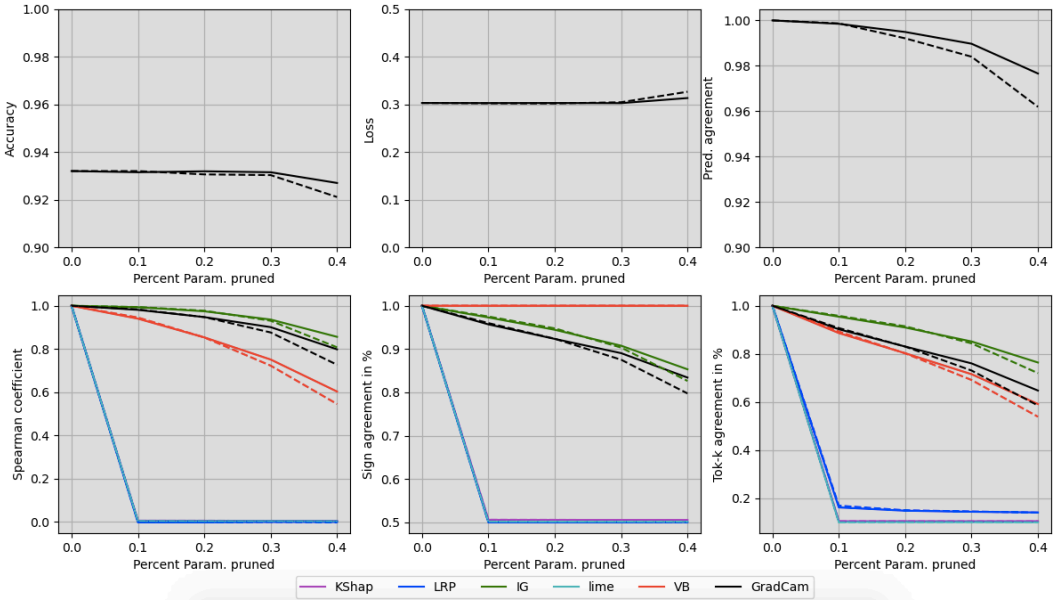

Figure 3.3.: Comparisons of explanation similarity of ResNet18 trained on CIFAR-10. The models are pruned with OBD (solid lines) and FOTaylor (dashed lines).

where Imp_{m-1} is the global average attribution of the first $m-1$ observable minibatches, β is an adjustable weighting factor, and Imp_m is the average attribution of the m^{th} minibatch. In our experiments, we prune off 10% of the model parameters when the entire dataset is trained once, i.e., $10\% \times \frac{10}{N_B}$ of the parameters are pruned for each minibatch, where N_B is the total number of the training batches.

As demonstrated in the results in Fig. 3.4, we observe that OBD-based fine-tuning pruning dramatically diminishes the stability of the explanations. Explainability methods such as VB and IG, which have shown relative consistency in non-fine-tuning pruning, are enormously more sensitive to fine-tuning pruning. Overall, VB is almost completely irrelevant to the explanation of the original model after pruning 10% of the parameters, while IG even shows a negative correlation. Interestingly, the Top-k agreement of VB and IG remains moderate, which indicates that fine-tuning pruning has less impact on the pixels with the largest attributions in the explanations compared to the rest of the pixels. In addition, we note that GradCAM is much more sensitive to fine-tuning pruning in terms of sign agreement than Spearman's correlation, which we believe is due to the sign flipping of the attributions for most insignificant pixels. For FOTaylor, parts of the explainability methods such as VB and IG instead slightly increase the robustness in comparison to non-fine-tuned pruning. Nevertheless, for the rest of the explainability methods, a noticeable decrease in the stability of the explanations can still be observed.

We perform the identical experiment on ResNet18 trained on CIFAR-10 and present the results in Fig. 3.5. It can be seen that on the more complex model ResNet18, for most of the gradient-based explainability methods (e.g., VB, IG, and GradCAM), despite that

			$\rho_S < 0.8$						$SA < 0.9$						$TA < 0.8$					
			VB	IG	LRP	GC	LM	KS	VB	IG	LRP	GC	LM	KS	VB	IG	LRP	GC	LM	KS
w.o. Fine-tuning	MC	OBD	0.4	0.4	0.2	0.3	**0.1**	**0.1**	0.3	0.3	0.3	0.3	**0.1**	**0.1**	0.3	0.3	0.2	0.2	**0.1**	**0.1**
		FOT	0.3	0.4	0.2	0.3	**0.1**	**0.1**	0.5	0.3	0.2	0.2	**0.1**	**0.1**	0.2	0.4	0.2	0.2	**0.1**	**0.1**
	RN18	OBD	0.3	/	**0.1**	/	**0.1**	**0.1**	/	0.4	**0.1**	0.3	**0.1**	**0.1**	0.3	0.4	**0.1**	0.3	**0.1**	**0.1**
		FOT	0.3	/	**0.1**	0.4	**0.1**	**0.1**	/	0.4	**0.1**	0.3	**0.1**	**0.1**	0.3	0.4	**0.1**	0.3	**0.1**	**0.1**
w. Fine-tuning	MC	OBD	**0.1**	**0.1**	**0.1**	**0.1**	**0.1**	**0.1**	**0.1**	**0.1**	**0.1**	**0.1**	**0.1**	**0.1**	**0.1**	**0.1**	**0.1**	**0.1**	**0.1**	**0.1**
		FOT	0.4	/	**0.1**	**0.1**	**0.1**	**0.1**	/	0.4	**0.1**	**0.1**	**0.1**	**0.1**	0.3	/	**0.1**	**0.1**	**0.1**	**0.1**
	RN18	OBD	0.4	/	**0.1**	/	**0.1**	**0.1**	/	/	**0.1**	0.4	**0.1**	**0.1**	0.3	/	**0.1**	0.4	**0.1**	**0.1**
		FOT	0.4	/	**0.1**	/	**0.1**	**0.1**	/	/	**0.1**	0.4	**0.1**	**0.1**	0.3	/	**0.1**	0.3	**0.1**	**0.1**

Table 3.1.: A summary of the robustness of explainability methods to pruning. We set a threshold for Spearman's correlation coefficient, sign, and Top-k agreement below which a change is considered "significant". The threshold is 80% of the reasonable variation interval for the metric ($\rho < 0.8$, $SA < 0.9$ and $TA < 0.8$). We show in the table that how many proportions of the parameter are pruned when the change in explanation is considered significant. Those results that are particularly sensitive to pruning are bolded. In the table, GC, LM and KS denote GradCAM, LIME and KernelSHAP, respectively. MC and RN18 are ModelNetCNN and ResNet18, respectively.

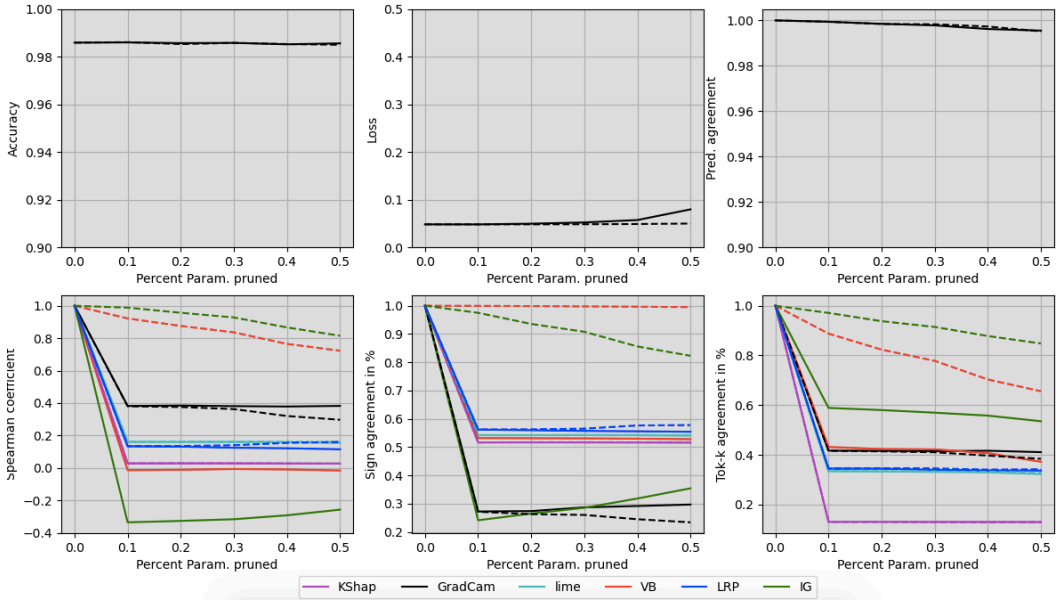

Figure 3.4.: Comparison of rubostness of explanations generated by fine-tuning pruning methods for ModelNetCNN on the MNIST handwritten dataset. The solid and dashed lines are the results of computing the attributions of neurons with OBD and FOTaylor, respectively.

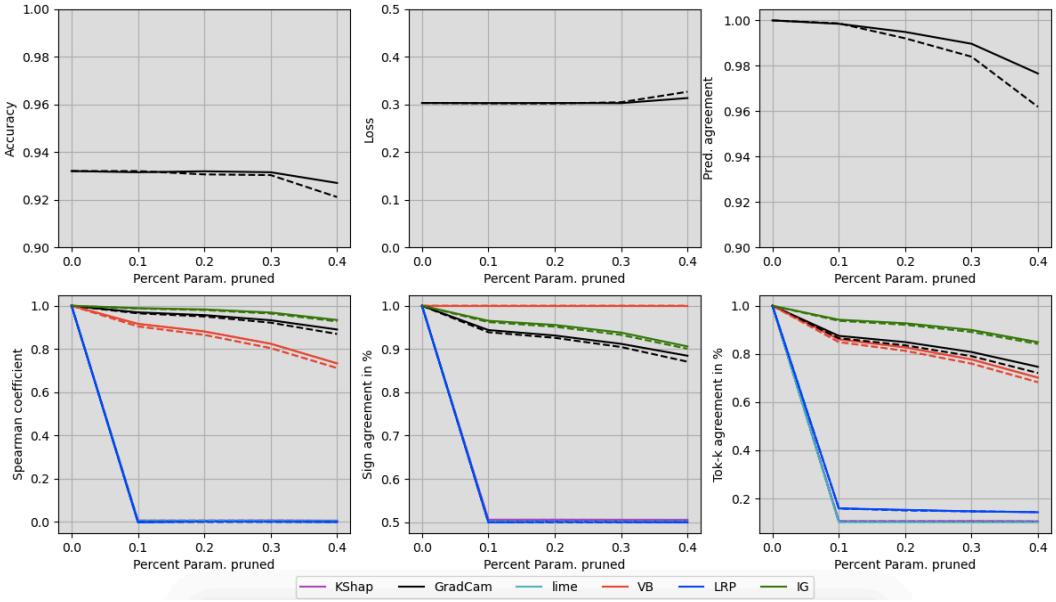

Figure 3.5.: Fine-tuning pruning on CIFAR-10 and assessing the robustness of the explanations. Again, solid and dashed lines represent the use of OBD and FOTaylor pruning methods, respectively.

the explanations generated after pruning are more similar to the original ones in terms of Spearman's correlation coefficients, the discrepancies in similarity are still considered noticeable as compared to the convergence in the prediction performance. Moreover, the perturbation-based explainability methods (LIME and KShap) and LRP fail to exhibit robustness regardless of which pruning method is applied: their similarities to the original explanations collapse entirely when 10% of the parameters are pruned. For sign agreement, VB performs consistently on this dataset, which is to be expected as the prediction performance of the model is likewise relatively stable. Notably, the robustness of the explanations on ResNet18 pruned by FOTaylor shows no collapse as it suffers on ModelNetCNN pruned by OBD. This may indicate a huge discrepancy in the pruning robustness of the explainability methods to different network structures. Further investigation on the effect of network architecture on the explanation robustness for pruning may be considered as one of the potential future works.

Tabulated summary. For intuition, we summarize the evaluation of the explanation's robustness in table 3.1.

3.5. Conclusion

In this chapter, we demonstrate that popular explainability methods suffer from a lack of robustness to model pruning. We perform various degrees of pruning on two models trained on different datasets with two pruning methods and explain the pruned mod-

els with popular explainability methods and then observed how the explanations vary according to the percentage of parameters pruned. The results indicate that eliminating only those neurons that are least important collapse the explanations generated by explainability approaches while having almost no impact on the predictions. This observation questions the fidelity of the explanations to the model parameters from a novel perspective. In the next chapter, we extend this fidelity evaluation to the explanations and model parameters as well as the predictions, proposing the concept named sensitivity consistency.

4. Evaluating Sensitivity Consistency of Explanations

In the previous chapter we evaluate the plausibility of explanations by observing their robustness to model parameter pruning. The essence of this observation is based on the idea that explanations should be faithful to the model parameters. In this chapter, we extend this fidelity to the level of the correlation between explanations, predictions and model parameters, to propose a broader evaluation concept, named sensitivity consistency. The intuition behind is that features and parameters that strongly impact the predictions and explanations should be highly consistent and vice versa. The evaluation of sensitivity consistency addresses the challenge in the ablation test mentioned in Sec. 1.2.2 where explanations may fail to cover all high-attribution features, thereby enabling the explanations to fulfill the key property of sufficiency. Extensive experiments on different datasets and models evaluate popular explainability methods while providing qualitative and quantitative results. Our approach further complements the existing evaluation systems and aims to facilitate the proposal of an acknowledged explanation evaluation methodology.

4.1. Introduction

In Sec. 1.2.2, we point out that existing sensitivity tests carry the risk of failing to cover all high attribution features. A feasible approach to this issue is to consider the sensitivity of explanations and predictions together, and analyze their consistency. Specifically, this chapter proposes a novel evaluation metric for explanations, called Sensitivity Consistency (SenC), based on the idea that predictions and explanations are expected to be sensitive to the identical input features. In addition, we extend this perspective to model parameters, assessing sensitivity consistency by observing whether those groups of neurons that play important roles in prediction would have similar impacts on explanations. Our contributions are mainly as follows:

- We propose a novel explanation evaluation metric SenC that assesses whether an explanation is reliable by comparing the discrepancy between the sensitivity of predictions and explanations to input features or neurons. SenC is a black-box approach that is applicable to all explainability methods.

- We quantitatively evaluate popular explainability methods on various datasets and models through extensive experiments, and verify the consistency of SenC with human intuition through a user study.

4.2. Related Work

Evaluation metrics for explanations. Due to the absence of ground truth, there is no acknowledged metrics. Sensitivity is one of the most intuitive metrics that assesses the fidelity of explanations by comparing the difference in confidence between the predictions after removing the most attributed feature in the explanation from the input (or insert it in an uninformative baseline) and the original predictions (baselines) [35], [56], [60], [86], [101], [105]. [103] argued that hard removals may disrupt the data distribution, resulting that the model is incapable of predicting effectively for data that has never been seen before. They propose RemOve And Retrain to mitigate the OOD problem by retraining after removing features. However, this in turn raises concerns about explanation fidelity to the model. Explanation robustness is another assessment perspective, where [86] argues that explanations given to similar inputs should also share a high degree of similarity. Another perspective that drew attention was the sensitivity to model parameters, as [84] found that the quality of the explanations from part of the methods is not seriously impaired when the model parameters are highly randomized. Besides, there are approaches that are not widely employed, such as Pointing Game [98], Generalizability [5], semantic-level perturbations and synthetic ground truth [191]. User assessment [79], [80], [128] is a convincing alternative, which is nonetheless costly and lacks reproducibility due to human subjectivity. In addition, a latest research [190] integrates explainability evaluation toolkits to facilitate assessment.

4.3. Methods

4.3.1. Sensitivity Consistency (SenC)

Existing studies indicate that different explainability methods may provide different explanations for identical models and inputs [84], [105]. In addition, recent research argue that there is a "Rashomon" of explanations, whereby explanations that reasonably demonstrate prediction attributions may not be unique [182], [192]. Due to the lack of ground truth, it is challenging to authenticate the credibility of inconsistent explanations. However, by considering inputs and explanations as a black-box system, we can assess the plausibility of explanations by observing the relationship between their variations. We rely on the argument that prediction confidences and explanations are expected to be sensitive to identical prediction bases, which is termed sensitivity consistency. The prediction bases are attributed to two factors, the input features and the model parameters, which are two aspects of the proposed assessment method. An overview of SenC is presented in Fig 4.1.

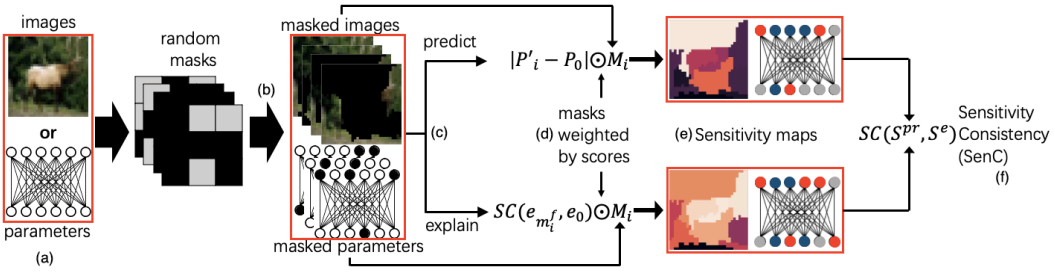

Figure 4.1.: An overview of SenC. SenC contains the following main components:(a) Selection of the input features or parameters to be perturbed. (b) Perturbation of the selected target by randomly generating an extensive number of masks. (c) Re-prediction and re-explanation with perturbed conponents (inputs or parameters) respectively. (d) Score each mask based on the difference in predictions and similarity in explanations. (e) Summing the product of all masks with their scores yields the prediction and explanation sensitivity maps, respectively. (f)SenC is derived by comparing the correlation of the two sensitive maps. Note that the red box in the figure indicates that SenC applies to either the input or the parameter, rather than in parallel with both.

4.3.2. Data Sensitivity Consistency

Data sensitivity consistency refers to the proximity of the degree to which the prediction confidence and explanation are impacted when part of the features in the input data vary. Elaborately, if a modification (removal or perturbation) of a feature significantly interferes with the prediction, while no serious explanation corruption occurs, the explanation is considered to lack sensitive consistency, and vice versa. To statistically measure the proximity of the impacts, we leverage a method based on random mask perturbations, which is inspired by [95]. We segment the input image into several partitions as input features with image segmentation algorithms to avoid the overwhelming computational intensity for processing pixel-wise features. Subsequently, we generate a massive amount of masks to randomly eliminate a fraction of the features. We predict the original and masked inputs to yield confidence scores P^o and P', respectively, and denote their differences as $\Delta P = |P^o - P'|$. With an enormous number of masks weighted by ΔP and summed, the final prediction sensitivity of the k^{th} feature is formulated as:

$$S_{f_k}^{pr} = \sum_{i=1}^{n} (1 - \Delta P_i) \odot M_i^f \tag{4.1}$$

where M_i^f and ΔP_i denote the i^{th} feature mask and its confidence discrepancy, respectively. The sensitivity of all m input features to the prediction can be summarized as $S_f^{pr} = \left\{ S_{f_1}^{pr}, ..., S_{f_m}^{pr} \right\}$.

Explanation sensitivity is the degree to which the input features impact the generated explanations. Similar to predictive sensitivity, we randomize an equivalent number of masks to eliminate input features and explain the masked inputs by a specific explainability method. We weight each mask by comparing the similarity of explanations from

the original and masked inputs, and eventually derive the feature sensitivity to explanations from the weighted sum. The explanation sensitivity of input features is $S_f^e = \left\{S_{f_1}^e, ..., S_{f_m}^e\right\}$, where the k^{th} feature is represented as:

$$S_{f_k}^e = \sum_{i=1}^{n} \rho_S(e^o, e_{m_i^f}) \odot M_i^f \tag{4.2}$$

where e^o, $e_{m_i^f}$ denote the explanations from the original and masked inputs, respectively. ρ_S denotes Spearman's correlation coefficient, which is formulated as:

$$\rho_S(e, e') = \frac{COV(R(e), R(e'))}{\sigma_{R(e)}\sigma_{R(e')}} \tag{4.3}$$

where $R(*)$, COV and ρ_S are the rank function, the covariance and the standard deviation, respectively.

Finally, by comparing the proximity of the prediction and explanation sensitivities, we derive the feature sensitivity consistency:

$$\rho_S^f = \rho_S(S_f^e, S_f^{pr}) \tag{4.4}$$

The detailed algorithm for data SenC is shown in Algorithm 1. ρ_S^f is essentially a Spearman's correlation coefficient, hence its domain of values is $[-1, 1]$. However, our experiments are statistical and the probability of presenting opposite correlations can be ignored. Therefore, the statistical mean of SenC has a value domain of [0,1], where 0 represents the absence of sensitivity consistency and 1 represents absolute consistency.

Algorithm 1: Data Sensitivity Consistency (SenC) for a given data

Input : An input data x, a well-trained model $F(\cdot)$, an explainability method $H(F, x)$ for the model F and the number of masks K

Output: Data SenC $\rho_S^{f_x}$ of H for input x

$P_x^o = F(x)$ # Original prediction

$e_x^o = H(F(x), x)$ # Original explanation

$S_{f_x}^{pr}, S_{f_x}^e = zeros\_like(x)$ # Initialization

for $k = 1$ to K do: #Generating masks

  $M_k^f = random\_like(x)$

  $x'_k = x \odot M_k^f$ #The k^{th} perturbation

  $S_{f_x}^{pr_k}$ += $(1 - |P_x^o - F(x'_k)|) \odot M_k^f$ #Scored by prediction variation

  $S_{f_x}^{e_k}$ += $\rho_S(e_x^o, H(F, x'_k) \odot M_k^f$ #Scored by explanation similarity

end for

$\rho_S^{f_x} = \sum_{k=1}^{K} \rho_S(S_{f_x}^{e_k}, S_{f_x}^{pr_k})$ # Sum all scores

For input features, we generally pay more attention on those relevant components, such as the set of pixels containing objects. Therefore, beside the overall consistency, we evaluate two additional metrics, which are Top-1 and Top-3 agreement. Top-K indicates the percent of overlap for the K features that are most sensitive to predictions and explanations, which is formulated as:

$$TA_k = \frac{\left|\left\{Top_k(S_f^e) \cap Top_k(S_f^{pr})\right\}\right|}{\left|\left\{Top_k(S_f^e)\right\}\right|} \tag{4.5}$$

where $Top_k(S)$ represents the set of the first k elements from sorted S. The Top-K metric mitigates the interference caused by background pixels to some extent, thereby concentrating more on the sensitivity assessment of the target objects.

4.3.3. Parameter Sensitivity Consistency

Apart from features, parameters are expected to be consistently sensitive as well, which ensures that predictions and explanations made by the model on a given input are mainly attributed to the same set of neurons. In contrast to images (typically $C \times W \times H$), parameters are higher dimensional, which encompass diverse architectural units and rendering the perturbation more challenging. To avoid explosive computational intensity, we draw the following compromises without sacrificing too much performance:

- We compute the parameter similarity for each layer individually and derive the global similarity by averaging.
- As the majority of the parameters belong to the weights, we only evaluate the parameters on the weights of the feature extraction layers (convolutional and fully-connected layers), ignoring the biases.
- The quantity of parameters on the weights is far greater than the image pixels, thus requiring a remarkably larger number of masks to accurately assess the sensitivity, which causes enormous time and computation costs. To enable the experiment being feasible, we group the parameters according to the type of the layer to diminish the amount of masks required for perturbation. For a convolutional layer with size $D_{in} \times D_{out} \times C_w \times C_h$, we treat an output channel as a perturbation unit with the size of $D_{in} \times C_w \times C_h$. The reason for not considering an input channel as a perturbation unit is to avoid the possibility that only one perturbable term exists when processing uni-channel data (e.g. MNIST). For a fully connected layer $D_{in} \times D_{out}$, we select an input channel as a perturbation unit with dimension D_{out}. The argument against treating the output channel as unit is to prevent masking the label channel when perturbing the last layer and thereby losing gradients.

The process of calculating parameter sensitivity is analogous to that of feature sensitivity. We randomly generate masks on parameters groups, which partially eliminate the

original weights and turn into new models. Comparing the differences in prediction confidence and explanation similarity between the original and new models for the identical inputs allows for the calculation of sensitivity consistency for each parameter group. The parameter sensitivity consistency is formulated as:

$$\rho_S^{pa} = \rho_S(S_{pa}^e, S_{pa}^{pr}) \tag{4.6}$$

where $S_{pa}^{pr} = \{S_{pa_1}^{pr}, ..., S_{pa_2}^{pr}\}$ denotes the parameter sensitivity for the prediction, in which pa_k is the k^{th} parameter group in a certain layer, and $S_{pa_k}^{pr}$ is calculated by the prediction differences, also formulated as $S_{pa_k}^{pr} = \sum_{i=1}^{n}(1 - \Delta P_i) \odot M_i^{pa}$ (M_i^{pa} is the i^{th} mask on the parameter groups). S_{pa}^e is the parameter sensitivity of the explanation, obtained by $S_{pa}^e = \sum_{i=1}^{n} \rho_S(e^o, e_{m_i^{pa}}) \odot M_i^{pa}$, where $e_{m_i^{pa}}$ is the explanation generated by the model under mask M_i^{pa}. The details of parameter SenC is demostrated in Algorithm 2. Note that though we need to access the model parameters when evaluating parameter SenC, we are still interested in the correlation between input and output, and thus SenC is still considered to be a black box in a broad sense.

Intuitively, both predictions and explanations are supposed to follow variations in the same set of features and parameters. Therefore, explanations with higher sensitivity consistency are considered more plausible. Lastly, due to the lack of reference, we randomly generate a nonsensical explanation for each data and equally perform the sensitivity consistency evaluation on it as the baseline.

Algorithm 2: Parameter Sensitivity Consistency (SenC)

Input : An input data x, a well-trained model $F(\cdot)$ with layer-wise parameters $\{w_1, ..., w_n\}$, an explainability method $H(F, x)$ and the number of masks K
Output: Parameter SenC ρ_S^{pa} of H
$P_x^o = F(x)$ #
$e_x^o = H(F(x), x)$ #
for $m = 1$ to n do: #Layer-wise process
  $S_{w_m}^{pr}, S_{w_m}^e = zeros\_like(w_m)$ # Initialization with the shape of corresponding parameters
  for $k = 1$ to K do:
    $M_k^{w_m} = random\_like(w_m)$
    $w_m'^k = w_m \odot M_k^{w_m}$
    $F'_{w_m'^k} = \{w_1, ..., w_m'^k, ..., w_n\}$ #k^{th} perturbation on $m^{th} layer$
    $S_{pa_m}^{pr_k} \mathrel{+}= 1 - \left| P_x^o - F'_{w_m'^k}(x) \right| \odot M_k^{w_m}$
    $S_{pa_m}^{e_k} \mathrel{+}= \rho_S(e_x^o, H(F'_{w_m'^k}, x) \odot M_k^{w_m}$
  end for
  $\bar{\rho}_S^{pa} = \sum_{k=1}^{K} \rho_S(S_{pa_m}^{e_k}, S_{pa_m}^{pr_k})$
$\rho_S^{pa} = \sum_{m=1}^{n} \frac{\bar{\rho}_S^{pa}}{n}$ #Global parameter SenC

4.4. Experiments

In this section we demonstrate the experimental results. Our experiments are conducted on three datasets with different complexities, which are the MNIST handwritten dataset, CIFAR10 and a real-world dataset called *German Traffic Sign Recognition Benchmark* (GTSRB) [27], respectively. For better prediction performance, we train models with different structures on each of the three datasets. For MNIST, we train a simple four-layer neural network, noted as ModelCNN, whose structure can be simply summarized as $Conv1 \rightarrow MP1 \rightarrow Conv2 \rightarrow MP2 \rightarrow FC1 \rightarrow FC2$, where $Conv$, MP and FC denote convolutional, max-pooling and fully connected layers, respectively. ModelCNN achieves 98.5% accuracy on the MNIST test set. For CIFAR10 and GTSRB, we train a ResNet18 [48] and a MobileNetV3 [104] as the classifiers, respectively, which achieve 93.2% and 97.7% accuracy on the test set, respectively. Finally, we conduct a user study on ImageNet to verify whether the evaluation of SenC is consistent with human cognition.

We select the following explainability approaches as candidates to be evaluated: Vanilla Back-propagation (VB) [28], Guided Back-propagation (GB) [33], Integrated Gradients (IG) [78], Layer-wise Relevance Propagation (LRP) [35], GradCAM [75] and DeepLift [76]. Since perturbation-based evaluation consumes a significant amount of time to generate and process masks, while surrogate models methods such as LIME [54] also demand extensive perturbation samples for their explanations, this category will not be evaluated considering the time and computational costs. In the experiments, all explainability methods are implemented based on Captum toolkit [147].

The experimental configurations are as follows: When evaluating the data SenC, we randomly select 1000 instances from the test set, and for each instance the number of perturbation masks generated is 5000. When evaluating the parameter SenC, we select only 100 instances from the test set but generate 10000 random masks for each layer of the model to be evaluated, as the number of perturbable channels of the parameter is significantly larger than the number of hyperpixels in the image. In segmenting the image, we utilize *slic* in the *scikit-image* package to roughly split every 50 pixels into one superpixel, which maintains the independence of local features without excessively raising the computational intensity. We chose 0.8 as the masking rate for the generated masks. The masking rate is a flexible hyperparameter with appropriate values will not significantly impact the evaluation results.

4.4.1. Sensitivity visualization

In this section we demonstrate two visualization examples for data and parameter sensitivity consistency, respectively.

Data sensitivity. We show the data sensitivity of a random image in CIFAR10 in Fig. 4.2. The sensitive areas of prediction can be seen to be centered on the front part of the body and head of the deer. However, for explainability methods, the sensitive fields are

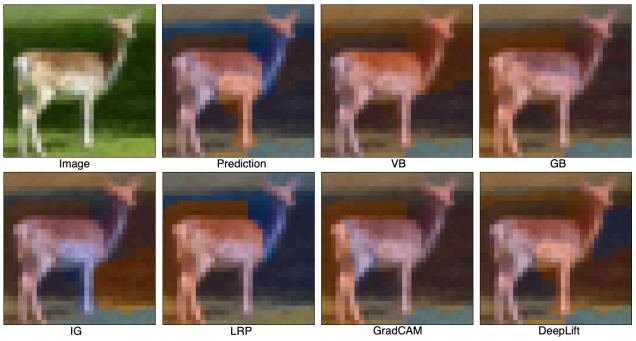

Figure 4.2.: Visualization of data sensitivity from CIFAR-10 dataset. Areas rendered in red represent high sensitivity while those in blue indicate low sensitivity.

partially different: e.g. the neck of the deer is included in the sensitive regions by VB, as well as the upper part of the background is labeled as more sensitive for VB, IG, GradCAM and DeepLift. The closest match to the prediction sensitivity is LRP, so which therefore yields the highest sensitivity consistency for this instance.

Parameter sensitivity. For a simple illustration, we choose the first convolutional layer ($conv1$) of ModelCNN as the object to visualize the parameter sensitivities, which is shown in Fig. 4.3. This layer contains 16 output channels, each represented by a square in the figure, where red and blue indicate high and low sensitivity, respectively. It can be observed that for prediction, the first, third and sixteenth channels are most sensitive. For the explainability methods, all except Deeplift label the third channel as highly sensitive, while among them VB,GB and LRP exhibit high sensitivity on all three channels simultaneously, hence their sensitivity consistency for the particular input on this layer is relatively high.

4.4.2. Quantitative SenC evaluation

MNIST

MNIST is the simplest image dataset, which consists of 60,000 train and 10,000 test instances, each with the size of 784 (28×28). For efficient evaluation, we restrict the number of hyperpixels to 15.

Data SenC. The results of the qualitative assessment of data SenC are demonstrated in (a) of Fig. 4.7. As a reference, we generate a random mask for each instance as a baseline explanation, whose average SenC is expected to be zero. The results illustrate that for data with low complexity, almost all explainability methods exhibit consistent sensitivities (mean SenC $\bar{\rho_S} > 0.6$), except for GradCAM, whose variance is slightly higher ($\sigma^2(\rho_S) = 0.13$).

Furthermore, we present the agreement of features with Top-1 and Top-3 sensitivities in (a) and (d) of Fig. 4.4, respectively. All explainability methods except GradCAM are

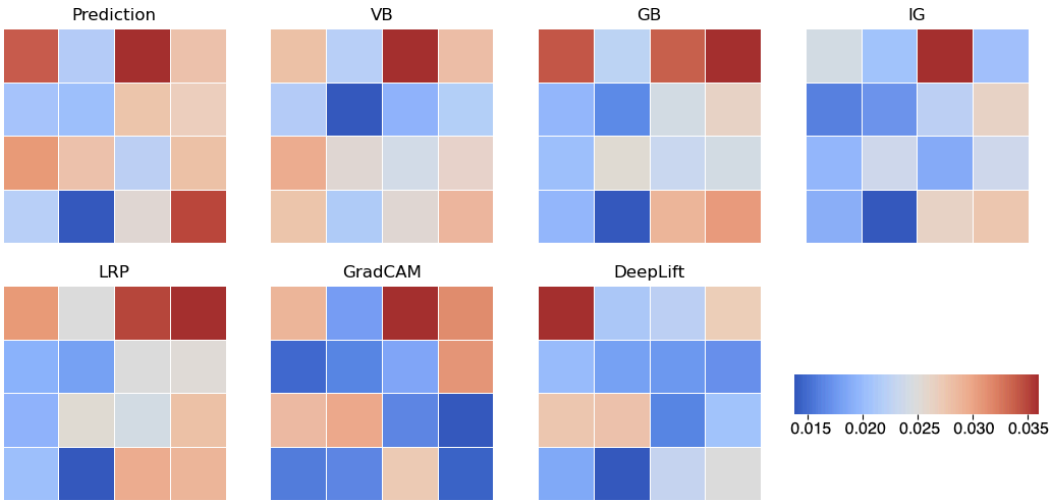

Figure 4.3.: Visualization of parameter sensitivity. The layer being visualized is the first convolutional layer of the ModelCNN, which contains 16 output channels corresponding to the 16 squares in the figure. The redder the color of the squares, the higher the sensitivity and vice versa.

capable of reaching a Top-1 agreement of around 80%. In the Top-3 metric, again with the exception of GradCAM, the probability that all three of the most sensitive features in the explanations generated by the rest of the methods are all in agreement is higher than 20%, while there are barely any cases where no intersection exists (percent $p < 5\%$). In general, for simple datasets like MNIST, almost all explainability methods achieve highly sensitivity consistency of input features for predictions and explanations.

Parameter SenC. (a) of Fig. 4.6 shows the average SenC of different explainability methods on all layers of the model (also including the randomized baseline explanation). In the results, GB and IG achieve higher levels of consistency ($\bar{\rho_S} = 0.29$ and 0.32, respecitively), whereas the consistency of GradCAM is relatively low ($\bar{\rho_S} = 0.14$) and unstable ($\sigma^2(\rho_S) = 0.014$). The remaining methods reveal an intermediate level of sensitivity consistency ($\bar{\rho_S} \in [0.2, 0.3]$). Moreover, we analyze each layer individually and present the results in Fig. 4.5. We note that the sensitivity consistency of convolutional layers is far superior to that of fully-connected layers, regardless of which explainability method is applied. We attribute the reason to two points: a) Convolutional layers are more intuitive as they directly extract features from the adjacent areas of images, whereas fully-connected layers are typically treated as latent features, which are highly abstract and may be activated by features in different regions. b) The channels of convolutional layers are more independent of each other compared to fully-connected layers, and thus are less impacted under individual perturbations. Therefore, we recommend choosing the top convolutional layer as the target for evaluating parameter consistency.

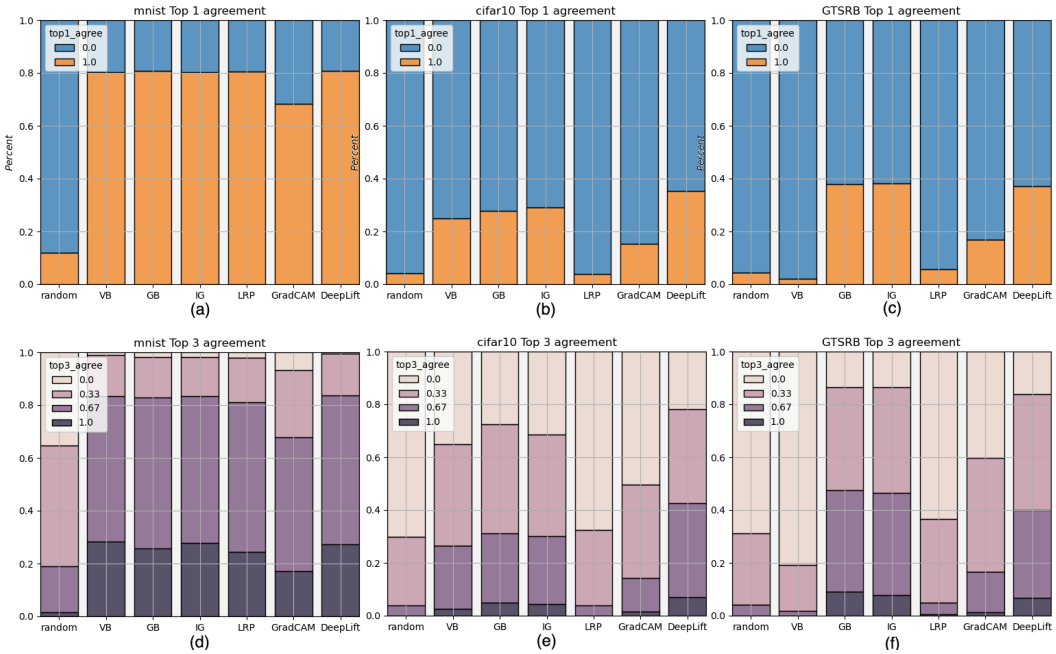

Figure 4.4.: From up to bottom are Top-1 and Top-3 agreement, respectively. The x-axis in all plots represents different explainability methods. The y-axis in agreement indicates percentages. In Top-1 agreement, larger proportion of 1.0 (orange) fractions signify a higher percentage of agreement on the most sensitive features (better). In Top-3 agreement, higher percentage of darker color sections indicates better agreement.

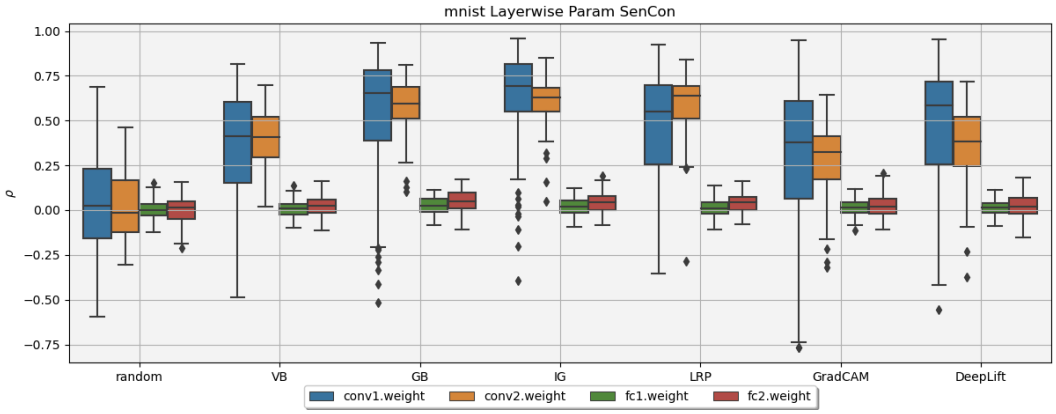

Figure 4.5.: Layer-wise parameter sensitivity consistency assessment of ModelCNN trained on MNIST dataset. The x-coordinates are the different explainability methods, the y-coordinates are the Spearman correlation coefficients for the parameter sensitivities, and each box in the figure represents a specific layer.

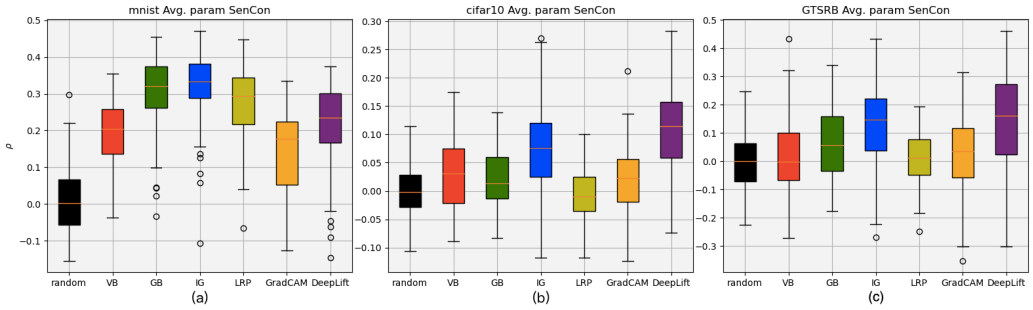

Figure 4.6.: Average parameter sensitivity consistency over all selected layers. From left to right are the MNIST,CIFAR-10 and GTSRB datasets, respectively.

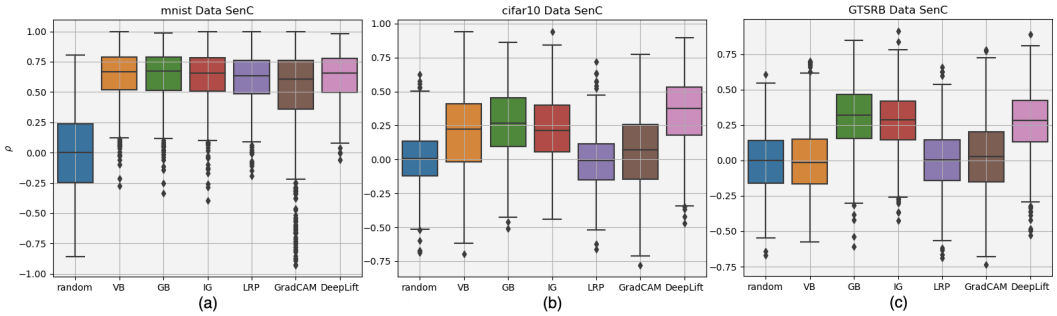

Figure 4.7.: Evaluation of data sensitivity consistency. From top to bottom are the evaluation results on MNIST,CIFAR-10 and GTSRB datasets, respectively. The x-axis in all plots represents different explainability methods, the y-axis represents Spearman's correlation coefficient ρ_S. Higher ρ_S denotes more consistent sensitivity.

CIFAR-10

CIFAR10 is a small-size (32×32) image dataset consisting of 10 categories, which contains 50000 training and 10000 test data. When splitting the hyperpixels, again every 50 pixels are split into a group with a total number of 20 hyperpixels per image.

Data SenC. The data SenC is illustrated in (b) of Fig. 4.7. Compared to MNIST dataset with low complexity, the data SenC of CIFAR10 exhibits a significant collapse, especially for LRP, whose $\bar{\rho_S}$ plummets from 0.609 to -0.012, which implies that when explaining structurally complex data and models, LRP barely reveals consistency of sensitivity between predictions and explanations. Besides, GradCAM still suffers from low consistency $\bar{\rho_S} = 0.052$, which is almost on par with randomly generated explanations. In contrast, despite the substantial reduction, DeepLift maintains a relatively high consistency ($\bar{\rho_S} = 0.351$) and is therefore considered to be the more stable explainability method.

We again evaluate the Top-1 and Top-3 SenC and show the results in (b) and (e) of Fig. 4.4, respectively. Agreements for all explainability methods decline significantly on CIFAR-10, with the most dramatic drop being for LRP, whose Top-1 agreement falls from nearly

80% on MNIST to the same level as the randomized explanations. The performance of Top-3 agreement is analogous to that of Top-1, where VB, GB, IG and DeepLift outperform, with over 60% of their Top-3 sensitivity features sharing at least one agreement. DeepLift still maintains the best agreement with more than 40% probability that at least two of its Top-3 features overlap. As a conclusion, the complexity of CIFAR-10 is elevated compared to MNIST, resulting in a certain decrease in sensitivity consistency for all explainability methods, while DeepLift maintains the highest consistency.

Parameter SenC. Due to the complicated structure of ResNet, for clarity, we only demonstrate the consistency of parameter sensitivities of the first convolutional layer, the last fully-connected layer, and all the intermediate hidden layers belonging to "layer1". The parameter sensitivity consistency of ResNet18 is demonstrated in (b) of Fig. 4.6. In comparison to MNIST, all explainability methods suffer from various degrees of reduction in consistency, most notably GB and LRP, which both exhibit a decrease in average parameter consistency of 0.27, and LRP, whose consistency is already lowered to a comparable level to that of the random explanation. The consistency of VB and GradCAM degrades 0.17 and 0.13, respectively, to a relatively insignificant degree, due to their unprominent performance on MNIST. Despite declining 0.25 and 0.1, respectively, IG and DeepLift remain in relatively high consistency, with the average of DeepLift remaining at a high level above 0.1. The conclusions are aligned with those in MNIST, where convolutional layers are more consistent compared to fully-connected layers, and the deeper the layer the more difficult their sensitivities are to be consistent. Additionally, IG and DeepLift again outperform the parameter SenC, especially for the layers $conv1$ and $layer1.0.conv1$, which are remarkably higher than all other explainability methods.

GTSRB

To test the reliability of explainability methods on real-world datasets, we conduct experiments on the GTSRB dataset [27]. GTSRB is a dataset consisting of photographs of 43 different types of traffic signs, which includes 39209 and 12630 training and test data for learning and prediction, respectively.

Data SenC. The data SenC of GTSRB is shown in (c) of Fig. 4.7 and (c), (f) in Fig. 4.4. No significant fluctuations are observed for all explainability methods compared to CIFAR-10, except for a significantly decline in SenC for VB. The trends in Top-1 and Top-3 agreement evaluations are roughly equivalent, with all methods maintaining comparable levels except for VB, which rapidly collapses. In summary, for data SenC on GTSRB, GB, IG and DeepLift perform relatively better compared to other explainability methods.

Parameter SenC. The average parameter sensitivity consistency of GTSRB is exhibited in (c) of Fig. 4.6. The results for the average parameter SenC for GTSRB are comparable to those of CIFAR-10, except for a relatively noticeable gain in GB while IG and DeepLift still remain superior. For the layer-wise evaluation, again due to the computational intensity, we choose only the first 2 convolutional layers and the last 2 fully connected layers of the network. The final conclusion remains broadly uniform, that IG and DeepLift exhibit

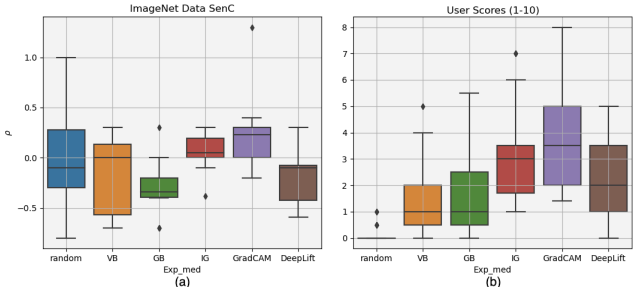

Figure 4.8.: Quantitative evaluation and user study on ImageNet. (a) SenC evaluation results and (b) User scores.

relatively better consistency in the first two layers, whereas the consistency of VB, LRP, and GradCAM fails to show an advantage over randomized explanations.

4.4.3. Complex model (dataset) and user study

For practical purposes, we perform an evaluation of data SenC on the SOTA large model while conducting a light scale user study. We chose VIT [135] as the classifier, train it on ImageNet and achieves 80.7% accuracy in the testset. We randomly select 10 out of 1000 categories from ImageNet, first generate explanations with the selected explainability methods (Vanilla LRP is discarded due to incompatibility with VIT), and evaluate their data SenC. In parallel, we send the generated explanations to the users and let them score each explanation subjectively based on their experience and intuition. We invite 21 participants for the study by showing them the original images and the explanations generated by the explainability methods, and asking them to rate each explanation. For those users who do not have basic knowledge in XAI, we briefly introduce the idea of explainability methods and their functionality. Explanations are rated on a scale of 0 to 10, with 0 representing barely able to provide any information, to 10 where users believe they can fully understand the basis of model predictions.

We finally combine the evaluations of SenC with user feedbacks in Fig. 4.8 to observe whether they are correlated. The result demonstrates that while IG and GradCAM receive relatively high ratings from users, their SenC also wins in quantitative evaluations. On the contrary, VB, GB and DeepLift suffer from flawed SenC in this dataset, as well as they receive lower user scores. The result indicates that the SenC evaluation results are to some extent consistent with the intuitive perceptions of humans.

4.5. Limitations

This work presents a novel perspective for evaluating explainability methods. However, we acknowledge that a few deficiencies remain non-negligible, which fall broadly into the following three points:

- **OOD Perturbations.** Hard perturbation that may disrupt the distribution of data is one of the largest challenges for explainability studies. Though alternatives have been proposed in recent studies such as [103], they are not widely accepted and applied due to computational intensity and fidelity issues. In the assessment of SenC, perturbing data or parameters with randomly generated masks is considered as hard perturbation, which is one of the factors that may threaten the reliability of the evaluations.
- **Perturbation channels for FC layers.** FC layers exhibit much less consistency than convolutional layers in the evaluation, partially due to the issue of segmenting the channels of FC layers. On the one hand, the FC layers contain an enormous number of parameters, which if perturbed discretely would require an incalculable amount of masks, rendering them almost impossible. On the other hand, unlike convolutional layers with well separated channels, neurons in FC layers are densely connected, and hard masking any part may severely impair the remaining ones. Therefore, a proper splitting and perturbation approach is desired to balance the assessment accuracy and computational intensity.
- **Computation time costs.** Similar to other perturbation-based methods, SenC sacrifices efficiency for the black-box property. When evaluating structurally complex data or models, a relatively large quantity of hyperpixels or channels is required, which leads to an explosive demand for the number of masks. Reducing the amount of hyperpixels or channels effectively mitigates this issue, however will result in a compromise in evaluation precision, which is a tradeoff that also needs to be addressed.

4.6. Conclusion

This chapter proposes a novel perspective to evaluate explainability methods. By generating a large number of masks to perturb inputs or parameters to identify which components are sensitive to the perturbation and assess whether they are consistent. We conduct experiments with three different datasets and models, as well as a user study on a more complex dataset. The result reveals that the SenC assessment is to some extent consistent with human intuition.

Another major issue with sensitivity test is that it may disrupt the input feature distribution. In the next chapter, we propose an autoencoder-based evaluation method that eliminates the need to perturb features, thereby ensuring the integrity of the feature distribution.

5. The Generalizability of Explanations

In the previous two chapters, we focused on the issue of the explanation fidelity to the participants in the prediction. Nonetheless, explanations that are faithful to the model parameters or inputs may not directly indicate high quality, as they are still, for example, difficult to perceive and comprehend, inconsistent or noisy. Furthermore, as mentioned in Sec. 1.2.2, the ablation test inevitably faces the risk that the feature distribution is disrupted, and the credibility of the evaluation results may thus be overestimated. Therefore, evaluating explanations on the premise of not disrupting the feature distribution is a promising research direction. In this chapter, we start from the properties of the explanations themselves, and assess the regularity and learnability of their distributions through a deep neural network with promising reconstruction capabilities, denoted as the generalizability. We employ an encoding-decoding module to learn the distributions of the generated explanations and observe their learnability as well as the plausibility of the learned distributional features. This metric does not require perturbing features, and enables quantitative evaluation of explanations while maintaining the feature distribution of the input. First we briefly demonstrate the evaluation idea of the proposed approach for LIME, and then quantitatively evaluate multiple popular explainability methods. We also find that smoothing the explanations with SmoothGrad can significantly enhance their generalizability.

5.1. Introduction

As mentioned in Sec. 1.2.2, existing sensitivity tests suffer from the risk of disrupting the feature distribution, which may directly compromise the reliability of evaluations. This chapter proposes a novel evaluation approach from the perspective of generalizability, which preserves the input feature distribution. Our assumption is that a plausible explanation should share a proximate distribution with the original data and we train an encoding-decoding model that is adequate for the complexity of the data and observe the performance while reconstructing the explanations. Our approach is intuitive, applicable to all explanations in the form of saliency maps (both for gradient- and perturbation-based). Our contributions are as follows:

- We propose a novel method for evaluating explanations that validates whether they possess information consistent with the original data from the perspective of generalizability.
- We evaluate popular saliency map-based explainability methods.

- We show the interesting observation that SmoothGrad can effectively enhance the generalizability of explanations.

The structure of this chapter is as follows: In Section 5.2 we present the relevant studies, in Section 5.3 we detail the proposed approach, and in Section 5.4 we show the experimental results. Finally, we conclude and describe future work in Section 5.5.

5.2. Related Work

In this section, we introduce existing approaches for evaluating explanations.

Evaluation Approaches: There are two existing mainstream evaluation methods, the sensitivities to input perturbations and model parameters [176]. The former operates by perturbing the features of the input and observing the changes in the prediction. For a plausible explanation, perturbations to the feature with the largest attribution lead to severe corruption in the prediction confidence. These approaches are intuitive and therefore widely applied to justify the reliability of explanations [35], [56], [60], [86], [101], [105]. However, it has been scepticized that such perturbations neglect the features (pixels) correlation or feed a distribution that the model has never learned before, which impairs the reliability of the evaluation [103], [152]. [103] proposes RemOve And Retrain (ROAR), which retrains the model with the perturbed dataset and observes the magnitude of the performance degradation, to some extent eliminating the out-of-distribution issue of the perturbed data.

The sensitivity of the model parameters originates from a concern about the explanation sanity. [83], [84] reveal that by randomizing the parameters of the model, several explainability methods remain unaffected, which raises questions about whether the explanations are faithful to the model.

Besides, several other approaches are proposed, such as Pointing Game [98], which counts the number of times the point with the largest attribution in the explanation is inside the target object in the image. User study is also an important evaluation methodology. Although the explanation needs to be human-friendly, however, this method is costly and subjective, which may not reveal the true basis of the prediction [95]. Moreover, several studies raise concerns about the robustness or stability of explainability methods [86], [105], [180]. Although they experimentally demonstrate certain deficiencies of explanations, such as the lack of linear invariance [105] and Lipschitz continuity [86], they cannot be used as quantitative evaluation approaches.

5.3. Methods

In this section we present a novel approach to evaluating explanations. Our approach is applicable to all reconstructable data types, such as images, text and tabular data, etc.

Unlike existing sensitivity measures, this method targets the generalizability of the explanations across the entire data set.

Consider a image dataset $X = \{x_1, ..., x_n\} \subseteq \mathbb{R}^{W \times H}$ with a well-trained model $H(\cdot)$. We obtain the explanation set $E = \{e_1, ..., e_n\} \subseteq \mathbb{R}^{W \times H}$ by the explainability method H. The explainability approach can be regarded as a mapping function, i.e. $e_i = H(F, x_i)$ (only local methods are considered here). Therefore, there should be a certain degree of distributional similarity between X and E: $X \dot{\sim} E$. Intuitively, this can be understood as "proximity", i.e., the model predicts the input as a class based on a certain set of features. Existing studies had mentioned similar concerns, they perturb the input and measured the local continuity of the explanations with *Local Lipschitz Continuity*, which is formulated as:

$$\|H(F, x_i) - H(F, (x_i + \epsilon)\| \leq L \|x_i - (x_i + \epsilon)\| \tag{5.1}$$

where ϵ is the perturbation matrix and L is a constant. However, the perturbations may disrupt the data distribution. Analogous to sensitivity tests [103], out-of-distribution data that are never seen by the model may impair the evaluation performance.

Our approach is more inclined to "reconstruct" rather than "perturb". For a well-trained model F, a certain rule R_m should exist with regard to the prediction, for instance, when multiple horse images are input, the model takes similar features as prediction bases(e.g., legs or tails). A plausible explainability method H should exhibit those rules in its explanations E. However, R_m is agnostic due to the opaqueness of the black box model. Therefore, under the premise that the model is well-trained, we can verify whether H implies the rule R_m by analyzing E. Inspired by early researches, which have shown that a network with sufficient parameters can fit any function [18], [20], we train an appropriately structured encoding-decoding neural network G to simulate R_m. G takes the original images x_i as input and outputs the corresponding (synthetic) explanation e'_i ($e'_i \approx e_i = H(F, x_i)$), and the reliability of the explainability method H is evaluated by observing the performance of $G(X)$ on the whole dataset. The evaluation via G is based on two factors:

- Distribution Learnability: H should not be erratic. If the performance of the trained G is inferior, it indicates that H is *unlearnable* and thus *erratic*. For example, G cannot learn any rule from randomly distributed saliency maps, which leads to a nearly non-decreasing loss curve. Note that the reverse of this factor does not hold. One trick is to generate simple and identical explanations for all inputs, which can be easily learned by the network, whereas they cannot be considered as reliable explanations.
- Variance Proximity: As a complement to the previous point, P' should possess similar statistical properties as X. Intuitively, the gap between explanations in the same class should be much smaller than that between different classes. This complement prevents the aforementioned "simple and identical" explanations from being regarded as plausible, since they are not consistent with the distribution of the original images.

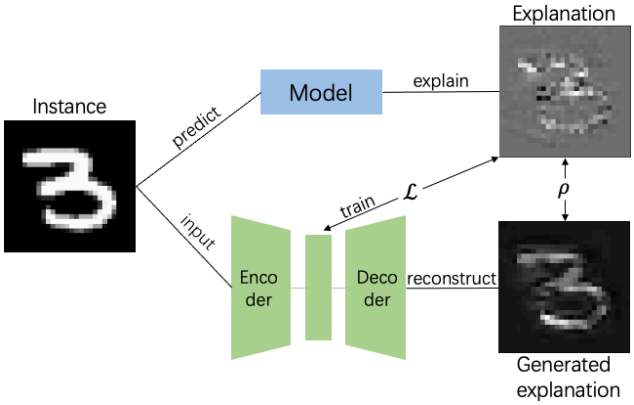

Figure 5.1.: An overview of the evaluation methods. We first select the explainability method to be evaluated and generate explanations utilizing the classification model and the original inputs. Subsequently, we train a generative model that takes the original image as input and attempts a reconstruction of the generated explanations. Finally, we compare the distributional relationships between the reconstructed instances and the explanations.

Subsequently, we elaborate on the details of the proposed evaluation approach. The general structure of the approach can be seen in Fig. 5.1. Our evaluation method consists of two components, which verify generalizability and proximity, respectively.

Distribution Learnability: The essential task of the simulating model is to learn the rules of the explainability method. Therefore, the model requires sufficient learning capability (i.e., structural complexity). We search for Autoencoders with different architectures until it can reconstruct the input image with high quality, which indicates that the structure is competent for the complexity of the dataset. We train this autoencoder with the original images and the explanations as inputs and labels, respectively, with L_1 as the target loss function until the loss curve converges. We denote this Autoencoder as AE_i for easy reuse in subsequent processes. We divide the dataset X into training (X_{tr}) and test sets (X_{te}), and we train the Autoencoder corresponding to each explainability method (denoted as AE_H, where H represents a certain explainability method) with X_{tr} according to the framework in Fig. 5.1, and then input the X_{te} into AE_H to obtain the reconstructed explanations E'_{te}. According to the reconstruction performance the plausibility of the explanations can be observed: Autoencoder is more likely to reconstruct those explanations that possess more typical rules, while on the contrary, erratic and random explanations cannot be well-learned. We show an example for the generalizability evaluation in Section 5.4.1.

To quantitatively assess the learnability of the explanations, we calculate the reconstruction performance of E'_{te} by six different measurements:

L_1&L_2: The L_1 and L_2 distances provide an intuitive indication of the pixel-wise similarity, however they may be affected by extreme values and those points with little attribution occupy the same weight.

Structural Similarity Index Measure (SSIM): In particular, for images, we introduce SSIM as a similarity measure, which measures the statistical likelihood of two explanations regarding the mean and variance in terms of luminance, contrast and structure. SSIM is formulated as:

$$SSIM(P, P') = \frac{(2\mu_P\mu_{P'} + c_1)(2\sigma_{PP'} + c_2)}{(\mu_P^2 + \mu_{P'}^2 + c_1)(\sigma_P^2 + \sigma_{P'}^2 + c_2)} \tag{5.2}$$

where μ and σ denote the mean and (co)variance, respectively. Note that the core of SSIM is the approximation of the statistics, existing studies show that it is problematic to assess the perceptual proximity [150], so we only consider it as a reference.

Pearson & Spearman's rank correlation coefficient: Pearson correlation coefficient (PC) measures the linear correlation of two explanations, which is formulated as $\rho_R = \frac{COV(E,E')}{\sigma_E\sigma_{E'}}$. When comparing explanations, one may focus more on the ranking of feature attributions than on the specific values. Spearman's correlation coefficient (SC) is the ranked version of the *Pearson correlation coefficient*, which can be formulated as:

$$\rho_S(E, E') = \frac{COV(R(E), R(E'))}{\sigma_{R(E)}\sigma_{R(E')}} \tag{5.3}$$

where $R(*)$ is the rank function, COV is the covariance and σ is standard deviation.

Top-k Accuracy (TA): For explanations, humans tend to be concerned only with the features that yield larger attributions. Therefore, we perform a Pointing Game on the pixels with the top-K attributions. Specifying a percentage K (all 25% in the experiment), we calculate how many pixels in E_{te} ranked in the top-K attributions appear in the top-K attributions of the pixels in E'_{te}. It can be formulated as:

$$TA = \frac{|TopK(E) \cap TopK(E')|}{|TopK(E')|} \tag{5.4}$$

TA is in the range $[0, 1]$, with higher values representing that the pixels with large attribution are more similar in the two explanations. Note that the TA of two sufficiently large random sequences converge to K (0.25 in this experiment).

The first three of the above measure the numerical (or statistical) distances. However, for explanations, we are more interested in the discrepancies of rankings, therefore in the experiments we mainly monitor the last three metrics for evaluation.

Variance Proximity: High learnability is not sufficient to conclude the plausibility of the explanations. Simple rules can be easily learned by the Autoencoder (e.g. focusing the attribution on a fixed pixel), however they may not provide reliable explanations. Therefore, we exclude these uninformative ones by evaluating the proximity of the distributions between the reconstructions and the original images. Our assessment is based on the idea that the variation of the reconstructed explanations within a same class should be much smaller than those between different classes. In this regard we consider two proximity measurements, the pixel and latent-wise discrepancies. For the former, calculating the

absolute value is insignificant, since the reconstructed explanations may not be in the same order of magnitude as the original image. Therefore, we also employ *Spearman's rank correlation coefficient* to measure pixel-wise proximity. For the latter, we constructed a latent distance measurement for validation with the pixel-wise approach, which can be considered as a variant of *Fréchet inception distance (FID)*. We take the encoder of the original Autoencoder (AE_i) as the "Inception" network, input the images and explanations respectively to obtain the latent vectors, and compute their 2-Wasserstein distances (both are considered as multidimensional Gaussian distributions)

$$\begin{aligned} d_H(L_E(\mu_E, \sigma_E), L_{E'}(\mu_{E'}, \sigma_{E'})) &= \|\mu_E - \mu_{E'}\|_2^2 \\ + tr\left(\sigma_E + \sigma_{E'} - 2\left(\sigma_E^{\frac{1}{2}} \cdot \sigma_{E'} \cdot \sigma_E^{\frac{1}{2}}\right)^{\frac{1}{2}}\right) \end{aligned} \tag{5.5}$$

In measuring the reconstructed similarity within classes, we take a certain number of samples from each class for a pairwise comparison. For inter-class, we try all combinations of different classes and take same number of samples from each class for a pairwise comparison. Our Variance Proximity scores (VP) can be calculated as:

$$VP = \Delta\overline{\rho_{S_a^r}} + \Delta\left\|\overline{FID_a^r}\right\| \tag{5.6}$$

where $\Delta\overline{\rho_{S_a^r}}$ is the difference between the mean of the inter- and intra-class Spearman coefficients, which can be formulated as:

$$\Delta\overline{\rho_{S_a^r}} = \overline{\rho_{S_a}} - \overline{\rho_{S_r}} \tag{5.7}$$

and $\Delta\left\|\overline{FID_a^r}\right\|$ denotes the (normalized) difference between the averages of the inter- and intra-class FIDs, which can be expressed as:

$$\Delta\left\|\overline{FID_a^r}\right\| = (\overline{FID_r} - \overline{FID_a})/\overline{FID_a} \tag{5.8}$$

Finally, the quantitative score of corresponding explainability methods is:

$$S_{EM} = VP \times DL \tag{5.9}$$

where VP is the variance proximity (see equation 5.6), and DL denotes the Distribution Learnability, which is the average of Top-k Accuracy ($\overline{TA}$), Spearman's Correlation ($\overline{\rho_S}$) and Pearson Correlation ($\overline{\rho_R}$) coefficients (the overline denotes that all three metrics are calculated by averaging over the last 10 training epochs).

5.4. Experiments

In this section, we first show an example on assessing LIME (Section 5.4.1), followed by a quantitative evaluation of the popular explainability methods. We choose MNIST

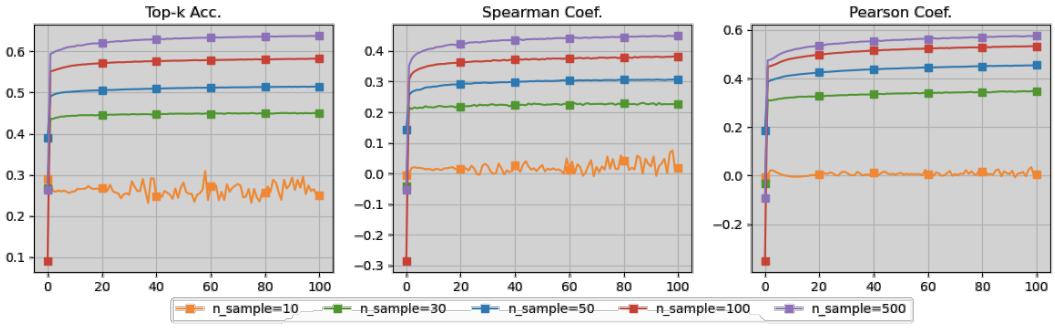

Figure 5.2.: The training curves of LIME with perturbed sample numbers of 10 (yellow), 30 (green), 50 (blue), 100 (red), and 500 (purple), respectively. The y-axis is the value of the corresponding metrics and the x-axis is the training epoch numbers.

handwritten dataset for our experiments. The accuracy of the classification model to be explained is 98.5% on the test set. The structure of Autoencoders applied for reconstruction is identical, whose latent vector has dimension 128. We train each for 100 epochs with a learning rate of $1e-5$. The Autoencoder that reconstructs the original image (AE_i) achieves $L_1loss \approx 0.01$, $MSEloss \approx 0.001$ and $SSIM > 0.99$. In the evaluation of Variance Proximity, we choose $S = 500$. For reproducibility, our code is available at `https://github.com/Explain3D/EXPGeneralizability`.

5.4.1. Case study:The Generalizability of LIME

To intuitively demonstrate the correlation between the generalizability and the quality of explanations, we first show a straightforward evaluation example. We choose an explainability approach by tuning the parameters to generate explanations with different qualities. LIME [54] is an ideal candidate as it relies on perturbing the instances to be explained to try out local decision boundaries, while the number of perturbed samples affects the quality of the explanation. The prior knowledge already exists that LIME performs poorly with few perturbed samples and improves as they increase until saturation. We segment the input into 50 super-pixels, and apply LIME with a total number of perturbed samples ($n\_samples$) of 10, 30, 50, 100 and 500 for explaining the classification model respectively, and evaluate the generalizability of the explanations according to the proposed metric.

The results, as illustrated in Fig. 5.2, indicate that insufficient number of perturbation samples may prevent Autoencoder from learning the distribution of the explanations. For instance, when $n\_samples = 10$, the distribution is barely learned: the pixel accuracy of top-k ($\overline{TA}$) converges to 25% ($26 \pm 3.6\%$), which is approximately identical to the probability of a random sampling. Besides, both Spearman and Pearson coefficients (SC and PC) oscillate near zero (0.021 ± 0.054 and 0.020 ± 0.056, respectively), implying there is hardly (ranked) linear correlation between the learned distribution and original explanation. As n_sample increases, the distribution learned by Autoencoder from the

n_sample	10	30	50	100	500
$\overline{TA} \uparrow$	0.267	0.449	0.514	0.582	**0.637**
$\overline{\rho_S} \uparrow$	0.032	0.227	0.306	0.381	**0.448**
$\overline{\rho_R} \uparrow$	0.010	0.347	0.454	0.532	**0.574**
$DL \uparrow$	0.103	0.341	0.425	0.498	**0.553**
$\Delta\overline{\rho_{S_a^r}} \uparrow$	0.019	0.120	**0.125**	0.111	0.124
$\Delta\overline{FID_a^r} \uparrow$	0.297	0.757	**0.892**	0.889	0.849
$VP \uparrow$	0.316	0.877	**1.017**	1.000	0.973
$S_{EM} \uparrow$	0.031	0.296	0.429	0.498	**0.536**

Table 5.1.: Detailed quantitative results of the proposed evaluation method for LIME with different n_sample. From top to bottom are the average of: Top-k Accuracy, Spearman's and Pearson's coefficients, the Distribution Learnability, the difference of Spearman's coefficient and Fréchet inception distance between intra and inter-class, the Variance Proximity and the final score of the generalizability. The up arrow indicates that the higher the value, the better the performance.

explanations grows more accurate. It can be observed that when n_sample increases to 30, the top-k accuracy, Spearman and Pearson coefficients of the generated samples converge to $44.9 \pm 0.1\%$, 0.227 ± 0.003 and 0.347 ± 0.002, respectively. As n_sample further grows to 50, these metrics are raised to $51.4 \pm 0.1\%$, 0.306 ± 0.002 and 0.454 ± 0.003, respectively. However, the benefits from a continued increase in n_sample are limited ($\overline{TA}$, $\overline{\rho_S}$ and $\overline{\rho_R}$ converge to $63.7 \pm 0.1\%$, 0.448 ± 0.001 and 0.574 ± 0.002, respectively when $n\_sample = 500$).

To avoid the trap of "simple rules", we also evaluate the Variance Proximity between the generated samples. Fig. 5.3 depicts the point-wise ranking (left) and latent space (right) proximity of the generated samples for inter and intra-classes. We observe that as the number of perturbed samples grows, the Spearman coefficients (SC) increase in varying degrees according to the class relationship. When the number of perturbation samples is minimal ($n\_sample = 10$), the similarity of intra- and inter-classes is approximated. The average of SC for inter and intra-class are $\overline{\rho_{S_r}} = 0.46$ and $\overline{\rho_{S_a}} = 0.49$, respectively, whose difference is $\Delta|\overline{\rho_{S_a^r}}| = 0.02$. Similarly, the means of FID for inter and intra-class are $\overline{FID_r} = 1.4e-5$ and $\overline{FID_a} = 1.8e-5$, respectively, the corresponding difference is $\Delta\left|\overline{FID_a^r}\right| = 3.7e-6$). It can be observed that the gaps between inter and intra-classes obviously increase with the growth of n_sample, which implies that the Autoencoder begins to learn the commonality within classes and the divergence between classes from the explanations.

We finally report the quantitative scores of LIME with $n\_sample = 10$, 30, 50, 100 and 500 as 0.031, 0.296, 0.429, 0.498, 0.536, which implies that the more perturbed samples result in better explanation performance. The tabulated results are shown in Table 5.1.

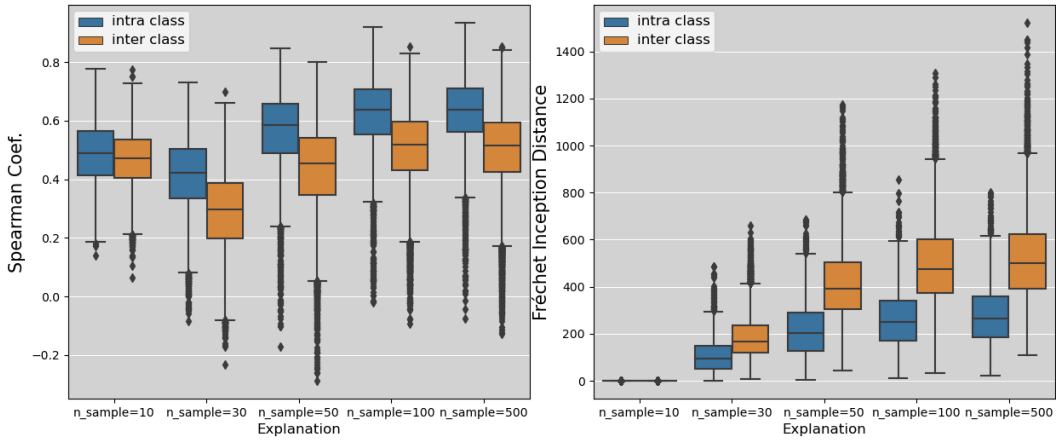

Figure 5.3.: The intra (blue) and inter (orange) class similarity (discrepancy) of the samples generated by Autoencoder based on the explanations of LIME with different number of perturbations. The x-axis from left to right shows the LIME for 10, 30, 50, 100 and 500 perturbed samples, respectively, and the y-axis is the Spearman coefficient (left) and Fréchet Inception Distance (right), respectively. Note that large Spearman coefficients represent similar distributions, while FIDs are the opposite.

5.4.2. The Generalizability of Explainability Methods

A further step is to extend the proposed evaluation approach to more explainability methods. We choose several popular gradient-based and perturbation-based methods, including Vanilla Gradients (V) [28], Guided Back-propagation (GB) [33], Input×Gradients (IxG) [57], Integrated Gradients (IG) [78], Layer-wise Relevance Propagation (LRP) [35], DeepLift [57], LIME ($n\_sample = 100$) [54] and KernelSHAP (KSHAP) [65]. As a reference, we additionally introduce a randomly generated noise explanation. Again, the explanations are generated from all above approaches, then learned and reconstructed by Autoencoder.

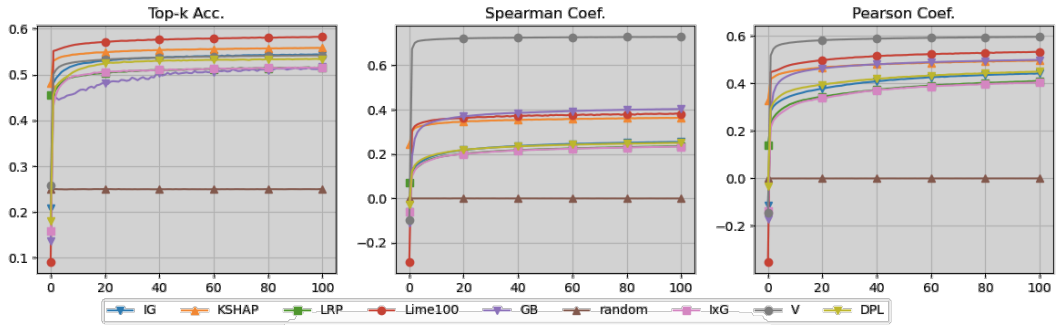

Figure 5.4.: The training curves of Vanilla Gradients, GB, IxG, IG, LRP, DeepLift, LIME, KernelSHAP and random explanation, respectively. The y-axis is the value of the corresponding metrics and the x-axis is the training epoch numbers.

	V	GB	IxG	IG	LRP	DPL	LIME	KSHAP	Random
$\overline{TA} \uparrow$	0.541	0.513	0.515	0.543	0.515	0.534	**0.582**	0.558	0.250
$\overline{\rho_S} \uparrow$	**0.727**	0.402	0.232	0.254	0.234	0.249	0.381	0.362	$-8e-5$
$\overline{\rho_R} \uparrow$	**0.596**	0.499	0.403	0.441	0.409	0.449	0.532	0.496	$2e-4$
$DL \uparrow$	**0.622**	0.471	0.383	0.413	0.386	0.411	0.498	0.472	0.083
$\Delta\overline{\rho_{S_a^r}} \uparrow$	0.025	**0.193**	0.111	0.107	0.110	0.105	0.109	0.140	0.035
$\Delta\overline{FID_a^r} \uparrow$	0.364	0.508	0.401	0.378	0.418	0.386	0.854	**0.876**	0.204
$VP \uparrow$	0.389	0.701	0.512	0.485	0.528	0.491	0.963	**1.016**	0.239
$S_{EM} \uparrow$	0.241	0.330	0.196	0.200	0.203	0.201	**0.479**	**0.479**	0.019

Table 5.2.: Detailed quantitative results of the proposed evaluation method for popular explainability methods. From left to right are: Vanilla Gradients, Guided Backpropagation, Input×Gradients, Integrated Gradients, Layer-wise Relevance Propagation, DeepLift, LIME, KernelSHAP and random generated explanations.

As the evaluation results illustrated in Fig. 5.4, all the explainability methods far outperformed the random explanation in terms of Top-K accuracy, Spearman and Pearson coefficients, which indicates that none of the explanations generated by these methods is erratic. Among these approaches, Vanilla Gradients ($\overline{TA} = 0.541 \pm 2e-4$, $\overline{\rho_S} = 0.727 \pm 2e-4$ and $\overline{\rho_R} = 0.596 \pm 4e-4$), LIME ($\overline{TA} = 0.582 \pm 5e-4$, $\overline{\rho_S} = 0.381 \pm 1e-3$ and $\overline{\rho_R} = 0.532 \pm 8e-4$) and KernelSHAP ($\overline{TA} = 0.558 \pm 4e-4$, $\overline{\rho_S} = 0.362 \pm 8e-4$ and $\overline{\rho_R} = 0.496 \pm 1e-3$) slightly outperform in terms of distribution learnability. Comparatively, IxG and LRP are less learnable, with all three metrics being relatively inferior ($\overline{TA} = 0.515 \pm 8e-4$, $\overline{\rho_S} = 0.232 \pm 8e-4$, $\overline{\rho_R} = 0.403 \pm 1e-3$ and $\overline{TA} = 0.515 \pm 7e-4$, $\overline{\rho_S} = 0.234 \pm 9e-4$, $\overline{\rho_R} = 0.409 \pm 1e-3$, respectively). Furthermore, we observe that the generalizability of perturbation-based methods ($\overline{DL_{overall}} = 0.485$) is superior to that of the majority of gradient-based methods ($\overline{DL_{overall}} = 0.447$).

Again, we assess the Variance Proximity between the generated explanations. As demonstrated in Fig. 5.5, the reconstructed explanations of all explainability methods outperform the random ones, which represents that their explanations are more distinguishable in terms of intra and inter-class discrepancies. However, Vanilla Gradients ($\Delta\overline{\rho_{S_a^r}} = 0.025$, $\Delta\overline{FID_a^r} = 0.364$ and $VP = 0.389$), Integrated Gradients ($\Delta\overline{\rho_{S_a^r}} = 0.105$, $\Delta\overline{FID_a^r} = 0.386$ and $VP = 0.491$) and DeepLift ($\Delta\overline{\rho_{S_a^r}} = 0.107$, $\Delta\overline{FID_a^r} = 0.378$ and $VP = 0.485$) are relatively inferior. Considering the excellent performance of Vanilla Gradients ($DL = 0.622$) in Distribution Learnability, we believe that the explanation distributions are relatively homogeneous, in line with the aforementioned "simple rule" trap. IxG, IG and DeepLift perform mediocrely both in Distributional Learnability ($DL = 0.383, 0.413$ and 0.411, respectively) and Variance Proximity ($VP = 0.512, 0.485$ and 0.491, respectively), and therefore we consider their explanations as slightly noisy, which interfere with the learning of the Autoencoder. In addition, we observe that the perturbation-based approaches also perform better in the assessment of Variance Proximity ($VP = 0.963$ for LIME and 1.016 for KernelSHAP). This is mainly attributed to the

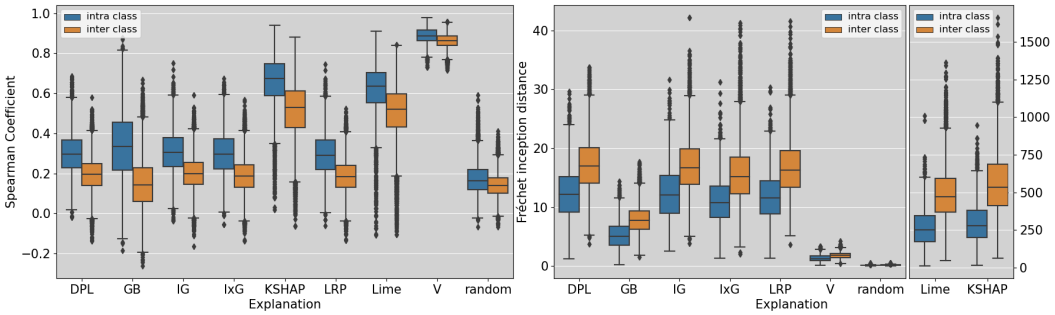

Figure 5.5.: The intra (blue) and inter (orange) class similarity (discrepancy) of the samples generated by Autoencoder based on the explanations of various explainability approaches. DPL, GB, IG IxG, KSHAP, and V denote DeepLift, Guided Backpropagation, Integrated Gradients, Input×Gradients, KernelSHAP, and Vanilla Gradients, respectively, and the y-axis is the Spearman coefficient (left) and Fréchet Inception Distance (right), respectively. The FIDs of the perturbation-based explanations are separated since they are not in the same order of magnitude as the rest.

randomness and complexity of the perturbation-based explanation generating process, which prevents Autoencoder from summarizing fixed and simplistic rules.

As a conclusion, we report the final score of the explainability approaches in terms of generalizability (S_{EM} in descending order): LIME and KSHAP (0.479), GB (0.330), V (0.241), LRP (0.203), DPL (0.201), IG (0.200) and IxG (0.196). For a clearer comparison, tabulated results for methods are presented in Table 5.2.

5.4.3. Smoothed vs. Unsmoothed Maps

SmoothGrad [77] is a technique that finds the gradient (or other method to obtain the explanations) after introducing noise to the original image multiple times and takes the average value. SmoothGrad-processed explanations are considered cleaner and more comprehensible. Interestingly, we found that SmoothGrad not only enhances visual consciousness, but also increases the generalizability of the explanations. We choose three gradient-based methods as baselines, namely Vanilla Gradients (V), Input×Gradients (IxG) and Integrated Gradients (IG), and implement SmoothGrad on them respectively.

As shown in Fig. 5.6, regarding the Distribution Learnability (DL), the SmoothGrad-applied versions all outperform the original baselines (The differences in DL ΔDL are 0.130, 0.041 and 0.031 for V, IxG and IG, respectively), which implies that explanations with SmoothGrad are more generalizable. Moreover, as Fig. 5.7 illustrated, SmoothGrad significantly strengthens the Variance Proximity (VP) of the explanations. The VPs of the three explainability methods increase by 0.295, 0.468 and 0.475 respectively. As a consequence, SmoothGrad raises the final scores of V, IxG, and IG from 0.241, 0.196, and 0.200 to 0.514, 0.415, and 0.426, respectively, which is consistent with human intuition: [77] concludes that SmoothGrad significantly reduces the noise in the saliency maps and

	V	V_s	$I \times G$	$I \times G\_s$	IG	IG_s
$\overline{TA} \uparrow$	0.541	**0.629**	**0.515**	0.488	**0.543**	0.508
$\overline{\rho_S} \uparrow$	0.727	**0.847**	0.232	**0.332**	0.254	**0.351**
$\overline{\rho_R} \uparrow$	0.596	**0.782**	0.403	**0.452**	0.441	**0.473**
$DL \uparrow$	0.622	**0.752**	0.383	**0.424**	0.413	**0.444**
$\Delta\overline{\rho_{S_a^r}} \uparrow$	0.025	**0.030**	0.111	**0.299**	0.107	**0.304**
$\Delta\overline{FID_a^r} \uparrow$	0.364	**0.654**	0.401	**0.681**	0.378	**0.656**
$VP \uparrow$	0.389	**0.684**	0.512	**0.980**	0.485	**0.960**
$S_{EM} \uparrow$	0.241	**0.514**	0.196	**0.415**	0.200	**0.426**

Table 5.3.: Detailed quantitative results of the proposed evaluation method for Vanilla Gradients, Input×Gradients, Integrated Gradients and their SmoothGrad versions.

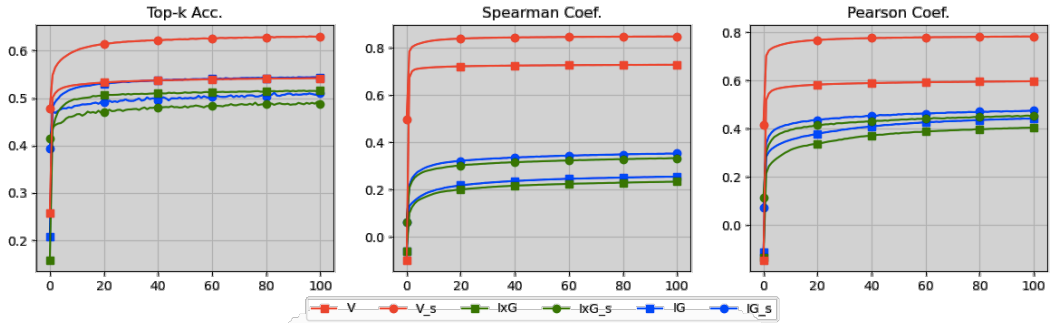

Figure 5.6.: The training curves of Vanilla Gradients (V), Input$times$Gradients (IxG), Integrated Gradients (IG) and their corresponding SmoothGrad versions (with suffix "_s"), respectively. The y-axis is the value of the corresponding metrics and the x-axis is the training epoch numbers.

thus augments their comprehensibility. We present the detailed quantitative results in Table 5.3.

5.5. Conclusion

This chapter provides a novel perspective for quantitatively evaluating explainability methods: generalizability. We argue that the distributions of good explanations should have clearer regularities and learnabilities, and possess distributional approximations and variations within and between classes. We demonstrate the evaluation of multiple explainability methods with the proposed approach.

In this part, we present all the works in this dissertation that address the challenges mentioned in Sec. 1.2.2, comprising four novel approaches that provide new insights for evaluating explainability methods from different perspectives. In the next part, we extend the applicability of the explainability approaches to point cloud models, addressing the research gaps mentioned in Sec. 1.3.2.

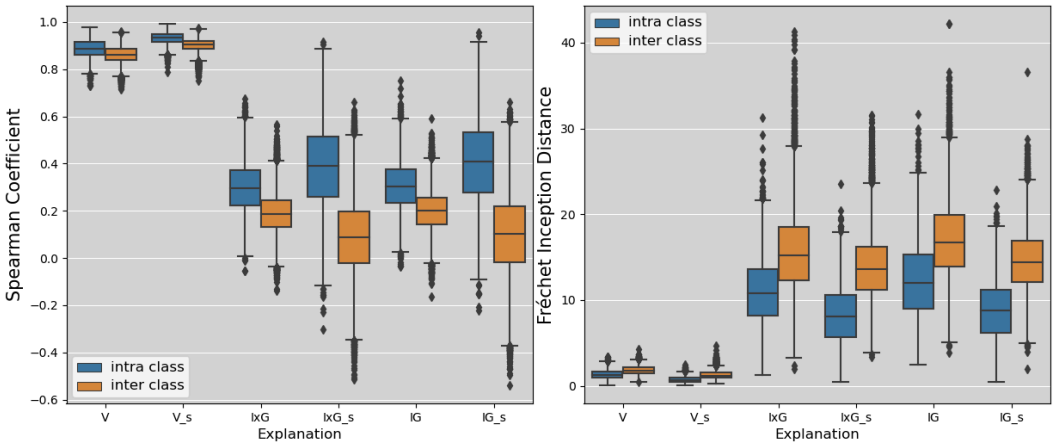

Figure 5.7.: The intra (blue) and inter (orange) class similarity (discrepancy) of the samples generated by Autoencoder based on the explanations of Vanilla Gradients (V), Input$times$Gradients (IxG), Integrated Gradients (IG) and their corresponding SmoothGrad versions (with suffix "_s"), respectively, and the y-axis is the Spearman coefficient (left) and Fréchet Inception Distance (right), respectively.

Part III.

Explainable applications on 3D vision

6. Surrogate Model-Based Explainability Methods for Point Cloud NNs

The preceding discussion has established the critical need for comprehensive and standardized evaluation methodologies to assess the fidelity, robustness, and trustworthiness of XAI techniques. While these evaluation methods provide the necessary theoretical and empirical foundation to verify the quality of an explanation, their true value is realized when applied to complex, high-stakes domains. Among the most challenging and rapidly evolving domains in machine learning is the analysis of 3D point cloud data. Unlike images or tabular data, the intrinsic properties of point clouds—such as unstructured nature, permutation invariance, and sparsity—present unique obstacles for both the application and subsequent evaluation of existing XAI methods. Therefore, building upon the principles and metrics of assessment developed in the first part of this work, the focus now shifts to practically addressing these application-specific challenges. The subsequent chapters will specifically delve into improving the effectiveness and relevance of XAI techniques for point cloud networks, thereby moving from the general task of assessing explainability to the specialized task of applying and optimizing it within the critical domain of 3D perception.

This chapter starts with a point cloud-applicable explainability approach to shed light on the rationale behind predictions made by point cloud models. The proposed method is based on perturbations and local surrogate models that reveal which points (sets) play an essential role in the prediction. Moreover, we propose quantitative fidelity validations for generated explanations that enhance the persuasive power of explainability and compare the plausibility of different existing point cloud-applicable explainability methods. Our new explainability approach provides a fairly accurate, more semantically coherent and widely applicable explanation for point cloud classification tasks.

6.1. INTRODUCTION

Point cloud data, as the raw data of most mainstream sensors, has a significant advantage in real-time scenarios compared to other 3D data formats and therefore has become a popular research direction in recent years. Point clouds exhibit higher structural complexity than 2D images. For instance, convolution kernels are easily applied to images due to their regularity, but they are not directly applicable to point clouds. Due to the lack of adjacency of the point cloud data, neighboring points in the point cloud matrix have a high probability of being irrelevant to the 3D spatial adjacency, which leads to the

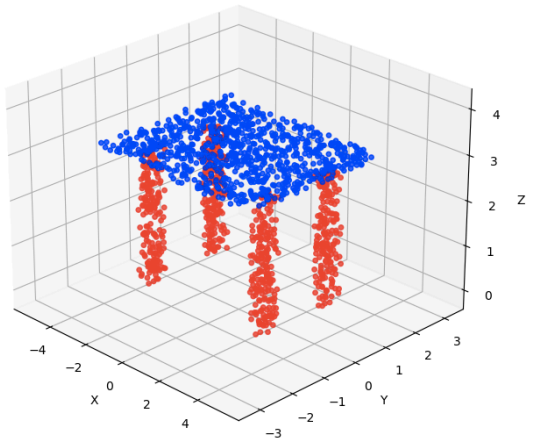

Figure 6.1.: A point cloud table example consisting of 1000 points.

invalidity of traditional convolution kernels. Specifically, a point cloud instance generally consists of hundreds or thousands of individual points, which can be represented as follows

$$P = \{p_1, p_2, ..., p_n\} \tag{6.1}$$

where, p_1 to p_n denote the n individual points that compose the instance. p_i is a one-dimensional vector whose dimensionality is normally determined by the number of features included, and it is expressed as

$$p_i = (x_i, y_i, z_i, r_i, g_i, b_i, f_1, f_2, ..., f_n) \tag{6.2}$$

where x_i, y_i and z_i are the spatial coordinate information, r_i, g_i and b_i denote the RGB values, and f_1 to f_n are the vectors representing other features. Overall, a point cloud instance can be represented by the following formula

$$P = \begin{bmatrix} x_1 & y_1 & z_1 & r_1 & g_1 & b_1 & f_{11} & f_{12} & \cdots & f_{1m} \\ x_2 & y_2 & z_2 & r_2 & g_2 & b_2 & f_{21} & f_{22} & \cdots & f_{2m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ x_n & y_n & z_n & r_n & g_n & b_n & f_{n1} & f_{n2} & \cdots & f_{nm} \end{bmatrix} \tag{6.3}$$

An example of a table composed of a point cloud is shown in Fig. 6.1. Remarkably, point cloud data possesses disorderliness: Swapping the order of any two points in the point cloud does not change the final spatial structure, i.e.,

$$P = \{p_1, p_j, p_k, ..., p_n\} \equiv \{p_1, p_k, p_j, , ..., p_n\} \tag{6.4}$$

Disorderliness distinguishes point clouds from image data in that they cannot extract localized information through convolutional kernels of size more than one.

Addressing this issue, [70], [71], [72] bring up solutions for point feature extraction and make point clouds suitable for convolutional neural networks. So far, most of the latest point cloud models follow the overall architecture [184]:

$$g_i = \mathcal{A}(\Phi(f_{i,j})|j = 1, \cdots, K) \tag{6.5}$$

where $\mathcal{A}$, Φ and f are global symmetric functions (various pooling layers), local feature extractor (MLP [69], [73], CNN [93], Residual [184], Attention [177], etc.) and local grouping function (MSG [73], Graphs [119], etc.) respectively.

However, in contrast to the field of 2D image processing with a large number of explainability studies [87], [123], [130], [157], most point cloud-compatible DNNs currently remain black-boxes due to the paucity of research investigating their working principles [138]. This indicates an indispensable need for the explainability research on DNNs dealing with point cloud data to ensure transparency of decisions made by robots and autonomous vehicles.

As for the reliability examination of explainability methods, to date, there is no acknowledged evaluation criterion. Most of the previous work validates the explanation results subjectively based on human interpretation, which easily leads to bias in the evaluation of explainability approaches. Therefore, quantitative evaluations are increasingly recognized as an essential requirement in the explainable machine learning domain.

This work proposes a point cloud-applicable *local surrogate model-based* approach [54], [65] investigating the explainability and reliability of point cloud neural networks. With the help of explanations, humans gain a better awareness of the underlying reasons for misclassification cases. Besides, we quantify the plausibility of the explanations for point cloud data through fidelity and accuracy verification methods instead of a subjective approach based on human interpretation. Our contributions are primarily summarized as follows:

- We propose a local surrogate model-based explainability approach for point cloud DNNs based on LIME [54], which is more widely applicable than gradient-based methods [138].
- We provide two quantitative evaluations for 3D explanations: fidelity metrics and cluster flipping, which are applied to validate the fidelity and plausibility of surrogate model-based and all 3D explainability approaches, respectively.
- We present quantitative comparisons of our proposed method with existing approaches for point cloud data using the proposed evaluation approach. Besides, we demonstrate an interesting viewpoint on the misclassified cases through our proposed approach.

The overall structure of this paper takes the form of five sections: In section 6.2, we introduce the outline of existing explainability methods and 3D neural networks, and the possibility of validating explainability approaches. Section 6.3 sets out details of our explainability approaches and the corresponding verification metrics. In section 6.4, we present the qualitative and quantitative results of our proposed methods. In section 6.5, we conclude a brief summary and suggest future research directions.

6.2. RELATED WORK

This section reviews the current widely used explainability approaches, summarizes the classical point cloud neural network, presents existing explainability methods for point cloud DNNs, and identifies the current possibilities for verifying the explainability approaches.

Explainability in 3D DNNs: Few studies have attempted to investigate the explainability of 3D DNNs. Although [160] refers to explainable point cloud classification, their work addresses the disorderly properties of point clouds using PointHop Units to adapt them to classical classifiers, which is part of the pre-processing rather than post-hoc explanations. [125] obtains point saliency maps by simply dropping points, which is not relevant to the explainability approaches. [138], the pioneer study of utilizing explainability approaches to point clouds remains crucial to our understanding of feature sparsity of 3D models. However, they only show sparse explanations that emphasize the importance of points at edges and corners, which is lack-of-semantics, and the evaluation criterion of the explanations is absent. In addition, the gradient-based methods are not adapted to models without gradients, such as tree-based models. In contrast, local surrogate model-based approaches are completely model-agnostic.

Explanation plausibility verification: Although there are many studies in the literature on the outcome of explainability methods, an acknowledged quantitative assessment for those approaches is absent [133] because explanations are subjective to humans. [129] argues that a feasible explanation should be sensitive to the weights of models and the data generating process, and proposed an alternative evaluation approach by randomizing the network weights as well as the labels and inspecting the sensitivity of the saliency maps. However, this approach tends to only benefit the gradient-based explainability methods and validates invalidity instead of feasibility. [137] strive to observe the improvement of the core performance of the network and the confidence they can generate for the users of the system when processing image data. [35], [74], [112] propose an intuitive and efficient pattern to verify the explanations by flipping the pixels that contribute positively or negatively (or approximately zero) to a particular class and record the verified prediction scores. Nevertheless, the flipping operation of this method could be optimized to some extent while processing point cloud data, which we will discuss in section 6.3.

6.3. EXPLAINABILITY APPROACHES FOR POINT CLOUDS

A significant advantage of surrogate model-based methods is that they are more widely applicable. In this section, we describe in detail our explainability approach, i.e. local surrogate model-based method for point cloud data based on LIME [54]. In addition, we elaborate the quantitative evaluation approach for point cloud explanations, which consider the local fidelity and plausibility of existing point cloud explainability methods. Note that as the surrogate model-based explainability approach is model-independent, the evaluation metrics are suitable for other data types besides point clouds as well.

6.3.1. Local surrogate model-based explainability approaches for Point Clouds

Local surrogate model-based explainability approaches aim to generate an explanation for a classifier f and a specific instance x from the data set X. To apply these methods to point cloud data, some pre-processing is necessary.

Algorithm 3: Pre-processing of Local surrogate model-based methods for point clouds

Input: $P \rightarrow N \times D$ point clouds
$n_c \rightarrow$ number of clusters
$maxIter \rightarrow$ Max iterations
Output: $C \rightarrow 1 \times N$ matrix ; `// Output indicates which cluster each point belongs to`
Function `3D K-Means with FPS(`$P, n_c, maxIter$`)`:
    `// Sample` n_c `points from` P `using FPS`
    $Centers \leftarrow$ FPS(c from P);
    `// Find the nearest cluster center for each point`
    while $maxIter$ **do**
        for i *in* n_c **do**
            $EDMatrix \leftarrow \|P, Centers\|_2$;
        end
        $minDis \leftarrow \arg\min(EDMatrix)$;
        `// Point belongs to the nearest cluster, update the centers`
        for j *in* n_c **do**
            $P[C_j] \leftarrow$ Where $minDis == j$;
            $newCenters \leftarrow$ Mean($P[C_j]$);
        end
        $Centers \leftarrow newCenters$;
    end
    return C
end

Pre-processing with FPS

For explaining a point cloud input with size P utilizing local surrogate model-based explainability approaches, each point $p \in P$ is considered as a feature individually. However, to avoid explosive computational complexity and to organize the disordered point cloud data, we group the points into super-points C as features to be perturbed. We initialize a user-defined parameter n_c, indicating the number of clusters. To ensure uniformity and strengthen semantics we employ Farthest Point Sampling (FPS) to select n_c points from P and group all p according to spatial coordinates using 3D K-Means Clustering such that $\forall p \in P : p \in C_i$. The pseudo-code is presented in Algorithm 3.

Vanilla LIME applied for point clouds

Same as processing 2D images [54], LIME for point cloud data also satisfies the following constraint:

$$\xi(x) = \underset{g \in G}{argmin}\ L(f, g, \pi_x) + \Omega(g) \tag{6.6}$$

where f and g denote the classifier and the explainable model for a local instance x respectively, π_x denotes the proximity measure between samples z to the input x (locality around x), and $\Omega(g)$ denotes the complexity of the explainable model. LIME tries to minimize the locality-dependent loss $argminL(f, g, \pi_x)$ by approximating g to f. It takes samples z around x and feeds the perturbed samples z' into f to obtain a faithful surrogate model g that approximates f, also, it regularizes the complexity of the surrogate model g to guarantee that it is still explainable to humans.

As with most 2D image datasets, our 3D dataset for experiments has 40 different label categories, in which explainability is hardly guaranteed even for linear surrogate classifiers. Therefore, we train a linear regressor that approximates the prediction score of the corresponding category from the neural network. We sample $z \in Z$ by randomly flipping component clusters from x and feed the perturbed samples Z into the regressor g to obtain the predictions $g(z)$. To minimize $L(f, g, \pi_x)$, a kernel filters the generated samples Z around x based on the similarity between z and x proportionally (the fewer clusters being flipped the higher the weight). The surrogate model is subsequently trained with the weighted samples Z using linear regression. Due to the simplicity and transparency of linear models, it is explainable and understandable to humans and intuitive as to which parts (clusters) have positive/negative attributions to a particular prediction according to the parameters of the surrogate linear regressor.

Variable input size flipping (VISF)

LIME generates adjacent perturbation samples by flipping the corresponding clusters of the original instance. There are three widely-used flipping methods for a target cluster regarding 2D images: zero clearing all the included pixels, replacing those pixels with the average of the selected cluster (or the whole image), or reversing the sign of their

coordinates. However, we argue that the above three operations can barely eliminate all information of the target cluster. For instance, although all pixel values are zeroed out, the contours formed by zeros remain on the data matrix and may still be learned by the neural networks. For a point cloud instance $P \in \mathbb{R}^{N\times D}$, the pixel values represent 3D spatial coordinates, and the above alternatives are likely to form a highly overlapping point set, resulting in uncertainty to determine whether the prediction fluctuations of the neural network are merely caused by flipping operations.

To address the above problem, we simply discard the points contained in the target cluster c_i from the original instance as a means to completely ablate the information of the target cluster, i.e. $s_i = P \backslash c_i \in \mathbb{R}^{(N-\|c_i\|)\times D}$. This approach is only applicable for point cloud neural networks. Recall the architecture of point cloud networks, where the final symmetric function (i.e. the max-pooling layer) is used to extract the *global* feature from disordered point clouds while the *local* features in the lower layer are weighted by numerous 1×1 convolutional kernels, which allows the input size of the network to be arbitrarily reshaped without obstructing the inference. Notably, the Variable Input Size Flipping (VISF) is both extendable in explaining and verification process (section 6.3.2).

Attribution summarizing

For explainability methods that return the importance of each spatial coordinate axis, the most popular and intuitive attribution summarization process is to simply sum them up:

$$C_p = \sum (C_1, C_2, C_3) \tag{6.7}$$

Where $C_{1\sim3}$ stand for the attributions in each of the three spatial axes. Different summarizing patterns have varied impacts on the explanations, which is worthy of further exploration.

6.3.2. Plausibility verification for 3D explanations

Local fidelity metrics

The fidelity indicates the prediction coherence between the original black-box model and the surrogate one, which is formulated as:

$$Fid = \frac{\sum \mathbb{1}\left(f(Z) = g(Z)\right)}{\|Z\|} \tag{6.8}$$

Nevertheless, instead of a classifier we utilize a linear regressor as the surrogate model, which returns the prediction score only associated with the predicted class. We thus compare the batched similarity between regression scores $g(Z) \in \mathbb{R}^{|Z|}$ and the prediction scores of the corresponding logits unit of the network $f(Z)$ via several loss and coefficient measurements:

- Mean loss: $L_m = \left| \sum_i^{\|Z\|} (\frac{f(Z)_i}{\|Z\|}) - \sum_i^{\|Z\|} (\frac{g(Z)_i}{\|Z\|}) \right|$
- Mean L_1 and L_2 loss: $L_1 = \sum_i^{\|Z\|} (\frac{|f(Z) - g(Z)|}{\|Z\|})$

  and $L_2 = \sum_i^{\|Z\|} (\frac{(f(Z)_i - g(Z)_i)^2}{\|Z\|})$
- Weighted L_1 and L_2 loss:

  $L_1^\omega = \sum_i^{\|Z\|} (\frac{|f(Z)_i - g(Z)_i| \cdot \omega}{\|Z\|})$

  and $L_2^\omega = \sum_i^{\|Z\|} (\frac{(f(Z)_i - g(Z)_i)^2 \cdot \omega}{\|Z\|})$
- Weighted coefficient of determination:
  $R_\omega^2 = 1 - \frac{\sum_i^{\|Z\|} (f(Z)_i - g(Z)_i)^2}{\sum_i^{\|Z\|} (f(Z)_i - \overline{f_\omega(Z)})^2}$
- Weighted adjusted coefficient of determination:

  $\hat{R}_\omega^2 = 1 - (1 - R_\omega^2) \left[\frac{\|Z\| - 1}{\|Z\| - \|g\| - 1} \right]$

where ω indicates the weights derived from the kernel, $\|Z\|$ denotes the number of observed samples, $\|g\|$ is the number of parameters of g and $\overline{f_\omega(Z)}$ indicates the weighted average. L_m,$L_1^{(\omega)}$ and $L_2^{(\omega)}$ measure the discrepancy in predicted scores while R_ω^2 indicates the correlation between the prediction scores of the proxy model and the output of the neural network. In general, better agent approximations possess lower loss and higher decision coefficients with the predictions of neural networks. However, R_ω^2 is sensitive to the number of samples and prone to positive bias under a small sample size [16]. We therefore introduce $\hat{R}_\omega^2$, which takes into account the size of variables and samples. Note that $\hat{R}_\omega^2$ has the meaningful range between $(-\infty, 1]$ under the assumption that $\|Z\| > \|g\|$, while the opposite case may exist in our experiments, it is therefore only referable for the case $\|Z\| > \|g\|$ in the experiment.

Method-independent explanation verification

Fidelity metrics are only suitable for surrogate model approaches. Additional measuring methodologies are required for the reliability of non-surrogate-based explainability methods such as gradient-based saliency maps. According to the hypothesis of the local accuracy of additive feature attribution [65]:

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x_i \tag{6.9}$$

the output of original model $f(x)$ is composed by linear summation of the individual feature attributions ϕ_i. One of the most intuitive ways to verify an explainability approach is

to eliminate features with certain attributions ϕ_i (normally positive or negative) according to their generated explanations and observe whether the output of the model $f(x)$ exhibits corresponding variations:

$$f\left(x\right) - f\left(x \backslash i\right) \begin{cases} \geqslant 0 & \text{if } \phi_i x_i \geqslant 0 \\ \leqslant 0 & \text{if } \phi_i x_i \leqslant 0 \end{cases}, i \in M \tag{6.10}$$

where $\phi_i x_i$ denotes the attribution of flipped feature and $f\left(x \backslash i\right)$ denotes the output of the model after flipping feature i.

Nevertheless, in point cloud DNNs, the sensitivity of prediction scores for batch data is difficult to observe quantitatively due to the unevenly distributed prediction scores (the logits before the softmax) from different instances. Therefore, we normalize the variability of the predicted scores to facilitate its presentation in the form of an average prediction scoreline, which is formulated as

$$S_{avg} = \frac{1}{n} \sum_{i=0}^{n} \frac{S_i - S_{i_{min}}}{S_{i_{max}}} \tag{6.11}$$

where S_i denotes each score in the ith test (including positive, negative and random perturbation series), $S_{i_{min}}$ and $S_{i_{max}}$ denote the minimum and maximum values in the corresponding evaluation run respectively.

In addition, due to the use of clustered points, we determine the averaged attributions of clusters $\phi_c x_c$ in our work rather than of individual points $\phi_i c_i$, where

$$\phi_c x_c = \sum_{i=1}^{c} \phi_i x_i \tag{6.12}$$

, which lead to fluctuations in the prediction scores. The issue can be alleviated by increasing the number of clusters. We discuss this further in Section 6.4.

To quantitatively compare the plausibility among all types of explainability methods, we record the prediction scores of flipping the positive, negative and random contributing clusters respectively, denoted as S_{pos}, S_{neg} and S_{rdm}. The plausibility of the corresponding explanation can be formulated as:

$$\bar{p} = -\frac{\sum_{i}^{\|Z\|} \rho_{S_i}(S_{pos} - S_{rdm}, S_{neg} - S_{rdm})}{\|Z\|} \tag{6.13}$$

where $\rho_S(a, b)$ denotes the correlation coefficient between a and b. Intuitively, flipping positive clusters results in a decline of predicted scores while flipping negative clusters lifts them up. Flipping random clusters represents the impact of eliminating neutral clusters independent of attributions, as randomly selected clusters may consist of both positive and negative points and are therefore considered indifferent. $S_{pos} - S_{rdm}$ and

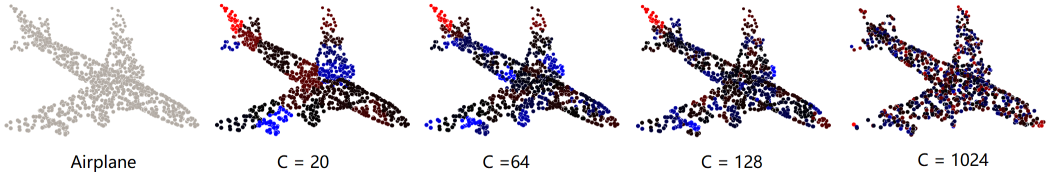

Figure 6.2.: Examples of explanations with 1000 perturbation samples. C denotes the number of clusters. Brighter red points represent more positive contributions and, conversely, brighter blue points represent more negative contributions and dim points indicate zero contributions to the corresponding classification labels.

$S_{neg} - S_{rdm}$ are then approximations of unbiased attribution-flipping processes. A plausible explanation should have exactly opposite sensitivities to contrary attributions, and therefore its correlation coefficient of the prediction score series is expected to be as small as possible, i.e., a high score of $\bar{p}$. We consider this value as a succinct description of the plausibility of the explainability method.

6.4. EXPERIMENT

In this section, we present the qualitative results of 3D surrogate model-based explainability methods (Sec. 6.4.1), evaluate and compare it with other 3D-applicable approaches utilizing the quantitative verifications proposed in section 6.3.2 (Sec. 6.4.2) and show how the explanations help to analyse the samples classified incorrectly by the classifier. (Sec. 6.4.3). In our experiments[1], 1000 test instances are selected from Modelnet40 [44], which contains 12311 CAD models in 40 common categories and is currently the most widely-applied point cloud classification data set. We choose PointNet [70] as the model to be explained, which achieves an overall accuracy of 89.2% on Modelnet40. We sample 1024 points from each instance as input to the network. Additionally, we choose Exponential Smoothing kernel for training linear regressor, denoted as

$$K = \sqrt{e^{\frac{-d^2}{w^2}}} \tag{6.14}$$

where d denotes the distance from samples to the instances to be explained, and w denotes the kernel width which has an impact on the explanations. We therefore conduct a sensitivity experiment of the kernel widths from 0.05 to 0.3.

6.4.1. Qualitative explanation visualisation

Examples of explanations generated by PointNet, as well as their original point cloud structures are shown in Figure 6.2. What stands out in the figure is that explanations with different C are consistent overall except the one with $C = 1024$. We believe that the reason is that 1000 samples are insufficient for such a large number of clusters (1024)

[1] Our code is available at `https://github.com/Explain3D/LIME-3D`

and therefore the surrogate model is not well-trained. Explanations based on clusters suffer from contribution neutralization. A cluster may consist of positive and negative contributing points simultaneously, aggregating them as an entity obscures the individual contribution of each point ($C = 20$,64 and 128). The neutralization can be alleviated by increasing the number of clusters, with the side effect of requiring more training samples and processing time ($C = 1024$).

6.4.2. Quantitative verification of explanation plausibility

Assessing the explanations by intuition is not quantitatively verifiable and is vulnerable to bias. This section mainly demonstrates the results of plausibility verification experiments i.e. local fidelity metrics in subsection 6.4.2 and the method-independent verification approach in subsection 6.4.2. There are two hyper-parameters for the proposed explainability method: Number of clusters C and number of perturbation samples S. In this section, we choose $C = 128$ and $S = 10^3$ as the standard performance of the proposed explainability method, since $C = 128$ is experimentally proven to achieve the best quantitative performance while maintaining the qualitative semantics. $S = 10^3$ generates high-qualified explanations within an acceptable processing time and thus is considered as the best configuration.

Local fidelity metrics

Local fidelity metrics address measuring the prediction similarity between the original black-box model and the surrogate one, which play a pivotal role in verifying the plausibility of local surrogate model-based explainability methods. Due to the absence of related results as a reference, we treat the unmodified LIME (hard transplanted to point clouds) as the baseline. Table 6.1 compares the local fidelity of different explaining mechanisms, i.e. whether Farthest Points Sampling (FPS) is used or whether Variable Input Size Flipping (VISF) is employed. Corresponding metric symbols refer to section 6.3.2. According to the results, both FPS and VISF facilitate the improvement of local fidelity compared to the vanilla 3D LIME (baseline) as our LIME(FPS + VISF) improvement outperforms others in terms of most fidelity metrics. Note that the local fidelity only measures how closely the surrogate model approximates the black-box model. One drawback of the metric is that it is only applicable to explainability methods based on local surrogate models. Popular explainability methods (already proposed for point clouds) such as gradient-based ones are not compatible with these metrics, which confuses the user in choosing the most appropriate explainability method for specific tasks.

Method-independent plausibility verification

To address the aforementioned drawback we instead compare all existing point cloud-applicable explainability approaches utilizing the method-independent verification proposed in section 6.3.2. Again, we set $C = 128$ and $S = 10^3$ as the "competitor" of our

	L_m	L_1	L_1^ω	L_2	L_2^ω	R_ω^2	$\tilde{R}_\omega^2$
LIME (baseline)	1.40×10^{-2}	1.11×10^{-1}	8.66×10^{-2}	1.06×10^{-1}	6.53×10^{-2}	0.338	0.241
LIME (FPS)	1.22×10^{-2}	9.80×10^{-2}	7.66×10^{-2}	8.67×10^{-2}	5.35×10^{-2}	**0.353**	**0.257**
LIME (VISF)	1.18×10^{-2}	1.01×10^{-1}	7.90×10^{-2}	9.68×10^{-2}	5.95×10^{-2}	0.335	0.237
LIME (FPS + VISF)	$\mathbf{1.03 \times 10^{-2}}$	$\mathbf{8.89 \times 10^{-2}}$	$\mathbf{6.95 \times 10^{-2}}$	$\mathbf{7.84 \times 10^{-2}}$	$\mathbf{4.82 \times 10^{-2}}$	0.346	0.249

Table 6.1.: Local fidelities of different explaining mechanics for point cloud data, where FPS denotes employing Farthest Point Sampling instead of randomly choose clusters and VISF denotes the Variable input size flipping mechanism. The unmodified application of LIME to point clouds is regarded as the baseline.

	$\bar{p}_{.15}$	$\bar{p}_{.3}$	$\bar{p}_{.5}$
Vanilla Gradients	-0.574	-0.569	-0.672
Guided Back-propagation	-0.741	-0.695	-0.623
Integrated Gradients	0.484	0.366	0.236
KernelSHAP	-0.205	-0.257	-0.256
LIME (FPS + VISF)	**0.622**	**0.531**	**0.372**

Table 6.2.: Plausibility $\bar{p}$ of flipping top-%15,%30 and %50 attributed points.

proposed method. Besides positive and negative attributions, we also flip the same percentage of randomly-selected points as the baseline of prediction scores.

Figure 6.3 and Table 6.2 depict the trends of prediction scores and the correlation coefficient $\bar{p}$ between different existing 3D-applicable explainability methods. As the gradient-based approaches yield individual attributions for each point, we calculate coefficients for different percentages of points for fairness, i.e. top-%15,%30 and %50 positive ones. What stands out in the results is that the explanations generated by 3D LIME and Integrated Gradients behave robustly. Their average prediction scores deteriorated rapidly after the gradual flipping of the most positive contribution points and conversely tended to increase when the negative contribution points are flipped.

On the other hand, Vanilla Gradients, Guided Back-propagation and 3D KernelSHAP are unable to distinguish between points with different contributions, resulting in gradient maps being less uniform than Integrated Gradients [138]. Interestingly, KernelSHAP is a variant of LIME based on Shapley value, differing from the latter solely in the choice of kernels. KernelSHAP assigns high weights to perturbation samples with only a minority of clusters remained, which severely impairs the global structure of the instance. Empirically, we find that such kernels may be more suitable for black-box model structures on other data types, but with limited performance in explaining point clouds.

We also compare the plausibility among all explanation mechanisms, the corresponding scores are presented in table 6.3. The proposed method also dominates which is consistent with the results in section 6.4.2.

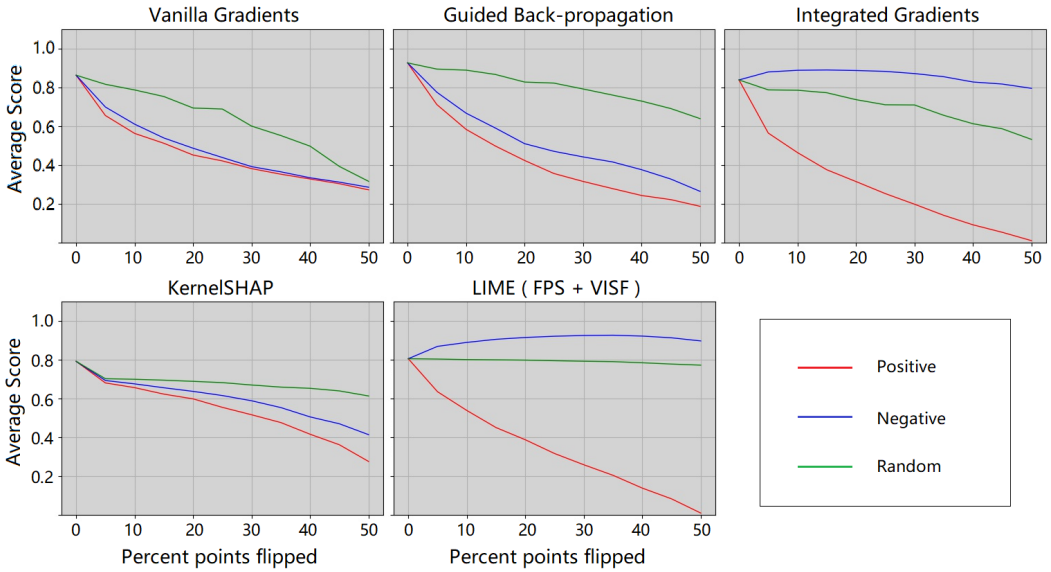

Figure 6.3.: Variation trends of the prediction scores (y-axis) by flipping and re-inference. The scores are the average of normalized prediction scores of 1000 test instances. Red and blue lines indicate the trend of flipping positive and negative contribution points, respectively, the green line indicates flipping random points that are independent of contribution. The x-axis indicates the percentage of flipped points for a given instance.

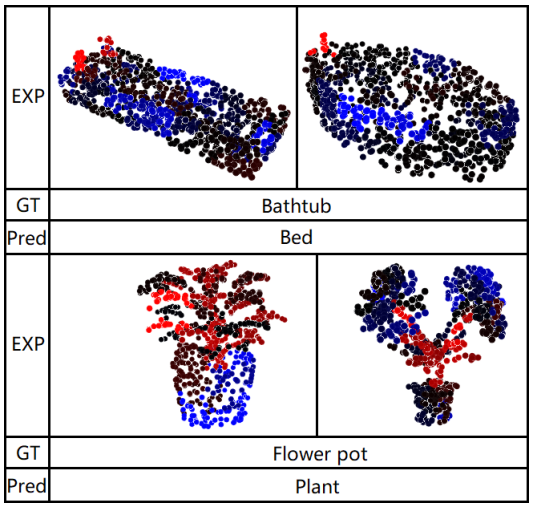

Figure 6.4.: Explanation of the misclassified examples. Brighter red points indicate more positive contributions, while brighter blue points indicate more negative contributions and dim points indicate zero contributions. All contributions are concerning the prediction class (wrong class instead of the ground truth).

	$\bar{p}_{.15}$	$\bar{p}_{.3}$	$\bar{p}_{.5}$
LIME (baseline)	0.598	0.520	0.341
LIME (FPS)	0.615	0.513	0.361
LIME (VISF)	0.593	0.514	0.345
LIME (FPS + VISF)	**0.622**	**0.531**	**0.372**

Table 6.3.: Plausibility of different explaining mechanics for enhancing the explanation quality.

6.4.3. Applying local surrogate model-based explainability methods for failure analysis

A potentially applicable prospect of local surrogate model-based explainability methods is failure analysis. This analysis has important implications for understanding the erroneous attention paid by the classifier and provides opportunity for further research, e.g. 3D model revision. Figure 6.4 shows examples of the attributions to the misclassified instances.

As can be seen from Figure 6.4, a majority of the misclassifications were caused by misdirected attention. In the first two examples, the faucet instead of the bathtub itself possesses the most positive attribution, misleading the model to neglect the structure of the primary target object. We believe this is because only a tiny fraction of the bathtubs in the training data is accompanied by a faucet. Another similar type of error frequently occurs with the class "Flower pot". The major attention of the model is drawn to the plant above rather than the pot below, resulting in a prediction of 'Plant' instead of "Flower pot" (the pot even draws a negative contribution to the ground truth label). From a human perspective, this type of data is ambiguously labeled, as both classes "Plant" and "Flower pot" are reasonable ground truth. Towards a more accurate model, this labeling type should be avoided whenever possible.

6.5. CONCLUSION

Although point cloud neural networks have received critical attention in recent years, so far, there have been few studies on their explainability. This chapter proposes an explainability approach for point clouds based on LIME [54]. We also provided the possibility to quantitatively validate the point cloud explanations. We evaluated and compared the performance of our approach against different existing explainability methods for point cloud data. The evaluation comparison revealed that our local surrogate model-based approaches as well as Integrated Gradients yield relatively plausible explanations and outperform other methods such as Guided Back-propagation. Our results also demonstrated that a larger amount of clusters and more perturbed samples are required to avoid compromising fidelity, which however consumes more processing time. Moreover, we provided intuitive analyses for misclassified samples by utilizing the proposed method. The analyses showed that part of the misclassified cases can be attributed to the anomalous

structural distributions or ambiguous labels of the input data, misdirecting the attention of the classifier.

The approach proposed in this chapter is local and applies only to specific inputs and predictions. In the next two chapters, we present two different global explainability methods for point cloud models based on a technique for visualizing image latent features, named Activation Maximization (AM), which generate explanations that are representative of a particular class of objects across the entire dataset.

7. Visualizing Global Explanations of Point Cloud DNNs

In the previous chapter we introduced a local explainability method which is applicable to point clouds. Local approaches, while providing detailed explanations for particular predictions, are unable to visualize the decision principles of the model over the entire dataset. Starting from this chapter, we propose two explainability methods for visualizing the global explanations of point clouds, which produce intuitive, representative and diverse global instances leveraging generative models to help users understand the contours of representative objects learned by the classifiers. In this chapter, we first 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.

7.1. Introduction

The disorderliness and sparsity of point clouds make it difficult to directly apply 2D XAI methods (as detailed in Chapter 6), and to date, no global explainability methods applicable for point clouds have been proposed. In this chapter, we propose an autoencoder-based global explanation generation method for point clouds. Specifically, we strive to investigate the global explanations 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 7.1), 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: PCAMS, 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 7.2 introduces explainability methods for point clouds, especially AM and corresponding evaluation methods. Section 7.3 provides the proposed generative AM approaches for point clouds as well as a more persuasive evaluation metric. Section 7.4 demonstrates our experimental results and we summarize our work in Section 7.5.

7.2. 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.

Activation Maximization (AM): AM is a high-level feature visualization technique that was first proposed by [25]. 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 [39]. Optimization constrains, such as L_2-norm [28], Gaussian blur [46], Total Variation [50] or priors, such as average image initialization [53] and patch dataset [38], [43], 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. [52], [81], [110], [168] utilized auto-encoders and GANs to produce high quality AM images. [66] proposed Plug & Play embedding generative networks that simultaneously address the high-resolution and diversity of synthesized AM images. Additionally, [122] 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. [66] accessed the definition and diversity of AM images via Inception Score (IS) [55]. [110] incorporated Fréchet inception distance (FID) [64] to estimate the similarity between generated AM examples and real instances in latent spaces. AM score, another evaluation metric proposed by [81], 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. [125] traced the critical points to generate saliency maps of the point cloud network by dropping points. [139] was the first work to incorporate explainability methods, who started an observation of the intrinsic feature of point cloud networks via IG. A follow-on study was conducted by [1], 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.

7.3. Methods

In this section, we demonstrate our AM approach for point clouds (Section 7.3.1) as well as the proposed evaluation metric for point cloud AM (Section 7.3.2).

7.3.1. 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 [47] or by training a global surrogate model [185]. 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 [185]. 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 [116]. To visualize such an activation in DNNs, [25] proposed the AM, which is formulated as:

$$x^* = \underset{x}{\operatorname{argmax}} \left(a_i^l(\theta, x)\right) \tag{7.1}$$

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 [39]. Several studies have investigated regularizing AM examples with non-generative priors, such as L_2 Norm, Gaussian blur and Total variation [28], [46], [50]. 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 7.1). We believe that on the one hand, the

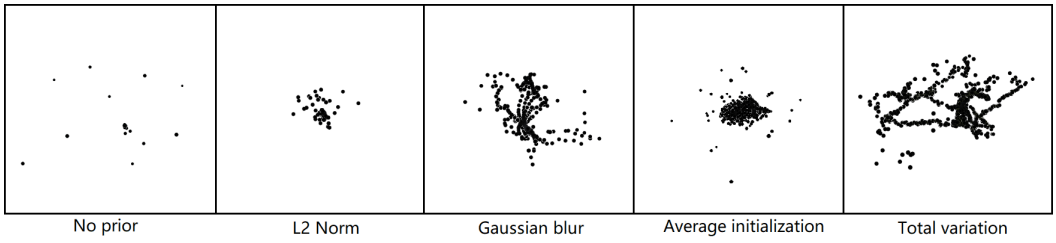

Figure 7.1.: AM for point clouds without generative priors (class "car"). Due to the specific architecture of the point cloud network, traditional regularization priors (for 2D images) are incapable of generating human-perceivable global explanations.

features of point cloud networks are comparatively sparse and the global structure information of instances is seriously impaired [139], 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 7.1) 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. 7.2. 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_{PC} 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 $F_{PC}(P) \in \mathbb{R}^{N_c \times 1}$, where N_c denotes the number of classes. Our goal is to build a module which outputs a P_g that $\text{argmax} F^i_{PC}(P_g)$, and $O_g \sim P_x$, where $F^i_{PC}(P_g)$ denotes the i^{th} activation of the logits and P_x denotes the real instances from dataset. 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 [82] 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 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 $P_g \sim P_x$. When training, we mea-

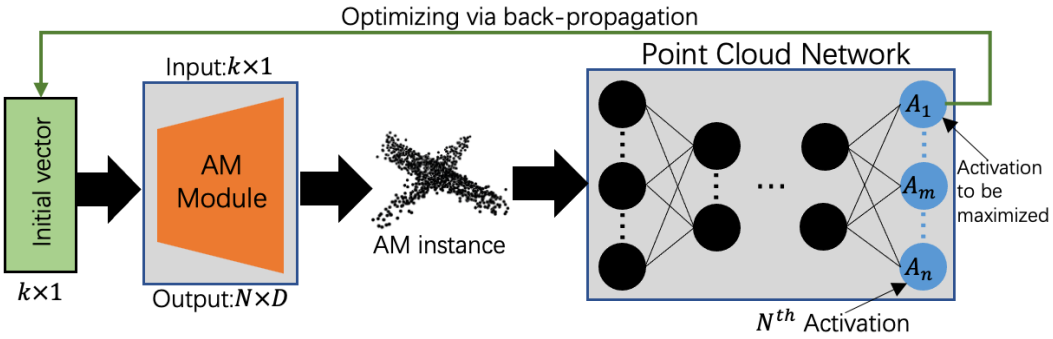

Figure 7.2.: General overview of the architecture for point cloud AM. The green and gray bars represent vectors and networks, respectively. In the point cloud network, the black and blue circles represent the neurons in the middle layer and the last layer (the activations), respectively. The thick black arrows and thin green arrows represent forward inference and backward propagation, respectively.

sure the gap between the generated examples and the original data with Unidirectional Chamfer Distance (CD) loss D^1_{CH}, which is formulated as:

$$L_{g_{AE}} = D^1_{CH}(P_x, P_g) = \frac{1}{|P_g|} \sum_{p_m \in P_x} \min_{p_n \in P_g} \|p_m - p_n\|_2 \tag{7.2}$$

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 [164]. 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 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 $L^d_{D_{AED}}$ with D_c for the discriminator, which is formulated as:

$$L^d_{D_{AED}} = D_c(P_g) - D_c(P_x) \tag{7.3}$$

Note that the value domain of $L^d_{D_{AED}}$ is $[-1, 1]$. We observe that since the performance of D_c easily outperforms g_{AED} ($L^d_{D_{AED}} \to -1$) during training, the latter struggles to be further optimized [61]. We therefore train only one of them alternately for each batch: If $L^d_{D_{AED}} < 0$, we train the g_{AED} only and vice versa. Furthermore, if the discriminator is overperforming ($L^d_{D_{AED}} < -0.75$), we add Gaussian noise to its parameters to disrupt the performance.

The latent distance loss 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:

$$L_F = \frac{1}{|V_g^c|} \sum_{v_m \in V_x^c} \min_{V_n \in V_g^c} \|v_m - v_n\| \tag{7.4}$$

where V_x^c and V_g^c represent the output of the second convolutional layer in the encoder, computed with real instances and generated examples, respectively. L_F can be regarded as the CD computed on the latent space.

The final generative loss of AED is formulated as:

$$L_{g_{AED}} = L_{g_{AE}} + w_F L_F - w_D L^g_{D_{AED}} \tag{7.5}$$

where w_F, w_D are the corresponding weights and $L^g_{D_{AED}}$ denotes the loss for the generator to deceive the discriminator, which is $-D_c(P_g)$.

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. 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 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. 7.3, 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:

$$L_{g_{NAED}} = (L_{g_{AE}} + w_F L_F + w_{F2} L_{F2}) - w_D L^g_{D_{NAED}} \tag{7.6}$$

where w_{F2} is the weights of L_{F2} and $L^g_{D_{NAED}}$ is the discriminative loss of NAED and is calculated identically as $L^g_{D_{AED}}$.

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 $L_{D_{(N)AED}}$, which forces the generated examples to be close to real ones (No $L_{D_{AE}}$ exists for AE, but for fairness, we repeat this encoding and decoding process as well). We then

obtain the target activation value $F^t_{PC}(P'_{ini})$ and optimize V_{ini} via back-propagation. The general term of the AM optimization loss is:

$$-(F^t_{PC}(P'_{ini}) + L_{D_{(N))AED}}) \tag{7.7}$$

Moreover, inspired by [53], 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.

7.3.2. 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 [55] or AM Score [81], 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 [154] which compared the global features from the PointNet architecture. Nonetheless, we observe that the metric is vulnerable for AM (see table 7.1: 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.

Point Cloud-Activation Maximization Score (PCAMS): Targeting the limitations of the aforementioned methods, we propose a composite AM evaluation metric: PCAMS. Our PCAMS is formulated as:

$$PCAMS = IS_m - \frac{(log(FID_{PN}) + log(D^1_{CH}(P_g, P_i))}{2} \tag{7.8}$$

IS_m denotes the modified Inception Score (m-IS) [63], which is formulated as:

$$IS_m = e^{\mathbb{E}_{P_i}[\mathbb{E}_{P_j}[(\mathbb{KL}(p(y|P_i)||p(y|P_j))]]} \tag{7.9}$$

where P_i and P_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:

$$FID_{PN} = \|\mu_r - \mu_g\|^2 + Tr(\sigma_r + \sigma_g - 2(\sigma_r\sigma_g)^{\frac{1}{2}}) \tag{7.10}$$

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 [154], 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:

$$CD = D^1_{CH}(P_g, P_i) = \frac{1}{|P_g|} \sum_{p_m \in P_g} \min_{p_n \in P_i} \|p_m - p_n\|_2 \tag{7.11}$$

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 PCAMS is $[-\infty, N_c]$. In addition, we introduce another point-wise distance used for comparison in Sec 7.4: Earth Mover's Distance (EMD), which is formulated as:

$$EMD(P_g, P_i) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} Pr_{i,j} d_{i,j}}{\sum_{i=1}^{m}\sum_{j=1}^{n} Pr_{i,j}} \tag{7.12}$$

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

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

7.4. Experiments

In this section, we qualitatively demonstrate the generated examples of our proposed point cloud-applicable AM (section 7.4.1), and show the quantitative evaluations of existing point cloud AM approaches (section 7.4.2). Additionally, we also provide an example of application scenes of proposed methods for prediction examination in section 7.4.3. In our experiments, we choose ModelNet40 [44] 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 [36], 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++ [73] and DGCNN [119]. 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.

7.4.1. Point Cloud AM Visualization

Perceptibility: Figure 7.3 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 (L_2, Gaussian blur and Total variation) also fail to yield globally

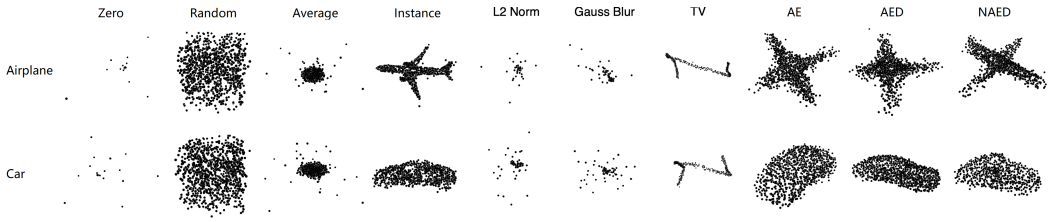

Figure 7.3.: AM results of different approaches. From left to right: Zero initialization, random initialization, initialized with the average of the test data per class, initialized from a specific instance, regularization with L_2 Norm, Gaussian Blur and Total Variation, and our proposed AE, AED and 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.

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

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

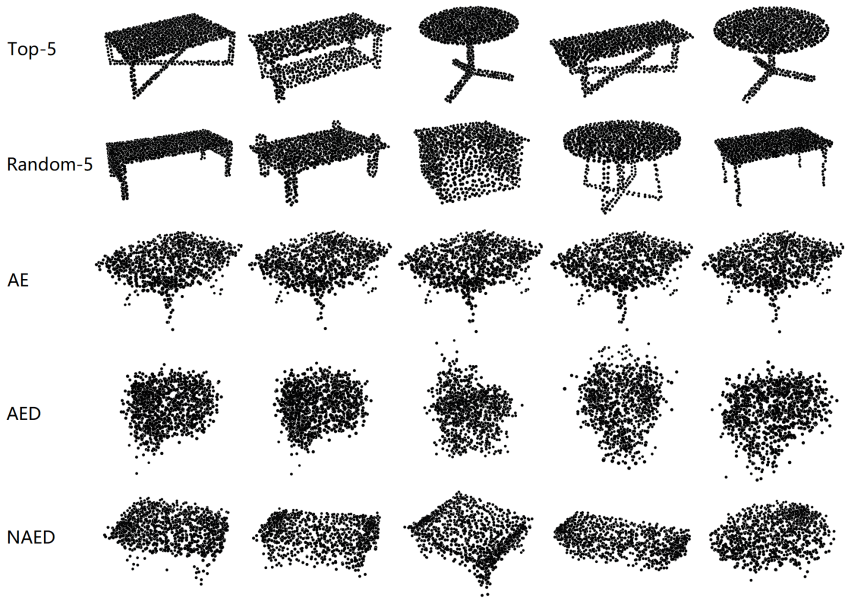

Figure 7.4.: Diversity of AM generations. We choose 5 examples from instances that (from top to bottom) 1) most hightly activate the neuron 2) are selected randomly 3) are from the generations of AE 4) of AED 5) of NAED. It can be seen that although the examples generated by AE are stable, they are severely 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.

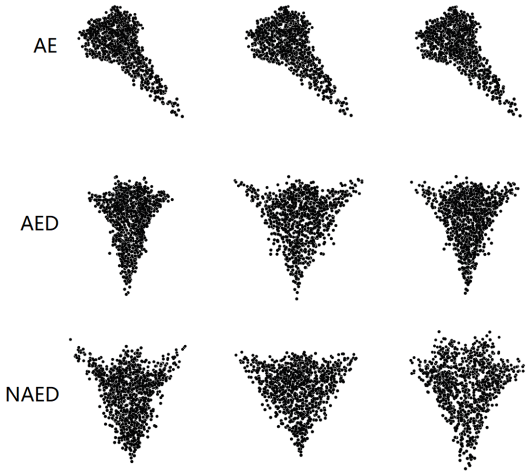

Figure 7.5.: AM examples from AE, AED and NAED respectively of class "airplane" in ShapeNet. The qualitative performance of AE, AED and NAED is comparable to those on ModelNet40.

7.5. 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.

7.4.2. 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 7.1. 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 PCAMS. 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 7.4). In addition, figure 7.3 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 PCAMS, and present the results in table 7.2. As a reference, we show an example of the corresponding visualization in figure 7.6. We notice that AED performs unstably, especially when explaining PointNet++, which occasionally fails to generate

		m-IS	FID	D^1_{CH}	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	L_2 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 7.1.: PCAMS 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.

		m-IS	FID	D^1_{CH}	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 7.2.: PCAMS evaluations for different point cloud models, where PN and PN++ denotes PointNet and PointNet++.

perceptible structures (middle plot of the second row). This is also revealed in PCAMS: in table 7.2, the lowest score is obtained by explaining PointNet++ with AED.

Another interesting observation we noticed is that the global feature-based FID proposed by [154], 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 $P_n \sim \mathcal{N}(0, \sigma^2)$, 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, FID (the brighter blue line) dramatically decreases with

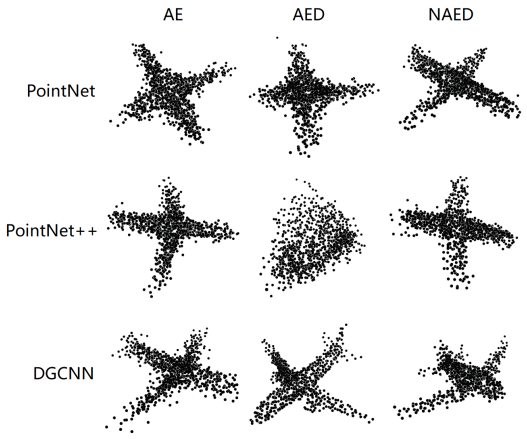

Figure 7.6.: AM visualization for the most popular point cloud networks: PointNet, PointNet++ and DGCNN. The proposed method is applicable to all point cloud networks.

	m-IS	FID	D^1_{CH}	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 7.3.: Quantitative evaluations on ShapeNet.

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.

7.4.3. 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 7.7. 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 [1] (the last column). The conclusions of the two explanations are approximately analogous, and the

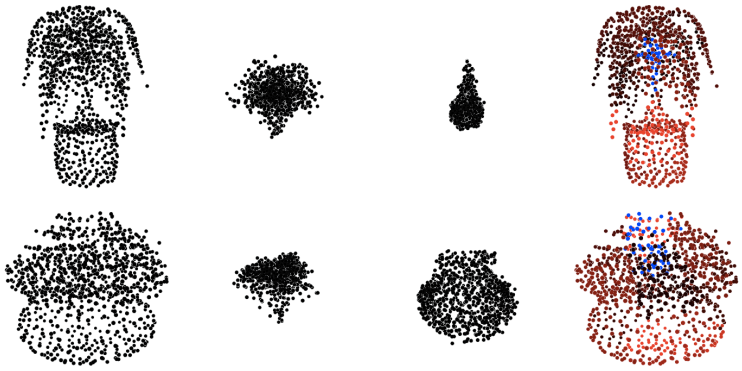

Figure 7.7.: An example of reviewing the inaccuracies of the dataset. The first column shows the instances in the dataset that are labeled as ”plant” but are classified as ”vase”. The second and third columns demonstrate the AM output for the categories ”plants” and ”vases” respectively. The last column exhibits an explanation generated from 3D LIME, where brighter red points represent more positive attributions while conversely brighter blue points represent more negative attributions, neutral attributed points are colored as black.

explanation given by the model is consistent with its predicted label in human perception.

7.5. Conclusion

This chapter 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. In the next chapter, we propose an upgrade version, which incorporates the latest point cloud diffusion models to further improve the quality of global explanations and reduce the processing time, while also visualizing the spreading process of the critical points in order to intuitively demonstrate how those points, which are crucial for predictions, are formed.

8. DAM: Diffusion Activation Maximization for 3D Global Explanations

In the previous chapter we propose a point cloud global explainability method that regularizes the gradient of the AM during optimization with the help of the prior learned by an autoencoder from the data distribution. The latest diffusion models open up new potential for such an approach. Compared to autoencoders, diffusion models have recently made a big splash in generative computer vision models. Replacing autoencoders with diffusion models further enhances the perceptibility of global explanations while significantly shortening the runtime. In this chapter, we propose a DDPM-based global explainability method (DAM) for point clouds, which leverages the superior synthesis capability of diffusion methods as well as a novel proposed point symmetric model, Point Diffusion Transformer (PDT), to generate high-quality global explanations under the guidance of dual classifiers. In addition, an adapted path gradient integration method (IGD) for DAM is proposed, which demonstrates how the critical points in the instances are moving during the synthesis process. Compared to the common integrated gradient approach, IGD exhibits remarkable faithfulness, stability and continuity superiority. Extensive experiments indicate that our method outperforms existing ones in terms of perceptibility, representativeness, and diversity, with a significant reduction in explaining time.

8.1. Introduction

While the autoencoder-based AM method proposed in Chapter 7 is capable of generating perceivable global explanations, there is still room for improvement in terms of its expla-

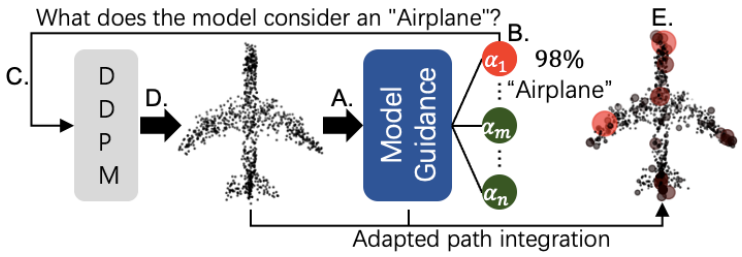

Figure 8.1.: Overview of DAM. A. Feeding an initial explanation into the classifier. B. Obtaining the target activation and its gradients. C. Maximizing the target activation under the prior of a DDPM. D. Repeating from A to C until high-quality explanations are generated. and E. Visualizing global saliency map through IGD.

nation quality and processing time. In this chapter, we further propose a novel method for visualizing global explanations of point cloud models, named Diffusion Activation Maximization (DAM). DAM is a variant of Activation Maximization (AM) [25], which is a technique for visualizing representative inputs to models. We incorporate the diffusion model into AM as the prior and exhibit through extensive experiments that DAM comprehensively outperforms existing methods (e.g., Autoencoder-based methods [6]) in terms of representativeness, perceivability, and diversity. An overview is shown in Fig. 8.1. Moreover, critical points play a crucial role in point cloud samples as they can easily alter the results of model predictions [126], [139], [7]. Spotlighting critical points may provide new insight into how these points are characterized. By adapting the path attribution method *Path Integrated Gradients* [78], we propose Integrated Gradient for Diffusion (IGD), which combines the properties of diffusion and gradient integration to more intuitively demonstrate the continuous motion of the critical points during the synthesis process. In summary, our contributions are as follows:

- We propose a novel global explainability method for point clouds named DAM based on Denoising Diffusion Probabilistic Models (DDPM). By incorporating DDPM into AM, the explanations generated by DAM outperform existing Autoencoder-based methods in all three aspects: representativeness, perceptibility and diversity with significantly reduced processing time. To the best of our knowledge, this is the first work incorporating DDPM models for generating global explanations.
- We propose IGD, an improved gradient integration method that demonstrates the varying states of critical points during the explanation generation process. IGD possesses significant superiority in terms of faithfulness, stability and continuity compared to the integrated gradients baseline.

8.2. Related Work

Activation Maximization (AM) was first proposed by [25]. A straightforward AM application to neural networks fails to produce perceptible images [39]. Subsequent studies suggested that incorporating constraints or priors enhances the perceptibility of the explanations, e.g. performing L2-norm [28], Total Variation [50] and Gaussian blur [46] on gradients, or starting optimization from the average of the dataset [38]. Higher quality AM explanations are achieved with the introduction of generative models. Alternately adding the gradients of GANs or Autoencoders during optimization effectively guides the AM explanations closer to real images [52], [66], [81].

Explainability for point clouds: Compared to images, the explainability of point cloud models has not been adequately addressed. Current point cloud explainability can be mainly categorized into transparent models, gradient-, perturbation- and example-based approaches [197]. Transparent models leverage interpretable substitute models instead of black-box DNNs to enhance transparency without significant performance degradation, examples include prototype-based models [181], linear or tree-structured models [159],

[3], etc. Gradient- [139], [186], [198] and perturbation-based methods [175], [1], [188] are extended from images to be applicable to point clouds, which are not significantly different in principle from the image explanations. Example-based methods provide users with visual illustrations that are highly representative of the model and a sample of a portion of the dataset [167], [6], [195]. The most relevant to this work is [6], which exploits autoencoders as priors to constrain the AM process in order to generate explanations with realistic contours. We demonstrate in experiments that DAM is capable of generating higher-quality explanations in reduced time, with a more transparent generation process.

8.3. Methods

8.3.1. Preliminaries

Consider a point cloud dataset $\mathcal{P} = \{P_1, ...P_m \mid P_i \in \mathbb{R}^{N \times D}\}$, where N and D denotes the number of points composing an instance and its dimension, respectively. A well-trained classifier F_{PC} is the model to be explained, which can be formulated as $F_{PC} : \mathbb{R}^{N \times D} \mapsto [0, 1]^{1 \times N_C}$.

Denoising Diffusion Probabilistic Model (DDPM): DDPM was first proposed by [40], [142], which is a denoising generative model based on Markov chains. The diffusion model is composed of two phases, with the forward being the diffusion phase, where Gaussian noise is gradually added from a real instance x_0:

$$q(x_{1:T}|x_0) = \prod_{a}^{b} q(x_t|x_{t-1}) \tag{8.1}$$

with the kernel:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}) \tag{8.2}$$

where β_t denotes the variance schedule at step t. The reverse is the sampling phase, which starts from a Gaussian distributed noise $x_T = \mathcal{N}(0, \mathbf{I})$:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{8.3}$$

with

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{8.4}$$

where μ_θ and Σ_θ are estimated by the model with parameter θ. There are numerous subsequent improvements to DDPM [165], including point cloud applicable ones [171], [173], [183], [187].

Path Integrated Gradients: Path Integrated Gradients [78] is a series of gradient-based explainability methods. By determining an uninformative baseline and a path $\gamma : [0, 1] \rightarrow$

$\mathbb{R}^{N\times D}$, the gradient from the baseline to the input is accumulated along the path in order to observe which are the critical features for the prediction. The general form of Path Integrated Gradients is:

$$PathIG_i^{\gamma}(x) = \int_{\alpha=0}^{1} \frac{\partial F(\gamma(\alpha))}{\partial \gamma_i(\alpha)} \frac{\partial \gamma_i(\alpha)}{\partial \alpha} d\alpha \tag{8.5}$$

where $\alpha = 0$ and 1 indicate the baseline and input, respectively.

8.3.2. Diffusion Activation Maximization (DAM)

DAM consists of two components, generative training and explanation sampling. An overview of the structure can be seen in Fig. 8.2.

Generative training: We leverage DDPM (equation 8.1 to 8.4) to filter the perceptible $p_G(x) \sim \mathcal{P}$ from $\mathbb{R}^{N\times D}$. Compared to [6] which utilizes Autoencoders, the advantages of DDPM are twofold:

- DDPM possesses strong reconstruction capabilities and significantly outperforms other generative models (e.g., GAN, Autoencoder, etc.) in various experiments [165].
- DDPM handles noise for each point independently, which is ideally suited to the disorderly nature of point clouds. This property enables the follow-up research such as critical attribution analysis.

The training process is roughly analogous to image DDPM, and we follow and adapt [171], exploiting the following objective as the training loss for point cloud DDPM

$$\begin{aligned} L(\theta,\varphi,\alpha) =& \mathbb{E}_q \,[\, D_{KL}(q_\varphi(z|x_0)|||p_\omega(\omega)\cdot\left|\det\frac{\partial F_\alpha}{\partial\omega}\right|^{-1}) \\ &+ \sum_{t>1}\sum_{i=1}^{n} D_{KL}(q(x_{t-1}|x_t,x_0)||p_\theta(x_{t-1}|x_t,z,l)) \\ &- \sum_{i=1}^{n}\log(p_\theta(x_0|x_1,z,l))\,] \end{aligned} \tag{8.6}$$

where x_t denotes the input at time point t, D_{KL} indicates Kullback–Leibler divergence and $q_\varphi(z|x_0)$ is a variant of PointNet [69] that serves as an encoder whose output is the mean and variance of x_0. $p_\omega(\omega)\cdot\left|\det\frac{\partial F_\alpha}{\partial\omega}\right|^{-1}$ are affine coupling layers, which project isotropic Gaussian distributions $p_\omega(\omega)$ onto more complex distributions via a trainable bijective F_α as the training priors [171] (B in Fig. 8.2).

To enhance performance and interpretability, we incorporate the following improvements:

- Following [174], to prevent x_t from collapsing into pure noise at a small t, the noise schedule α_t is optimized as

$$\alpha_t = \frac{f(t)}{f(0)}, f(t) = \cos\left(\frac{t/T + \beta_1}{1 + \beta_1} \cdot \frac{\pi}{2}\right)^2 \tag{8.7}$$

- Label l is embedded in the training process. Class information guides the sampling better towards a specific category. We simply convert the labels into one-hot vectors and concatenate them with the coordinate data (part A in Fig. 8.2).

- We propose a novel model: Point Diffusion Transformer (PDT) (part C in Fig. 8.2). Transformers are shown to be powerful architectures for learning latent representatives, which has already been introduced into point cloud DDPM [187]. However, we argue that the existing models (including non-Transformers as [171]) neglect the "symmetry property" , which was first proposed by [69], that the output of a point cloud model should be independent of the inputs sequence. Especially with DDPM, the noise added is typically isometric [142]. Thus, the utilization of asymmetric components such as multi-size convolutional kernels or fully connected layers for non-global features should be minimized. Moreover, eliminating correlations between points results in cleaner gradients in AM iterations and hence reinforces the representativeness.

  To address the above objective, we concatenate the point-wise coordinates, priors and label vector and obtain the input $x^* \in \mathbb{R}^{N\times(D+D_p+D_l)}$ (D_p and D_l are the dimensions of the prior and label vectors, respectively, C)). It is subsequently fed into a Point Diffusion Encoder (PDE), which is a multi-headed self-attention module whose inputs of Query, Key and Value are x^*. The following module is a Point Diffusion Decoder (PDD), which shares a similar structure to PDE, except that the inputs of Query and Key are the residuals $x^* + PDE(x^*)$ while the input Value is x^*. The most crucial property of PDT is that the entire model utilizes only the convolution kernel of $D_{in} \times 1$, ensuring that each point is independent of the input order during noise prediction and AM optimization. This structure not only strengthens the performance of the global explanations themselves, but also eliminates interference when calculating gradient integration, resulting in cleaner saliency maps (see Sec. 8.3.3).

Explanation sampling consists of two parts, sampling and explaining processes, corresponding to perceivability and representativeness of the explanations, respectively.

The sampling process is a reversed denoising Markov chain where the Gaussian noise x_T is fed into the trained diffusion model p_θ, and a perceptible synthetic sample x_0 is obtained after multi-step denoising. The reverse diffusion process can be formulated as

$$p_\theta(x_{(0:t)}|z,l) = p(x_t)\prod_{t=0}^{t}\mathcal{N}(x_{t-1}|\mu_\theta(x_t,t,z,l),\beta_t\mathbf{I}) \tag{8.8}$$

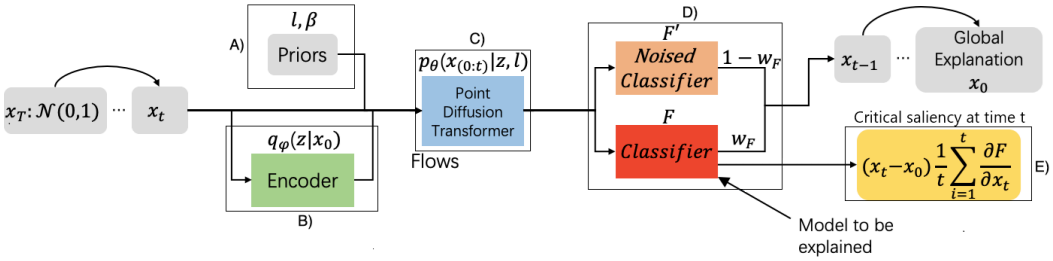

Figure 8.2.: Overview of the DAM flow at time point t. A). Embedded priors. B). Training priors encoded by PointNet-like submodules. C). Denoising x_t by the PDT model. D). Gradient guidance with dual classifiers. E). Obtain global gradients with IGD. Note there are two main explanations, one for the globally explainable sample x_0 (gray block on the right), and the other for the saliency map of the diffusion process (yellow block below).

Practically, we randomize an x_r and input it into $q_\varphi(z|x_0)$ to obtain encoded flow z_μ and z_σ, then reparameterize them to z, and generate x_T via equation 8.8. The advantage of this initialization is that it employs $q_\varphi(z|x_0)$, which is well-trained to normalize x_T to better approximate z of real data.

In the explaining phase, following Equation 7.1, we force the reversed diffusion process in the direction that highly activates a certain neuron a_i^l of the classifier F_{PC} by implanting a guidance gradient $x_t = x_t + \frac{\partial \alpha_i^l}{\partial x_t}$. The challenge of incorporating DDPM model is that, as t approaches T, the diffusion samples approximate pure Gaussian distributions $x_t \to \mathcal{N}(0, \mathbf{I})$, F_{PC} may never see analogous inputs and the obtained guidance gradients may be biased. Inspired by [165], we enable a twin classifier F'_{PC} trained on a noised dataset X' as a transition (part D in Fig. 8.2). F'_{PC} shares the identical architecture as F_{PC}, and is acquired by continuing training on X' after F_{PC} converges on X, where X' is the noisy version of X, which is transformed from the forward diffusion process $q(x_{1:t}|x_0)$. It contains samples of various noise levels based on the time information t. We train F'_{PC} by fusing binarized vector t with the coordinates information of X' so that it better guides the diffusion gradients. As t approaches 0, the sample outline is gradually regularized and the guiding classifier needs to be switched to F_{PC} (the model to be explained). We schedule two weights $W_{F_{PC}}$ and $W_{F'_{PC}}$ such that $W_{F_{PC}} + W_{F'_{PC}} = 1$, which weight the guidance gradients of F_{PC} and F'_{PC}, respectively: $W_{F'_{PC}}$ converges to 1 as t approaches T, while $W_{F_{PC}} = 1$ when $t = 0$. During optimization, we choose log(SoftMax) as the target activation α, which significantly enhances the explanation performance. A general overview of our sampling approach can be found in Algo. 8.1.

8.3.3. Integrated Gradients for Diffusion (IGD)

Numerous existing studies have shown that critical points are decisive in point cloud model predictions [126], [139], [7]. Tracking critical points in the synthesis process may shed interesting light on what properties these decisive points share and how they are

Algorithm 8.1 Sampling algorithm of DAM, given a diffusion model $(\mu_\theta(x_t, l), \Sigma_\theta(x_t, l))$, a noised classifier $F'_{PC}(x, t)$ and the model to be explained $F_{PC}(x)$

Input : Class label l, guidance weights $W_{F_{PC}}$ and guidance scale s
Output: Global explanation x_0
$Sample\ x_T \sim \mathcal{N}(0, \mathbf{I})$ **for all** t from T to 1 **do**
μ_t,$\sum_t \leftarrow (\mu_\theta(x_t, l), \Sigma_\theta(x_t, l))$
$Sample\ x_{t-1} \sim (\mathcal{N}(\mu_t + s\Sigma_t(W_{F_{PC}} \bigtriangledown_{x_t} \log(F_{PC}(x, t)) + (1 - W_{F_{PC}}) \bigtriangledown_{x_t} \log(F'_{PC}(x, t))), \Sigma_t))$
end for
return x_0

formed and changing. For this purpose, we propose IGD (part E in Fig. 8.2), which can be formulated as

$$IGD = (x_0 - x_T) \times \sum_{t=0}^{T} \frac{\partial F_{PC}(p_\theta(x_{(0:t)})}{\partial x_t} \times \frac{1}{T} \tag{8.9}$$

The two elements of Path Integrated Gradients are the baseline and the path, respectively [78], which we adapt as

- **Baseline**: The baseline is defined as "uninformative" [152], which ensures that the integrated gradient captures the whole attributional variation of the model. We consider x_T as baseline as it is sampled from $\mathcal{N}(0, \mathbf{I})$ and does not contain any information.
- **Path**: Various options are available from the baseline to input, with *linear paths* ($x' + \alpha \times (x - x')$, $\alpha \in [0, 1]$) being the frequent option. In IGD, we leverage the diffusion sampling process itself $p_\theta(x_{(0:t)})$ as the path.

In the sampling process, we have calculated exactly the guide gradients of F_{PC}: $\frac{\partial F_{PC}(p_\theta(x_{(0:t)})}{\partial x_t}$, thus there is no need to recalculate the gradients and simply integrate them from x_T to x_0. The procedure of IGD is described in Algorithm 8.2.

Algorithm 8.2 Integrated Gradients for DAM (IGD), given a DAM model $p_\theta(x_{0:T})$ and the model to be explained $F_{PC}(x)$

Output: Saliency maps IGD for arbitrary time step t
$\Delta_g = 0$
for all t from T to 1 **do** #Sampling starts
$\Delta_g \leftarrow \Delta_g + \frac{\partial F_{PC}(x_t)}{\partial x_t}$
$IGD \leftarrow (x_t - x_T) \times \Delta_g \times \frac{1}{T-t}$
Output(IGD) #IGD can output in arbitrary loop
$x_{t-1} \leftarrow p_\theta(x_t)$ #Then perform DAM Sampling process
end for

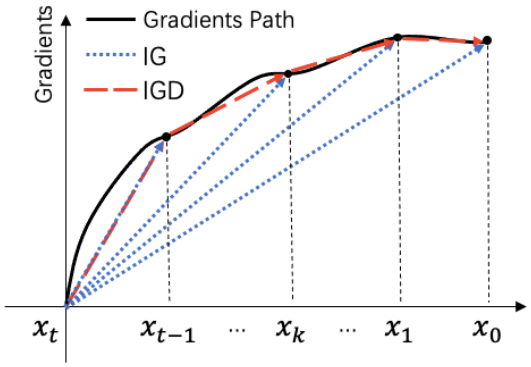

Figure 8.3.: Visual comparison of IGD and IG. The gradient path in the diffusion process is the integration from x_0 to x_T (the black curve). IG paths for x_t are the linear integration of x_0 to x_t, which may lead to bias, while the path of IGD for x_t is the integration from x_{t-1} to x_t, which better approximates the real path.

Note that **IGD is not a hard application of IG on DAM**. Compared to standard IG, IGD is more flexible for gradients integration in diffusion processes. As shown in Fig. 8.3, assume that the path of the sample gradients from a diffusion process is given by the black curve. For all x_t, the typical IG integrates the linear path of the gradients starting from the baseline each time (the blue line), which results in a bias between the final integration and the real one. This bias causes unfaithfulness and large fluctuations in the generated saliency maps. In comparison, the issue is significantly alleviated by IGD. IGD integrates the gradients of x_{t-1} and x_t with a linear path, which minimizes the bias under the precondition that the true gradient path is unavailable. We quantitatively compare the performance of the two path methods in terms of coherence and sensitivity (faithfulness) in Sec. 8.4.3. From an explainability perspective, IGD offers two advantages: a) It provides inductive exhibitions of feature attributions from a global perspective (rather than local specific inputs), b) The high confidence of examples ensured by AM enables the attributions to be globally representative of the corresponding categories.

8.4. Experiments

In this section we present the qualitative demonstrations (Sec. 8.4.1) and quantitative evaluations (Sec. 8.4.2) for DAM, and visualization of IGD and the corresponding quantitative assessments (Sec. 8.4.3). We employ ModelNet40 [45] as the primary experimental dataset (default dataset, unless specifically mentioned), it contains 12311 CAD models, of which 9843 are used for training and 2468 for testing. In addition, we validate the performance of our method on ShapeNet, a larger database of 3D objects containing a total number of 45969 samples in 55 categories, of which 35708 are used for training and 10261 for test. During the sampling phase, we generate 10 samples containing 1024 points for each class and randomly select 5 instances from the real dataset $P_1, ..., P_5$ as

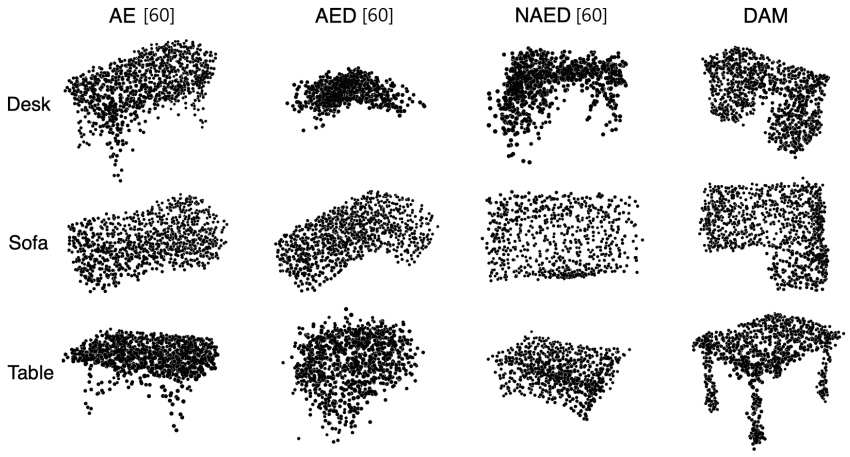

Figure 8.4.: Global explanations of 3 classes generated by DAM. For comparison, we present the identical amount of explanations generated by AE, AED and NAED [6]. As can be observed, DAM significantly outperforms AED and NAED in terms of perceptibility (see categories "Desk" and "Table"). Compared to AE, although no clear advantage is visible in terms of perceptibility, DAM far exceeds it in terms of diversity, which is analyzed in detail in Sec. 8.4.2.

the benchmark for calculating the Unidirectional Chamfer Distance D^1_{CH} and Fréchet Inception Distances (CD and FID) for quantitative evaluations.

Note that DAM is a **global explainability method**, rather than a simple point cloud synthesis or completion approach. DAM requires a simultaneous balance of representativeness to the model to be explained, demanding it to compare with synthetic methods in terms of generative performance is unfair. Thus we do not consider other point cloud synthesis methods as competitors. In the experiments, we take the approaches proposed by [6], which is currently the sole work that investigates global explanation of point clouds, for comparison.

8.4.1. Qualitative Visualizations for DAM

Perceptibility is the degree to which the generated explanation can be comprehended by humans. Generally, complete and high-quality explanations are more perceptible. We select common classes from the 40 categories of ModelNet40 and generate global explanations with DAM (illustrated in Figure 8.4). We also qualitatively compare the results of DAM with AE, AED and NAED [6]. Overall, the geometric structure of DAM-generated explanations is more robust and thus more easily perceived by humans.

Diversity is also one of the essential properties for explanations. Abundant diversity provides humans with different perspectives of explanations to gain better comprehension [66]. Fig. 8.5 illustrates the diversity of explanations generated by DAM. It can be observed that DAM is able to generate diversified and qualified global explanations.

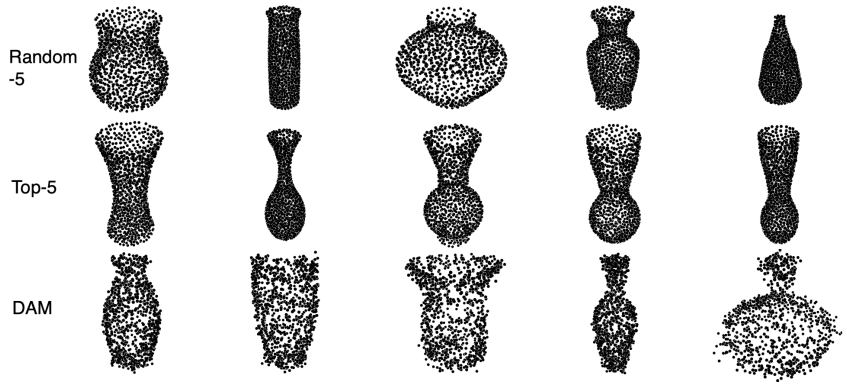

Figure 8.5.: Diversity examples. We randomly generated 5 explanations for the category "vase". For intuition, we also show 5 randomly chosen objects of the same class from the dataset (Random-5), and 5 samples that most highly activate the neuron "vase" (Top-5).

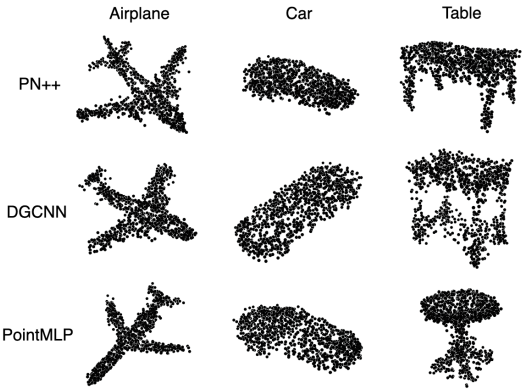

Figure 8.6.: Global explanations of other models generated by DAM. From top to bottom are PointNet++, DGCNN and PointMLP.

Explanations on other models and datasets: We test DAM on other popular or state-of-the-art point cloud models besides PointNet, including PointNet++ [73], DGCNN [119], and PointMLP [184]. Fig. 8.6 depicts the visualization of their global explanations generated by DAM. It can be observed that the performance of DAM in explaining models is independent of their internal architectures. In conclusion, our approach achieves both performance and diversity on different datasets.

8.4.2. Quantitative Evaluation for DAM

Quantifying the explainability of AM are typically based on three aspects: **higher representation** (activation degree of a certain neuron), **better perceptibility** (proximity to real objects) and **richer diversity** [66], [6]. An existing acknowledged point cloud AM

Data	Method	Model	m-IS↑	FID(10^{-3})↓	D^1_{CH}(10^{-3})↓	EMD↓	PCAMS* ↑
	Baseline	/	1.08 / 1.04	16 / 18	245 / 273	413.5 / 442.8	3.85 / 3.69
M40		AE [6]	1.08 / 1.01	16 / 17	**44** / **47**	143.1 / 147.8	4.71 / 4.57
/	AE	AED [6]	1.12 / 1.14	18 / 12	86 / 76	241.3 / 208.0	4.37 / 4.65
SN		NAED [6]	1.46 / 1.15	14 / 11	74 / 67	207.6 / 203.7	4.89 / 4.75
	DDPM	DAM (ours)	**1.78** / **1.70**	**9** / **10**	45 / 54	**133.9** / **146.2**	**5.68** / **5.45**

Table 8.1.: Quantitative evaluation of DAM compared with the models proposed in [6] on ModelNet40 (M40) and ShapeNet (SN), respectively. For reference, we additionally introduce Earth Mover's Distance. The up and down arrows denote that higher and lower values are better, respectively. The baseline is the random initialization without any regularization. *__Note that except for PCAMS, all the other metrics only reflect only one aspect and thus cannot be considered as an objective evaluation of the comprehensive performance of global explanations.__

	AE [6]	AED [6]	NAED [6]	DAM
$\hat{t}$ (s)	47.75	458.69	201.27	**12.35**

Table 8.2.: Average time $\hat{t}$ required to generate an explanation. Note that we report the processing time for comparable performance rather than identical number of iterations.

evaluation matrix known as PCAMS [6] is available, thus we employ it as the metric for evaluating DAM.

We demonstrate the quantitative results of DAM in Tab. 8.1. For comparison, we also exhibit the assessment of the explanations generated by existing studies: AE, AED and NAED [6]. The results indicate that DAM outperforms all existing point cloud global explainability methods in each metric, except for sacrificing a minimal point-wise distance to balance the diversity compared with AE. Approximate outcomes are yielded from the evaluation on ShapeNet.

In terms of computational complexity, as the interpolation times in the diffusion process are restricted, the processing time is significantly reduced. Tab. 8.2 details the average time consumption of generating an explanation, again, we compare with AE, AED and NAED from [6].

Tab. 8.3 shows the quantitative evaluations of explaining other models generated by DAM. It can be seen that DAM outperforms existing methods [6] on all metrics except for few ones on PointNet++ where DAM is slightly inferior.

8.4.3. Visualizations and assessments for IGD

In this section we illustrate the results of IGD. In the total number of 250 diffusion steps, we integrate the gradients every 50 steps and calculate the corresponding saliency maps according to Algorithm 8.2, and randomly choose an example from class "Airplane" to be illustrated in Fig. 8.7.

	Model	m-IS↑	FID(10^{-3})↓	D^1_{CH}(10^{-3})↓	EMD↓	PCAMS↑
PN2	AE [6]	1.10	**8**	**41**	**134.1**	5.12
	AED [6]	1.10	20	122	255.4	4.12
	NAED [6]	**1.86**	11	72	236.4	5.43
	DAM	1.69	**8**	48	134.4	**5.62**
DGC	AE [6]	1.02	**10**	105	252.8	4.43
	AED [6]	1.35	13	109	343.1	4.63
	NAED [6]	1.31	15	109	335.5	4.52
	DAM (ours)	**1.75**	**10**	**47**	**130.5**	**5.58**
PML	DAM (ours)	1.49	9	47	129.7	5.37

Table 8.3.: Quantitative evaluations of global explanations generated by DAM on other point cloud models. In the first column, PN2, DGC and PML indicate the experiment results on PointNet++, DGCNN and PointMLP, respectively.

	Faithfulness		Global Stability	Local Continuity		
	$S_F^{j=0.5}\uparrow$	$S_F^{j=1.0}\uparrow$	$L_{var}\downarrow$	$L_D\downarrow$	$L_W\downarrow$	$L_{\rho_S}\uparrow$
RDM	-0.120	1.064	2.411	1.087×10^{-3}	5464.640	-8.784×10^{-4}
IG (raw)	1.438	4.226	0.037	1.989×10^{-4}	364.996	**0.986**
IGD (ours)	**38.974**	**91.061**	$\mathbf{2.630\times10^{-8}}$	$\mathbf{1.892\times10^{-6}}$	**0.107**	0.753

Table 8.4.: Quantitative evaluation of attributions in diffusion. RDM is a set of randomly generated attributions for reference. IG (raw) and IGD are the conventional IG with linear paths and the gradient integration with diffusion paths proposed in this paper, respectively.

The saliency map reveals that the sparse nature [139], [7] of the point cloud attributions is already formed at the beginning of the reverse diffusion process, and those critical points are also identified at an early stage and their attributions are almost invariant. Moreover, we observe that the critical features within a category are analogous. Interestingly, those critical points with the greatest attribution appear only at the tips of noses, wings, and tail, while the points in the centre of the fuselage exhibit relatively smaller attributions. Note that the four examples are generated from models (including the DDPM model, the classifier and its noised version) trained on two different datasets (ModelNet40 and ShapeNet), which indicates the classifier learns similar features from different data source that contribute most to the predictions.

For the validity of IGD, we quantitatively evaluate the performance from two aspects, faithfulness and coherence.

Faithfulness, also known as sensitivity, is one of the most important metrics for explanations. The theory behind faithfulness evaluation is that the confidence of the model prediction decreases dramatically after ablating those features with most positive attributions, and vice versa. In our experiments, We conduct MoRF and LeRF test [35], [56]. For each generated saliency map ψ, we recursively ablate 5% of the points with the highest attributions $(x_t, \psi^p_{.05}), (x_t, \psi^p_j)$ and the lowest attributions $(x_t, \psi^n_{.05}), (x_t, \psi^n_j)$ (j is the maximum ablation rate), respectively. We then predict the confidences of these ablated inputs individually with F, i.e. $F(x_t/(x_t, \psi^p_{.05})), \cdots, F(x_t/(x_t, \psi^p_j))$ and $F(x_t/(x_t, \psi^n_{.05})), \cdots, F(x_t/(x_t, \psi^n_j))$ (A/B denotes the ablation of B from A). We evaluate the faithfulness of the saliency maps by measuring the areas between the two confidence sequences:

$$S^j_F = \int_{i=0}^{j} F(\psi^p_i) - F(\psi^n_i) \tag{8.10}$$

Coherence is a novel metric proposed specifically for explanations in diffusion processes. Recall the DDPM sampling process, where the introduced noise from x_t to x_{t+1} is minor, enabling the parameterization of the neural networks for backward diffusion [142], which indicates that x_{t+1} has significant distributional similarity to x_t, with the exception of a small amount of noise. Meanwhile, existing researches [86], [141], [143] suggest that explainability methods should perform robustly, i.e., the explanations generated for neighboring inputs are supposed to be analogous. Thus, for the explanations of diffusion processes, we leverage numerical new metrics, i.e., the discrepancy between ψ_{t_1} and ψ_{t_2} is negligible when t_1 is approaching t_2. Quantitatively, we assess two aspects, global stability, which evaluates the statistical robustness of the attributions throughout the diffusion process, and local continuity, which computes the coherence of the explanations for two adjacent sampled t.

For global stability, we compute the variance of all input attributions over the diffusion process: $L_{var} = \frac{\sum_{t=T}^{0} var(\psi_t)}{j}$. We assess local continuity with three metrics: the Difference of Predecessor (DF) and the Sliding-Window Average (SWA), which focus on the

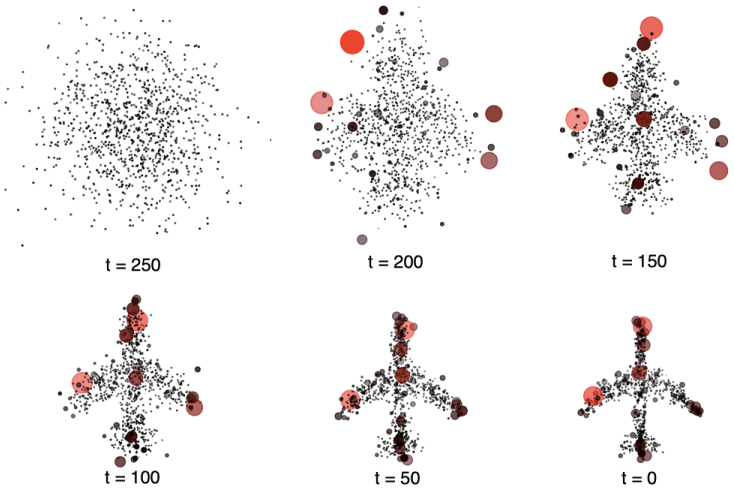

Figure 8.7.: Saliency maps for diffusion process. We integrate the gradients every 50 steps and calculate the attributions with Integrated Gradients. The redder and larger the points, the more positive the attributions in the prediction.

smoothness of the numerical values, and the Spearman Coefficient Average, which emphasizes the consistency of the rankings. The DF is simply the average of all differences between temporally neighboring attributions $L_D = \frac{\sum_{t=T}^{1} |\psi_t - \psi_{t-1}|}{T}$. For SWA, we calculate the average of the attributions for three consecutive diffusion samples $\overline{W_t} = \frac{\psi_{t-1}, \psi_t, \psi_{t+1}}{3}$ (When $t = T$ or 0, ψ_{t+1} and ψ_{t-1} are ignored, respectively). SWA (L_W) is the mean of the difference between all ψ_t and $\overline{W_t}$: $L_W = \frac{\sum_{t=T}^{0} \psi_t - \overline{W_t}}{T}$. Similarly, for Spearman Coefficient Average, we compute the mean of the Spearman Coefficients of all attributions with their predecessors $L_{\rho_S} = \frac{\sum_{t=T}^{1} \rho_S(\psi_t, \psi_{t-1})}{T}$, where $\rho_S(a, b)$ denotes the Spearman's Coefficient between a and b.

Tab. 8.4 demonstrates the results of the quantitative comparison of IG and IGD. In terms of faithfulness, IGD significantly outperforms IG for both 50% (j = 0.5) and 100% (j = 1.0) ablation, which verifies that integrating the gradients along the diffusion path is more faithful to the model. For coherence, IGD is remarkably more robust, owing to that IGD simply requires additional gradient integration from x_t to x_{t-1}, whereas IG recomputes the linear integration from the baseline x_T to x_{t-1}, which disrupts the continuity of neighboring samples in diffusion processes.

Interestingly, IG is consistent with the Spearman's coefficients, almost identifying the critical points at x_T period ($L_{\rho_S} = 0.986$). However, numerically IG does not exhibit a corresponding consistency (L_D, L_{var} and L_W), as the vast majority of attributions are centralized to a minority of points, which is in line with the conclusion from existing studies [139], [7]. We argue that such a sparse attribution may be biased as there exists an alternative integration path that yields saliency maps with significantly higher faithfulness than the linear path from the typical IG.

8.5. Conclusion

This chapter is the first attempt to employ DDPM for generating high-quality AM global explanations. In addition, we propose a diffusion path-based attribution approach, which alleviates the bias of the typical Integrated Gradients. Through extensive experiments, we demonstrate that diffusion model-based AM significantly improves the quality of the global explanations, and in addition reduces the processing time dramatically.

In the first three chapters of this part, we have presented three adaptive extensions of explainability methods to point cloud models. Starting from the next chapter, we design specialized modules to investigate the learning mechanism of point cloud models from their own properties. We further reveal how point cloud models make predictions by designing interpretable classifiers, analyzing the sensitivity of critical points, and deconstructing the distribution of latent features in the intermediate layers.

9. Fractal Projection Forest: Fast and Explainable Point Cloud Classifier

The previous chapters in this part have all made adaptive improvements for point clouds based on general XAI methods. Starting from this chapter, we will conduct point cloud-specific explainability analysis, which is based on the structural characteristics of point cloud data and models, thereby pioneering a novel perspective for unveiling the decision-making mechanisms of point cloud models.

Besides post-hoc explainability methods, interpretable models are likewise one of the most essential approaches to enhance the trustworthiness of predictions. Unlike post-hoc approaches, interpretable models do not require the incorporation of additional modules to explain the prediction results, rather, the original black-box model is replaced with one whose performance is approximated and more transparent to render the decisions more comprehensible. In this chapter, we leverage the spatial geometric properties of point clouds to design interpretable point cloud classifiers that only require the incorporation of random forests for accurate prediction. We propose a new pipeline named Fractal Projection Forest (FPF) that exploits fractal features to enable traditional machine learning models to achieve competitive performance with DNNs on classification tasks. Though compromising by a few percentages in accuracy, FPF is faster, more interpretable, and easily extendable. We hope that FPF may provide the community with a novel view of point cloud classification.

9.1. Introduction

In addition to post-hoc explainability approaches for point clouds proposed in Chapter 6, 7 and 8, the employment of interpretable models also facilitates plausibility [100]. One study suggests that if appropriate features are chosen, even with simpler and interpretable models, there will be no significant drop in accuracy compared to black-box ones [115]. Fortunately, recent post-hoc attribution research reveals that the features of point clouds are sparse [139], [1]. Comparable accuracy can be achieved using simple interpretable models if these features are filtered out from point cloud input in advance.

In this chapter, we propose a new pipeline of point cloud classification by extracting the input projections through multi-size fractal windows. This approach jumps out from the race of adding tricks to DNNs, and performs classification through simpler models with only few percent accuracy compromised. Moreover, our method enables the training of

the entire ModelNet40 in tens of seconds (several minutes if pre-processing is included) with a CPU, which is barely possible for DNNs. Besides, our model provides two explanations, i.e., intrinsic Gini impurity and perturbation-based attributions. For the latter, our model allows the ablation of entire grouped features, thus avoiding the concern of feature correlation and out-of-distribution issues [103] as in raw point clouds. Additionally, our approach is more extensible as it can adapt to different classification scenarios by manually adding appropriate features or switching models. In summary, our contributions are primarily summarized as follows:

- We propose a non-DL pipeline Fractal Projection Forest (FPF), that converts point cloud classification, which is previously only solvable with DNNs, into traditional machine learning tasks. Compared with DNNs, FPF is faster, more interpretable and extensible, while compromising only few percent of accuracy.
- We demonstrate two explanations of FPF, both of which are challenging to accomplish on DNNs.

The overall structure of this paper is as follows: In Section 9.2, we introduce popular point cloud models and corresponding explainability methods. Section 9.3 elaborates the ideas and technical details of FPF. In Section 9.4, we show the performance of the proposed method and the corresponding explanations. In Section 9.5, we give a short summary and propose future research directions.

9.2. Related Work

In this section, we introduce the widely applied classification models (9.2.1)and the research that promotes the explainability for point clouds (9.2.2).

9.2.1. Classification models for point clouds

Before the advent of models that act directly on raw point clouds, there were typically two ways for the classification tasks, namely the voxel-based [37], [106], [151] and the multi-view-based approaches [42], [90], [91], [166]. The voxel-based approaches are dedicated to organizing irregular point clouds so that they are spatially ordered, which enables the extraction of adjacent features between points using 3D convolutional kernels. The voxel-based methods, as early solutions to address the irregularity, suffer from limited processing speed due to the incorporation of additional pre-processings, such as voxelization [37]. The multi-view-based methods project point clouds onto planes, subtly downscaling them to two-dimensional images. Interestingly, in recent research [166], a simple 6-view-method achieves almost state-of-the-art accuracy by introducing tricks into the training process. With the proposal of PointNet [69], a series of classification methods that operate directly on raw point clouds come into view. PointNet establishes a pipeline for following studies, i.e., Eq. 6.1. Most subsequent models based on raw point

clouds [73], [93], [119], [177] are subject to this pipeline, with more complex tricks attached to the individual modules to achieve higher accuracies. However, all the aboves are based on neural networks, which are black-box models, and humans struggle to understand their decision-making principles [185].

9.2.2. Explainability research for point clouds

Post-hoc explainability methods: There are currently only few explainability studies on point cloud models. [139] and [1] explain the decision attributions by incorporating gradient-based and surrogate model-based explainability methods [185] to point clouds respectively. For the research on model intrinsic attribution, [126] observes changes in the prediction confidence by filtering and incrementally flipping the key points. However, all the aforementioned methodologies are post hoc attributions, and the explanations obtained may be biased due to flaws in explainability or perturbation approaches [103], [105], [156].

Interpretable models: Before DL was widely applied, point cloud classification was usually accomplished by manually extracting features and trivial ML models, e.g. LDA [59] or Markov network [23]. Nevertheless, these approaches were promptly replaced by newly emerged point cloud DNNs due to the limited performance. To address interpretability, a recent study [181] proposes a prototype-based model that accomplishes classification by clustering the features in the latent space. Although the decision basis of their approach is intuitive, it is less extensible, and we argue that the extraction of latent features with neural networks aggravates the opacity. In addition, PointHop [159] is also a non-DL method that utilizes a random forest to learn adjacent features of points extracted by a module called PointHop Unit. However, although PointHop improves the interpretability of the model, the geometric features between points are still incomprehensible for humans. In addition, PointHop performs significantly degraded on real scanned datasets.

9.3. Methods

In this section, we introduce the mechanism and structure of FPF. Section 9.3.1 outlines how fractal-based features operate, Section 9.3.2 details the internal structure of FPF, and Section 9.3.3 illustrates how FPF generates reliable explanations.

9.3.1. Fractal features

This work is inspired by *Hausdorff dimension* [12]. Hausdorff dimension reflects the smoothness of geometry and can be estimated by counting the total number of fractals in different sizes and applying an exponential fit. Similarly, we create features of point clouds manually with multi-size fractal windows. Fig. 9.1 illustrates an overview of fractal features. With progressively increased window sizes, the point set is sampled into different subgroups. We then extract relevant statistical information from each subgroup,

which varies according the distribution of points within the subgroups. For instance, from the two point sets (first column), we can extract the feature sequences $[4, 2, 2]$, $[1, 2, 2]$ (top) and $[4, 4, 4]$, $[1, 1, 1]$ (bottom) with respect to n and $\bar{p}$, respectively. Note that we only list the total number of windows (n) and the average number of points contained ($\bar{p}$) here, additional information will be discussed in Sec. 9.3.2.

9.3.2. Model architecture

Fig. 2 shows the architecture of FPF. FPF consists of the following components: projection, fractal sampling, feature generation & concatenation and prediction modules. Consider the input as a D dimensional instance with N points: $P = \{p_i | i = 1, \ldots, N\} \in \mathbb{R}^{N \times D}$, and we illustrate with the simplest case, i.e., $N = 3$.

Projection: Inspired by the multi-view classification methods [42], [97], [166], we first project the point cloud on each axis (x,y and z) and planes (xy, xz and yz). Subsequently, three 1-D and 2-D projections are obtained, they are denoted as: $P_1 \in \mathbb{R}^N$ and $P_2 \in \mathbb{R}^{N^2}$.

Fractal sampling: According to the mechanism in 9.3.1, we perform fractal sampling for each of the six projections. Let the number of multi-scale sampling be I. The start of the sampling window size is half of the entire spatial value range, i.e. $W_{max} = \frac{1}{2}\{\max x_i - \min x_i\}$, which is scalable and 1 for ModelNet40 [45]. The window size at the i-th sampling is:

$$W_i = e^{(-i \times \alpha)} \tag{9.1}$$

where α is a smoothing coefficient that flattens the changes in the sampling window sizes. We chose exponential because the points sampled by the equidistant window size are prone to alter drastically when i is small and almost constant when i is large. The return of the sampling is which window each point belongs to, and two sampling results are obtained, $S_1 \in \mathbb{N}^N$ and $S_2 \in \mathbb{N}^{N^2}$ for P_1 and P_2, respectively. Note that the records of those windows that do not sample any points are marked as 0, which are also important features.

Feature generation & concatenation: Compared with raw point clouds, the advantage of tabularized features is that they are more intuitive when explaining attributions. Therefore, we extract statistics from point clouds in the form of tabular features as training data. Similar to other DNN models, FPF requires global and local features for prediction as well.

Global features: Three global features are involved: the number of non-zero fractal windows: $G_n = |\{s \in S_1, S_2 | s > 0\}|$, basic statistics for all sampling windows, and the Gaussian parameters. The base statistics include the maximum and minimum (non-zero) values of points within a single window, and the corresponding window indexes. To obtain the parameters, we employ Gaussian fits to approximate the distribution of the sampled points in corresponding projection space. This feature is based on the assumption that most objects are normalized to be located in the middle of the spatial coordinates. We

estimate the parameters using one- and two-dimensional Gaussian models, respectively. The former contains two parameters, i.e., the mean (μ) and the variation (σ), and the latter consists of five, i.e., the means along each axis μ_x and μ_y, the standard deviations σ_x and σ_y, and an amplitude bias α.

Local features: Two local features are involved: (averaged) distributions of the points within all windows as well as several specified windows. For the former, we calculate the extremes, means and variances of the points within each window separately and obtain their global averages. For the latter, we selectively monitor the windows of particular indexes and compute the internal distributions of points. However, since the total number of windows obtained after sampling with different sizes is inconsistent, we specify a proportion $m \in [0, 1)$ and select $|S| \times m$ ($|S|$ is total number of windows) as the monitored window. We record the number, mean and variance of the points in this window as features. In addition, multiple monitored targets can be appended.

Feature concatenation: We simply concatenate global and local features as final inputs. Note that our features are arbitrarily expandable, and different statistical features can be explored for specific datasets to achieve optimal accuracu.

Others: FPF incorporates other modules to further enhance performance.

Rotation augmentation: Rotating objects R times to create R new training data is an augmentation method proposed in [69], [73]. In this work, we set the angle of each rotation to $2\pi/R$, i.e., the object is rotated by an identical angle R times, and the $R + 1st$ time returns to the original one. There are two types of rotation augmentation available in FPF: *feature* and *quantity* augmentation. We first perform the same tabular feature extraction on the rotated object to obtain R additional feature series. In feature augmentation, we horizontally concatenate R sequences ($R + 1$ times lengthened), which provides each training data with additional rotation information and enhances rotation invariance. The quantity augmentation is similar as in [69], [73]. The rotated features are vertically aggregated to the dataset as new training data, which increases the amount of training data to mitigate overfitting.

Attribution filtering: In decision making, there are features that contain either positive or negative attributions. Positive attributions reinforce the predictions while negative ones diminish the confidence. Therefore, after training a raw model, we obtain attributions of each feature based on the explanation (see Sec. 9.3.3) and filter out those features whose attributions are below a threshold TH_p to further enhance the accuracy.

9.3.3. Explainability

FPF offers two perturbation-based explanations: Gini Importance and grouped feature ablation.

Gini Importance is calculated as the average of the decreasing impurity of all nodes over all trees [26]. The advantage of this approach is the calling simplicity, which can be

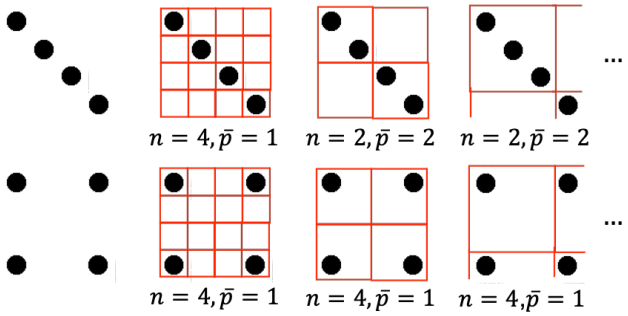

Figure 9.1.: A simple illustration of fractal feature extraction. Sampling the input using fractal windows of different sizes yields sequences of statistics, which vary depending on the distribution of input points.

obtained directly through $RF.feature\_importances\_$ from *sklearn* in few milliseconds. One drawback of impurity attribution is the bias towards continuous or high cardinality variables [94]. Another concern is the bias in datasets containing highly correlated features, which leads to sub-optimal predictor variables being assigned greater weights [22], [24].

Grouped feature ablation is an alternative method that addresses the above issues. Traditional feature ablation approaches suffer from out-of-distribution problem [103], i.e., when certain features are ablated, the data consisting of the remaining ones are outside the distribution from original dataset, resulting in unreliable prediction analyses. Therefore, we first group features according to types, i.e. features belonging to 1-D projection or sampling window size 1. Subsequently, we ablate all features belonging to the same type t to avoid any information residual. According to the methodology proposed by [103], we retrain the ablated dataset and record the accuracy of the entire testset as $Acc'_{\bar{t}}$. The attribution of features in type t can be represented as:

$$Atr_t \propto Acc - Acc'_{\bar{t}} \tag{9.2}$$

where Acc is the accuracy of the original model on the unablated testset. Grouped feature ablation alleviates the bias in the explanations, while prolongs the processing time as the model has to be retrained as many times as the number of feature groups.

9.4. Experiments

In this section, we report the quantitative results of FPF through extensive experiments. Section 9.4.1 presents the performance of FPF on multiple datasets, and Section 9.4.2 provides visual explanations for the feature importances. In our experiments, we choose ModelNet40 [45] as the main dataset, which contains 9,843 and 2,468 CAD models belonging to 40 classes in the training and test sets, respectively. In addition, we test our approach on ScanObjectNN [118] and ShapeNet [36]. The former is a real-world dataset containing 15,000 objects in 15 categories, and the latter is a dataset containing 35,708

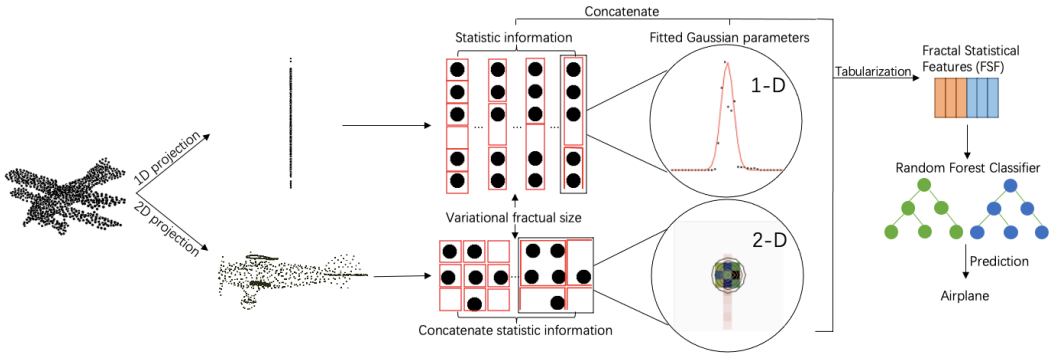

Figure 9.2.: An overview of FPF architecture. Given a point cloud object P, FPF first projects P onto the three axes and planes separately. For each projection, we record fractal feature extracted by the multi-size fractal windows. Under the assumption that the points in the fractal window overall subject to Gaussian distribution, we estimate the Gaussian parameters and concatenate the statistical information of other fractal windows together as features to train a random forest.

and 10,261 training and testing instances, respectively. Specifically, ShapeNet is originally proposed for 3D reconstruction and is extended to classification tasks in our experiments. All calculations are performed on an Intel(R) Core(TM) i7-4650U CPU @ 1.70GHz CPU except for the training of DNNs for the processing time comparison. For FPF, we set the smoothing coefficient α to 0.135, the number of fractal windows I to 30, and the number of rotational enhancements R to 3, which empirically yields the best efficiency-performance trade-off.

9.4.1. Quantitative Results

Results on ModelNet40. We compare FPF with two NN baselines (FC and CNN), four DL-based models (PointNet [69], PointNet++ [73], DGCNN [119] and PointMLP [184]) and one non-DL model. The selected DNN-based models are representative, where PointNet can be considered as a baseline for deep learning (the components in Equ. 6.1 are max-pooling, MLP and no point-wise correlation, respectively). PointNet++ replaces the last term with Multi-scale grouping (MSG), while DGCNN upgrades MLP to dynamic graph networks. PointMLP is the state-of-the-art model for point cloud classification. As reported in Table 9.1, our non-deep learning model FPF approximates DNNs in terms of accuracy (4.1% and 7.5% lower than PointNet and PointMLP respectively), with a significantly reduced processing time. However, FPF predicts the entire test set (2,468 instances) in less than 1 second with CPU, and the training time is approaching 1 minute. In addition, the raw FPF is enhanced in accuracy and speed with attribution filtering (empirically setting the threshold to $TH_p = 7e-5$), demonstrating the necessity of filtering out negative attribution features, which is unachievable in neural networks. Compared with the non-DL model PointHop, though FPF is slightly inferior in accuracy, its prediction is faster and the model size is almost 40 times shrunk.

	Models	OA(%)	mAcc(%)	F1	T_{tr}	T_{te}	T_{avg}	Size
DL	Baseline FC	74.2	65.5	64.0	/	0.21	8.50×10^{-5}	6.8 MB
	Baseline CNN	81.4	75.0	74.4	/	4.22	1.71×10^{-3}	**4.2 MB**
	PointNet [69]	90.4	85.5	85.6	/	3068.07	1.22	39.8 MB
	PointNet++ [73]	92.5	89.8	89.5	/	3004.10	1.20	20.1 MB
	DGCNN [119]	92.7	89.6	89.7	/	968.99	0.39	6.9 MB
	PointMLP w/o vot. [184]	**93.8**	**90.2**	**89.9**	/	2039.05	0.83	101.3 MB
Non-DL	PointHop [159]	87.3	81.5	78.9	301.03	6.91	2.8×10^{-3}	3.16 GB
	FPF (raw)	85.4	79.0	79.7	68.88	0.31	1.25×10^{-4}	77.5 MB
	FPF (fltd)	86.3	80.4	81.1	**68.36**	**0.29**	$\mathbf{1.16 \times 10^{-4}}$	76.1 MB

Table 9.1.: Classification results on ModelNet40. T_{tr}, T_{te}, T_{avg} and Size are the processing time for training the whole train set, validating the whole test set, the average time for predicting a single instance and the model size, respectively. FPF (fltd) and FPF (raw) denote FPF with/without attribution filtering, respectively.

Model	DL				Non-DL		
	PN	PN++	DGCNN	PointMLP	PointHop	FPF(raw)	FPF(fltd)
OA(%)	68.0	77.9	78.1	**85.4**	54.2	68.2	68.8

Table 9.2.: Comparison of the overall accuracy on the PB_T50_RS of ScanObjectNN (the hardest task). PN and PN++ denote PointNet and PointNet++, respectively.

Additionally, we attempt to train simple neural networks to learn the fractal features. We train a simple FC network and a CNN with the features extracted from the fractal windows as NN baselines (the structure can be found in Section A). Interestingly, the performance of NNs is inferior to random forest. Interestingly, NNs perform inferior to random forests, but they are thousands of times faster compared to other DL methods. One possibility is that better network structures may exist that outperform random forests, but this would not only compromise speed, but also sacrifice interpretability.

Results on ScanObjectNN. We select the hardest variant (PB_T50_RS) of ScannObjectNN as the training and test set, in which most of the objects are incomplete and retain only surface information. Table 9.2 reveals the quantitative results on ScanObjectNN. Note that FPF is potentially extensible and can concatenate more appropriate features for different data to achieve better performances. Here we follow the hand-crafted features in ModelNet40, which may not be optimal for ScanObjectNN, nevertheless, the accuracy of FPF outperforms PointNet, which is considered as the baseline for DNNs. We leave the methodology of crafting the most suitable features for different data as future work. Moreover, PointHop underperforms on this dataset (14% lower). We believe this is because the local region features extracted by the PointHop Unit are only suitable for hand-crafted datasets, and not for difficult ones like ScanobjectNN (most samples are available with surface information only). This restriction does not exist for FPF, since the hypothesis of FPF presupposes only that the distribution of the points from the train and test sets are similar.

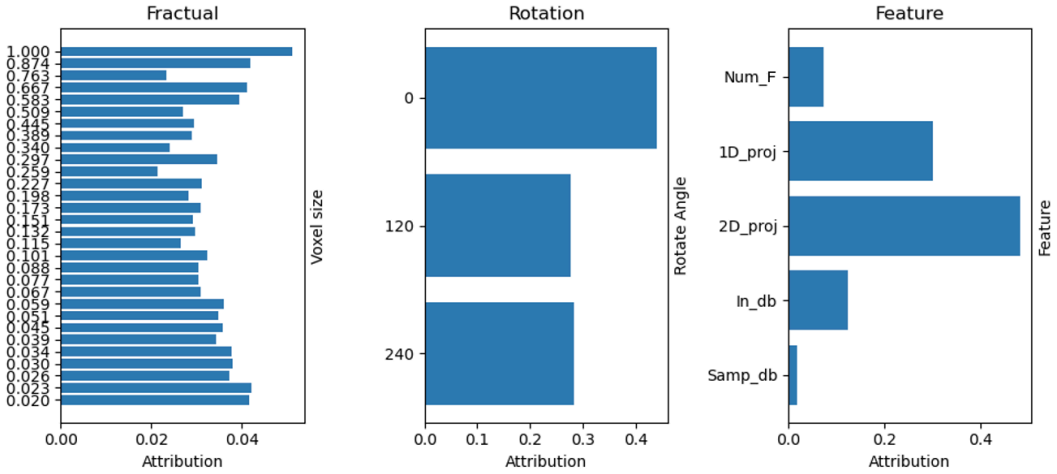

Figure 9.3.: Intrinsic feature importance explanations based on Gini impurity. Fractal, Rotation and Feature represent the feature attributions of each fractal sizes, rotation degrees and hand-crafted features, respectively. Among Features, Num_F, $1D\_proj$, $2D\_proj$, In_db and $Samp\_db$ indicate the number of fractions, statistics and estimated Gaussian parameters for 1D and 2D projections, statistics over all fractions and inside specifically sampled fractions, respectively.

Results on ShapeNet. Since ShapeNet is not a benchmark for classification, we only observe whether FPF suffers from accuracy collapse on different datasets. We report the accuracy of raw FPF as 79.1, and the accuracy of FPF after (empirically) filtering with $TH_p = 8e - 5$ as 79.3.

9.4.2. Explanations for Feature Importance

In addition to rapidity, a more important advantage of FPF is the interpretability. Aside from replacing DNNs with traditional models like PointHop [159] to enhance interpretability, we provide two explanations according to the properties of FPF, i.e., *Gini Importance* and *Grouped feature ablation*, which are shown in Fig. 9.3 and Fig. 9.4, respectively. The two explanations agree on the attribution of rotations, with one divergence in the attribution of features and a distributional discrepancy in the attribution of window sizes. The discrepancy stems mainly from the inherent deficiencies of existing explainability methods. Feature importance based on Gini impurity tends to assign more attributions to features with large cardinality [94], such as the fractal window sizes. In contrast, the importance of features with smaller cardinalities, such as the point statistics in windows, are prone to be underestimated (see Fig. 9.3, in "In_db" of the third subplots). The advantages of this explainability method are the simplicity and rapidity of invocation, which can be called directly in *sklearn*, and consumes an average computation time of approximately 0.03 seconds.

The post-hoc explanation based on grouped feature ablation is more plausible. The feature groups are independent of each other and the dataset is retrained based on ROAR

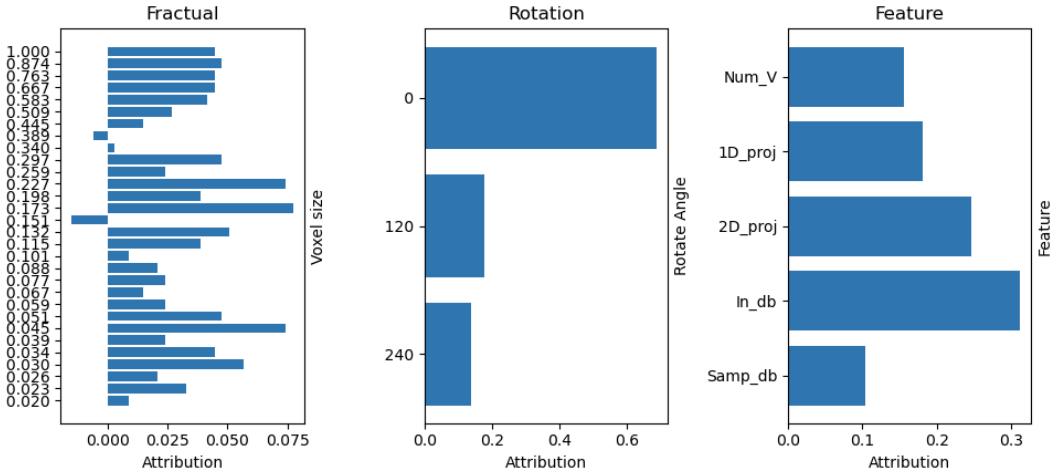

Figure 9.4.: Explanation of Grouped feature ablation.

	Random Baselines		Fractal Features				
	Baseline_U	*Baseline_W*	*Num_F*	*1D_Proj*	*2D_Proj*	*In_db*	*Sample_db*
Accuracy	2.5	3.0	65.5	65.4	66.5	63.3	61.1
Precision	2.6	2.4	59.8	56.8	57.7	54.4	54.1
Recall	2.4	2.4	60.1	57.5	58.4	55.4	54.4

Table 9.3.: Feasibility tests for fractal features. The baselines are: Uniform random guess and weighted random guess. The fractal features from left to right: number of fractions, 1D projection, 2D projection, intra-fraction distribution, sampled distributions.

[103] after each ablation round to prevent the out-of-distribution issue. However, ROAR is also controversial. The feature importance of ROAR depends on the magnitude of the decline in accuracies of the retrained models, while several studies questioned whether the explanations should be faithful to the original model or to the data [145], [152], [155]. One of the difficulties of explainability methods is the lack of ground truth, and exploring more accurate and plausible explanations is the potential research direction.

9.4.3. Others

In this section we present additional results of FPF, including the feasibility of fractal features, ablation studies and salinity checks for the explanations.

Feasibility of fractal features. To further confirm the effectiveness of fractal features, we train a simple decision tree ($max\_depth = 20$) with multiple fractal features and observe whether the accuracies outperform the baselines. We consider two baselines, uniform and weighted random guesses. The former assumes that each label has the same probability of being guessed, while the latter is weighted according to the amount of data in each class. As shown in Table 9.3, each fractal feature significantly outperforms the random guess baseline.

Module	Augmentations		Fractal features					
	~~Multi_Frac~~	*~~Rotat_Aug~~*	*~~Num_F~~*	*~~1D_Proj~~*	*~~2D_proj~~*	*~~In_db~~*	*~~Sample_db~~*	*All*
OA(%)	81.9	85.4	85.8	85.7	85.5	85.3	85.9	**86.3**
mAcc(%)	75.0	79.2	79.9	79.4	79.4	79.2	79.6	**80.4**
F1	75.3	79.8	80.3	80.2	79.9	79.5	79.9	**81.1**

Table 9.4.: Ablation study for modules. From left to right, the absent modules are: Multiple fractal series, rotation augmentation, number of fractions, 1D projection, 2D projection, intra-fraction distribution, sampled distributions. The last column indicates that all modules are integrated.

Rotation vote. Rotation vote is a technique to further improve accuracy by rotating and predicting an object multiple times and then performing a majority voting, which exhibited superior results in several point cloud models [73], [109]. For FPF, we consider two candidates: rotation votes with/without rotation augmentation (see the last subsection of 9.3.2), which represent whether rotating information of the object is incorporated in the training, respectively. Note that since the selection of rotation features is only possible before the attribution filtering, we only compare the performance with *raw* model. Surprisingly, rotation vote yields no performance boost for FPF and the accuracy even collapses if rotation augmentation is not employed. We believe the reason is that the fractal features dramatically change with rotations, evidenced by the degradation of performance after ablation of the rotation augmentation in Table 9.4. The solution is to learn more fractal features at different rotation angles, which however leads to more time consuming.

Ablation study. We decompose and ablate each module of FPF in turn, and record the corresponding accuracies. For ablating multiple fractal series, we calculate the results for fractal windows with different sizes individually and take the average. For the rotation enhancement, we simply set $R = 1$. For the remaining fractal features, we remove them sequentially from the aggregated features. We train a new random forest model after each ablation in order to avoid the out-of-distribution issue. The results are demonstrated in Table 9.4. The absence of each module results in a degradation of accuracy, while the drop is more significant when employing single size of fractal windows.

Sanity checks. Due to the lack of ground truth, there are few metrics available to assess the plausibility of explanations. Among them, salinity check [156] is an important indicator. The fundamental idea is that the generated explanation should be relevant to the model, and the collapse of the explanation is supposed to be observed as the model is randomized.

Due to the utilization of random forest, we can hardly modify arbitrary layers of the model as in [156]. Instead, we randomize certain percentages (from 10% to 100%) of decision trees that are randomly selected from the forest. However, it is challenging to edit the weights of decision trees directly, we therefore randomize the chosen trees by retraining

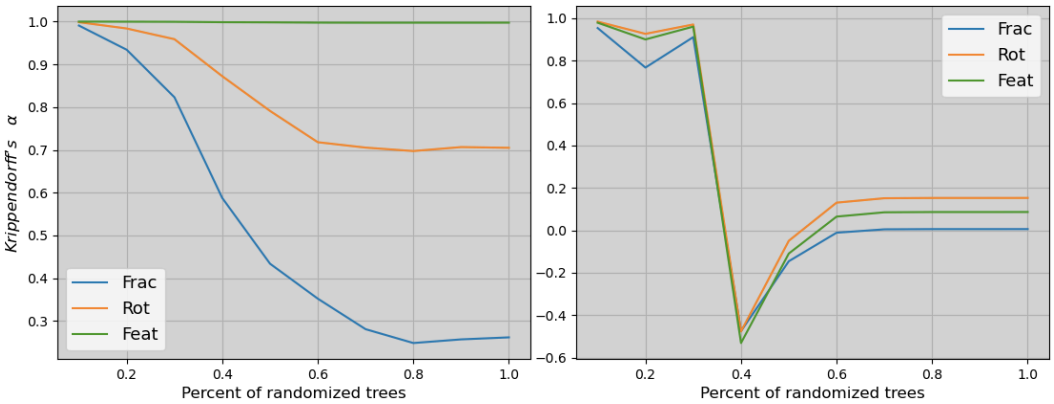

Figure 9.5.: Results of sanity checks. The left side is the intrinsic feature importance, and the right side is the group feature ablation. The blue, orange and green lines denote fractal windows, rotation and hand-crafted features respectively.The x and y axes denote the percentage of randomized trees and Krippendorff's α score, respectively.

them with random labels. For evaluating the similarity of explanations, we follow [156] using *Krippendorff's* α [92], which is formulated as:

$$\alpha = 1 - \frac{D_o}{D_e} \tag{9.3}$$

where D_o and D_e are the disagreements observed and expected by chance respectively. If the Krippendorff's α approaches 1, the two explanations are highly analogous, while if it approaches 0, they are almost independent (Negative values indicate an inverse proximity). As the results in Fig. 9.5 demonstrate, the grouped feature ablation passed the salinity check, where the more trees are randomized, the closer Krippendorff's α converges to zero. However, the intrinsic feature importance is shown to be flawed. Though the α-value of the fractal window feature is rapidly corrupted, there is minimal decrease with respect to the rotation feature, and almost no changes in the hand-crafted feature. This can be attributed to that the interpretation based on Gini impurity suffers from neglect of low cardinality features, while the correlation between features also raises problem due to the absence of retraining.

9.5. Conclusion

In this chapter, we propose a non-deep learning point cloud classification pipeline FPF. By extracting statistical features from fractal windows, FPF enables traditional machine learning models to achieve comparable performance to deep learning. Through experiments on different datasets, we demonstrate that compared to DNNs, FPF are not only faster but, more importantly, possess better interpretability.

Starting from the next chapter, we analyze the internal properties of point cloud classifiers and reveal the interior workings of point cloud models through the lens of those

interesting observations. We begin with an adversarial analysis of commonly employed models and datasets, addressing the issue of the sensitivity of critical points to predictions.

10. Explainability-Aware One Point Attack for Point Cloud Neural Networks

In Chapter 6, we introduce the method of generating saliency maps of point clouds, and it responds to which critical points in the inputs are important for the prediction results. Nonetheless, a discussion of the crucial degree of these points, i.e., the extent to which they can impact model predictions, is absent. In this chapter, we address this issue with a thorough analysis of the sensitivity of predictions to critical points with adversarial attacks. We propose two adversarial methods: One Point Attack (OPA) and Critical Traversal Attack (CTA), which target the points crucial to predictions more precisely by incorporating explainability methods. Our results show that popular point cloud networks can be deceived with high success rate by shifting only one point from the input instance. We also demonstrate the interesting impact of different point attribution distributions on the adversarial robustness of point cloud networks. We discuss how our approaches facilitate the explainability study for point cloud networks. To the best of our knowledge, this is the first point-cloud-based adversarial approach concerning explainability.

10.1. Introduction

Addressing the analysis of critical point sensitivity mentioned in Sec. 1.3.2, several attempts have been made to investigate the adversarial attacks on point cloud networks, e.g., [148] and [121]. The first series (*shape-perceptible*), represented by [148], although produce geometrically continuous adversarial examples with external generative models, fundamentally aim at deceiving the human eyes and therefore ignore the constraints on the perturbation dimensions. Another series (*point-shifting*) represented by [121] has shown a possibility that moving (dropping or adding) points at crucial positions can successfully fool the classifier. Nevertheless, most of such studies have only focused on minimizing perturbation distances for imperceptibility. Conversely, we start from a different perspective by exploring attacks on point cloud networks with a minimal number of perturbed points. Additionally, we argue that existing choices of critical points could be further optimized incorporating explainability approaches.

In comparison to previous studies, our work is motivated by the following reasons:

Model operating principle: Part of the point-shifting methods also deceived the classifier by perturbing critical points, however, we argue that their selection of critical points is flawed. Since most of the selection methods for critical points are based on gradients

only, and existing studies [58], [84] have demonstrated that raw gradients suffer from saturation issues and are therefore biased. On the other hand, [139] demonstrated that feature attributions for point cloud classification networks are extremely sparse, while no work has specifically studied how these attributions are distributed among the critical points as well as their impact on the prediction sensitivity.

Potential for explainability: Another possibility of one-dimensional perturbations is explainability. The explainability method called counterfactuals alters the prediction label by perturbing the input features to provide a convincing explanation to the users. Previous research has documented that humans are more receptive to counterfactuals with sparse-dimensional perturbations [14], [21], [146]. For high-dimensional decision boundaries like point clouds, reduction of perturbation dimension is an important way to enhance the comprehensibility of the vicinity, which can be regarded as "cutting the input space using very low-dimensional slices" [117]. Furthermore, by incorporating part semantics, our approach has the potential to be extended for generating high-quality counterfactuals. Moreover, ours require only the access of gradients and no additional generative models, and are therefore more intrinsically explainable.

Altogether, the contributions of this work can be summarized as follows:

- We propose two explainability methods-based adversarial attacks: One Point Attack (OPA) and Critical Traversal Attack (CTA). Incorporating the attribution from explainable AI, our methods fool the popular point cloud networks with high success rate. Supported by extensive experiments, a significant margin is established with existing approaches in terms of the perturbation sparsity.
- We investigate diverse pooling architectures as alternatives to existing point cloud networks, which have an impact on the internal vulnerability against critical points shifting.
- We discuss the research potential of adversarial attacks from an explainability perspective, and present an application of our methods on facilitating the evaluation of explainability approaches.

The rest of the chapter is organized as follows: We introduce the related research of point cloud attacks in Sec. 10.2, then we detailed our proposed methods in Sec. 10.3. In Sec. 10.4, we present the visualization of the adversarial examples and demonstrate comparative results with existing studies. In Sec. 10.5 we discuss interesting observations derived from experiments with respect to robustness and explainability. We finally summarize our work in Sec. 10.6.

10.2. Related Work

As the first work [29] on adversarial examples was presented, an increasing variety of attacks against 2D image neural networks followed [30], [49], [51], [62], [68], [88]. However, due to the structural distinctions with point cloud networks, we do not elaborate on

the attack methods of image DNNs. Relevant information about image adversarial examples refers to [85]. It is notable that [117] investigated one-pixel attack for fooling image DNNs and also aimed at exploring the boundary of inputs. Nevertheless, their approach is a black-box attack based on an evolutionary algorithm, which is essentially distinct from ours.

Existing point cloud attacks are generally categorized into two classes: (i) *Shape-perceptible* generation, which generates human-recognizable adversarial examples with consecutive surfaces or meshes via generative models or spacial geometric transformations [140], [148], [149], [158], [161], [162], [170], [178]. (ii) *Point-shifting* perturbation, which regularize the distance or dimension of the point-wise shifting via perturbing or gradient-aware white-box algorithms [108], [120], [121], [126], [153], [169]. Point-wise perturbations, especially gradient-aware attack methodologies, enable more intrinsic explorations of the model such as stabilities and decision boundaries. On the other hand, from the perspective of explainability, the majority of generative models contain complex network structures that are inherently unexplainable. Utilizing their output to interpret another model is counter-intuitive.

The conception "**critical points**" has been discussed by several previous studies as well as the PointNet proposer [69], which forms the skeletons of the input instances in the classification processes. Existing methods [120], [121], [126], [169] extract the critical points by tracing the ones that remain active from the pooling layer, or by observing the gradient-based saliency maps. While such approaches succeed in generating adversarial examples with minor perturbation distances and sparse shift dimensions, we argue that their modules for selecting critical points can be further optimized. Due to the subsequent FC layers, it is difficult to determine whether the surviving points from the pooling layer conclusively make significant contributions to the prediction. Besides, saliency maps based on raw gradients are defective [58], [84]. The above factors may result in the involvement of fake critical points or omission of real ones during the perturbation process, which severely impairs the performance of the adversarial algorithms.

Explainability has been gaining attention in recent years. Popular explainability methods can be broadly categorized into gradient-based [28], [33], [35], [76], [77], [78], which requires the access of the gradient information, and perturbation-based [54], [65], [96], which is model-agnostic. In addition, counterfactual explanations [99] is proposed for tabular data by modifying selected features to induce the model to make different predictions. The properties of counterfactuals are identical to the adversarial examples, therefore attack methods may possess similar explainability potentials [132].

10.3. Methods

In this section, we formulate the adversarial problem in general and introduce the critical points set (Subsec. 10.3.1). We present our new attack approaches (Subsec. 10.3.2).

10.3.1. Problem Statement

Let $P \in \mathbb{R}^{N \times D}$ denotes the given point cloud instance, $F_{PC} : P \rightarrow y$ denotes the chosen point cloud neural network and $M(a, b) : \mathbb{R}^{n_a \times d} \times \mathbb{R}^{n_b \times d}$ denotes the perturbation matrix between instance a and b. The goal of this work is to generate an adversarial examples $P' \in \mathbb{R}^{n' \times d}$ which satisfies:

$$\begin{aligned} argmin(\big|\{m \in M(P, P'))|m \neq 0\}\big| \\ + \big\|M(P, P'))\big\|) : F_{PC}(P') \neq F_{PC}(P) \end{aligned} \tag{10.1}$$

Note that among the three popular attack methods for point cloud data: point adding ($n' > n$), point detaching ($n' < n$) and point shifting ($n' = n$), this work considers point shifting only.

We address the adversarial task in equation 10.1 as a gradient optimization problem. We minimize the loss on the input point cloud instance while freezing all parameters of the network:

$$L = \alpha \times Z[F_{PC}(P)] + \beta \times D(P, P') \tag{10.2}$$

where α indicates the optimization rate, $Z[F_{PC}(P)]$ indicates the neuron unit corresponding to the prediction $F_{PC}(P)$ which guaranties the alteration of prediction, $D(P, P')$ represents the quantized imperceptibility between the input P and the adversarial example P' and β is the distance penalizing weight. The imperceptibility has two major components, namely the perturbation magnitude and the perturbation sparsity. The perturbation magnitude can be constrained in three ways: Chamfer distance (equation 10.3), Hausdorff distance (equation 10.4) or simply Euclidean distance. We ensure perturbation sparsity by simply masking the gradient matrix, and with the help of the saliency map derived by the explainability method we only need to shift those points that contribute positively to the prediction to change the classification results, which are termed as "critical points set".

Critical points set: The concept was first discussed by its proposer [69], which contributes to the features of the max-pooling layer and summarizes the skeleton shape of the input objects. They demonstrated an *upper-bound* construction and proved that corruptions falling between the *critical set* and the *upper-bound* shape pose no impact on the predictions of the model. However, the impairment of shifting those critical points is not sufficiently discussed. Previous adversarial researches studied the model robustness by perturbing or dropping critical points set identified through monitoring the max-pooling layer or accumulating loss of gradients [120], [121], [126], [169]. Nevertheless, capturing the output of the max-pooling layer struggles to identify the real critical points set due to the lack of transparency in the high-level structures (e.g., multiple MLPs following the pooling layer), while saliency maps based on raw gradients suffer from saturation [58], [84], both of which severely compromise the filtering of the critical point set. We therefore introduce IG [78], the state-of-the-art gradient-based explainability approach, to further investigate the sensitivity and robustness to the critical points set. The formulation of IG is summarized in equation 2.1.

Similarity metrics for point cloud data: Due to the irregularity of point clouds, Manhattan and Euclidean distance are both no longer applicable when measuring the similarity between point cloud instances. Several previous works introduce Chamfer [121], [124], [148], [149], [162], [169], [178] and Hausdorff [121], [124], [127], [149], [162], [169] distances to represent the imperceptibility of adversarial examples. The measurements are formulated as:

- Bidirectional Chamfer distance

$$\begin{aligned} D_{CH}^2(P_a, P_b) = &\frac{1}{|P_a|} \sum_{p_m \in P_a} \min_{p_n \in P_b} \|p_m - p_n\|_2 \\ &+ \frac{1}{|P_b|} \sum_{p_n \in P_b} \min_{p_m \in P_a} \|p_n - p_m\|_2 \end{aligned} \tag{10.3}$$

- Bidirectional Hausdorff distance

$$\begin{aligned} D_h(P_a, P_b) = \max(&\max(\min_{p_n \in P_b} \|p_m - p_n\|_2), \\ &max(\min_{p_m \in P_a} \|p_n - p_m\|_2)) \end{aligned} \tag{10.4}$$

10.3.2. Attack Algorithms

One-Point Attack (OPA): Specifically, OPA (see algorithm 10.1 for pseudo-code) is an extreme of restricting the number of perturbed points, which requires:

$$\left|\{m \in M(P, P')) | m \neq 0\}\right| = 1 \tag{10.5}$$

We acquire the gradients that minimize the activation unit corresponding to the original prediction, and a saliency map based on the input point cloud instance from the explanation generated by IG. We sort the saliency map and select the point with the top-n attribution as the critical points ($n = 1$ for OPA), and mask all points excluding the critical one on the gradients matrix according to its index. Subsequently the critical points are shifted with an iterative optimization process. An optional distance penalty term can be inserted into the optimization objective to regularize the perturbation magnitude and enhance the imperceptibility of the adversarial examples. We choose Adam [31] as the optimizer, which exhibits better performance for optimization experiments. The optimization process may stagnate by falling into a local optimum, hence we treat every 25 steps as a recording period, and the masked Gaussian noise weighted by W_n is introduced into the perturbed points when the average of the target activation at period $k+1$ is greater than at period k. For the consideration of time cost, the optimization process is terminated when certain conditions are fulfilled and the attack to the current instance is deemed as a failure.

Critical Traversal Attack (CTA): Due to the uneven vulnerability of different point cloud instances, heuristically setting a uniform sparsity restriction for the critical points perturbation is challenging. CTA (pseudo-code presented in algorithm 10.2) enables the constraint of perturbation sparsity to be adaptive by attempting the minimum number of

perturbed points for each instance subject to a successful attack. The idea of CTA is starting with the number of perturbed points n as 1 and increasing by 1 for each local failure until the prediction is successfully attacked or globally failed. Similarly, we consider the saliency map generated by IG as the selection criterion for critical points, and the alternative perturbed points are incremented from top-1 to all positively attributed points. Again, for accelerating optimization we also select Adam [31] as the optimizer. Since most point cloud instances can be successfully attacked by one-point shifting through the introduction of Gaussian noise in the optimization process, we discarded the noise-adding mechanism in CTA to distinguish the experiment results from OPA. The aforementioned local failure stands for terminating the current n-points attack and starting another $n + 1$ round, while the global failure indicates that for the current instance the attack has failed.

Algorithm 10.1 N_p-critical Point(s) Attack. ($N_p = 1$ for OPA)

Input: $P \rightarrow N \times D$ point cloud data, $f \rightarrow$ point cloud neural network, $\alpha \rightarrow$ Optimizing rate, $\beta \rightarrow$ Weight for constrain the perturbing distance(optional), $D \rightarrow$ Distance calculating function(optional), $N_p \rightarrow$ Number of shifting points(1 for *One-point attack*), $W_n \rightarrow$ Gaussian noise weights

Output: $P_{adv} \rightarrow N \times D$ Adversarial example

$A_{idx} = Argsort(IG(P, f))$ // Get IG mask of P

$R_s = list()$ // Activation Recorder

$I_{cur} = 1$ // Current iteration

while *True* **do**

  $a_p \leftarrow F_{PC}(P)$ // Current activation of predicted class

  $G = \alpha * A_p + \beta * D(P_{adv}, P)$ // Add distance constrain(Optional)

  $P_{adv} = Adam(P_{adv}, G[A_{idx}[0 : N_p]])$ // Adam optimizing N points

  I_{cur} += 1

  $R_s.append(a_p)$

  /* Add masked Gaussian random noise if activation descending stopped */

  if $R_s[t] < R_s[t + 1]$ **then**

    P_{adv} += $W_n \times GaussianRandom(P_\delta)[A_{idx}[0 : N_p]]$

  end

  /* Success if predict class changed */

  if $max(a)! = pred$ **then**

    return P_{adv}

  end

  /* Fails if the stopping conditions related to R_a and I_{cur} are fulfilled */

  if *Stopping criteria are fulfilled* **then**

    return *Failed*

  end

end

Algorithm 10.2 Critical Traversal Attack (CTA)

Input: $P \rightarrow N \times D$ point cloud data, $f \rightarrow$ point cloud neural network, $\alpha \rightarrow$ Optimizing rate, $\beta \rightarrow$ Weight for constrain the perturbing distance(optional), $D \rightarrow$ Distance measuring function(optional)

Output: $P_{adv} \rightarrow N \times D$ Adversarial example

$A_{idx} = Argsort(IG(P, f))$ // Get IG mask of P

$Num_{pos} = count(IG(P, f) > 0)$ // # Points with attribution >0

$R_s = list()$

$I_{cur} = 1$

for N_p *from* 1 *to* Num_{pos} **do**

  while *True* **do**

    $a_p \leftarrow F_{PC}(P)$ // Activation of predicted class

    $G = \alpha * A_p + \beta * D(P_{adv}, P)$ // Add distance constrain(Optional)

    $P_{adv} = Adam(P_{adv}, G[A_{idx}[0 : N_p]])$ // Adam optimizing N points

    I_{cur} += 1

    $R_s.append(a_p)$

    /* Success if predict class changed */

    if $argmax(a_p) != pred$ **then**

      return P_{adv}

    end

    /* Current N_p round fails if the local stopping conditions related to R_a and I_{cur} are fulfilled */

    if ***Local stopping criteria*** *fulfilled* **then**

      break;

    end

  end

  /* Current instance fails if the global stopping conditions are fulfilled */

  if ***Global stopping criteria*** *fulfilled* **then**

    return *Failed*

  end

  return *Failed*

end

10.4. Experiments

In this section, we present and analyze the results of the proposed attack approaches. We demonstrate quantitative adversarial examples in Subsec. 10.4.2 and scrutinize the qualitative result in Subsec. 10.4.1. Our experiments[2] are primarily conducted on PointNet [69], which in general achieves an overall accuracy of 87.1% for the classification task on ModelNet40. Moreover, we extended our approaches on the most popular point cloud network PointNet++ [73] and DGCNN [119], which outperform the point cloud classification task with 90.7% and 92.2% accuracies respectively. We also experiment on PointMLP [184], the state-of-the-art classification model to date, which achieves the best accuracy of 94.5% on ModelNet40. Modelnet40 [45], our main experimental dataset, contains 12311 CAD models (9843 for training and 2468 for evaluation) from 40 common categories, and is currently the most widely-applied point cloud classification dataset. We randomly sampled 25 instances for each class from the test set, and then selected those instances that are correctly predicted by the model as our victim samples. When configuring parameters, the optimization rate α is empirically set to 10^{-6}, which performs as the most suitable step size for PointNet after grid search. Specifically for OPA, we set the Gaussian weight W_n to 10^{-1}, which proved to be the most suitable configuration. For CTA, we investigate both $\beta = 0$ and $1e-3$. We also validate our methods on ShapeNet [36] dataset. All attacks performed in this section are non-targeted unless specifically mentioned. In all experiments, we only compare the performance among **point-shifting** attacks, motivated by exploring the peculiarities of point cloud networks. Though previous shape-perceptible approaches such as [140], [148], [161], [162], [178] also addressed adversarial studies of point clouds, they were devoted to generate adversarial instances with human-perceptible geometries. Therefore, comparison of perturbation distances and dimensions with their works is not relevant.

10.4.1. Quantitative evaluations and comparisons

In this section, we compare the imperceptibility of proposed methods with existing attacks via measuring Hausdorff and Chamfer distances as well as the number of shifted points, and demonstrate their transferability among different popular point cloud networks. Additionally, we show that CTA maintains a remarkably high success rate even after converting to targeted attacks.

Imperceptibility: We compare the quality of generated adversarial examples with other **point-shifting** researches under the aspect of success rate, Chamfer and Hausdorff distances, and the number of points perturbed. As Tab. 10.1 shows, compared to the approaches favoring to restrict the perturbation magnitude, despite the relative laxity in controlling the distance between the adversarial examples and the input instances, our methods prevail significantly in terms of the sparsity of the perturbation matrix. Simultaneously, our methods achieve a higher success rate, implying that the network can be fooled for almost all point cloud instances by shifting a minuscule amount of points (even

[2] Our code is available at `https://github.com/Explain3D/Exp-One-Point-Atk-PC`

	S(↑)	$D^2_{CH}(\downarrow)$	$D_h(\downarrow)$	$N_p(\downarrow)$
L_p Norm [121]	85.9	1.77×10^{-4}	2.38×10^{-2}	967
Minimal selection [169]	89.4	1.55×10^{-4}	1.88×10^{-2}	36
Adversarial sink [149]	88.3	7.65×10^{-3}	1.92×10^{-1}	1024
Adversarial stick [149]	83.7	4.93×10^{-3}	1.49×10^{-1}	210
Random selection [120]	55.6	7.47×10^{-4}	2.49×10^{-3}	413
Critical selection [120]	19.0	$\mathbf{1.15 \times 10^{-4}}$	9.39×10^{-3}	50
Critical frequency [126]	63.2	5.72×10^{-4}	2.50×10^{-3}	303
Saliency map/L [126]	56.0	6.47×10^{-4}	2.50×10^{-3}	358
Saliency map/H [126]	58.4	7.52×10^{-4}	$\mathbf{2.48 \times 10^{-3}}$	424
Ours (OPA)	98.7	8.64×10^{-4}	8.45×10^{-1}	**1**
Ours ($CTA_{\beta=0}$)	**100**	8.92×10^{-4}	8.19×10^{-1}	2
Ours ($CTA_{\beta=1e-3}$)	99.6	7.73×10^{-4}	6.68×10^{-1}	6

Table 10.1.: Comparison of existing point-shifting adversarial generation approaches for PointNet, where S, D^2_{CH}, D_h and N_p denote the success rate, Chamfer and Hausdorff distances and the number of shifted points respectively. Part of the records sourced from [169]. It is worth noting that we only compare the gradient-based **point-shifting** competitors. The upward (↑) and downward (↓) arrows indicate whether a larger or smaller value is better, respectively.

	Dataset	S(↑)	$D^2_{CH}(\downarrow)$	$D_h(\downarrow)$	$N_p(\downarrow)$
OPA	ModelNet40	98.7	8.64×10^{-4}	8.45×10^{-1}	1
	ShapeNet	95.1	8.39×10^{-4}	8.06×10^{-1}	1
CTA	ModelNet40	100	8.92×10^{-4}	8.19×10^{-1}	2
	ShapeNet	100	8.91×10^{-4}	7.26×10^{-1}	3

Table 10.2.: Comparison of attack results with ModelNet40 and ShapeNet dataset.

one). Note that while calculating D^2_{CH} and D_h, we employ the L_2-norm. Therefore, despite the large Hausdorff distance, the average perturbation magnitude along each axis is $\mathbf{0.488}$. Considering that each axis of ModelNet40 is regularized into the interval $[-1, 1]$, this magnitude occupies $\mathbf{24.4}\%$ of the interval, which corresponds to an average perturbation of 62.2 gray values in 2D grayscale images. We thus consider the perturbation magnitude to be acceptable.

To eliminate potential bias, we also test the proposed attack methods with ShapeNet [36] dataset. As Tab. 10.2 presents, our approaches perform similarly on the two different datasets, and therefore the vulnerable bias in the data distribution of ModelNet40 can be basically excluded.

In addition to PointNet, we also tested the performance of our proposed methods on point cloud networks with different architectures. Tab. 10.3 summarize the result of attack PointNet, PointNet++, DGCNN and PointMLP with both OPA and CTA respectively. Both OPA and CTA achieve high success rate fooling those networks while only a single-

	$Model$	$S(\uparrow)$	$D^2_{CH}(\downarrow)$	$D_h(\downarrow)$	$N_p(\downarrow)$
OPA	PN [69]	98.7	8.45×10^{-4}	8.64×10^{-1}	1
	PN++ [73]	99.1	1.58×10^{-2}	1.61×10^{1}	1
	DGCNN [119]	90.9	1.70×10^{-3}	1.69	1
	PointMLP [184]	52.9	1.91×10^{-3}	1.90	1
CTA	PN [69]	100	8.92×10^{-4}	8.19×10^{-1}	6
	PN++ [73]	99.5	1.22×10^{-2}	8.90	6
	DGCNN [119]	100	2.13×10^{-3}	1.48	3
	PointMLP [184]	99.8	3.77×10^{-3}	9.83×10^{-1}	13

Table 10.3.: Comparison of attack results on PN(PointNet), PN++(PointNet++), DGCNN and PointMLP.

digit number of points are shifted. PointMLP seems to be the most stable model, and we speculate that this is attributed to the affine module of relative position [184]. Intuitively, point cloud neural networks appear to be more vulnerable compared to image CNNs ([117] is a roughly comparable study since they also performed one-pixel attack with the highest success rate of 71.66%. An opposite conclusion has been drawn by [121], they trained the PointNet with 2D data and compared its robustness with 2D CNNs against adversarial images. Nevertheless, we argue that the adversarial examples are generated by attacking a 2D CNN, such attacks may not be aggressive for PointNet, which is specifically designed for point clouds.

Transferability: We investigate the transferability of proposed attacks across different point cloud networks by feeding the adversarial examples generated by one network to the others and recording the success rate. Fig. 10.1 presents the adversarial transferability between PointNet, PointNet++, DGCNN and PointMLP. What stands out in the figure is that PointNet++, DGCNN, PointMLP show strong stability against the adversarial examples from PointNet. We believe this is because the aggregated adjacency features disperse the attribution of a single point. Recall the *EdgeConv* [119] in DGCNN, which extracts adjacent features in both point and latent spaces, while PointNet++ possesses a similar module that aggregates neighboring points [69], which can be considered as a point-space-only *EdgeConv*. Such an integration distributes the feature contribution to multiple adjacent points, and a modest shifting of one point has limited impacts on the aggregated cluster. For PointMLP, this module transforms into affines of relative-position encodings [184]. Despite the involvement of adjacent points information, the features of relative positions may be severely corrupted if the centroids are perturbed. However, the feature extractor in PointNet can also be regarded as a special *EdgeConv* with $K = 1$, preserving the location information of the central point only, and therefore is more sensitive to the perturbation. Surprisingly, PointNet++ performs stably against adversarial examples from DGCNN and PointMLP, while the opposite fails. We consider the stability of PointNet++ stems from the *multi-scale(resolution) grouping*, where latent features are concatenated by grouping layers at different scales, resulting in more points involved in the aggregation.

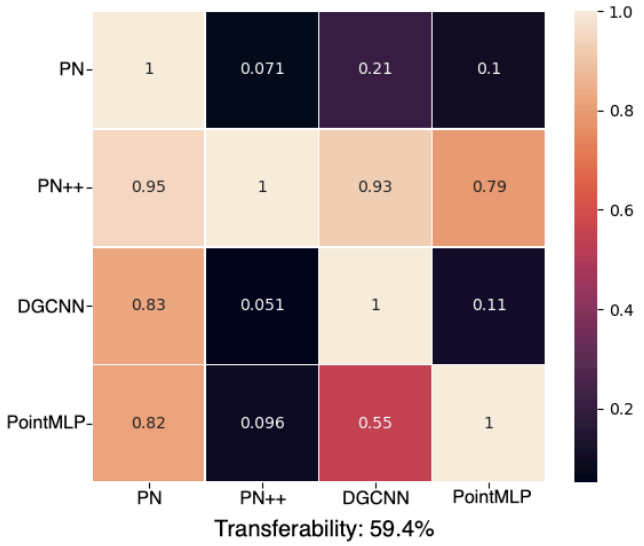

Figure 10.1.: Transferability for PointNet, PointNet++, DGCNN and PointMLP. Networks on the rows and columns denote from which victim networks the adversarial examples are generated and to which those examples are transferred respectively. Brighter squares denote higher transferabilities. The total transferabilities under the matrices are the averages of the off-diagonal values of corresponding methods.

Targeted attack: We also extend the proposed methods to targeted attacks. To alleviate redundant experiment procedures, we employ three alternatives of conducting ergodic targeted attack: *random*, *lowest* and *second-largest* activation attack. In the random activation attack we choose one stochastic target from the 39 labels (excluding the ground-truth one) as the optimization destination. In the lowest and second-largest activation attack, we decrease the activation of ground truth while boosting the lowest or second-largest activation respectively until it becomes the largest one in the logits. The results, as shown in Tab. 10.4, indicate that though the performance of OPA is deteriorated when converting to targeted attacks due to the rigid restriction on the perturbation dimension, CTA survived even the worst case (the lowest activation attack) with a remarkably high success rate and a minuscule number of perturbation points. We also demonstrate the results from LG-GAN [162], which also dedicates to targeted attack for point cloud networks. In comparison, CTA achieves an approximated success rate with a much smaller D^2_{CH}. Note that their approach is based on generative models and the comparison is for reference only.

10.4.2. Adversarial examples visualization

Fig. 10.2 visualizes two adversarial examples for OPA and CTA respectively. Interestingly, in CTA, regardless of the absence of the restriction on the perturbation dimension, there are instances (e.g. the car in CTA) where only one-point shifting is required for an adversarial example.

	Pattern	S(↑)	$D^2_{CH}(\downarrow)$	$D_h(\downarrow)$	$N_p(\downarrow)$
O	Second-largest	58.5	9.49×10^{-4}	9.33×10^{-1}	1
P	Random	20.9	1.06×10^{-2}	1.08×10^{1}	1
A	Lowest	6.3	4.73×10^{-3}	4.80	1
C	Second-largest	99.5	1.55×10^{-3}	8.14×10^{-1}	5
T	Random	97.7	5.75×10^{-3}	2.31	10
A	Lowest	99.0	8.52×10^{-3}	3.06	13
LG-GAN [162]		98.3	3.80×10^{-2}	-	-

Table 10.4.: Targeted OPA and CTA on PointNet. Targeting all labels for each instance in the test set is time-consuming. Therefore, we generalize it with three substitutes: random, the second-largest and the lowest activation in the logits. We also show the results of LG-GAN as a reference.

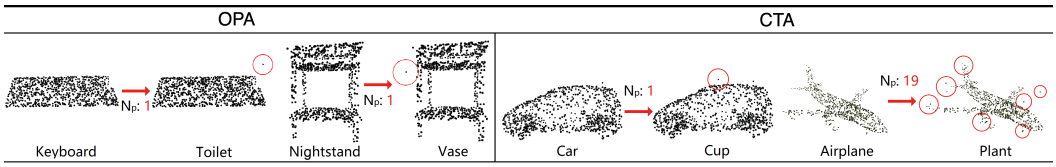

Figure 10.2.: Adversarial examples for OPA and CTA. N_p denotes how many points are shifted.

10.5. Discussion

In this section, we present our viewpoint concerning the robustness of point cloud networks (10.5.1) and discuss the potential of OPA from the viewpoint of explainability (10.5.2).

10.5.1. Structural stability of point cloud networks

Plenty of researches have discussed defense strategies against intentional attacks for point cloud networks [108], [124], [127], [149], [153], [162], [169], [178], the majority of which were with respect to embedded defense modules, such as outlier removal. However, there has been little discussion about the stability of the intrinsic architectures. Inspired by [153] who investigated the impacts of different pooling layers on the robustness, we replace the max-pooling in PointNet with multifarious pooling layers. As Tab. 10.5 shows, although PointNet with average and sum-pooling sacrifice 3.3% and 10.4% accuracies in the classification task, the success rates of OPA on them plummet from 98.7% to 44.8% and 16.7% respectively, and the requested perturbation magnitudes are dramatically increased, which stands for enhanced stabilization. We speculate that it depends on how many points from the input instances the model employs as bases for predictions. We calculate the normalized IG contributions of all points from the instances correctly predicted among the 2468 test instances, and we also introduce the Gini coefficient [15] to quantify the dispersion of the absolute attributions which is formulated as:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} ||a_i| - |a_j||}{2n^2 |\bar{a}|} \tag{10.6}$$

	Acc.	S↑)	$D_c(\downarrow)$	$D_h(\downarrow)$	N_{pos}	Gini.
Max-pooling	87.1	98.7	8.45×10^{-4}	8.64×10^{-1}	397.2	0.91
Average-pooling	83.8	44.8	2.94×10^{-3}	2.96	718.5	0.53
Median-pooling	74.5	0.9	1.28×10^{-4}	9.55×10^{-2}	548.1	0.57
Sum-pooling	76.7	16.7	2.50×10^{-3}	2.53	868.2	0.49

Table 10.5.: Model accuracies, success attacking rates, average Chamfer and Hausdorff distances of OPA on PointNet with max, average, median and sum-pooling on the last layer respectively. The evaluation accuracy is also presented in the second column. N_{pos} denotes how many points are positively attributed to the prediction, and Gini. denotes the Gini coefficient of the corresponding attribution distributions.

where a is the attribution mask generated by IG. We demonstrate the corresponding results in Tab. 10.5 and 10.6. There are significant distributional distinctions between the max, average and sum-pooling architectures. PointNet with average and sum-poolings adopt 70.18% (718.5 points) and 84.78% (868.2 points) of the points to positively sustain the corresponding predictions, where the percentages of points attributed to the top 20% are 0.65% (6.7 points) and 1.16% (11.9 points), respectively, while these proportions are only 38.79% (397.2 points) and 0.15% (1.5 points) in the max-pooling structured PointNet. Moreover, the Gini coefficients reveal that in comparison to the more even distribution of attributions in average (0.53) and sum-pooling (0.49), the dominant majority of attributions in PointNet with max-pooling are concentrated in a minuscule number of points (0.91). Hence, it could conceivably be hypothesized that for point cloud networks, involving and apportioning the attribution across more points in prediction would somewhat alleviate the impact of corruption at individual points on decision outcomes, and thus facilitate the robustness of the networks. Surprisingly, median-pooling appears to be an exception. While the success rate of OPA is as low as 0.9%, the generated adversarial examples only require perturbing $D_h = 9.55 \times 10^{-2}$ in average (all experiments sharing the same parameters, i.e. without any distance penalty attached). On the other hand, despite that merely 53.53% (548.1) points are positively attributed to the corresponding predictions, with only 0.23% (2.4 points) of them belonging to the top 20%, which is significantly lower than the average and sum-pooling architectures, median-pooling is almost completely immune to the deception of OPA. We believe that median-pooling is insensitive to extreme values, therefore the stability to perturbations of a single point is dramatically reinforced.

10.5.2. Towards explainable point cloud models

Despite the massive number of adversarial methods that contribute to the model robustness for computer vision tasks, to our best knowledge, none has discussed the explainability of point cloud networks. However, we believe that the adversarial methods can

	Top 20%	Top 40%	Positive
Max-pooling	0.15%	0.23%	38.79%
Average-pooling	0.65%	2.12%	70.18%
Median-pooling	0.23%	0.59%	53.53%
Sum-pooling	1.16%	4.53%	84.78%

Table 10.6.: Overview of the percentage of top-20%, top-40% and positive attributed points with four different pooling layers.

facilitate the explainability of the models to some extent. Recall the roles of counterfactuals in investigating the explainability of models processing tabular data [99]. Counterfactuals provide explanations for chosen decisions by describing what changes on the input would lead to an alternative prediction while minimizing the magnitude of the changes to preserve the fidelity, which is identical to the process of generating adversarial examples [134]. Unfortunately, owing to the multidimensional geometric information that is unacceptable to the human brain, existing image-oriented approaches addressed the counterfactual explanations only at the semantic level [102], [157].

Several studies have documented that a better counterfactual needs to be sparse because of the limitations on human category learning [146] and working memory [14], [21]. In addition, previous adversarial studies on images have also suggested that unidimensional perturbations contribute to depicting relatively perceptible vicinities and boundaries [117]. Fig. 10.3 compares the visualization of multidimensional and unidimensional perturbations. The latter, though larger in magnitude, shows more clearly the perturbation process of the prediction from ”car” to ”radio”, and makes it easier to perceive the decision boundary. Conversely, while higher dimensional perturbations perform better on imperceptibility for humans, they are more difficult for understanding the working principles of the model.

In addition, we found another application where the proposed method facilitate the explainability. Evaluating explanations is a major challenge for explainability studies due to the lack of ground truth [163]. An intuitive idea is sensitivity testing, i.e., perturbing features in the explanation that possess high attributions and observing whether the prediction results dramatically change. Theoretically, in our methods, a more accurate explanation induces a more precise selection of critical points, and therefore a higher success rate when perturbing them for generating adversarial examples. Tab. 10.7 presents the attack performance utilizing gradient-based explainability methods: Vanilla Gradients [32], Guided Back-propagation [41] and IG as the critical identifier respectively. Our results are consistent with [139] and [1], the performance of IG is comparatively better than that of Vanilla Gradients and Back-propagation.

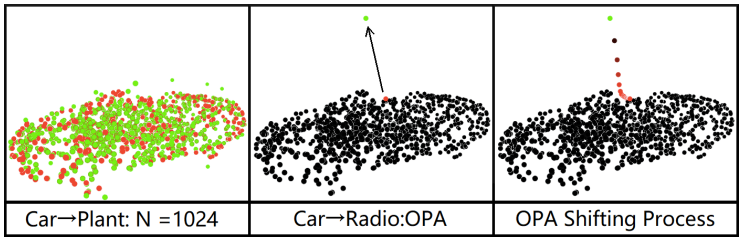

Figure 10.3.: Intuitive visualization of multidimensional shifting(left), unidimensional OPA shifting(middle) and the shifting process of OPA(right). In the right plot, the redder the point the higher the confidence for label "car". In the right plot the green point indicates that the prediction is altered.

Mtds.	S(↑)	$D_c(\downarrow)$	$D_h(\downarrow)$	$N_p(\downarrow)$
VG	82.5	8.20×10^{-4}	8.01×10^{-1}	1
GB	83.4	8.21×10^{-4}	8.02×10^{-1}	1
IG	**98.7**	8.64×10^{-4}	8.45×10^{-1}	1

Table 10.7.: OPA performance utilizing various gradient-based explainability methods to identify the critical points, where VG, GB and IG denote Vanilla Gradients [32], Guided Backpropagation [41] and Integrated Gradients respectively.

10.6. Conclusion

As the first attack methods for point cloud networks incorporating explainability, we demonstrate the importance of individual critical points by analyzing the impact of perturbations of critical points in the input point cloud on the prediction. Such importance raises the question of whether critical points are the sole basis for point cloud models to make decisions? In the next chapter, we delve into the intermediate layers inside the model to shed light on this question. We analyze the correlations between the learned spatial geometric contours and the latent feature in the intermediate layers of the model, and illustrate an observation that the contours of objects are learned by the model and recorded in the activation distribution of the intermediate layers. Based on this observation, we propose an activation maximization method that does not rely on generative models, while still reconstructing the contours of the real objects with high quality.

11. Do Point Cloud Models Learn Object Contour Features?

In the previous chapter we analyze the impact of critical points on prediction through an adversarial black-box approach, which demonstrates that the prediction of point cloud models strongly depends on the location of those critical points. This observation raises the question: Are point cloud models learning the spatial geometry contours of objects, or are they simply predicting based on critical points? In this chapter, we propose a novel observation: the point cloud models with PointNet-like structure effectively learn the contours of 3D objects in their intermediate layers. This observation presents a counterfactual for the existing view that point cloud models make decisions based on critical points. Besides, it analyzes the learned contour features of point cloud models and provides a novel inspiration for point cloud synthesis. Based on this observation, we further propose an application called "Latent Activation Maximization" (Latent AM). Latent AM is a point cloud global explainability method which requires only the classifier to be explained and generates much higher quality explanations than other non-generative model-based AMs. We believe this observation will be an important contribution to future research in the areas of point cloud network transparency as well as 3D instance synthesis.

11.1. Introduction

In Sec. 1.3.2, we discussed the necessity of analyzing the internal structure of point cloud models, and due to the architectural differences of point cloud models (see Chapter 6), investigating the latent features learned by these models may bring new perspectives to explainability research for point clouds. In this chapter, we present a novel observation: the intermediate layers of point cloud models with PointNet-like structure [69] effectively learn the contours of 3D objects. We reconstruct the profile of the corresponding category by aligning the latent vectors in the intermediate layers of the classifier without any external module (e.g., incorporating generative models).

To exhibit the potential of this observation, we further propose an explainability application, namely Latent AM. Latent AM is a global explainability method based on Activation Maximization (AM) [25], which significantly improves the quality of explanations compared to other non-generative model-based point cloud AM methods, while reducing the computational intensity and technical barriers to training generative models, and more

importantly, eliminating doubts about the information sources in the explanations compared to other generative model-based approaches. In summary, our contributions are as follows:

- We present an important observation that the intermediate layers of PointNet-like structured models effectively learn the contours of 3D objects. This observation has two important implications as follows: a) it provides a counterfactual to existing studies which consider that the predictions of point cloud models highly depend on critical points [126], [136], [139], [7], b) it offers a new potential for explainability studies of point cloud models based on latent features, and c) it opens up a novel way to synthesize human-perceptible point cloud instances without incorporating any generative model.
- Based on this observation, we provide an application, Latent AM, which generates global explanations with significantly higher quality than other non-generative model-based AM approaches. Despite the limited applicability, Latent AM still exhibits great potential of the proposed observation for explainability and synthesis.

11.2. Related Work

Point cloud explainability is so far not adequately investigated. [126] identify points that are critical for prediction through perturbation, and other studies [139], [1] attempt to transplant explainability methods from images so that they are also applicable to point clouds. [6] is the sole study of point cloud AM, which demonstrates that traditional regularizations (such as L_2-norm) are incapable of generating perceptible explanations, and proposes an autoencoder-based approach that samples and optimizes explanations from real object distributions. However, these approaches either treat the entire model as a black box, neglecting to delve into the internal properties of the classifier, or resort to external generative models, leading to uncertainty about the source of the information brought to the users. Moreover, existing transparency studies conclude that point cloud decisions are determined by a few critical points [126], [136], [139], [7]. We experimentally present a counterfactual that the model effectively learns the global contours of 3D objects and records the outline information in a fraction of its components.

11.3. Observation

11.3.1. Latent activation discrepancy (LAD)

We then elaborate on the observation of latent activation discrepancies. We train a PointNet [69] on ModelNet40 [45] and predict the instances from the testset. In predicting an instance P_c^1 from class c, we record the neuron activations of each intermediate layer, denoted as ψ_c^1. In parallel, we randomly select two other instances P_c^2 and $P_{c'}^3$ ($c \neq c'$) in the testset from the same and different categories as P_c^1, respectively, predict them

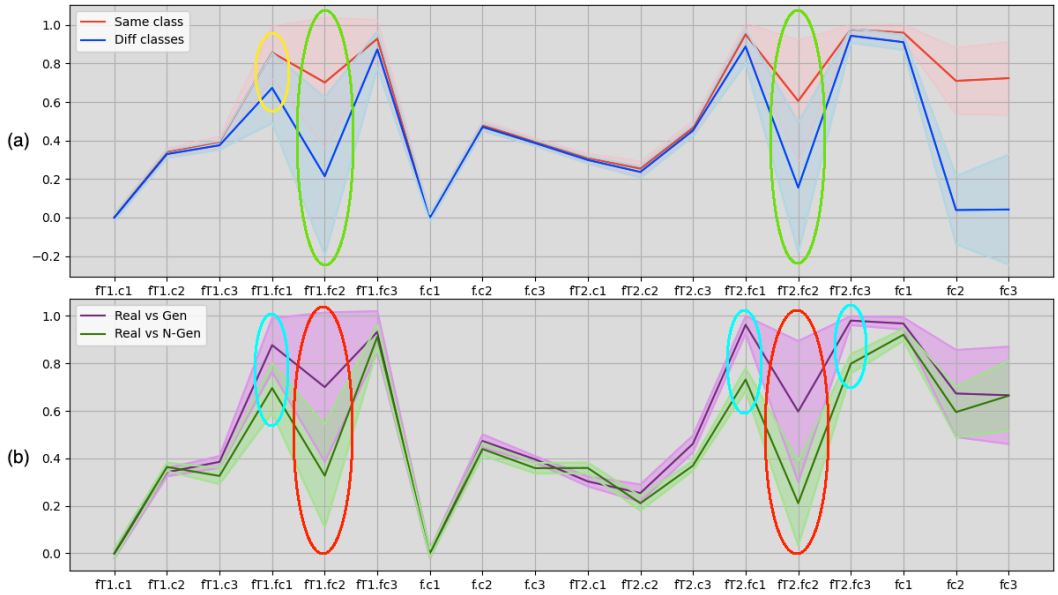

Figure 11.1.: Spearman's ρ in the activation of neurons within the model when predicting instances (a) from the same (red) and different classes (blue) and (b) from Real vs. N-Gen (green) and Real vs. Gen (purple), respectively. The y-axis indicates the cosine similarity and the x-axis is the name of each layer of PointNet. In the x-axis, *fT*, *fc*, and *c* denote feature Transform modules (T-net [69]), FC layers, and convolutional layers, respectively. For instance, *fT2.c3* represents the third convolutional layer in the second feature Transform module. Furthermore, we also exploit Cosine similarity and Pearson's coefficients to compute the similarity of activations at each layer, with results analogous to Spearman's ρ.

and obtain two additional activation sequences ψ_c^2 and $\psi_{c'}^3$, respectively. Subsequently, we assess the rankings similarities between $\{\psi_c^1, \psi_c^2\}$ and $\{\psi_c^1, \psi_{c'}^3\}$, respectively, with *Spearman's* ρ, obtaining two similarity ρ_{same}^1 and ρ_{diff}^1, which represent the similarity in the degree of neuron activations across all layers when the classifier predicts instances from the same class and different classes, respectively. We repeat the above process until all instances in the testset are evaluated once. Finally, we statistically average the two similarities respectively and demonstrate them in Fig. 11.1 (a).

As can be observed from the figure, the major difference in the activation of the intermediate layers in the classifier when predicting instances from the same and different categories is reflected in layer $fT1.fc2$ and $fT2.fc2$ (layers circled in green), and the secondary difference is reflected in the layer $fT1.fc1$ (layers circled in yellow). We term this difference as Latent activation discrepancy (LAD). We ignore the last two layers ($fc2$ and $fc3$) because they are related to differences in predictions (logits). The reason for the low similarity of all the convolutional layers is that they are 1×1 in size and serve to extract local point-wise features. In contrast, fully-connected (FC) layer is mostly set behind a max-pooling layer for extracting global features. Objects from the same class normally

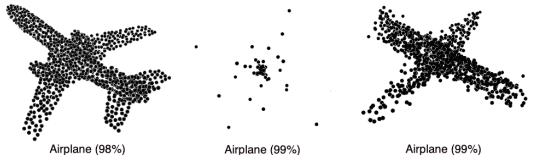

Figure 11.2.: Three different instance types, all of which are classified by the classifier as "Airplane" with high confidences. From left to right: real instances (Real), AM instances generated without regularization (N-Gen) and regularized by a generative model [6] (Gen), respectively.

have similar profiles and therefore share a higher intermediate activation similarity, and vice versa.

Based on the above observation we propose the conjecture that **contour information may be efficiently learned by those intermediate layers whose similarity varies widely in predicting inputs from same and different classes** (Conj. 1). We will verify the conjecture by further experiments in the next subsection.

11.3.2. LAD for synthesized vs. real data

To further verify the conjecture, we specialize in inputs that are classified by the model into an identical category. To preserve the predictions as consistent as possible, we synthesize point cloud instances leveraging the Activation Maximization (AM) [25] technique, which ensures that the synthesized instances are classified by the model with a confidence level that is approximate to or higher than real objects.

We obtain three types of data through sampling from the dataset and AM synthesis, which are: real objects (Real), objects synthesized by AM without regularization (N-Gen), and those regularized by an Autoencoder [6] (Gen). A group of examples is shown in Fig. 11.2. Note that the commonality of the above three inputs is that they are all classified into the same class by the classifier and win exceedingly high confidence. Their properties are that Real is sampled from the dataset and possesses realistic and natural contours, the shape of Gen is slightly flawed but approximately identical to Real and can be clearly recognized, whereas N-Gen does not have regular outline at all.

We set up two comparison groups, Real vs. N-Gen and Real vs. Gen, and repeat the experiments in Sec. 11.3.1 (again testing each instance in the testset), yielding the results shown in Fig. 11.1 (b). It can be observed that the activation similarity of Real vs. Gen is much higher than that of Real vs N-Gen on specific layers, and the layers where there is a large difference occur to be $fT1.fc2$ and $fT2.fc2$ as well (circled in red), consistent with the previous experiment. Minor differences exist in other layers, reflected in $fT1.fc1$, $fT2.fc1$ and $fT2.fc3$ (circled in cyan). Note that in this comparison, differences in predictions due to categories are almost ruled out, i.e., the ρ of the two curves at $fc2$ and $fc3$ are nearly identical. Additionally, we observe another interesting phenomenon, i.e., when we compare the two curves: instances from the same class with Real vs. Gen, and

instances from different classes with Real vs. N-Gen, respectively, we find that the two similarities are highly approximated.

From the above experiments we draw the following conclusions: for part of the intermediate layers in the classifier, a) when the classifier predicts instances from the same class, their activations are highly analogous, b) when the classifier predicts instances from different classes, they are activated with low similarity, c) when the classifier predicts real objects and instances synthesized with generative models, the result is similar to a), d) When the classifier predicts real objects and instances synthesized without generative models, the result is similar to b), and d) their activation cases are highly approximate when the classifier predicts instances of the same category and instances synthesized with generative models, as well as when predicting instances from different categories and those synthesized without generative models. With the a priori knowledge that instances of the same class share similar contours with those synthesized by generative models, and that contours differ more between instances of different classes and those synthesized by non-generative models, we verified Conj. 1. In addition, we propose a further conjecture: **Contours learned by the intermediate layers can be extracted through aligning those layers whose activations are dissimilar when the classifier predicts instances with distinct profiles** (Conj. 2).

11.3.3. PointNet-like architectures

We try various point cloud classifiers, but observe that only those with PointNet-like structures yield the above phenomena, specifically, PointNet [69] and PointNet++ [73]. We speculate the reason is that simpler structures retain a certain degree of transparency, whereas subsequently proposed classifiers lose the visibility of feature boundaries due to the overwhelming artificial features incorporated. Note that though this observation only occurs in individual classifiers, it might have great potential for explainability, as **the substructures of PointNet is not limited to serving as a classifier in subsequent researches**. Extensive existing works leverage the PointNet structure to extract point cloud latent features, e.g., Autoencoder [82], [6], generative quality assessment [154] and, more recently, point cloud DDPM models [172]. Our observations are also applicable to these extended models.

11.4. An Application: Latent AM

With the prior knowledge from the previous section, we propose a global explainability method for point clouds based on latent activation alignment: Latent AM. Two points need to be declared at the outset: a) **Latent AM is a derivative of analyzing the internal properties of PointNet**, which belongs to a part of aforementioned observation, hence it is only applicable to those classifiers with the observations described in Sec. 11.3.1 and 11.3.2. We recognize the limited applicability, but Latent AM is not the main contribution of this work. Latent AM is simply an application that demonstrates the potential of our

observations, leaving more extensions for future work. b) Latent AM requires access to the entire testset. This does not exceed the restrictions of non-generative model-based AM approaches. On the one hand, the global explanations of AM should be representative of the classifiers and the entire dataset, on the other hand, extensive image AM methods exploit real images to optimize the quality of their global explanations [46], [52], [53], [67]. Therefore, **the access to real objects does not imply that Latent AM possesses an unfair advantage over other non-generative model-based approaches**.

AM-generated global explanations must contain two factors, representativeness and perceptability. Representativeness indicates the extent to which an explanation reflects (for a given classifier) the general profile of a particular category, which is typically achieved by optimizing the input vectors until they highly activate a specific neuron in the final layer of the classifier. For example, the three instances in Fig. 11.2 all highly activate the neuron representing "Airplane", and thus they all exhibit strong representativeness. Perceptibility indicates the extent to which an explanation is recognizable by humans, i.e., its outline approximates real objects of the same class. For example, again in Fig. 11.2, the two instances on left and right possess strong perceptibility, while the one in the center barely does. Moreover, diversity is a plus for global explanations, which denotes the extent to which generated explanations differ from each other.

In order to fulfill the above criteria, Latent AM incorporates the following three modules:

I. Activating class neuron. Activating neurons of the target class (AM loss) optimizes the input vectors to be representative of a specific class. AM loss can be formulated as

$$L_{AM}^{c} = -a_i^l(F_{PC}, P) \tag{11.1}$$

where $-a_i^l(F_{PC}, P)$ denotes the i^{th} neuron (the i^{th} class) on the output layer l of the classifier F_{PC}.

II. Latent approximation of real objects. This module is an attempt at validation of Conj. 2. We begin by generating several imperceptible explanations for each category (whose profiles are analogous to the middle instances of Fig. 11.2) only utilizing Eq. 11.1 (raw AM). We then predict these generated instances with the classifier and record the intermediate layer activations by category, denoted as $\tilde{z}_c^S$, where S denotes all S^{th} layers and c denotes the class. Afterwards, we predict each object in the testset with the classifier and record the activations of all intermediate layers. Subsequently, we aggregate these records according to category and calculate the average activation for each class, denoted as z_c^S. According to Conj. 1, we compare $\tilde{z}_c^S$ and z_c^S and identify those layers $l \in S$ from which there are large discrepancies in activations. We then filter out the activation records of the corresponding layers z_c^l and $\tilde{z}_c^l$ from z_c^S and $\tilde{z}_c^S$, respectively.

The latent alignment regularization (Latent loss) in Latent AM is formulated as

$$L_{LA}^{c} = -\sum_{l_f \in l} \alpha_{l_f} \times CS(z_c^{l_f}, \tilde{z}^{l_f}) \tag{11.2}$$

where α_{l_f} denotes the regularization weight on layer l_f and CS denotes the cosine similarity. The rule for setting α_{l_f} is that the larger the difference in similarity, the larger the α_{l_f} corresponding to the layer, and vice versa.

III. Point continuity and smoothness. Eq. 11.1 tends to expand individual points outward during the optimization process, which causes the explanation to suffer from a small number of outliers that are severely off-center, with most of the rest (common points) clustered near the origin [6], [7] (see the middle instance of 11.2). We mitigate the issue in two aspects: limiting the expansion of outlying points and diminishing their distance from neighboring points, while shifting the rest of points away from the origin and spreading them out by increasing their distance from neighboring ones. We propose a continuity loss can be formulated as

$$L_C = \sum_{i=1}^{N_p} \left| B - (min\{\|p_i - p_j\|_2\}_{j \neq i} + \|p_i\|_2) \right| \tag{11.3}$$

where N_p and p_i denote the total number of points and the i^{th} point in the explanation, $min\{\|p_i - p_j\|_2\}_{j \neq i}$ is the distance from p_i to its nearest neighboring point, and B is the legal boundary of the dataset. The intuition behind continuous loss is illustrated in Fig. 11.3. Each point is impacted by two regularizations in the optimization process, i.e., absolute and relative distances, the former being $\|p_i\|_2$ in Eq. 11.3 which is the distance to the origin, and the latter being $min\{\|p_i - p_j\|_2\}_{j \neq i}$. For example, the location of point A in the figure is close to the legal limit ($B = 1$) of point cloud instances, with an absolute distance close to 1, as well as a large relative distance since the outer points are sparse. The optimizer reduces both its absolute and relative distances to relocate it back to the origin (the green arrow) and closer to its neighboring point C (the blue arrow). In contrast, the absolute and relative distances of the common point C are relatively small, and the optimizer raises them up so that the sum approaches 1. By balancing the two distances, points are evenly distributed on the surface of the generated explanations to enhance their perceptibility.

IV. Legal restriction. Point cloud datasets are subject to a legal spatial limitation, e.g., ModelNet40 is legal in the interval [-1,1], and instances containing points that exceed this limitation are treated as illegal inputs. To guarantee that most of the explanations are valid, we incorporate a simple regularization that restricts outer points from exceeding the legal limitation, which is frequently observed in point cloud AMs without regularization [7]. The legal restriction is formulated as

$$L_{LR} = \sum_{i=0}^{N_p} ReLU(|p_i| - B) \tag{11.4}$$

where B denotes the legal constrain of the dataset.

Finally, we aggregate the aforementioned loss terms as the optimization target for Latent AM. In practice, we observe that L_C and L_{LA}^c are conflicting to some extent, rendering

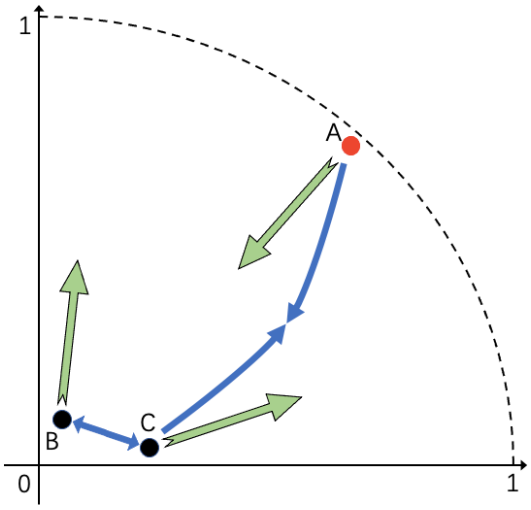

Figure 11.3.: The intuition of the point continuity loss. Point A is a critical point that tends to expand during AM, while points B and C are common points that are ignored by the gradient. The outer dashed curve is the boundary of a legal point cloud instance. The green and blue arrows indicate the direction of two different terms in the continuity loss, i.e., the distance to the origin and to the nearest neighboring point, respectively.

the optimization non-converging. We incorporate a trade-off coefficient β for these two terms, which is adjusted to balance the smoothness and closeness to real objects. The final loss function of Latent AM can be formulated as

$$L^c_{Latent} = L^c_{AM} + (1-\beta)L^c_{LA} + \beta L_C + L_{LR} \tag{11.5}$$

In addition to the loss terms, we employ the average initialization to improve the quality of the explanation [38]. We compute the average of the points in the test set for all objects by category and utilize it as the input to Latent AM, which improves the efficiency of the optimization process and the quality of the explanations. The overall process of Latent AM is demonstrated in Fig. 11.4 and Algo. 11.1.

11.5. Evaluation of Latent AM

In the experiments, we choose ModelNet40 and PointNet [69] as the main experimental datasets and models, respectively. ModelNet40 consists of 40 classes totaling 12,311 CAD models, where the training and test sets contain 9,843 and 2,468 object instances, respectively. PointNet is mainly composed of point-wise convolutional, pooling and fully connected layers, with a relatively straightforward structure, and is ideal for explainable analyses. We set the regularization weights in Eq. 11.2 to be $N_{l_f} \times 1e-9$, $1e-9$, $1e-1$, $1e-1$ and 1 for $fT1.fc1$, $fT1.fc2$, $fT2.fc1$, $fT2.fc2$ and $fc2$, respectively, where N_{l_f} is the total number of activations contained in this layer. In addition, we set $\beta = 0.1$ to

Algorithm 11.1 Latent AM with Latent alignment

Input : Classifier to be explained $F(\cdot)$, activations on latent layers $S = \{l_1, ..., l_k\}$ extraction module F: $H(F, S, \cdot)$, a set of real objects in class c: $R_c = \{r_c^1, ...r_c^n\}$, a threshold T, the learning rate θ and the number of optimization steps I

Output: Global explanation x_c^* for class c

x_c^o = Vanilla AM(random_Ini) #Raw prototype

$\dot{z}_c^S = H(F, S, x_c^o)$ # Raw activations

$z_c^S = \frac{\sum_{m=1}^{n} H(F,S,r_c^m)}{n}$ # Average of real latent activations

Select $l \in S$ s.t. $CS(\dot{z}_c^l, z^l) < T$ #Layers with low similarity as targets (CS: Cosine Similarity)

x_c^0 = Avg_Data_Ini() # Average initialization

for $i = 1$ to I do: #Flow optimization

$x_c^i = x_c^{i-1} + \theta * \frac{\partial L_{Latent}^c}{\partial x_c^i}$ # Loss from Eq. 11.5

end for

$x_c^* = x_c^I$

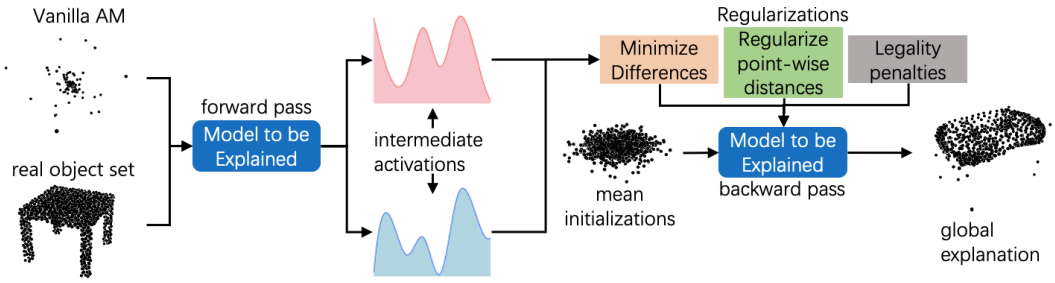

Figure 11.4.: Overview of the structure of Flow AM. Raw AM represents the explanations generated by AM without any regularization.

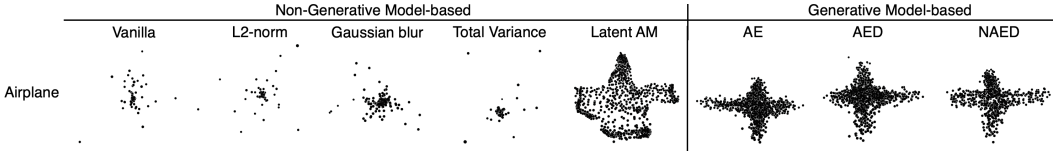

Figure 11.5.: Qualitative comparison of non-generative model-based global explanations. From left to right are, no regularization, L_2, Gaussian blur, total variance, latent regularizations (ours), AE [6], AED [6] and AED [6]. Note that the first five approaches are based on non-generative models while the last three are based on generative models.

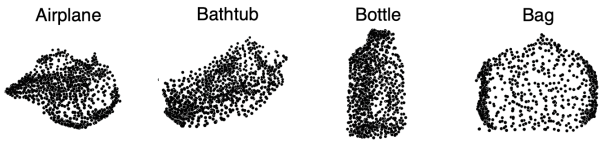

Figure 11.6.: Global explanations for PointNet trained on ShapeNet dataset.

balance the latent and point-wise distances and the number of optimization steps is set to 1000. As a reference, we choose the non-generative model-based as well as the generative model-based approaches proposed by [6] as competitors, which to the best of our knowledge are currently all the point cloud AM approaches that have been attempted by existing researches.

11.5.1. Qualitative comparison of global explanations

We select several common classes as target activations from ModelNet40 and visualize them utilizing the aforementioned AM methods. Fig. 11.5 illustrates the performance of existing regularization approaches and generative model-based AM methods. Among the non-generative model-based AM methods, none of them, except Latent AM, are able to generate perceptible global explanations. With the incorporation of latent alignments in the intermediate layers, our Latent AM successfully reconstructs the global profiles of objects and achieves almost comparable quality to that based on generative models with the help of continuity regularization. Furthermore, though generative model-based approaches yield high-quality explanations as well, our Latent AM does not rely on any information from external models, guaranteeing the fidelity of explanations to the classifier to be explained, while dramatically diminishing the computational intensity. We will discuss in detail the fidelity threat posed by external generative models to global explanations in Sec. 11.5.3.

We also generate global explanations for PointNet trained on ShapeNet, and select the first few categories to display in Fig 11.6. Note that the wing contours of the airplane are not as clear as those of ModelNet40, this is because the airplane class in ShapNet dataset contains a large percentage of delta-winged airplanes, which is the feature learned by the latent activation of the classifier.

11.5.2. Quantitative evaluations

We quantitatively evaluate the quality of the generated global explanations in the following three aspects:

Representativity reflects the ability to generalize about a particular category, i.e., the more distinctive the characteristics of the category that the explanation possesses, the more representative it is. Note that the representativity evaluator is the classifier to be explained, rather than humans. There are features that are semantically unavailable to humans but can highly activate higher-level activations of the classifier, such as the high-frequency mosaics in early feature visualizations [28]. We simply measure the neurons corresponding to specific classes in the Logits and Softmax layers as the metric for representativity assessment, where the former is considered to be absolute representativity, which is the absolute magnitude of activation of the neuron, while the latter is considered to be relative representativity, which takes the activation magnitudes of all the classes into account and calculates the class probability.

Point-wise perceptibility We employ Chamfer Distance (CD) as point-wise perceptibility metric.

Latent perceptibility We employ Fréchet inception distance (FID) as the metric of latent perceptibility and follow existing point cloud applications [154], [6] in selecting global features of PointNet as the recording layer for latent distances. When evaluating point-wise and latent perceptibilities, we randomly select five real objects from the same class in the dataset and average the corresponding distances between them and the generated explanations, respectively.

Quantitative evaluations are reported in Table 11.1. Methods based on generative models typically suffer from poor representativity, which is attributed to the additional prior embedded in the gradient by the generative model. In contrast, while methods based on non-generative models significantly enhance the magnitude of target activations, they perform poorly in terms of perceptibility (CD and FID), which is consistent with the conclusion shown in qualitative comparisons. Latent AM strikes a better balance, preserving the representativeness of the non-generative approaches at the expense of an acceptable level of perceptibility.

Other datasets and models. Besides ModelNet40 , we also test the performance of Latent AM on ShapeNet and report the results in Table 11.2. Again choosing non-generative methods as competitors, Latent AM also dominates on ShapeNet by a large margin, particularly on CD and FID, which indicates a better perceptibility. Further, we experiment on PointNet++ as well and demonstrate the results in Table 11.4. For PointNet++, we similarly select those layers with a large difference in activations when predicting real objects and raw AM instances and minimize the discrepancies. The test results are consistent with PointNet in that non-generative AMs except Latent AM fail to generate global explanations with realistic object outlines.

	Non-generative model-based					Generative model-based		
	Vanilla	L_2-norm	Gaussian blur	Total Variance	Latent (ours)	AE [6]	AED [6]	NAED [6]
Logit↑	16.6	6.3	6.2	6.0	**16.7**	10.5	7.0	8.1
Sftmax↑	$\mathbf{-4.5 \times 10^{-4}}$	-4.1×10^{-3}	-3.7×10^{-3}	-9.8×10^{-3}	-2.5×10^{-3}	-9.1×10^{-2}	-7.6×10^{-1}	-6.0×10^{-1}
D^1_{CH}↓	0.324	0.139	0.148	0.376	0.081	**0.044**	0.086	0.074
FID↓	0.176	0.256	0.420	0.092	0.077	0.016	0.018	**0.014**

Table 11.1.: Quantitative evaluation of existing point cloud AM methods. The metrics from top to bottom are the magnitude of the neurons for corresponding classes in the Logits and SoftMax layers, the Chamfer distance, and the Fréchet inception distance, respectively.

	Vanilla	L_2	GB	TV	Latent
Logit↑	9.5	14.6	9.4	7.8	**15.2**
Sftmax↑	-0.13	-0.29	-0.25	$\mathbf{-0.038}$	-0.043
D^1_{CH}↓	0.304	0.544	0.507	0.245	**0.067**
FID↓	2.714	11.022	3.450	0.232	**0.033**

Table 11.2.: Quantitative evaluations for Non-generative model-based AM explanations on ShapeNet. AM regularization methods from left to right are: vanilla, L_2-norm, Gaussian blur, total variation and our Latent AM.

11.5.3. Generative models: fidelity threat

Despite the higher quality of explanations generated by AM based on generative models, a potential concern is the fidelity to the classifier to be explained. Generative models are inherently strong reconstructionists, which interferes with the information source of the explanations and reduces their persuasiveness. We verify whether the information sources of the two types of global explanations are faithful to the classifiers by randomizing its parameters. We employ AE, i.e., the Autoencoder-based method proposed in [6], and our Latent AM, respectively, to generate global explanations of PointNet trained on ModelNet40. Subsequently, we set a dropout probability of up to 50% for each parameter of the classifier. All dropout classifiers suffer a certain degree of accuracy loss and prediction confidence due to the sparsity and the absence of parameters. We again explain the dropout classifiers with AE and Latent AM, respectively, and compare whether the explanations are corrupted after dropout. As shown in Fig. 11.7, generative model-based AM remains capable of generating well-outlined global explanations when suffering dropouts and thus there is a danger of infidelity of the classifiers to be explained. In contrast, Latent AM, where there is no external interference and all information is derived from the classifier, suffers from a severe deformation in the explanation after dropout and is no longer able to generalize the category. As a result, generative model-based AM may import extensive information from generative models, diminishing the fidelity of the classifier to be explained. Another observation is that the gradients of the generative model often conflict with the classifier to be explained, resulting in significantly lower activation magnitudes for the generative model-based explanations than for the Vanilla and Latent AMs in Table 11.1.

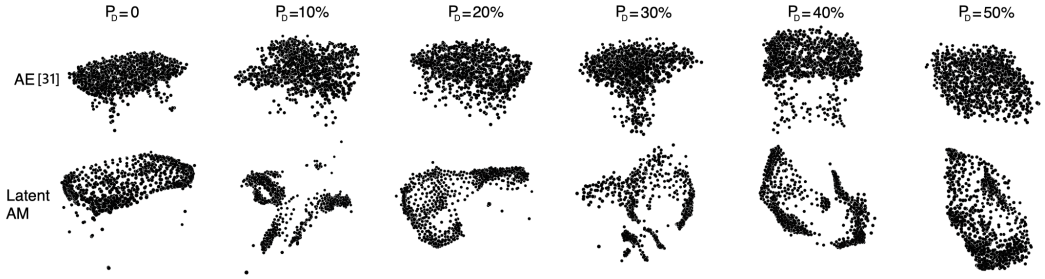

Figure 11.7.: Salinity check results for the selected category "Table". Above and below are the global explanations generated by AE from [6] and our proposed Latent AM, respectively. We show the extent to which the explanation collapses when the dropout probability is set from 0 to 50%, respectively.

	All	$\cancel{L_{LA}^{c}}$	$\cancel{L_C}$	$\cancel{L_{LR}}$
Logits↑	16.7	14.9	-11.2	**17.8**
Sftmax↑	$\mathbf{-2.5 \times 10^{-3}}$	-1.4×10^{-2}	-11.1	$\mathbf{-2.5 \times 10^{-3}}$
D_{CH}^{1}↓	**0.081**	0.209	/	0.087
FID↓	**0.077**	0.114	/	0.110
Legality% ↑	92.5%	**100**%	**100**%	5%

Table 11.3.: Quantitative evaluation of ablation test. From left to right are Latent AM with all components included, with latent approximation removed, with coherence removed, and with legal constraints removed, respectively. When L_C is eliminated, we do not record the distance to real objects of the same class since all the explanations are misclassified by the model.

11.5.4. Ablation Study

To disentangle and validate the effectiveness of individual components, we perform ablation tests on Latent AM. The main refinements of Latent AM are the latent approximation, the point continuity and the legal restriction. In the ablation test, we remove one of the above terms individually, and evaluate the generated explanations. As shown in Fig. 11.8, eliminating latent approximation or point continuity drastically ruins the perceptibility of the explanation from the perspective of humans. After ablating the latent approximation, the explanation is rendered spherical under the restraints of the point continuity and the outline of the object is lost. The removal of point continuity results in the majority of points shrinking and therefore losing semantics. This conclusion is also evidenced by the quantitative evaluation results in Table 11.3. Additionally, though the legal restriction has insignificant impact on the quality of the explanations, its removal results in only 5% of the explanations remain within the legal intervals of the dataset.

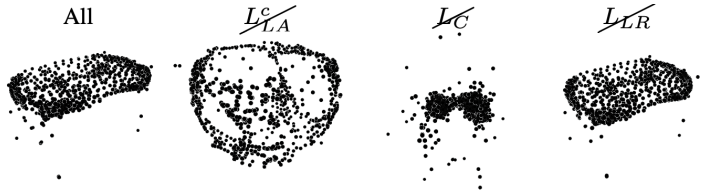

Figure 11.8.: Visualization of ablation test. From left to right are Latent AM with all components included, with latent approximation removed, with coherence removed, and with legal constraints removed, respectively.

	Vanilla	L_2	GB	TV	Latent
Logit↑	6.9	1.9	2.0	5.1	**15.4**
Sftmax↑	-4.0×10^{-4}	-0.24	-0.31	-0.01	$\mathbf{-5.4 \times 10^{-7}}$
D_{CH}^{1}↓	0.116	0.272	0.225	0.066	**0.049**
FID↓	**0.013**	0.052	0.047	0.020	0.020

Table 11.4.: Quantitative evaluation of the global explanations of PointNet++ trained on ModelNet40.

11.6. Conclusion

This chapter presents a novel observation that the intermediate layers of classifiers with PointNet-like effectively learn the contours of 3D objects, and these features can be extracted through latent alignment. We propose Latent AM based on this observation, which generates explanations that are far more perceptible than other non-generative model-based AM methods, while demonstrating higher fidelity to the classifier compared to generative model-based approaches. However, we only found this property on models of PointNet-like structures, and therefore acknowledge that the applicability is limited. Nevertheless, we believe that it still reveals great potential in point cloud explainability and 3D synthesis.

Part IV.

Conclusions

12. Conclusions

12.1. Main conclusions

In recent years, deep neural networks have achieved rapid performance gains due to innovative architectures and modules, but increased structural complexity has impaired model transparency and prediction trustworthiness. This deficiency is fatal in areas where human lives are at stake, such as autonomous driving and AI-assisted healthcare, where black-box model decisions lead to extreme difficulties in debugging and low user trust.

In this regard, a promising research direction is explainability. Explainability is an emerging research direction that strives to reveal the decision-making mechanisms of black-box models in order to enhance their reliability. Saliency maps are a common type of output for post-hoc explainability methods that assign higher attributions to input features that play dominant roles in the prediction. To date, several explainability methods based on saliency maps have been proposed, yet their performance is difficult to evaluate quantitatively. The challenge of the evaluation lies primarily in the absence of ground truth, forcing researchers to verify the validity of saliency maps from multiple different perspectives. Nonetheless, so far, these perspectives still remain to be complemented. More comprehensive evaluation perspectives ensure objectivity in evaluation and enable users to select their desired explainability methods more accurately.

On the other hand, studies on explainability methods are in their infancy, with the vast majority of relevant research involving only the most commonly utilized data types, i.e., tabular and image data, which fail to match the existing deep learning research process, where the explainability of many other categories of data and models is not sufficiently covered. Point clouds are one of the representative data types. Point clouds, the simplest 3D representations, are now widely applied in robotics, autonomous driving, and other scenarios involving three-dimensional spaces. Nonetheless, almost no existing explainability studies on point cloud models can be identified, even though they are not directly transferable from image models due to their structural specificity.

In the above context, this dissertation attempts to contribute to the explainability approaches and their applications in the following two aspects:

- To address the limitations of existing general explainability methods, including inconsistent baseline selection, lack of parameter robustness assessment, insufficiency of important features in explanations, and the potential risk of disruption of input features in ablation tests.

- To facilitate the application of explainability research in point clouds, including making adaptive improvements to general explainability methods to adapt to the specific structure of point cloud models, or proposing novel explainability analysis methods tailored to the characteristics of point clouds.

12.1.1. Evaluable explainability

Baseline refinement

In Chapter 2, a novel baseline is proposed named Maximum Entropy Baseline (MEB). MEB is theoretically the best baseline that fulfills the definition of "uninformative" for a given model. MEB is the input when the entropy of the model prediction is maximized, and is derived by performing only one gradient descent optimization for a specific model. We demonstrate both theoretically and experimentally that MEB enhances performance for some baseline-based explainability methods, while for those evaluation approaches that involve information erasure or feature replacement, MEB leads to more intuitive results compared to other baselines. In addition, by introducing entropy in information theory, we explain the reason why partial shifts of the baseline cause the resulting explanation to be irreversibly altered.

Parameter robustness

In Chapter 3, we extend the sensitivity test to the model parameter level, since as a critical contributor to the decision-making process, the robustness of model parameters has often been overlooked. This assessment is based on the hypothesis that predictions are highly correlated with those parameters in the model that are important and almost irrelevant to those that are not. Based on this assumption, we propose pruning robustness, i.e., when irrelevant parameters in the model are pruned, so that the model performance does not fluctuate significantly (and sometimes even gains slightly), and the explanations it generates should remain almost consistent with the original ones. We evaluate the pruning robustness of popular explainability methods with this criterion and find that the pruning robustness of the overall explanation distribution is superior for gradient-based explainability methods. However, the opposite is true for pixels with the highest attributions. For the perturbation-based explainability approaches, almost no pruning robustness is observed. This research highlights concerns about the fidelity of explainability methods to model parameters, i.e., the correlation of generated explanations with parameters that are irrelevant needs to be reconsidered as one of the prerequisites for fidelity.

Attribution sufficiency

In Chapter 4, a novel concept of explanation evaluation is proposed, termed Sensitivity Consistency. General sensitivity tests carry the risk of missing important features, as they only focus on the one-way sensitivity of predictions to explanations. We argue that

for a given model and input, the explanation should be highly and bidirectionally consistent with the prediction in multiple sensitivities, including input features and model parameters. We assess whether the explanations are faithful to the predictions by generating random masks to perturb the inputs and model parameters, and by comparing the consistency of the changes in the predictions with the explanations. The results indicate that explainability approaches that calculate attributions through baselines exhibit higher sensitivity consistency compared to other gradient-based or perturbation-based methods.

Feature distribution integrity

General sensitivity tests carry the risk of disrupting the feature distribution, thus their credibility is questionable. Chapter 5 proposes another novel idea of evaluation from the viewpoint of the regularity of the explanations. We propose an explanation evaluation framework based on an autoencoder, where the autoencoder can be regarded as an emulation of the explainability approach. In the training phase, we train the autoencoder using the original images in the training set as inputs and their corresponding explanations as labels. Subsequently, the original images in the test set are taken as test data to let the autoencoder reconstruct their explanations and analyze their similarity with the corresponding ground-truth explanations. This assessment avoids the problem of feature distribution corruption from perturbations that the vast majority of assessments confront.

12.1.2. Point cloud applicability

Local surrogate explainer for point clouds

In Chapter 6, we extend an explainability approach for tables and images, based on perturbation and surrogate models, to point clouds. Furthermore, an evaluation metric corresponding to the point cloud saliency map is expanded from images. On the basis of images, the above method combines the properties of point clouds by employing farthest point sampling to segment disordered points into super-points as features to reduce computational workload and a novel technique to perturb the point cloud instances without destroying the data distribution. We demonstrate through qualitative and quantitative experiments that the performance of the proposed improved explainability method outperforms hard transplantation of the method from images.

Visualizing global explanation of point cloud models

Chapters 7 and 8 propose two global explainability methods for point clouds based on Activation Maximization (AM). In the former, we propose three frameworks that utilize autoencoders to regularize AM gradients, including the pure autoencoder, the autoencoder with discriminator and the noisy autoencoder with discriminator. Moreover, we propose a metric for evaluating the performance of point cloud AM explanations named

Point Cloud Activation Maximization Score (PCAMS), which evaluates three different perspectives of the explanations, namely representativeness, perceptibility, and diversity. Through experiments we demonstrate the following two arguments: a) existing, non-generative model-based regularizations applicable to images fail to enable point cloud models to generate explanations with contours approximating real objects, and b) the three point cloud AM methods we propose are capable of generating such explanations, and that the explanations generated by the noisy autoencoder with discriminator achieve the best performance. We extend this idea further in Chapter 8 by leveraging a diffusion model instead of an autoencoder as the regularization of AM, which is currently the most powerful generative model for 3D reconstruction. Additionally, we find that the trajectories of critical points in the point cloud can be observed through gradient integration during the diffusion process, providing an intuitive visualization of their formation. We demonstrate in qualitative and quantitative experiments that the proposed method is able to generate higher-quality global explanations of point clouds within a shorter processing time compared to its predecessor.

Interpretable point cloud classifier

In Chapter 9, we propose a novel non-DNN-based classifier to enhance the interpretability of point cloud models, named Fractal Projection Forest (FPF). FPF projects the point cloud instances onto different dimensions, and then generates sliding fractal windows of different scales to statistically measure the distribution of points for each dimension. These statistical features can be used as inputs to a random forest for classification purposes. We demonstrate through quantitative experiments on different datasets that FPF sacrifices only a slight degree of accuracy while dramatically enhancing explainability and transparency compared to DNNs, along with a substantial increase in its training and inference speed.

Sensitivity research through adversarial attacks

Chapter 10 proposes an adversarial attack method based on saliency maps, which aims to analyze the importance of critical points for point cloud DNNs, as well as their stability. We exploit an explainability approach to generate saliency maps, perturb critical points in order of attribution from largest to smallest, and observe the perturbation magnitude required to alter the prediction. Experiments demonstrate that point cloud models are susceptible to being fooled by extremely sparse perturbations of critical points, and that only one point even needs to be perturbed to change a large proportion of the predictions. We find that replacing the max-pooling layer with other types that are insensitive to extreme values effectively mitigates this deficiency, reinforcing stability by averaging the attribution over multiple critical points while compromising an almost negligible accuracy.

Latent feature analysis of point cloud models

In Chapter 11, we observe that point cloud models can register the contour features of 3D objects through the activation distribution of intermediate layers. We analyze the difference in the activation distribution of the intermediate layers of the point cloud model when learning real instances, generated instances with realistic contours, and generated instances without realistic contours, and based on this analysis we propose a novel global explanation generating method that optimizes the input vectors by aligning the activation distributions in the intermediate layer with those of the real objects, forcing them to go one step towards the contours of the real objects. This observation not only helps to analyze the mechanism of point cloud models, but also provides a potential for generative methods that do not require incorporating additional modules.

12.2. Future Work & Open Issues

The open issues and future work of this dissertation are also twofold, with the first being the evaluation of explanations and the second being the application of explainability methods to point cloud models.

12.2.1. Explanation assessment

- **Feature distribution in perturbation.** As mentioned in Chapters 2 and 4, the majority of existing feature perturbation methods are based on the assumption that features are independent of each other. However, this assumption does not hold for data with complex structures, for instance when considering the correlation between age and income. Straightforward modification of features may lead to issues of distributional corruption, thus threatening the plausibility of explanations and their evaluations. Although several existing methods have proposed alternatives (e.g., retraining the model after perturbation), this incurs substantial computational costs and raises concerns about fidelity to the model being explained. An alternative possibility is to exploit generative models as regularization to force the distribution of features of the perturbed samples to be consistent with the dataset, or to exclude those instances with out-of-distribution features by analyzing the anomalies exhibited by the model in prediction.
- **Fidelity to Parameters.** Chapter 3 demonstrates that explainability methods suffer from infidelity to model parameters, especially those that are irrelevant to the predictions, where pruning them out has no remarkable impact on the predictions but causes a significant collapse in the explanations. We argue that fidelity to parameters should be included as one of the prerequisites for a plausible explainability approach in future studies. To accomplish this, the gradient can be computed by assigning larger weights to those neurons in the model that are important, or by ignoring model structures that are irrelevant to prediction. Another possible way is to align those explanations generated by the pruned model with those generated

by the original one, in order to ensure that the important neurons remain as the main contributors to the explanations. Ensuring that the explanations are faithful to the model parameters is critical, which prevents the explanations from suffering from the Rashomon phenomenon in models with comparable performances.

- **Enhancing sensitivity consistency**. Chapter 4 presents the observation that there may be inconsistencies between explanations and predictions in terms of sensitivity to input features or model parameters. One promising direction is to take it into account when designing explainability methods. A concrete approach might be to emphasize those more important ones by assigning weights to each feature and parameter when calculating the saliency map.
- **Unified and multi-perspective evaluation.** Although several evaluation methods have been proposed to date, they tend to be single-perspective and incomplete, thereby being difficult to unify across different studies. An urgent demand is to integrate the axioms of explanation assessment and to propose corresponding assessment metrics in order to enable a more comprehensive and uniform assessment of different explainability studies and to avoid the bias caused by the inconsistent selection of metrics.

12.2.2. Point cloud explainability

- **Regularized perturbation samples.** Chapter 6 proposes a perturbation-based point cloud explainability method. Likewise, even though there are no shifted points at perturbation, it is still not proven that the perturbed instances are still consistent in distribution with the original ones. Regularizing perturbed examples with generative models as prior is a promising work which ensures that the perturbed instances remain within the learned distribution for the classifier. Based on the observations presented in Chapter 11, another possible approach is to regularize the latent vectors of the perturbed samples at the intermediate layer to maintain the geometric profile of the perturbed samples close to the real objects.
- **Critical point stability.** Chapter 10 reports on the over-sensitivity of point cloud classifiers to critical point perturbations. Though we propose to replace the max-pooling layer with other types (insensitive to extreme values) of structures, the side effect is an observable decrease in accuracy. Thus, finding a balance between stability and classification accuracy is a key direction for future research. The core of improving stability against critical point perturbations is to spread the overly concentrated attribution over a larger number of points, for which the exploitation of local area features instead of point-wise features is another promising alternative.
- **A more comprehensive approach for point cloud explainability.** So far, research on the explainability of point clouds compared to images is still in its infancy. The vast majority of existing studies on the explainability of point clouds are based on the porting or adaptive modification from image models. However, the unique

structure of point cloud models endows them with many potential, independent research directions from image, such as the analysis of latent features that are representative of three-dimensional object contours in Chapter 11 and interpretable classifiers based on the design of fractal windows in Chapter 9. We believe that more unique perspectives remain to be proposed, which are capable of devising unique explainable mechanisms to reveal how the model learns geometric features from complex three-dimensional distributions, starting from the particular structure of point clouds.

Bibliography

[12] F. Hausdorff, "Dimension und äußeres maß," *Mathematische Annalen*, pp. 157–179, 1918.

[13] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, pp. 379–423, 1948.

[14] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information.," *Psychological review*, p. 81, 1956.

[15] R. Dorfman, "A formula for the gini coefficient," *The review of economics and statistics*, pp. 146–149, 1979.

[16] M. Ali, "Effect of sample size on the size of the coefficient of determination in simple linear regression," *Journal of Information and Optimization Sciences*, pp. 209–219, 1987.

[17] S. Hanson and L. Pratt, "Comparing biases for minimal network construction with back-propagation," *Advances in neural information processing systems*, 1988.

[18] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, pp. 303–314, 1989.

[19] Y. LeCun, J. Denker, and S. Solla, "Optimal brain damage," *Advances in neural information processing systems*, 1989.

[20] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, pp. 251–257, 1991.

[21] G. A. Alvarez and P. Cavanagh, "The capacity of visual short-term memory is set both by visual information load and by number of objects," *Psychological science*, pp. 106–111, 2004.

[22] C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC bioinformatics*, pp. 1–21, 2007.

[23] D. Munoz, N. Vandapel, and M. Hebert, "Directional associative markov network for 3-d point cloud classification," 2008.

[24] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC bioinformatics*, pp. 1–11, 2008.

[25] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *University of Montreal*, p. 1, 2009.

[26] W.-Y. Loh, "Classification and regression trees," *Wiley interdisciplinary reviews: data mining and knowledge discovery*, pp. 14–23, 2011.

[27] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: A multi-class classification competition," in *The 2011 international joint conference on neural networks*, IEEE, 2011, pp. 1453–1460.

[28] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[29] C. Szegedy et al., "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[30] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] K. Simonyan, A. Vedaldi, and A. Zisserman, *Deep inside convolutional networks: Visualising image classification models and saliency maps*, arXiv preprint, arXiv:1312.6034, 2014.

[33] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[34] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, Springer, 2014, pp. 818–833.

[35] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, e0130140, 2015.

[36] A. X. Chang et al., "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[37] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2015, pp. 922–928.

[38] A. Mordvintsev, C. Olah, and M. Tyka, "Inceptionism: Going deeper into neural networks," *Google research blog*, p. 5, 2015.

[39] A. Nguyen, J. Yosinski, and J. 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*, 2015, pp. 427–436.

[40] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*, PMLR, 2015, pp. 2256–2265.

[41] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, *Striving for simplicity: The all convolutional net*, arXiv preprint, arXiv:1412.6806, 2015.

[42] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.

[43] D. Wei, B. Zhou, A. Torrabla, and W. Freeman, "Understanding intra-class knowledge inside cnn," *arXiv preprint arXiv:1507.02379*, 2015.

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

[45] Z. Wu et al., "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[46] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *arXiv preprint arXiv:1506.06579*, 2015.

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

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[49] A. Kurakin, I. Goodfellow, S. Bengio, et al., *Adversarial examples in the physical world*, 2016.

[50] A. Mahendran and A. Vedaldi, "Visualizing deep convolutional neural networks using natural pre-images," *International Journal of Computer Vision*, pp. 233–255, 2016.

[51] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[52] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," *Advances in neural information processing systems*, 2016.

[53] A. Nguyen, J. Yosinski, and J. Clune, "Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks," *arXiv preprint arXiv:1602.03616*, 2016.

[54] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[55] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, pp. 2234–2242, 2016.

[56] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the visualization of what a deep neural network has learned," *IEEE transactions on neural networks and learning systems*, pp. 2660–2673, 2016.

[57] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences," *arXiv preprint arXiv:1605.01713*, 2016.

[58] M. Sundararajan, A. Taly, and Q. Yan, *Gradients of counterfactuals*, arXiv preprint, arXiv:1611.02639, 2016.

[59] Z. Zhang et al., "A multilevel point-cluster-based discriminative feature for als point cloud classification," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 3309–3321, 2016.

[60] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Towards better understanding of gradient-based attribution methods for deep neural networks," *arXiv preprint arXiv:1711.06104*, 2017.

[61] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, PMLR, 2017, pp. 214–223.

[62] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*, IEEE, 2017, pp. 39–57.

[63] S. Gurumurthy, R. 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*, 2017, pp. 166–174.

[64] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, 2017.

[65] S. Lundberg, "A unified approach to interpreting model predictions," *arXiv preprint arXiv:1705.07874*, 2017.

[66] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. 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*, 2017, pp. 4467–4477.

[67] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, e7, 2017.

[68] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506–519.

[69] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[70] C. R. Qi, H. Su, K. Mo, and L. 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 (CVPR)*, 2017.

[71] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," 2017, arXiv preprint, arXiv:1706.02413.

[72] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D Object Detection from {RGB-D} Data," *CoRR*, 2017, arXiv preprint, arXiv:1711.08488.

[73] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, 2017.

[74] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K. Müller, "Evaluating the visualization of what a deep neural network has learned," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 2660–2673, 2017.

[75] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[76] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International conference on machine learning*, PMlR, 2017, pp. 3145–3153.

[77] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: Removing noise by adding noise," *arXiv preprint arXiv:1706.03825*, 2017.

[78] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*, PMLR, 2017, pp. 3319–3328.

[79] K. Yu, S. Berkovsky, R. Taib, D. Conway, J. Zhou, and F. Chen, "User trust dynamics: An investigation driven by differences in system performance," in *Proceedings of the 22nd international conference on intelligent user interfaces*, 2017, pp. 307–317.

[80] J. Zhou, S. Z. Arshad, S. Luo, and F. Chen, "Effects of uncertainty and cognitive load on user trust in predictive decision making," in *Human-Computer Interaction–INTERACT 2017: 16th IFIP TC 13 International Conference, Mumbai, India, September 25-29, 2017, Proceedings, Part IV 16*, Springer, 2017, pp. 23–39.

[81] Z. Zhou et al., "Activation maximization generative adversarial nets," *arXiv preprint arXiv:1703.02000*, 2017.

[82] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3d point clouds," in *International conference on machine learning*, PMLR, 2018, pp. 40–49.

[83] J. Adebayo, J. Gilmer, I. Goodfellow, and B. Kim, "Local explanation methods for deep neural networks lack sensitivity to parameter values," *arXiv preprint arXiv:1810.03307*, 2018.

[84] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *Advances in neural information processing systems*, 2018.

[85] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *Ieee Access*, pp. 14 410–14 430, 2018.

[86] D. Alvarez-Melis and T. S. Jaakkola, "On the robustness of interpretability methods," *arXiv preprint arXiv:1806.08049*, 2018.

[87] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan, *L-shapley and c-shapley: Efficient model interpretation for structured data*, arXiv preprint, arXiv:1808.02610, 2018.

[88] Y. Dong et al., "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9185–9193.

[89] M. Du, N. Liu, Q. Song, and X. Hu, "Towards explanation of dnn-based prediction with guided feature inversion," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1358–1367.

[90] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "Gvcnn: Group-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 264–272.

[91] A. Kanezaki, Y. Matsushita, and Y. Nishida, "Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5010–5019.

[92] K. Krippendorff, *Content analysis: An introduction to its methodology*. Sage publications, 2018.

[93] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *Advances in neural information processing systems*, 2018.

[94] T. Parr, K. Turgutlu, C. Csiszar, and J. Howard. "Beware default random forest importances. "[Online]. Available: `https://explained.ai/rf-importance/index.html`.

[95] V. Petsiuk, "Rise: Randomized input sampling for explanation of black-box models," *arXiv preprint arXiv:1806.07421*, 2018.

[96] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proceedings of the AAAI conference on artificial intelligence*, 2018.

[97] H. You, Y. Feng, R. Ji, and Y. Gao, "Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1310–1318.

[98] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by excitation backprop," *International Journal of Computer Vision*, pp. 1084–1102, 2018.

[99] R. M. Byrne, "Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning.," in *IJCAI*, 2019, pp. 6276–6282.

[100] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Communications of the ACM*, pp. 68–77, 2019.

[101] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," in *Proceedings of the AAAI conference on artificial intelligence*, 2019, pp. 3681–3688.

[102] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee, "Counterfactual visual explanations," in *International Conference on Machine Learning*, PMLR, 2019, pp. 2376–2384.

[103] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, "A benchmark for interpretability methods in deep neural networks," *Advances in neural information processing systems*, 2019.

[104] A. Howard et al., "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[105] P.-J. Kindermans et al., "The (un) reliability of saliency methods," *Explainable AI: Interpreting, explaining and visualizing deep learning*, pp. 267–280, 2019.

[106] Y. Li, G. Tong, X. Li, L. Zhang, and H. Peng, "Mvf-cnn: Fusion of multilevel features for large-scale point cloud classification," *IEEE Access*, pp. 46 522–46 537, 2019.

[107] Z. Q. Lin, M. J. Shafiee, S. Bochkarev, M. S. Jules, X. Y. Wang, and A. Wong, "Do explanations reflect decisions? a machine-centric strategy to quantify the performance of explainability algorithms," *arXiv preprint arXiv:1910.07387*, 2019.

[108] D. Liu, R. Yu, and H. Su, "Extending adversarial attacks and defenses to deep 3d point cloud classifiers," in *2019 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2019, pp. 2279–2283.

[109] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8895–8904.

[110] S. Mishra, D. Stoller, E. Benetos, B. L. Sturm, and S. Dixon, "Gan-based generation and automatic selection of explanations for neural networks," *arXiv preprint arXiv:1904.09533*, 2019.

[111] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, "Importance estimation for neural network pruning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 264–11 272.

[112] G. Montavon, "Gradient-based vs. propagation-based explanations: An axiomatic comparison," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. 2019, pp. 253–265, ISBN: 978-3-030-28954-6.

[113] G. Montavon, "Gradient-based vs. propagation-based explanations: An axiomatic comparison," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, 2019, pp. 253–265.

[114] Z. Qi, S. Khorram, and F. Li, "Visualizing deep networks by optimizing with integrated gradients.," in *CVPR workshops*, 2019, pp. 1–4.

[115] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature machine intelligence*, pp. 206–215, 2019.

[116] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, *Explainable AI: interpreting, explaining and visualizing deep learning*. Springer Nature, 2019.

[117] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, pp. 828–841, 2019.

[118] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1588–1597.

[119] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (tog)*, pp. 1–12, 2019.

[120] M. Wicker and M. Kwiatkowska, "Robustness of 3d deep learning in an adversarial setting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 767–11 775.

[121] C. Xiang, C. R. Qi, and B. Li, "Generating 3d adversarial point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9136–9144.

[122] W. Xiao and G. Kreiman, "Gradient-free activation maximization for identifying effective stimuli," *arXiv preprint arXiv:1905.00378*, 2019.

[123] K. Young, G. Booth, B. Simpson, R. Dutton, and S. Shrapnel, "Deep neural network or dermatologist?" In *Interpretability of Machine Intelligence in Medical Image Computing and Multimodal Learning for Clinical Decision Support*, 2019, pp. 48–55, ISBN: 978-3-030-33850-3.

[124] Q. Zhang, J. Yang, R. Fang, B. Ni, J. Liu, and Q. Tian, "Adversarial attack and defense on point sets," *arXiv preprint arXiv:1902.10899*, 2019.

[125] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, "Pointcloud saliency maps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[126] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, "Pointcloud saliency maps," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1598–1606.

[127] H. Zhou, K. Chen, W. Zhang, H. Fang, W. Zhou, and N. Yu, "Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1961–1970.

[128] J. Zhou et al., "Effects of influence on user trust in predictive decision making," in *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–6.

[129] J. Adebayo et al., *Sanity checks for saliency maps*, arXiv preprint, arXiv:1810.03292, 2020.

[130] M. M. Ahsan et al., "Study of Different Deep Learning Approach with Explainable AI for Screening Patients with COVID-19 Symptoms: Using CT Scan and Chest X-ray Image Dataset," 2020, arXiv preprint, arXiv:2007.12525.

[131] L. Arras, A. Osman, and W. Samek, "Ground truth evaluation of neural network explanations with clevr-xai," *arXiv preprint arXiv:2003.07258*, 2020.

[132] K. Browne and B. Swift, "Semantics and explanation: Why counterfactual explanations produce adversarial examples in deep neural networks," *arXiv preprint arXiv:2012.10076*, 2020.

[133] N. Burkart and M. F. Huber, "A Survey on the Explainability of Supervised Machine Learning," pp. 1–74, 2020, arXiv preprint, arXiv:2011.07876.

[134] S. Dandl, C. Molnar, M. Binder, and B. Bischl, "Multi-objective counterfactual explanations," in *International conference on parallel problem solving from nature*, Springer, 2020, pp. 448–469.

[135] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[136] V. F. Figueiredo, G. L. Sandri, R. L. de Queiroz, and P. A. Chou, "Saliency maps for point clouds," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, 2020, pp. 1–5.

[137] V. Gemert, "Evaluating the performance of the LIME and Grad-CAM explanation methods on a LEGO multi-label image classification task," 2020, arXiv preprint, arXiv:2008.01584v1.

[138] A. Gupta, S. Watson, and H. Yin, "3d point cloud feature explanations using gradient-based methods," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.

[139] A. Gupta, S. Watson, and H. Yin, "3d point cloud feature explanations using gradient-based methods," in *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2020, pp. 1–8.

[140] A. Hamdi, S. Rojas, A. Thabet, and B. Ghanem, "Advpc: Transferable adversarial perturbations on 3d point clouds," in *European Conference on Computer Vision*, Springer, 2020, pp. 241–257.

[141] L. Hancox-Li, "Robustness in machine learning explanations: Does it matter?" In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 640–647.

[142] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, pp. 6840–6851, 2020.

[143] C.-Y. Hsieh et al., "Evaluations and methods for explanation through robustness analysis," *arXiv preprint arXiv:2006.00442*, 2020.

[144] Z. Huang and Y. Li, "Interpretable and accurate fine-grained recognition via region grouping," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8662–8672.

[145] D. Janzing, L. Minorics, and P. Blöbaum, "Feature relevance quantification in explainable ai: A causal problem," in *International Conference on artificial intelligence and statistics*, PMLR, 2020, pp. 2907–2916.

[146] M. T. Keane and B. Smyth, "Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable ai (xai)," in *International Conference on Case-Based Reasoning*, Springer, 2020, pp. 163–178.

[147] N. Kokhlikyan et al., "Captum: A unified and generic model interpretability library for pytorch," *arXiv preprint arXiv:2009.07896*, 2020.

[148] K. Lee, Z. Chen, X. Yan, R. Urtasun, and E. Yumer, "Shapeadv: Generating shape-aware adversarial 3d point clouds," *arXiv preprint arXiv:2005.11626*, 2020.

[149] D. Liu, R. Yu, and H. Su, "Adversarial shape perturbations on 3d point clouds," in *European Conference on Computer Vision*, Springer, 2020, pp. 88–104.

[150] J. Nilsson and T. Akenine-Möller, "Understanding SSIM," *arXiv preprint arXiv:2006.13846*, 2020.

[151] S. Shi et al., "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 529–10 538.

[152] P. Sturmfels, S. Lundberg, and S.-I. Lee, "Visualizing the impact of feature attribution baselines," *Distill*, e22, 2020.

[153] J. Sun, K. Koenig, Y. Cao, Q. A. Chen, and Z. M. Mao, "On the adversarial robustness of 3d point cloud classification," *arXiv preprint arXiv:2011.11922*, 2020.

[154] Y. Sun, Y. Wang, Z. Liu, J. Siegel, and S. Sarma, "Pointgrow: Autoregressively learned point cloud generation with self-attention," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 61–70.

[155] M. Sundararajan and A. Najmi, "The many shapley values for model explanation," in *International conference on machine learning*, PMLR, 2020, pp. 9269–9278.

[156] R. Tomsett, D. Harborne, S. Chakraborty, P. Gurram, and A. Preece, "Sanity checks for saliency metrics," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, pp. 6021–6029.

[157] T. Vermeire and D. Martens, *Explainable image classification with evidence counterfactual*, arXiv preprint, arXiv:2004.07511, 2020.

[158] Y. Wen, J. Lin, K. Chen, C. L. P. Chen, and K. Jia, *Geometry-aware generation of adversarial point clouds*, 2020.

[159] M. Zhang, H. You, P. Kadam, S. Liu, and C.-C. J. Kuo, "Pointhop: An explainable machine learning method for point cloud classification," *IEEE Transactions on Multimedia*, pp. 1744–1755, 2020.

[160] M. Zhang, H. You, P. Kadam, S. Liu, and C.-C. J. Kuo, "Pointhop: An explainable machine learning method for point cloud classification," *IEEE Transactions on Multimedia*, pp. 1744–1755, 2020, ISSN: 1941-0077.

[161] Y. Zhao, Y. Wu, C. Chen, and A. Lim, "On isometry robustness of deep 3d point cloud models under adversarial attacks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1201–1210.

[162] H. Zhou et al., "Lg-gan: Label guided adversarial network for flexible targeted attack of point cloud based deep networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 356–10 365.

[163] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," *Journal of Artificial Intelligence Research*, pp. 245–317, 2021.

[164] R. Confalonieri, L. Coba, B. Wagner, and T. R. Besold, "A historical perspective of explainable artificial intelligence," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, e1391, 2021.

[165] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, pp. 8780–8794, 2021.

[166] A. Goyal, H. Law, B. Liu, A. Newell, and J. Deng, "Revisiting point cloud shape classification with a simple and effective baseline," in *International Conference on Machine Learning*, PMLR, 2021, pp. 3809–3820.

[167] N. F. Heide, E. Müller, J. Petereit, and M. Heizmann, "X 3 seg: Model-agnostic explanations for the semantic segmentation of 3d point clouds with prototypes and criticism," in *2021 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2021, pp. 3687–3691.

[168] A. Katzmann, O. Taubmann, S. Ahmad, A. Mühlberg, M. Sühling, and H.-M. Groß, "Explaining clinical decision support systems in medical imaging using cycle-consistent activation maximization," *Neurocomputing*, pp. 141–156, 2021.

[169] J. Kim, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Minimal adversarial examples for deep learning on 3d point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7797–7806.

[170] X. Li et al., *Pointba: Towards backdoor attacks in 3d point cloud*, 2021.

[171] S. Luo and W. Hu, "Diffusion probabilistic models for 3d point cloud generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 2837–2845.

[172] S. Luo and W. Hu, "Diffusion probabilistic models for 3d point cloud generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2837–2845.

[173] Z. Lyu, Z. Kong, X. Xu, L. Pan, and D. Lin, "A conditional point diffusion-refinement paradigm for 3d point cloud completion," *arXiv preprint arXiv:2112.03530*, 2021.

[174] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*, PMLR, 2021, pp. 8162–8171.

[175] W. Shen, Q. Ren, D. Liu, and Q. Zhang, "Interpreting representation quality of dnns for 3d point cloud processing," *Advances in Neural Information Processing Systems*, pp. 8857–8870, 2021.

[176] G. Vilone and L. Longo, "Notions of explainability and evaluation approaches for explainable artificial intelligence," *Information Fusion*, pp. 89–106, 2021.

[177] J. Yu et al., "3d medical point transformer: Introducing convolution to attention networks for medical point cloud analysis," *arXiv preprint arXiv:2112.04863*, 2021.

[178] J. Zhang et al., "3d adversarial attacks beyond point cloud," *arXiv preprint arXiv:2104.12146*, 2021.

[179] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger, "Evaluating the quality of machine learning explanations: A survey on methods and metrics," *Electronics*, p. 593, 2021.

[180] A. Ajalloeian, S.-M. Moosavi-Dezfooli, M. Vlachos, and P. Frossard, "On Smoothed Explanations: Quality and Robustness," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 15–25.

[181] N. I. Arnold, P. Angelov, and P. M. Atkinson, "An improved explainable point cloud classifier (xpcc)," *IEEE Transactions on Artificial Intelligence*, pp. 71–80, 2022.

[182] A.-M. Leventi-Peetz and K. Weber, "Rashomon effect and consistency in explainable artificial intelligence (xai)," in *Proceedings of the Future Technologies Conference*, Springer, 2022, pp. 796–808.

[183] T. Li, Y. Fu, X. Han, H. Liang, J. J. Zhang, and J. Chang, "Diffusionpointlabel: Annotated point cloud generation with diffusion model," in *Computer Graphics Forum*, Wiley Online Library, 2022, pp. 131–139.

[184] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual mlp framework," *arXiv preprint arXiv:2202.07123*, 2022.

[185] C. Molnar, *Interpretable Machine Learning, A Guide for Making Black Box Models Explainable*, 2nd ed. 2022.

[186] R. N. Mulawade, C. Garth, and A. Wiebel, "Saliency clouds: Visual analysis of point cloud-oriented deep neural networks in deeprl for particle physics," 2022.

[187] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, "Point-e: A system for generating 3d point clouds from complex prompts," *arXiv preprint arXiv:2212.08751*, 2022.

[188] F. Verburg, "Exploring explainability and robustness of point cloud segmentation deep learning model by visualization," B.S. thesis, University of Twente, 2022.

[189] N. Hama, M. Mase, and A. B. Owen, "Deletion and insertion tests in regression models," *Journal of Machine Learning Research*, pp. 1–38, 2023.

[190] A. Hedström et al., "Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond," *Journal of Machine Learning Research*, pp. 1–11, 2023.

[191] R. Hesse, S. Schaub-Meyer, and S. Roth, "Funnybirds: A synthetic vision dataset for a part-based analysis of explainable ai methods," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3981–3991.

[192] S. Müller, V. Toborek, K. Beckh, M. Jakobs, C. Bauckhage, and P. Welke, "An empirical evaluation of the rashomon effect in explainable machine learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2023, pp. 462–478.

[193] Y. Rong et al., "Towards human-centered explainable ai: A survey of user studies for model explanations," *IEEE transactions on pattern analysis and machine intelligence*, 2023.

[194] Y. Zhang, S. Gu, J. Song, B. Pan, G. Bai, and L. Zhao, "Xai benchmark for visual explanation," *arXiv preprint arXiv:2310.08537*, 2023.

[195] M. E. Atik, Z. Duran, and D. Z. Seker, "Explainable artificial intelligence for machine learning-based photogrammetric point cloud classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024.

[196] M. Miró-Nicolau, A. Jaume-i-Capó, and G. Moyà-Alcover, "Assessing fidelity in xai post-hoc techniques: A comparative study with ground truth explanations datasets," *Artificial Intelligence*, p. 104 179, 2024.

[197] R. N. Mulawade, C. Garth, and A. Wiebel, "Explainable artificial intelligence (xai) for methods working on point cloud data: A survey," *IEEE Access*, pp. 146 830–146 851, 2024. DOI: 10.1109/ACCESS.2024.3472872.

[198] I. Romanelis, V. Fotis, K. Moustakas, and A. Munteanu, "Exppoint-mae: Better interpretability and performance for self-supervised point cloud transformers," *IEEE Access*, 2024.

[199] J. Wei, H. Turbé, and G. Mengaldo, "Revisiting the robustness of post-hoc interpretability methods," *arXiv preprint arXiv:2407.19683*, 2024.

[200] M. Miró-Nicolau, A. Jaume-i-Capó, and G. Moyà-Alcover, "A comprehensive study on fidelity metrics for xai," *Information Processing & Management*, p. 103 900, 2025.