POLITECNICO DI TORINO
Repository ISTITUZIONALE

BubblEX: An Explainable Deep Learning Framework for Point-Cloud Classification

Terms of use:

Publisher copyright

(Article begins on next page)

12 July 2024

BubblEX: An Explainable Deep Learning Framework for Point-Cloud Classification

Francesca Matrone , Marina Paolanti, *Member, IEEE*, Andrea Felicetti, Massimo Martini ,
and Roberto Pierdicca

Abstract—Point-cloud data are nowadays one of the major data sources for describing our environment. Recently, deep architectures have been proposed as a key step in understanding and retrieving semantic information. Despite the great contribution of deep learning in this field, the explainability of these models for 3-D data is still fairly unexplored. Explainability, identified as a potential weakness of deep neural networks (DNNs), can help researchers against skepticism, considering that these models are far from being self-explanatory. Although literature provides many examples on the exploitation of explainable artificial intelligence approaches with 2-D data, only a few studies have attempted to investigate it for 3-D DNNs. To overcome these limitations, BubblEX is proposed here, a novel multimodal fusion framework to learn the 3-D point features. BubblEX framework comprises two stages: "Visualization Module" for the visualization of features learned from the network in its hidden layers and "Interpretability Module," which aims at describing how the neighbor points are involved in the feature extraction. For our experiments, dynamic graph convolutional neural network has been used, trained on Modelnet40 dataset. The developed framework extends a method for obtaining saliency maps from image data, to deal with 3-D point-cloud data, allowing the analysis, comparison, and contrasting of multiple features. Besides, it permits the generation of visual explanations from any DNN-based network for 3-D point-cloud classification without requiring architectural changes or retraining. Our findings will be extremely useful for both scientists and nonexperts in understanding and improving future AI-based models.

Index Terms—Artificial intelligence (AI), deep learning, explainable artificial intelligence (XAI), explainability, point cloud.

I. INTRODUCTION

TERRESTRIAL and airborne sensors (both image or range based) are able to produce accurate 3-D point-clouds data, which represent one of the major sources for describing our environment [1]. Nowadays, geometrical information is accurate, and, despite still entrusted on the operator, the procedure

Francesca Matrone is with the Politecnico di Torino, Department of Environment, Land and Infrastructure Engineering (DIATI), 10129 Torino, Italy (e-mail: francesca.matrone@polito.it).

Marina Paolanti is with the Department of Political Sciences, Communication and International Relations, Università degli Studi di Macerata, 62100 Macerata, Italy (e-mail: m.paolanti@staff.univpm.it).

Andrea Felicetti and Massimo Martini are with the Department of Information Engineering, Università Politecnica delle Marche, 60121 Ancona, Italy (e-mail: a.felicetti@staff.univpm.it; m.martini@pm.univpm.it).

Roberto Pierdicca is with the Department of Architecture, Building and Civil Engineering (DICEA), Università Politecnica delle Marche, 60121 Ancona, Italy (e-mail: r.pierdicca@staff.univpm.it).

is straightforward and, to some extent, trustworthy. The same cannot be said for the semantic information linked to the 3-D data; in other words, supervised learning tasks are far from being off-the-shelf solutions. Point clouds, as raw data of most mainstream sensors, have a significant advantage in real-time scenarios compared to other 3-D data formats and, therefore, are gaining a lot of attention for research purposes, exhibiting higher structural complexity than 2-D images. They can be acquired from LiDAR scanners or RGB-D sensors with a density that depends on the sensor scanning pattern and the distance of the surface being scanned from the sensor head. With the advancement of artificial intelligence (AI), deep neural networks (DNNs) have been employed for point-cloud classification. For the last years, we have seen tremendous progress in supervised learning tasks like segmentation and classification [2], [3]. Researchers have developed expressive handcrafted features that can be extracted from a local neighborhood of each point, and they have adapted supervised classification techniques from machine learning to the processing of point clouds [4]. The consideration of context in the classification process by graphical models, e.g., conditional random fields, has further improved the accuracy that can be achieved, in particular for small objects. Thus, learning-based techniques can be considered promising and deserves further investigation. Considering the 2-D counterpart (e.g., image analysis [5] or trajectory data [6]), it can be stated that point-cloud processing is one step behind [7]. Indeed, the extraction of semantic information from images has been revolutionized by deep learning techniques, in particular by convolutional neural networks (CNNs), which have been shown to outperform other techniques. Albeit in the meantime, deep learning has been adapted for the interpretation and knowledge extraction of point clouds; the bulk of knowledge in this latter field is still struggling against a bottleneck. The reasons are manifold, but the recent literature provides sights and research directions: 1) images raise a higher attention from a commercial point of view, and the Big Data players are investing huge resources in collection information from the whole mankind; 2) dealing with point cloud is more complex, considering that the image-based technique cannot be exploited for the same task in 3-D; 3) there are less available datasets of point clouds, given that the class labeling process is difficult and time demanding. Besides the above-mentioned reasons, there is one further hidden motivation that is hampering the research to make the step forward from feature- to learning-based approaches: skepticism.

AI-based algorithms, especially DNNs, are transforming the way of approaching real-world tasks done by humans. DNN

architectures are increasingly being adopted in several domains from medical diagnosis [8] and retail [9] to autonomous driving [10] due to their competence to learn relevant abstractions from data. At first, these models were considered as "black box" operators, but as their popularity has grown, they need to be interpretable and explainable [11]–[13]. The perception of DNNs as "black box" algorithms makes it difficult to ethically justify their use in high-stake decisions, especially in case of failure [14]. The adverse of black-boxness is transparency; in fact, the search for a direct understanding of the mechanism by which a model works becomes difficult [15]. Consequently, humans are commonly hesitant to use techniques that are not straight interpretable, tractable, and trustworthy, given the increasing request for ethical AI [16].

The terms "interpretability" and "explainability" have been widely discussed in the AI community, mainly for DNN performance and ethics, thus raising important questions: will the domain experts more heavily weigh AI output with improved interpretability? Can the adoption of explainable methods increase model performance? These and other similar questions have been investigated for defining "explainability" and "interpretability" in AI [14]. "Interpretability" refers to understanding algorithm output for implementation, while "explainability" concerns techniques applied to explain and improve the AI system [13]. To address these issues, explainable artificial intelligence (XAI) proposes to make a shift toward more trustworthy AI. It aims at developing a suite of techniques that produce more explainable models while keeping high performance levels.

A. Nature and Scope

In recent years, a lot of efforts were devoted to develop XAI framework for explaining DNN decisions, especially when these models deal with image data [17]. Several explainability techniques were developed to generate heatmaps of salient regions and most of them are obtained from the backpropagation of the gradients, by checking how individual pixel perturbations affect the decision [18]. Moreover, in the context of XAI, saliency maps denote the pixels that are deemed important for the decision of the model under consideration. Features learned from 2-D data can be visualized and distinguished as images [19].

Although there are many works in the literature on the exploitation of XAI approaches with 2-D data [20], [21], few studies have attempted to investigate the explainability of 3-D DNNs [22]. Currently, DNN models designed for analyzing point-cloud data remain black boxes due to the lack of research investigating their inner explainability. Similar to images that are composed of pixels, point-cloud instances consist of individual points as the fundamental unit. In contrast to images, point position is represented through the coordinates, not completely in the index of the image. Thereby, the challenges faced by this study mainly dealt with the sparseness, discreteness, and disorderliness of the point data. Independently from the operators chosen and exploited by the different state-of-the-art DNN models to solve such issues, the proposed methodology could be applied to any of them; nevertheless, when applying Grad-CAM to 3-D data, some criticalities faced were the choice

of the best function to flatten the features (minimum, maximum, average, and median), the combination of features and gradient (before or after flattening), as well as the choice of a color scale to enhance the results interpretation. To improve the distinctiveness of point-cloud DNN frameworks, we, therefore, propose BubblEX, a new multimodal fusion framework to learn the 3-D point features. BubblEX framework consists of two stages: Visualization Module and Interpretability Module. First, t-distributed stochastic neighbor embedding (t-SNE) [23] and uniform manifold approximation and projection (UMAP) [24] are used as they attempt to preserve the clustering structure by considering a local neighbor [12]. Both t-SNE and UMAP contain hyperparameters that can impact the structures visible to the operator. Then, we move one step further to unfold the black box for the 3-D point-cloud features learning. For this reason, BubblEX comprises an Interpretability Module, which describes how the neighbor points are involved in the features extraction. For the development of this module, we are inspired by gradient-weighted class activation mapping (Grad-CAM) approach [25]. The reasons for these improvements lie in two aspects: The Grad-CAM can be applied to common 3-D DNNs but fails to sparse tensor convolution. The Grad-CAM demands a class label to evaluate the category- specified gradient, but the registration task does not enclose these class labels. For our experiments, we considered dynamic graph CNN (DGCNN) that builds dynamic connections among points in their feature level and updates point features based on their neighboring points in the feature space [26]. DGCNN for classification of 3-D point clouds has been trained both on Modelnet40 and ScanObjectNN datasets. The first is one of the recent available 3-D datasets [27], developed by the Princeton Vision & Robotics Labs and composed by 12 311 prealigned shapes, made up of 1024 points, subdivided into 40 categories. Whereas, the second contains about 15 000 objects, categorized into 15 categories [28].

The *key contributions* of this article are as follows:
1) the extension of a method developed for obtaining saliency maps from image data to deal with 3-D point-cloud data;
2) a framework[1] for understanding the process of 3-D point-cloud features learning for multiclass classification tasks;
3) a visual method that enables analyzing, comparing, and contrasting multiple features;
4) the generation of visual explanations from any DNN-based network for 3-D point-cloud classification without requiring architectural changes or retraining.

The article is organized as follows. Section II provides a description of the approaches that were adopted for the explainability of 3-D point cloud. Section III describes BubblEX framework for 3-D point-cloud XAI. In Section IV, an extensive comparative evaluation of our approach with respect to the state-of-the-art ModelNet40 and ScanObjectNN datasets is offered, as well as a detailed analysis of each component of our framework. Finally, in Sections V and VI, discussions on the obtained results are drawn, along with the conclusions and the definition of the future directions for this field of research.

---

[1]https://github.com/vrai-group/BubblEX

II. STATE OF THE ART

Explaining the decisions performed by DNNs require knowledge of the internal layers of DNNs, missing with non-AI-experts and end-users who are more focused on getting accurate and reliable results. Therefore, the ability to interpret AI decisions are often considered less important of achieving state-of-the-art performances. Recently, XAI has gained increasing importance [29], even from governments, particularly with the European General Data Protection Regulation [30]. In fact, following the guidelines of the European Regulation on AI[2], when designing the application, it is intended to apply the principle of transparency of the technology used, so as to make known to those concerned, and compatible with the European Ethics Guidelines for trustworthy AI, the technology model applied, the type of information potentially expected and the predefined regime of risk control and security protection. Most of the ongoing research on explainability pays attention to image classification tasks [17]. Current approaches to explainability of DNN models include the design of a saliency map, which allows to highlight and identify the valuable areas of the input space [31]. Moreover, for explaining the decision made by a DNN for image classification, popular methods are gradient-based and local surrogate model-based [32].

However, in literature, there are few works that have investigated the explainability of 3-D DNNs. This section aims at reviewing the existing XAI methods for point clouds.

One of the first attempts to make AI explainable was PointHop [33]. In this article, the authors proposed a method which consisted of two steps: The first one is a local-to-global attribute building through iterative one-hop information exchange, while the second one is classification and ensembles. They have explored ensemble methods to improve the classification performance. Their work addressed the disorderly properties of point clouds using PointHop units to adapt them to classical classifiers, which is part of the preprocessing rather than *post hoc* explanations.

The concept of saliency map was also adopted for point-cloud XAI. The construction of point-cloud saliency maps was done in [34]. The method assigned each point a score reflecting its contribution to the model-recognition loss. The saliency map explicitly explained which points are the key for model recognition.

In [35], the authors proposed an alternative evaluation approach by randomizing the network weights as well as the labels. They also claimed that a feasible explanation should be sensitive to the weights of models and the data generating process.

Another approach for explaining the decision of a DNN when it deals with 3-D data is described in [36]. The authors proposed a point-cloud applicable explainability method based on local surrogate model-based approach to demonstrate which components are responsible to the classification. Moreover, they quantified the validity of the explanations for point-cloud data through fidelity and accuracy verification methods instead of a subjective approach based on human impressions.
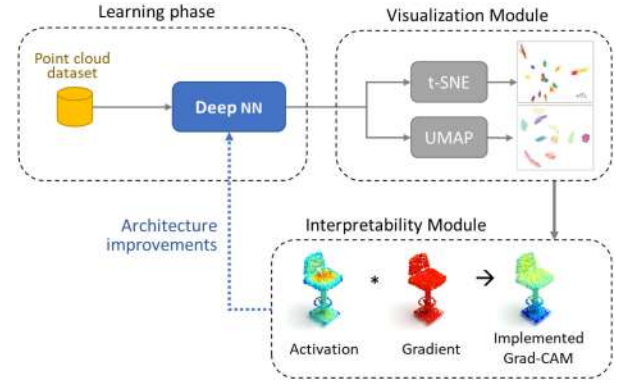
Fig. 1. BubblEX workflow. The features extracted from the trained DNN serve as input for the Visualization and Interpretability Modules. The output of these two core parts could help to improve network decisions.

The pioneer study of utilizing explainability approaches to point clouds remains crucial to understand feature sparsity of 3-D models [37]. Gupta et al. extended the saliency methods by analyzing the features in point clouds and voxel spaces. They have shown that edges and corners in 3-D data are considered as important features while planar surfaces are deemed less important. Driven by the insight that 3-D data is inherently sparse, they visualized the features learned by a voxel-based classification network and show that these features are also sparse and can be pruned relatively easily, leading to more efficient neural networks. However, they only demonstrated sparse explanations that emphasize the importance of points at edges and corners, which is lack of semantics, though the evaluation criterion of the explanations was absent. In addition, the gradient-based methods were not adapted to models without gradients, such as tree-based models.

Considering the state of the art in this context, BubblEX framework comprises a visualization step, performed by t-SNE and UMAP, followed by a recognition step in which the important features for DNN decisions are highlighted. To assess relatedness between clusters identified by t-SNE and UMAP essentially means building hierarchy and boundaries between the clusters and our Interpretability Module inspired by Grad-CAM can effectively recognize the features used for making decisions.

With respect to the above-mentioned state-of-the-art works, our approach does not need to iterate through XAI techniques (like [33]); so it is potentially faster. In addition, we improve the saliency map approach of [34] adding the third dimension and pointing out the intruders. Compared to [35], which performs tests by randomizing the weights of the network, we directly choose which layer to study (through visualization techniques), making our approach more punctual, precise, and effective. Finally, we do not have to train surrogate models like [36], but we can directly use the original classification approach, without the need to retrain it.

III. METHODOLOGY

The overall framework of BubblEX for learning 3-D point features is depicted in Fig. 1. As stated in Section I, BubblEX
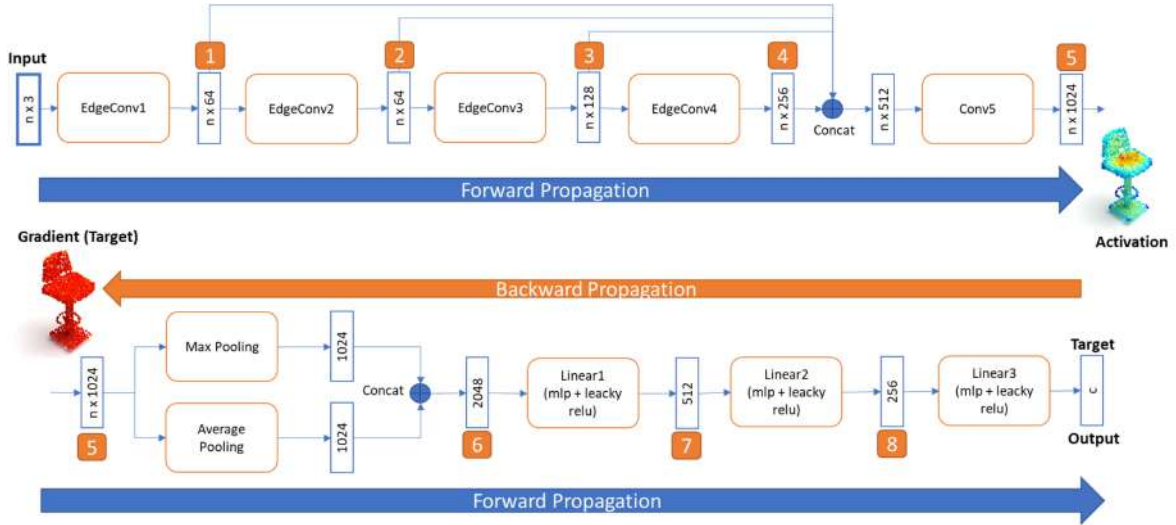
Fig. 2.   DGCNN Architecture. Numbers from 1 to 8 show the output layers for features extraction. Point 5 is the one selected for the Interpretation Module, while Point 8 for the Visualization Module.

framework consists of two stages: Visualization Module and Interpretability Module. First, a network trained on a dataset of point clouds is needed to solve a classification task and then a method for the extraction and visualization of the learned features by network's layers is selected. In this phase, the intuition lies on t-SNE [23] and UMAP [24] that are designed to principally preserve local structure that is to group neighboring data points together, providing informative visualization of heterogeneity for the 2-D data. In this article, we adapt them to the 3-D point clouds. Finally, the third step is fundamental to understand the decisions taken by the network to classify the extracted features in a given class. The union of these last two steps represents the fundamental interpretability core of the model trained in the initial steps; it allows to understand the decision errors taken by the network and consequently provides an idea of how to improve its training phase. We, thus, build an Interpretability Module, which describes how the neighbor points are involved in the feature extraction. For the development of this module, we were inspired by Grad-CAM approach. For our experiments, we considered DGCNN DNN [26]. DGCNN for classification of 3-D point clouds has been trained on Modelnet40 and ScanObjectNN datasets.

The proposed visualization approaches can also be useful to make the architecture of any network more efficient. In fact, in addition to testing the DGCNN, preliminary tests investigated the PointNet [38] architecture, trained with ModelNet10. Through the visualization methods, it was found that some layers could be excluded while maintaining high accuracy. These tests have not been detailed here as the performance enhancement was out of the scope of this research.

In the following subsections, we describe each part of our framework as well as the dataset used for evaluation.

A. Learning Phase

The first step of the approach concerns the training of a neural network for the task of point-clouds classification. As mentioned above, we chose the DGCNN as the basis for testing the proposed XAI approach since it is a well-known neural network and it has been already trained on the selected dataset [39]. Nevertheless, the proposed approach can be extended to any other architecture, being independent from the type of DNN picked out. Furthermore, since the DGCNN was also used by the authors for the semantic segmentation task [40], it provides the advantage that this task can also be tested as future works.

This method implements a dynamic graph neural network based on a particular group of layers, called Edge-Conv. This novel module captures local geometric structure while maintaining permutation invariance. It also allows to generate edge features that describe the relationship between a point and its neighbors instead of generating points' feature directly from embedding. With respect to the standard graph networks, DGCNN improves the learned graph in each of its levels [26].

In the two proposed modules, methods will be used to extract and study the features learned from the network layers. In Fig. 2, it can be noticed that the DGCNN architecture allows extracting features at global level only in the final layers. In fact, the proposed approach will exploit the features learned in the last MLP layer, before the classification layer. If feature extraction is desired in other intermediate layers, a global pooling layer should be added at the extraction point. However, this solution involves two problems that should be avoided: First, it will modify the architecture of the network; second, this type of layer, by performing the global pooling operation, will lead to a small loss of information, intrinsic in the operation carried out.

The analysis of the features can be performed both in the last layers and in the intermediate layers. For the analysis of the features, in the intermediate layers, the global pooling operation is necessary as global pooling layer is yet in the network (there is no modification of the network). The global pooling operation reduces information by flattening of activation vector. In our case, on the edgeconv5 layer, the activation is a vector of size 1024 features $\times$ 1024 points. In order to be displayed in the interpretability module, the feature dimension is flattened with
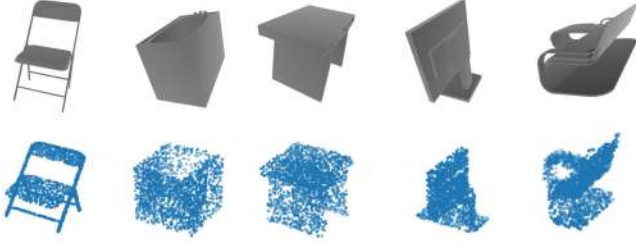
Fig. 3.   Mesh examples (top) and corresponding subsampled objects (bottom) of the ModelNet40 dataset.

a global pooling operation as detailed in Section 3.3. The result is a vector of size 1024 points.

The network has been trained using ModelNet40 dataset split into 80% for training (9843 objects) and 20% (2468 objects) for testing phase. Objects of this dataset are randomly sampled with 1024 points from the mesh faces; then their x, y, and z coordinates are used as input in all experiments of this work. Fig. 3 shows some point-cloud examples of the ModelNet40 dataset, after the subsampling phase.

B. Visualization Module

For the visualization of features learned from the network in its hidden layers, we deal with the large dimensionality of this data. One of the state-of-the-art techniques for dimensionality reduction is t-SNE [23]: particularly suited for the visualization of high-dimensional features and datasets. It is often used in the domain of imagery, but, in recent years, it has also been successfully applied with other types of data, such as point clouds [41], [42]. The t-SNE is a probabilistic technique, contrary to principal component analysis (PCA) that is a statistical one. It tries to minimize the divergence between two distributions: the former measures pairwise similarities of the input data, while the latter calculates pairwise similarities of the related low-dimensional data in the embedding. The main problem with this technique is that it can be computationally very onerous, when there are very high-dimensional data. A solution is to apply another dimensionality reduction technique before using t-SNE, i.e., PCA technique, which aims at reducing the number of dimensions of data preserving most information [43].

In recent years, a new dimensionality reduction technique, namely UMAP [24], has been introduced. UMAP is a learning technique for dimension reduction based on Riemannian geometry and algebraic topology. This approach is more competitive than the t-SNE in terms of dimensionality reduction and visualization quality since it allows most of the relationships between the input data to be preserved, while maintaining very fast processing times.

BubblEX adopts the aforementioned techniques for the visualization and study of the features learned in the intermediate layers of the DNN for point-clouds classification. This choice has been made considering that t-SNE and UMAP are proficient for studying features learned in a classification task. Each object in the dataset, in fact, is associated with a feature vector extracted from an intermediate layer. This vector can be given as input to one of these techniques, which will map it as a point in a 2-D space. The entire test dataset will be first given as input to the neural network, then transformed in feature vectors, and, finally, mapped inside a bidimensional world to be studied.

The main idea of this phase of the framework is to implement a generalizable method to understand how a generic 3-D object is transformed by any deep learning approach that aims to solve a particular task (Fig. 4). It begins with a 3-D object, described only by a point cloud composed of the points coordinates. The object is given as input to a DNN, which learns different types of features in different stages of the architecture. Each layer can define a certain type of features, which is able to describe either single points or the whole object. Once the network has been trained for a particular task, it can be used as a feature extractor, and it is also possible to choose at which point to extract the features. In fact, Fig. 4 shows the extraction from two different layers, which will generate n-dimensional features that can describe different characteristics of the same object. Finally, these n-dimensional features are clearly difficult to study, and for this reason, they are transformed into a 2-D space using the techniques described in this section. Hence, the initial object will be mapped as a single point within the graph of the chosen technique, and the color of the point is given according to which type of error should be studied, i.e., comparison with ground truth (GT) or the class predicted by DNN.

The advantage is the ability to transform a 3-D object into any n-dimensional space in order to learn any complex but useful feature for the task to be solved, and then return to a simple 2-D space to investigate its effectiveness.

After selecting the proper visualization technique, a cluster analysis is carried out on the objects of the dataset mapped in the 2-D generated graph. This analysis makes it possible to find all objects misclassified by the network, either due to the learned features or due to the final classification part. This analysis will be described in detail in Section IV-A.

C. Interpretability Module—From Learned Features to Decision Making Operations

After displaying in 2-D space the 3-D misclassified objects, for each of these "intruders," the closest objects in the "post t-SNE" space is selected, based on the Euclidean distance. This operation allows an in-depth analysis of these samples by studying the activation in the innermost layers. Specifically, we analyze activation and Grad-CAM in output from the latest layer, namely "conv5," corresponding to the fifth layer $n \times 1024$ in Fig. 2. This layer has a size of 1024 features $\times$ 1024 points. Furthermore, "conv5" is the last layer from which information relating to points can be extracted, after which the size of the points is flattened by the subsequent max pooling and average pooling layers.

The activation on this layer can be viewed using a point cloud in which points are colored through a scalar field representing the intensity of the activation. To this aim, the activation itself (a matrix of dimension 1024 features $\times$ 1024 points) is flattened to the dimension of the features through the median. In-depth
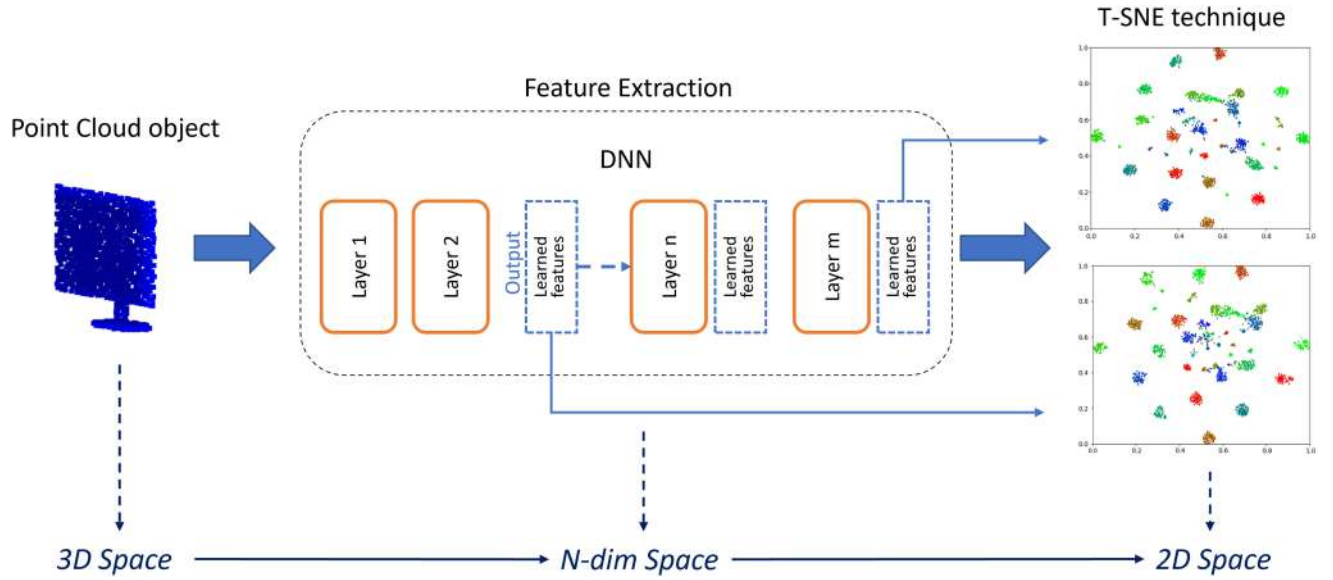
Fig. 4.　Workflow of spatial transformations of the data flow, from the input point cloud to XAI techniques.

analyses have been carried out for the choice of the median as a function to flatten the features (Section IV).

After that, the obtained values are linearly normalized between −1 and 1 based on the highest of the absolute minimum and maximum values. The result is a vector of size 1024 points with values between −1 and 1. Specifically, at least one point is mapped to 1 or −1 as a consequence of the original range.

The Grad-CAM, unlike activation, which is obtained as an output of an intermediate layer in function of the input, is parametric and must be calculated according to a target class. To obtain the Grad-CAM, in the same model and in the same "conv5" layer, the product between activation and gradient obtained from the retro-propagation of the output must be calculated (both with a size of 1024 features × 1024 points). To compute the gradient, a one-hot-encoding signal identifying the target class is multiplied by the output vector and retro-propagated to the "conv5" layer in analogy to the retro-propagation of the error during network training. The product between activation and gradient (always a matrix of dimension 1024 features × 1024 points) results accordingly, as already described for the activation, namely flattened, normalized, and "colored". An in-depth analysis was made to choose whether to multiply activation and gradient and then flatten, or flatten activation and gradient and then multiply (Section IV-B).

Activation is independent from the target class. The Grad-CAM modulates the activation with the gradient that is conditioned by the target class. The extraction of the activation and the gradient do not therefore affect the prediction: their evaluations can explain the reasons why an input object is correctly or incorrectly predicted.

IV. RESULTS

This section describes the tests carried out to understand the potential of feature visualization techniques as explainability techniques in the 3-D world. In particular, the techniques are

tested using deep learning approaches that exploit raw point clouds as input. The next subsection presents experiments on one of the fundamental tasks of this domain, that is, the classification of point clouds.

A. Visualization Module

The ModelNet40 classes of objects (bathtub, bed, chair, desk, dresser, monitor, night-stand, sofa, table, toilet, etc.) were split in the same way as the original paper. The data, provided as mesh files, were first subsampled to 1024 points for each object by performing a uniform random sampling. In addition, data augmentation techniques were performed for jittering and shuffling the data. The network was trained for 100 epochs, using the Adam optimizer with a learning rate of 0.001.

The trained network achieved a good performance in object classification, i.e., 93.27% of accuracy. In the second phase of the explainability approach, two different locations were chosen for the extraction of the learned features from the network. As described in Fig. 2, the extraction points concern the last two MLP layers. The next-to-last layer, called *MLP1*, allows 512 features to be extracted, while the last, called *MLP2*, generates 256 features. These two groups of features represent the seven and eight extraction points shown in Fig. 2.

The features extracted from this layer were given as input to the t-SNE and UMAP techniques, in order to map them in a 2-D space and understand if the network is discriminating well the different classes of the dataset within its architecture. The results of the feature visualization using the t-SNE technique are described in Fig. 5.

It can be seen from the results that in the first layer, the network is starting to discriminate well the various classes, but there are several misclassifications. It is only thanks to the last layer that the discrimination is almost optimal, obviously with some small margin of error since the accuracy of the network never reaches 100%. In fact, note that the labels used in the figure are those of
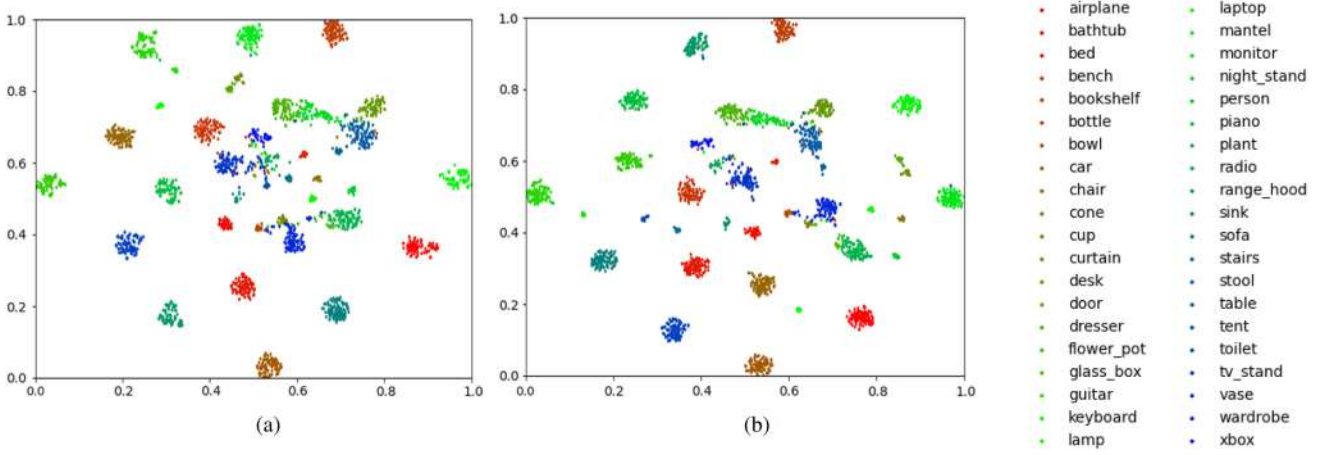
Fig. 5.   t-SNE technique applied on two different layers of DGCNN. (a) *t*-SNE on layer MLP1. (b) *t*-SNE on layer MLP2.
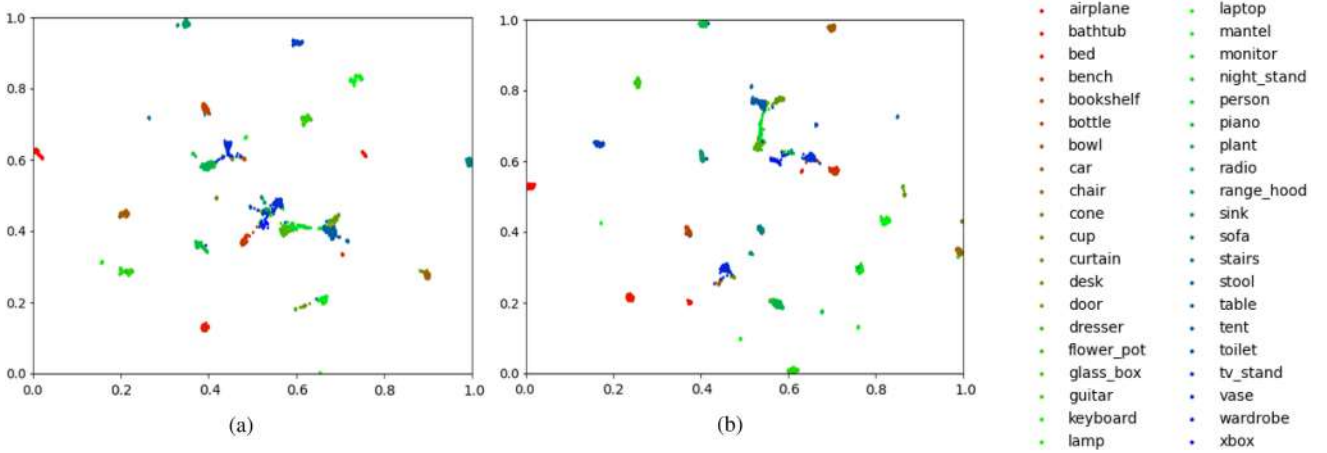

Fig. 6.   UMAP technique applied on two different layers of DGCNN. (a) UMAP on MLP1 layer. (b) UMAP on MLP2 layer.

GT, while the points are mapped using the features learned from the network. These explainability techniques allow to understand, looking at the generated plot, in which way the network associates the wrong classified objects, and to see which are the closest objects to them. They also allow to study the potential of the various intermediate layers, allowing to understand if a layer is essential for the classification or it could be removed, improving the performance both in terms of processing time and accuracy.

On the other side, the results of feature visualization using the UMAP technique are described in Fig. 6. The enormous potential of this technique compared to t-SNE is immediately evident. The results in the two intermediate layers follow the same trend of the previous results, but this technique allows a more evident discrimination of the classes and is easier to study. In fact, the objects of the various classes are distanced in a more explanatory way, allowing misclassified objects to be studied in a simpler way.

Once the visualization technique has been chosen, it is necessary to define how to map the objects within the generated diagram. There are two options, as shown in Fig. 7, and the choice depends on the type of error to be identified. Fig. 7(a) shows the mapping of objects according to the GT class. All clusters are mostly made up of points of the same color, but some are different. These "intruder" points represent objects of other classes but which, for some reason, were misleading to the network. In fact, the network has learned features from these intruders that are very similar to those of the selected cluster.

Instead, Fig. 7(b) describes the mapping of the objects according to the predicted class. It can be seen that the size of the clusters is identical to Fig. 7(a), but only the colors of some points change. The structure is identical since the position of the points derives from the same visualization technique chosen. In this case, however, the intruder points represent classification errors indicating that there is a divergence between the features learned from the network and the final decision made to classify those particular objects. In conclusion, the first type of error concerns the whole process of feature learning, while the second type describes the discrepancy between the features learned from the convolutional part of the network and the final decision taken by the last MLP layers of the approach.
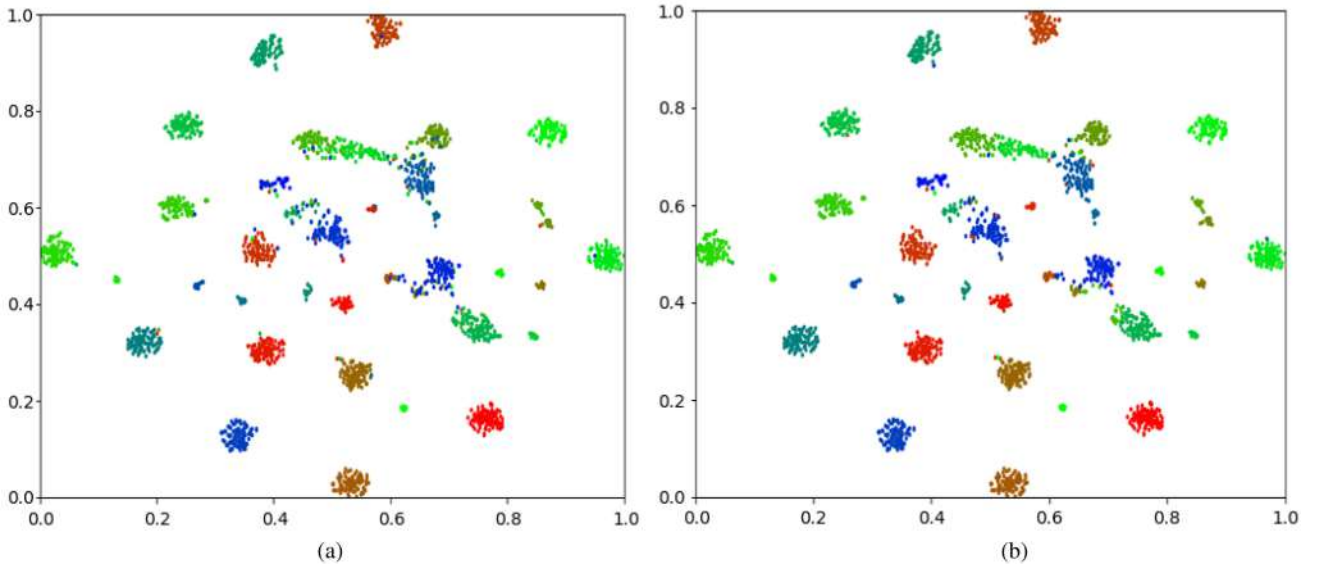
Fig. 7.   T-SNE results on MLP2 layer. (a) T-SNE results labelled by Ground Truth. (b) T-SNE results labelled by Prediction.
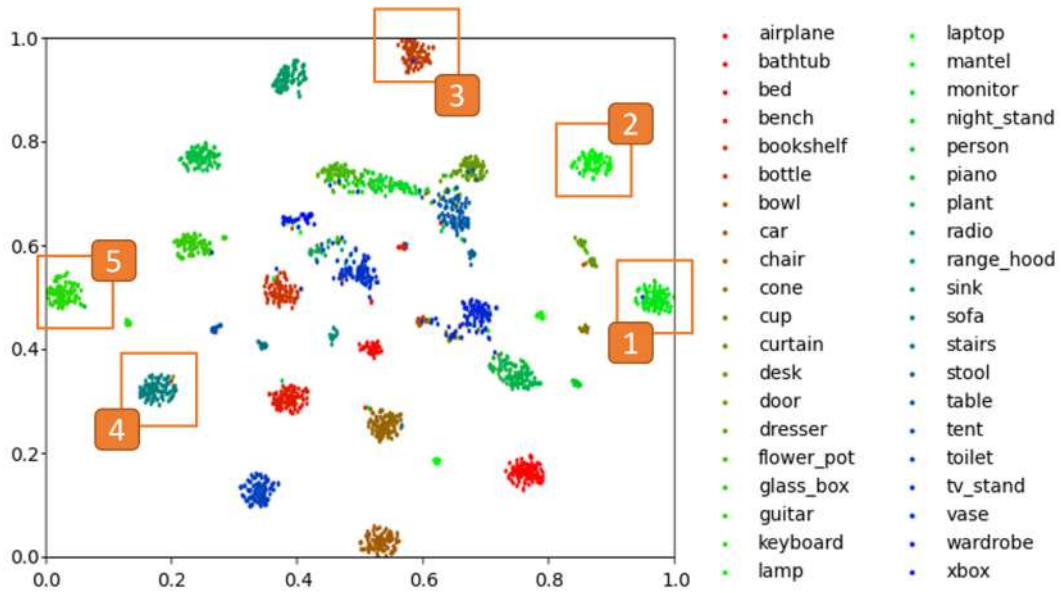

Fig. 8.   Five clusters isolated by the t-SNE technique and used to test the proposed framework.

1) XAI on Misclassified Objects: The starting point of this part of the framework is the cluster analysis got from the t-SNE output, applied to the features of the last part of the DGCNN network. This layer, named *MLP2*, allows to learn 256 features. T-SNE reduces the feature dimension from 256 to 2 components. The activation map of each object of the test set is then mapped within the new 2-D space, obtained after this transformation. In this phase, we chose to study the misclassified objects using the t-SNE visualization technique, mapping the objects by their GT label. The choice of t-SNE is motivated by the fact that the results are visually better studied than UMAP technique. Instead, the mapping respect GT was chosen because it allows to globally study the network errors according to the learned features.

As a first step, we need to identify and isolate a cluster containing at least one misclassified object. That cluster will consist of almost all objects mapped with the same class and at least one object of a different class. A cluster can be isolated by using a 2-D bounding box, as shown in Fig. 8. The proposed framework has been tested by isolating five different clusters.

The composition of the five isolated clusters is as follows.

1) Cluster 1 contains objects of the Monitor class and two intruder objects of the TV stand and Curtain classes (analyzed in Section IV-B3).
2) Cluster 2 contains objects of the Mantel class and one intruder object of the Piano class.
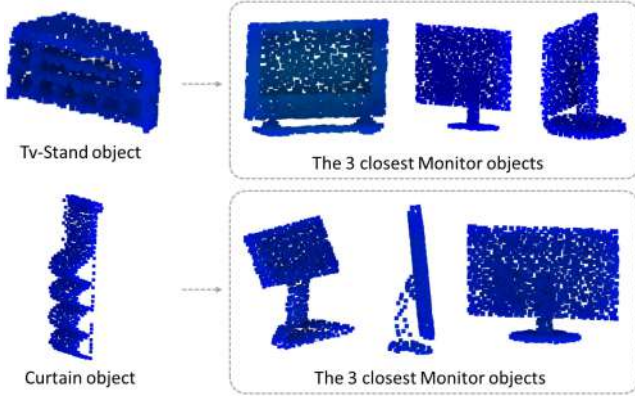
Fig. 9. TV stand intruder object and its three closest Monitor objects (up) and the Curtain intruder object and its three closest Monitor objects (bottom).

3) Cluster 3 contains objects of the Bottle class and one intruder object of the Vase class (analyzed in Section IV-B3).
4) Cluster 4 contains objects of the Sofa class and two intruder objects of the Desk and Bench classes.
5) Cluster 5 contains objects of the Guitar class and two intruder objects of the Keyboard and Plant classes (analyzed in Section IV-B3).

Specifically, for example, Cluster 1 is confined between the limits $0.9 < x < 1.0$ and $0.4 < y < 0.6$. This cluster contains all objects of class Monitor and two intruder objects of class TV stand and Curtain. By using the Euclidean distance, it is possible to extrapolate the three closest objects, in order to check which physical characteristics these objects have in common. The objective is to understand why the network has grouped this particular TV stand object and Curtain object with the Monitor cluster.

Fig. 9 shows the two intruder objects of the cluster described above. In particular, it shows the intruder object of classes TV stand and Curtain along with the three Monitor objects closest to it, in terms of Euclidean distance.

B. Interpretability Module

The implementation of Grad-CAM for this module required investigations to define the best setting to properly visualize and interpret the results. For this reason, further analyses were carried out to study the best function to flatten the size of the features (Section IV-B1) and the best combination between activation and gradient (Section IV-B2). Thanks to these results, it has been possible to suitably compare and evaluate the outcomes in Section IV-B3.

1) Choice of the Best Function to Flatten the Size of the Features: The tested domain functions have been the minimum, maximum, average (equivalent to the normalized sum), and median, namely those frequently used in the DNNs (pooling layer) to flatten the dimensions. A statistical analysis was carried out for the choice of the best function. Some samples have been selected, and, for these, the activation and gradient have been extracted to the "conv5" layer. For activation (matrix 1024 features × 1024 points), the distribution of the values between the features for each point was evaluated.

For each distribution (each of the 1024 points), the minimum (blue), maximum (orange), average (green), and median (red) values are calculated and displayed in Fig. 10.

The graph shows how the maximum function returns values almost five times (in absolute value) those returned by the minimum function. This phenomenon is due to the "LeakyReLU" activation functions with "negative slope" equal to 0.2. The values obtained can be summarized in the histograms and displayed as a point cloud in Fig. 11, in which the points of the object are colored according to same values.

The choice of the minimum and maximum as functions to flatten the size of the features was discarded because, very often, unipolar values are returned, limiting the display range from (–1 to 0) or (0 to 1) and making unusable at least half of the shades. The choice of the median, in combination with the normalization of the values adopted, with respect to the average, is the one that most frequently returns the widest display range and, consequently, gives a greater emphasis to the nuances between the points.

The same analyses have also been made for the gradient (the latter according to the targets from 0 to 40).

For each distribution (each of the 1024 points), the minimum (blue), maximum (orange), average (green), and median (red) values are calculated and summarized in Fig. 12.

From the graphs, it can be seen how the median, compared to the average, has a lower dispersion of the points around their central values. These values can be summarized in the following histograms and displayed as a point cloud, in which the points of the object are colored according to objects themselves (Fig. 13). In this case, by observing the histograms, it is clear that the functions minimum, maximum, and average often have a single peak around 0. The median is the only one that frequently presents significant peaks, and sometimes multiple ones, centered outside of 0. This feature makes it suitable to be multiplied with activation to modulate its values.

2) Choice of the Best Combination Between Activation and Gradient: A further analysis was performed to choose the best combination of activation and gradient to obtain the Grad-CAM. Two strategies were analyzed: 1) multiply activation and gradient and then flatten; 2) flatten both activation and gradient and then multiply. Fig. 14 shows the results of some tests performed.

The use of the mean, in both combinations, returns an almost monochromatic shade (green with isolated red and/or blue points) of the cloud due to the sparse distribution of the gradient values around 0. The use of the median for flattening both activation and gradient before multiplication does not add enough emphasis; the nuances obtained are almost identical to those of the activation alone. On the other hand, the use of the median to flatten the activation and gradient product creates a significant variation of the shades compared to those of activation alone.

Broadly speaking, the median, among the functions to flatten the size of the features, seems to give better results than the others, at least visually, especially in the representation of the Grad-CAM. It has been verified that the gradient between the features, at each point, has a very sparse distribution, and, for this reason, the average value is very susceptible to outliers, which move it away from the center of the distribution. While
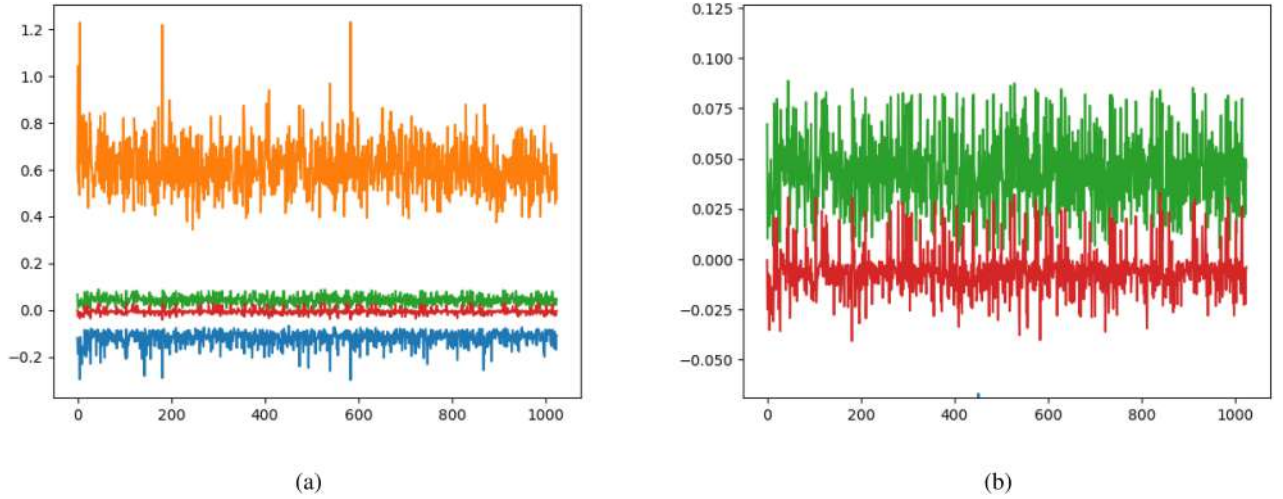
Fig. 10.   Statistics of activation for each point over all the features.
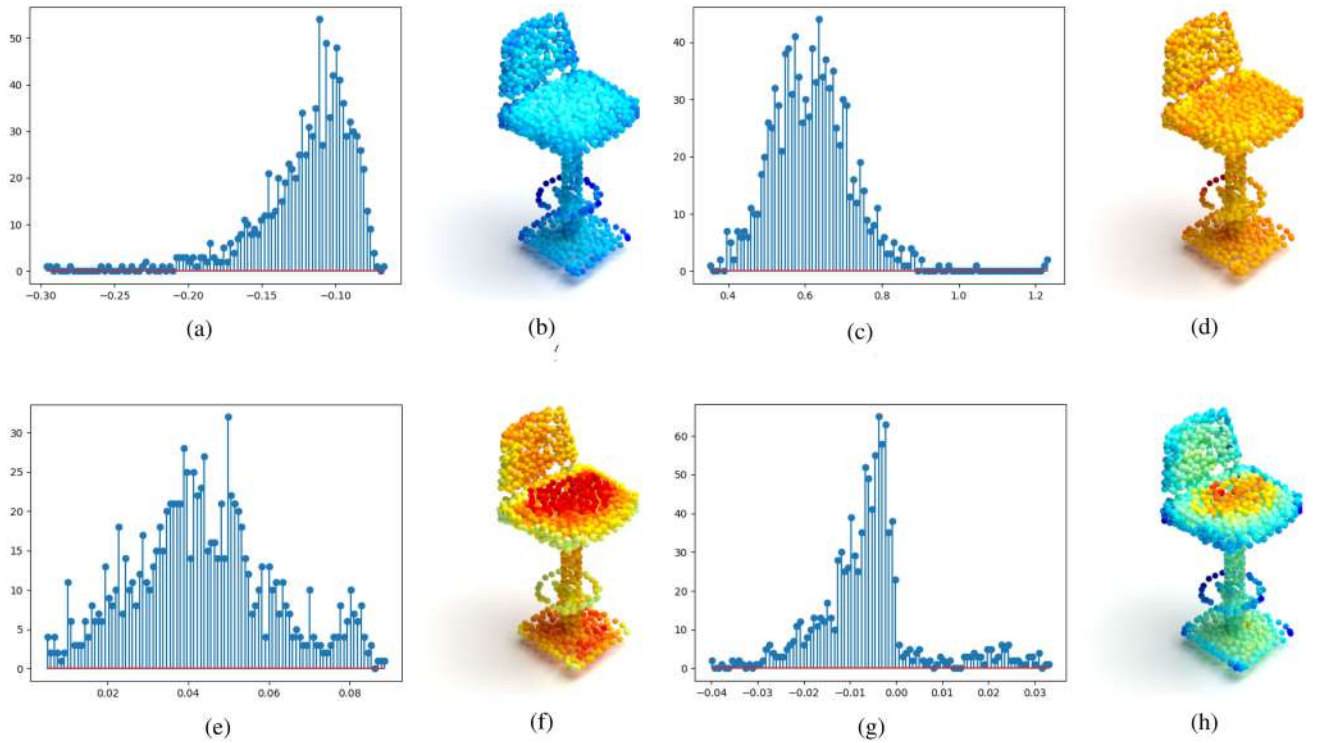

Fig. 11.   Histogram and 3-D plot of activation flattened by testing functions. (a) Histogram of *minimum* statistics of activation. (b) 3D plot of activation flatten by *minimum*. (c) Histogram of *maximum* statistics of activation. (d) 3D plot of activation flatten by *maximum*. (e) Histogram of mean statistics of activation. (f) 3D plot of activation flatten by *mean*. (g) Histogram of *median* statistics of activation. (h) 3D plot of activation flatten by *median*.

the median value is more robust, and, consequently, in the representation with the Grad-CAM, the difference in the nuances between the points is still appreciable.

3) Ablation Study: Below are represented the activation and the implemented Grad-CAM of four "intruder" objects, their closest object, and an object belonging to the cluster to which the "intruder" should belong. The Grad-CAM was evaluated only for targets equal to the GT class, post t-SNE membership

cluster, and prediction (only if different from GT and membership cluster). Each example compares an object among those classified incorrectly (prediction different from the GT). The same object is then compared to a point cloud with an equal GT class and to a GT object corresponding to the class predicted by the object under consideration. So, for each object, activation map and Grad-CAM for both classes (GT and predicted) are compared.
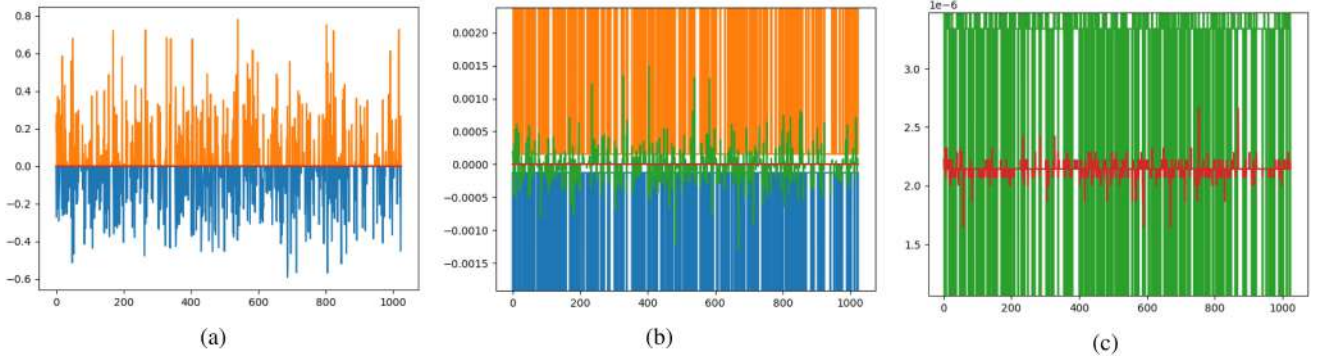
Fig. 12. Statistics of gradient for each point over all the features. Minimum (blue), maximum (orange), average (green), and median (red) values.
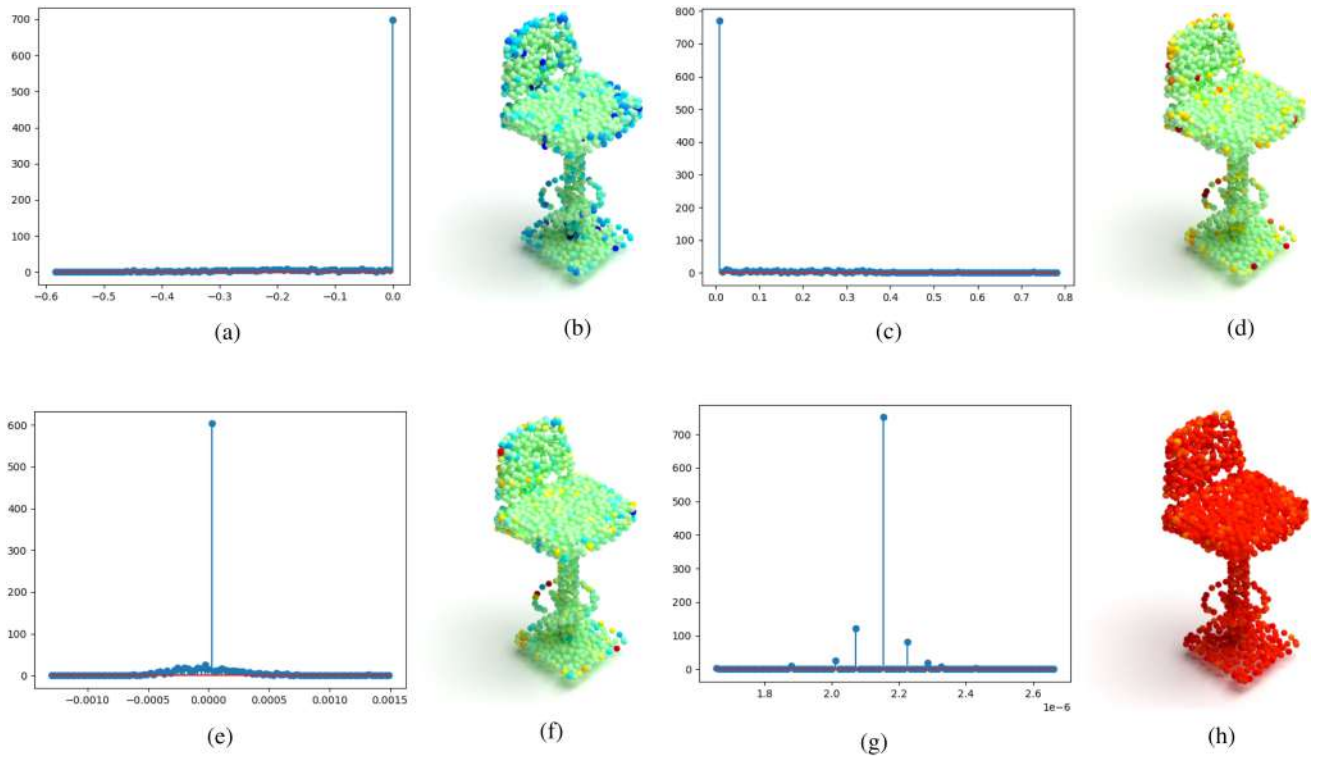

Fig. 13. Histogram and 3-D plot of gradient flattened by testing functions. (a) Histogram of *minimum* statistics of gradient. (b) 3D plot of gradient flatten by *minimum*. (c) Histogram of *maximum* statistics of gradient. (d) 3D plot of gradient flatten by *maximum*. (e) Histogram of mean statistics of gradient. (f) 3D plot of gradient flatten by mean. (g) Histogram of median statistics of gradient. (h) 3D plot of gradient flatten by median.

A colormap jet is used to emphasize the intensity of the values, where blue and red display the influential points, although their contribution is opposite. In fact, they map −1 and 1, respectively, while green is close to 0.

From Fig. 15, it can be seen how in the case of the TV stand as target, the Grad-CAM indicates as points to be considered (second column) those at the edges of the object, while the opposite behavior occurs for the target class of the Monitor (third column), where the central points of the object acquire greater importance and those at the edges lesser. If we now consider the second row, we see how the activation map of the misclassified TV stand is more similar to that of the monitor than to that of its own target class. It follows that for this object, the error is probably due to incorrectly learned features. Identical behavior was found in object 281 (Fig. 16 of the ScanObjectNN dataset. The intruder object is a display that has been incorrectly classified as a cabinet. If we consider the activation map, the one of the intruder object is certainly more similar to that of the correctly recognized display (ID 309); however, if we consider the target, it is more similar to that of the cabinet (ID 121) with the arrangement of the most relevant points mainly concentrated in the upper left corner of the vertical plane.

Similar reasoning can be made for the following example (Fig. 17). In fact, in this case, it can be clearly seen that to discriminate a Curtain (activation map of object 928, first row), the points that are taken into consideration are those at the top
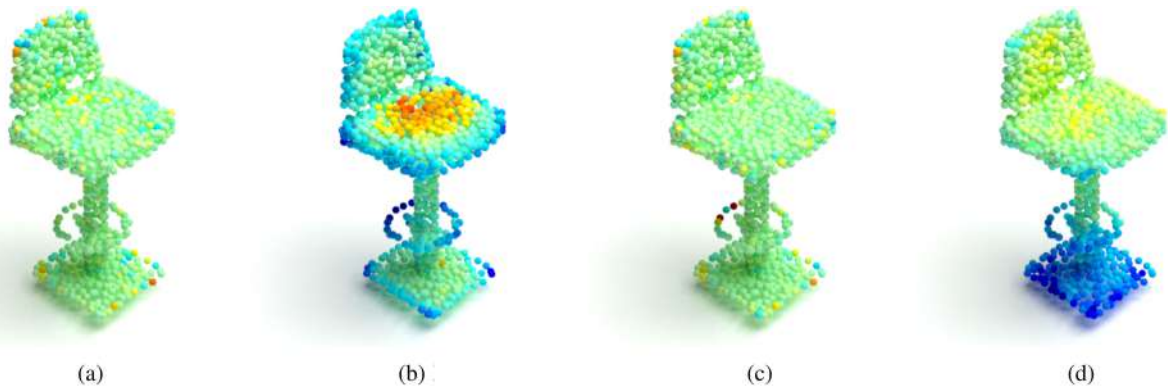
Fig. 14. Choice of the best gradient and activation combination. (a) Mean flatten before product. (b) Median flatten before product. (c) Mean flatten after product. (d) Median flatten after product.
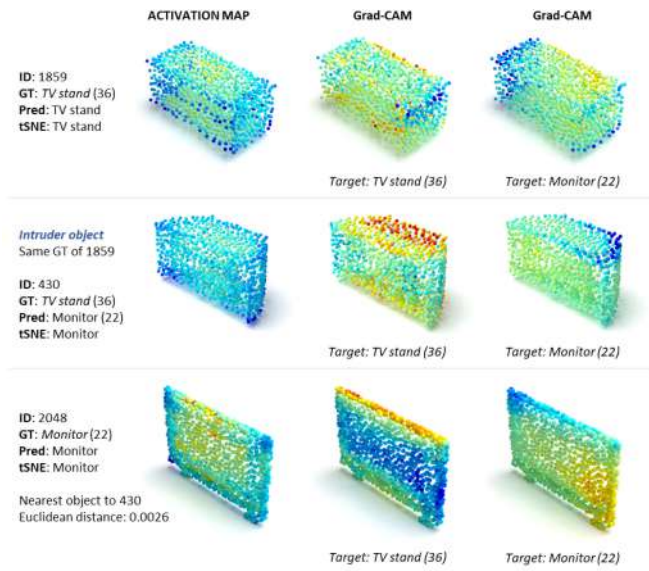

Fig. 15. Case study 1: ModelNet40. TV stand intruder object (second row) classified as Monitor both in the prediction and in the t-SNE. In brackets is the number of the object class.
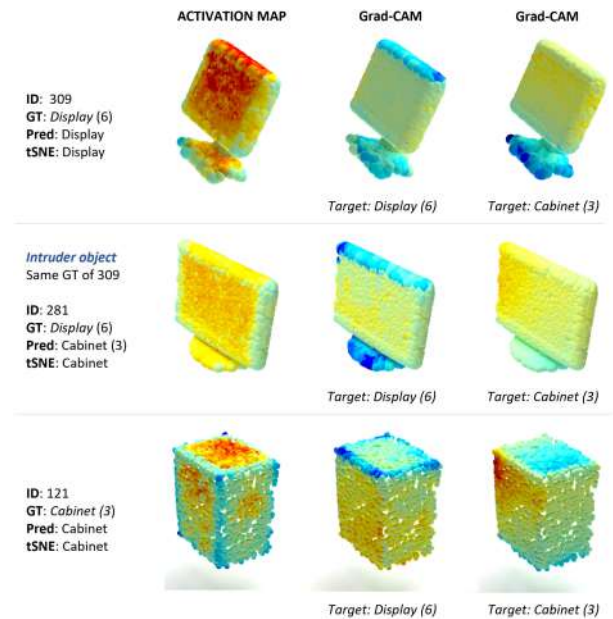

Fig. 16. Case study 1: ScanObjectNN. Display intruder object (second row) classified as Cabinet in the t-SNE. In brackets is the number of the object class.

or at the edges, with the central part activated uniformly. The intruder object (ID 600) is here a Curtain classified as a Stool in the prediction phase and as a Monitor in the Visualization Module of the t-SNE (second line). However, the classification error is due, unlike case study 1, to the geometry of the object rather than to the type of features learned. In fact, if we analyze the target classes and the closest objects in Euclidean space (ID object 2320, fourth row), we can observe how the activation points of the object 600 are similar to the activation map of the Stool and the Monitor as well as to their target classes.

A different interpretation must be made for case study 3. As shown in Fig. 18, the geometries and dimensions of the objects are very similar and the learned features do not help to discriminate correctly. In fact, both the vase and the bottle have as influential points both the top (cap or bottleneck in the case of the bottle) and the bottom part. In this case, the DNN cannot have tools for the correct identification of the object, regardless of how

it is trained, and only human knowledge, based on context, color or function can discern.

Anomalous behavior is instead recorded for the intruder object Toilet (ID 571) (Fig. 19 of the ScanObjectNN dataset). This object has been mistakenly classified as a Chair, but if we analyze its activation maps and the Grad-CAM with the Toilet target, they turn out to be very similar. The wrong result could, therefore, be due to an actual error in the network which, due to the similarity of the geometries, could have considered the central hole in the point cloud of the Toilet as a simple lack of data or an occlusion. Finally, case study 4 (Fig. 20) shows how a discrepancy in the input dataset can be detected thanks to the t-SNE. The intruder is, in fact, a plant (object ID 1675) which could have been erroneously flattened, having minimal differences in the value of the z coordinate. Precisely, for this
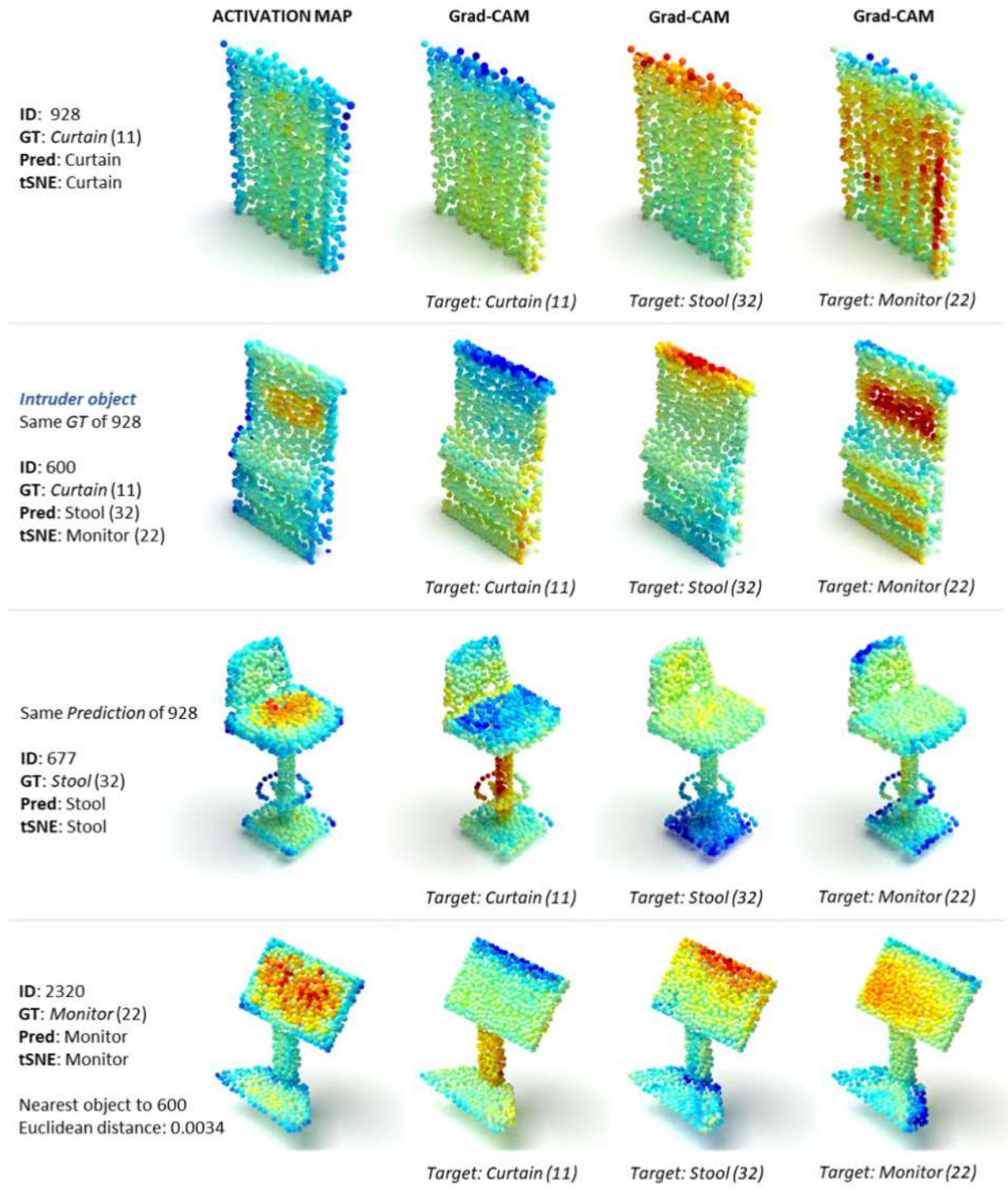
Fig. 17. Case study 2: ModelNet40. Curtain intruder object (second row) classified as Stool in the prediction phase and as Monitor in the t-SNE. In brackets is the number of the object class.

reason, it is classified as a Guitar, an almost flat object with a very similar activation map.

Broadly speaking, it can be, therefore, observed that, in most cases, objects of the same or similar class have analogous shades between their respective activation maps and Grad-CAM (given a target). Often for misclassified objects, the Grad-CAM with a target equal to the predicted class has shades more comparable to the Grad-CAM (same target class) of the object of the erroneously predicted class. On the contrary, the Grad-CAM with a target equal to the GT class of the wrongly classified object has shades less similar to the Grad-CAM (same target class) of the object of the same class.
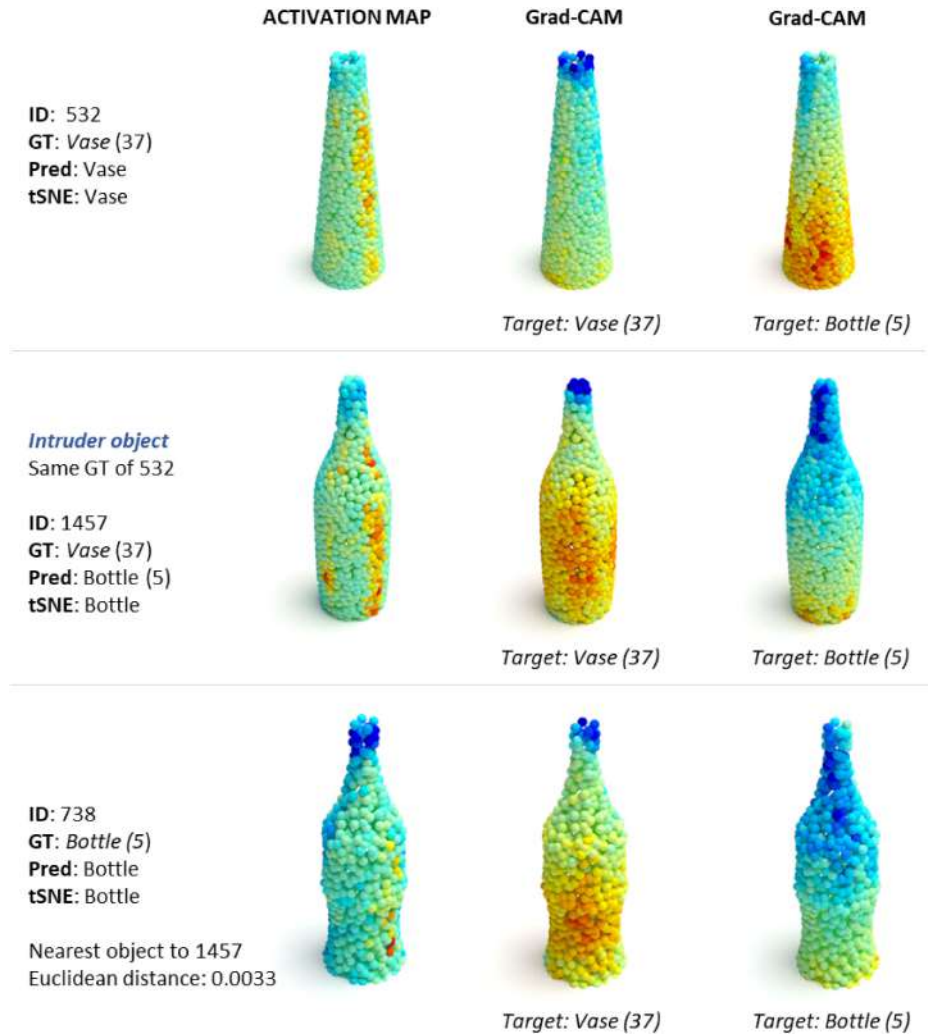
Fig. 18.   Case study 3: ModelNet40. Vase intruder object (second row) classified as Bottle both in the prediction and in the t-SNE. In brackets is the number of the object class.

V. DISCUSSIONS

As regards the Visualization Module, it was possible to adapt the UMAP and t-SNE 2-D techniques also to 3-D data such as point clouds. In particular, as regards the t-SNE, it displays clusters based on n-features and then it colors them according to the prediction or GT. In the first case, if we consider, for example, a cluster with a certain color (e.g., green, corresponding to class #1) and we notice that it comprises an object (a point) with a different color (e.g., blue, corresponding to class #2), this indicates that the network has predicted another class for that point, far from the cluster it is inserted in. Nevertheless, we would not be able to understand whether it is predicted incorrectly or not (we thus should investigate its GT). The three cases that therefore arise would allow us to understand.

1) If GT is equal to class #1, while prediction is equal to class #2, then the network generated features similar to class #1 but ultimately predicted #2. We can, thus, assume the features were correctly learned, but the classification layer of the network was wrong to predict.

2) If both GT and the prediction are equal to class #2, then the network has correctly predicted the class, but, in a way or other, its features are very similar to those of class #1. We can deduce that the features learned are not as discriminating as we think.

3) If the GT is bigger than class #2, while the prediction is equal to class #2, then the DNN has learned totally wrong features; so it should be retrained with different parameters or the input dataset could contain some issues.

If, on the other hand, we do not compare the result to the GT, we will study the predictions in an unsupervised way. In fact, if we consider the above described example, it would only indicate that features very similar to those of class #1 led to the prediction of another class. So, where does the error of this prediction lie? Is it the fault of the classification layer or is it the fault of the clustering made by t-SNE (so perhaps the small loss of information due to the transition from n-dimensions to 2D)? This question remains still unanswered, but future works will investigate it.

On the other side, if we just map the t-SNE on the GT (not on the prediction) and we notice that a point has a different color than the cluster in which it is inserted, it means that the point has features very similar to class #1, but then it was predicted as
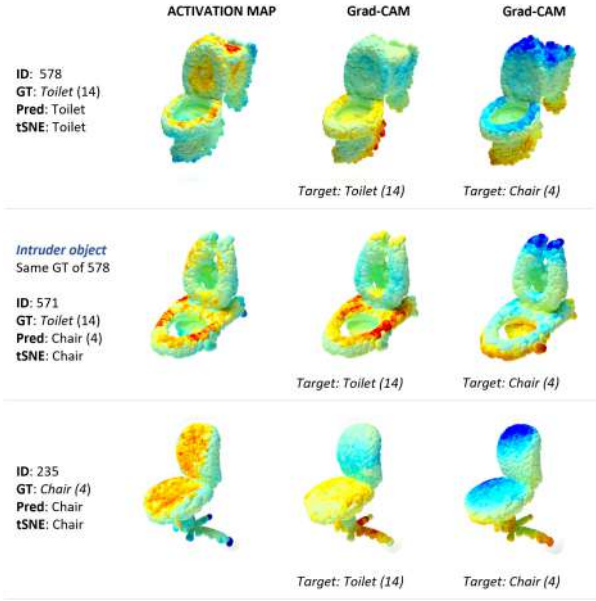
Fig. 19. Case study 2: ScanObjectNN. Toilet intruder object (second row) classified as Chair in the t-SNE. In brackets is the number of the object class.
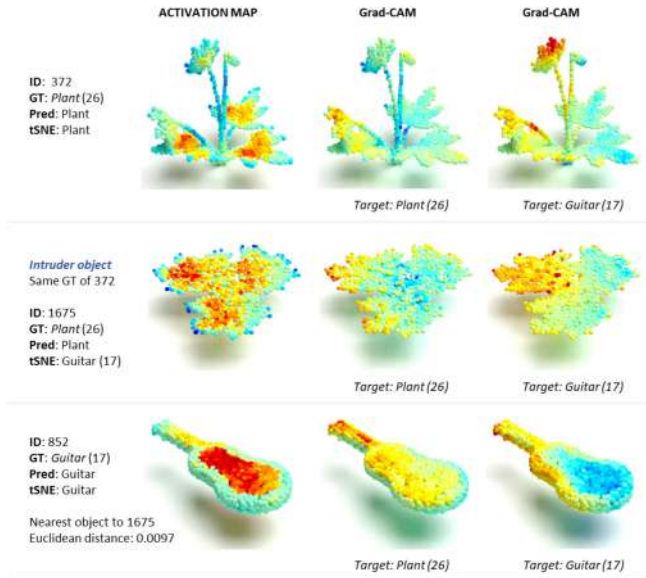

Fig. 20. Case study 4: ModelNet40. Plant intruder object (second row) correctly classified in the prediction phase but confused with a Guitar in the t-SNE. In brackets is the number of the object class.

class #2. So we can quickly interpret that the point constitutes an error entirely due to the network.

As for the Interpretability Module, the adaptation of Grad-CAM to the 3-D space and the comparison with the target class allowed to identify possible different causes for the wrong prediction of pilot objects. Some errors could be solved with a simple check of the input dataset or with an adjustment of the network training parameters. In other cases, however, a limitation of neural networks has been highlighted which, without human interpretation, are not able to correctly classify objects that are very similar to each other but belonging to different classes. In this way, the user is able to understand and investigate

the criticalities of the network, choosing the best solution to remedy.

VI. CONCLUSION AND FUTURE WORKS

The need to process ever-increasing large amounts of 3-D data is driving the geoscientific community to provide more accurate, less uncertain, and physically consistent models to interpret such complex data. Even if NNs for point-cloud processing achieved, to some extent, a high degree of attention, few studies are focusing on explainability. In this work, we proposed an explainability framework for point-cloud classification. The method here proposed is based on t-SNE [23] and UMAP [24] but adapted for 3-D data, preserving the local structure by grouping neighboring data points. This allows the identification of intruder classes by clustering the related points; besides, once the misclassified objects are identified, a gradient-based visualization module [25] plots, with a very high degree of interpretability, the misleading feature of such objects, defining a new interpretation method even for non-AI experts. This latter provided intuitive analyses for misclassified sample.

A future development of this study is its application to architectural-urban geospatial datasets such as SynthCity [44], S3DIS [45], Semantic3D [46], or ArCH [47] to investigate its behavior with larger objects and with a wider extension. This analysis could also be associated with a computation of the Euclidean distance on the point cloud itself to evaluate the effectiveness of the method and evaluate which objects, even if distant in the physical space, result close in the feature space, highlighting the abilities of the network to learn the proper features.

As for the reliability examination of explainability methods, most of the works validate the explanation results subjectively based on human interpretations. Therefore, quantitative evaluations are increasingly recognized as an essential requirement in the explainable deep learning domain. Some attempts have been done in the literature to define quantitative evaluation metrics of XAI [32] but cannot be adopted for BubblEX since it combines two different modules: visualization and interpretability. We will further investigate by defining new evaluation metrics used to compare the performance of our approach with the other proposed in the literature.

REFERENCES

[1] N. Marchi, F. Pirotti, and E. Lingua, "Airborne and terrestrial laser scanning data for the assessment of standing and lying deadwood: Current situation and new perspectives," *Remote Sens.*, vol. 10, no. 9, 2018, Art. no. 1356.

[2] X. Han, Z. Dong, and B. Yang, "A point-based deep learning network for semantic segmentation of MLS point clouds," *ISPRS J. Photogrammetry Remote Sens.*, vol. 175, pp. 199–214, 2021.

[3] Y. Li, B. Wu, and X. Ge, "Structural segmentation and classification of mobile laser scanning point clouds with large variations in point density," *ISPRS J. Photogrammetry Remote Sens.*, vol. 153, pp. 151–165, 2019.

[4] K. Mirzaei, M. Arashpour, E. Asadi, H. Masoumi, Y. Bai, and A. Behnood, "3D point cloud data processing with machine learning for construction and infrastructure applications: A comprehensive review," *Adv. Eng. Informat.*, vol. 51, 2022, Art. no. 101501.

[5] X. X. Zhu et al., "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Apr. 2017.

[6] L. Rossi, A. Ajmar, M. Paolanti, and R. Pierdicca, "Vehicle trajectory prediction and generation using LSTM models and GANs," *PLoS One*, vol. 16, no. 7, 2021, Art. no. e0253868.

[7] Y. Guo, W. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021.

[8] A. Esteva et al., "A guide to deep learning in healthcare," *Nat. Med.*, vol. 25, no. 1, pp. 24–29, 2019.

[9] M. Paolanti, R. Pietrini, A. Mancini, E. Frontoni, and P. Zingaretti, "Deep understanding of shopper behaviours and interactions using RGB-D vision," *Mach. Vis. Appl.*, vol. 31, no. 7, pp. 1–21, 2020.

[10] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *J. Field Robot.*, vol. 37, no. 3, pp. 362–386, 2020.

[11] Y. Xiao, J. Wu, Z. Lin, and X. Zhao, "A deep learning-based multi-model ensemble method for cancer prediction," *Comput. Methods Programs Biomed.*, vol. 153, pp. 1–9, 2018.

[12] H. Elhamdadi, S. Canavan, and P. Rosen, "AffectiveTDA: Using topological data analysis to improve analysis and explainability in affective computing," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 1, pp. 769–779, Jan. 2021.

[13] J. D. Fuhrman, N. Gorre, Q. Hu, H. Li, I. El Naqa, and M. L. Giger, "A review of explainable and interpretable AI with applications in COVID-19 imaging," *Med. Phys.*, vol. 49, pp. 1–14, 2021.

[14] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable AI: A review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, 2021, Art. no. 18.

[15] A. B. Arrieta et al., "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, 2020.

[16] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision-making and a "right to explanation"," *AI Mag.*, vol. 38, no. 3, pp. 50–57, 2017.

[17] M. Graziani, V. Andrearczyk, S. Marchand-Maillet, and H. Müller, "Concept attribution: Explaining CNN decisions to physicians," *Comput. Biol. Med.*, vol. 123, 2020, Art. no. 103865.

[18] P.-J. Kindermans et al., "The (un) reliability of saliency methods," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Berlin, Germany: Springer, 2019, pp. 267–280.

[19] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[20] 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:2007.12525*.

[21] 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*. Berlin, Germany: Springer, 2019, pp. 48–55.

[22] B. Zhao et al., "Evaluation of convolution operation based on the interpretation of deep learning on 3D point cloud," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5088–5101, Aug. 2020.

[23] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[24] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*.

[25] 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 *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.

[26] 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 Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.

[27] Z. Wu et al., "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.

[28] M. A. Uy, Q.-H. Pham, B.-S. Hua, D. T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1588–1597.

[29] S. Razavi, "Deep learning, explained: Fundamentals, explainability, and bridgeability to process-based modelling," *Environ. Modelling Softw.*, vol. 144, 2021, Art. no. 105159.

[30] European Commission, "High-level expert group on artificial intelligence," 2019. Accessed: March 20, 2022. [Online]. Available: https://www.aepd.es/sites/default/files/2019-12/ai-definition.pdf

[31] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *Proc. IEEE 5th Int. Conf. Data Sci. Adv. Analytics*, 2018, pp. 80–89.

[32] H. Tan and H. Kotthaus, "Surrogate model-based explainability methods for point cloud NNs," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 2239–2248.

[33] M. Zhang, H. You, P. Kadam, S. Liu, and C.-C. J. Kuo, "PointHop: An explainable machine learning method for point cloud classification," *IEEE Trans. Multimedia*, vol. 22, no. 7, pp. 1744–1755, Jul. 2020.

[34] T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren, "Pointcloud saliency maps," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1598–1606.

[35] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *Adv. Neural Inf. Process. Syst., 31*, vol. 31, 2018.

[36] T. Hanxiao and H. Kotthaus, "Surrogate model-based explainability methods for point cloud NNs," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 2239–2248.

[37] A. Gupta, S. Watson, and H. Yin, "3D point cloud feature explanations using gradient-based methods," in *Proc. Int. Joint Conf. Neural Netw.*, 2020, pp. 1–8.

[38] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[39] Y. Yang, J. Qiu, M. Song, D. Tao, and X. Wang, "Distilling knowledge from graph convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7074–7083.

[40] R. Pierdicca et al., "Point cloud semantic segmentation using a deep learning framework for cultural heritage," *Remote Sens.*, vol. 12, no. 6, 2020, Art. no. 1005.

[41] K. Zhang, M. Hao, J. Wang, C. W. de Silva, and C. Fu, "Linked dynamic graph CNN: Learning on point cloud via linking hierarchical features," 2019, *arXiv:1904.10014*.

[42] A. Cheraghian, S. Rahman, D. Campbell, and L. Petersson, "Transductive zero-shot learning for 3D point cloud classification," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 923–933.

[43] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdiscipl. Rev.: Comput. Statist.*, vol. 2, no. 4, pp. 433–459, 2010.

[44] D. Griffiths and J. Boehm, "SynthCity: A large scale synthetic point cloud," 2019, *arXiv:1907.04758*.

[45] I. Armeni et al., "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1534–1543.

[46] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "Semantic3D. net: A new large-scale point cloud classification benchmark," 2017, *arXiv:1704.03847*.

[47] F. Matrone et al., "A benchmark for large-scale heritage point cloud semantic segmentation," in *Proc. XXIV ISPRS Congr.*, 2020, vol. 43, pp. 1419–1426.

Francesca Matrone received the Ph.D. degree in urban and regional development from Politecnico di Torino, Turin, Italy, in 2021, with a thesis titled Deep Semantic Segmentation of Cultural Built Heritage Point Clouds.

She is currently a Postdoc Research Fellow with the Geomatics Laboratory, Department of Environment, Land and Infrastructure Engineering (DIATI), Polytechnic University of Turin, Turin. She is the main promoter of the ArCH (Architectural Cultural Heritage) dataset for the semantic segmentation on built heritage point clouds and organizer of international workshops and tutorials on this topic. She has been a Visiting Researcher with the Computer Science and Artificial Intelligence Lab (CSAIL), Massachusetts Institute of Technology (MIT), Boston, MA, USA, and the National Institute of Applied Sciences of Strasbourg, Strasbourg, France. Her research focuses on deep learning techniques applied to geospatial data.

Marina Paolanti (Member, IEEE) received the Ph.D. degree from the Department of Information Engineering (DII), Universitá Politecnica delle Marche, Ancona, Italy, with a thesis titled "Pattern Recognition for Challenging Computer Vision Applications."

She is currently an Assistant Professor with the Department of Political Sciences, Communication and International Relations, University of Macerata, Macerata, Italy, and an Adjunct Professor with the DII. During the Ph.D. studies, she has worked with GfK Verein, Nuremberg, Germany, for visual and textual sentiment analysis of brand-related social media pictures using deep convolutional neural networks. Her research interests include artificial intelligence and computer vision, with particular focus to specialized machine learning algorithms and deep learning architectures.

Prof. Paolanti is a Member of CVPL.

Massimo Martini received the Ph.D. degree in semantic segmentation of point clouds from the Department of Information Engineering (DII), Universitá Politecnica delle Marche, Ancona, Italy, in 2021.

His research interests include point-cloud semantic segmentation, reidentification, and social media intelligence.

Andrea Felicetti received the Ph.D. degree from the Department of Information Engineering (DII), Universitá Politecnica delle Marche, Ancona, Italy, in 2021.

He is currently a Postdoc Research Fellow with DII, Universitá Politecnica delle Marche. During his Ph.D. studies, he was hosted at the Universitat Politènica de València (UPV), Valencia, Spain, under the supervision of Prof. Josè Louis Lerma. His research focuses on GeoAI applications, i.e., artificial intelligence approaches for the interpretation of complex geomatics data.

Roberto Pierdicca received the Ph.D. degree in information engineering from the Universitá Politecnica delle Marche, Ancona, Italy, in 2017.

He is currently an Assistant Professor and Adjunct Professor of geomatics. He is the author of more than 100 papers in international conferences and journals and participates in several international projects related to cultural heritage, tourism management, education, geosciences, and archaeology. His research interests include geospatial intelligence, photogrammetry, remote sensing, and 3-D point clouds.