INTERPRETABLE POINT CLOUD CLASSIFICATION USING MULTIPLE INSTANCE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

3D image analysis is crucial in fields such as autonomous driving and biomedical research. However, existing 3D point cloud classification models lack interpretability, limiting trust and usability in safety-critical applications. To address this, we propose POINTMIL, an inherently locally interpretable point cloud classifier using Multiple Instance Learning (MIL). POINTMIL offers local interpretability, providing fine-grained point-specific explanations to point-based models without the need for *post-hoc* methods, addressing the limitations of global or imprecise interpretability approaches. We applied POINTMIL to four popular point cloud classifiers, PointNet, DGCNN, CurveNet, PointMLP and PointNeXt, and proposed a transformer-based backbone to extract high-quality point-specific features. POINTMIL made these models inherently interpretable while increasing predictive performance on standard benchmarks (ModelNet40, ShapeNetPart) and achieving state-of-the-art mACC (97.3%) and F1 (97.5%) on the IntrA biomedical data set, and another dataset of biological cells. To our knowledge, this is the first work to apply MIL to interpretable point cloud classification.

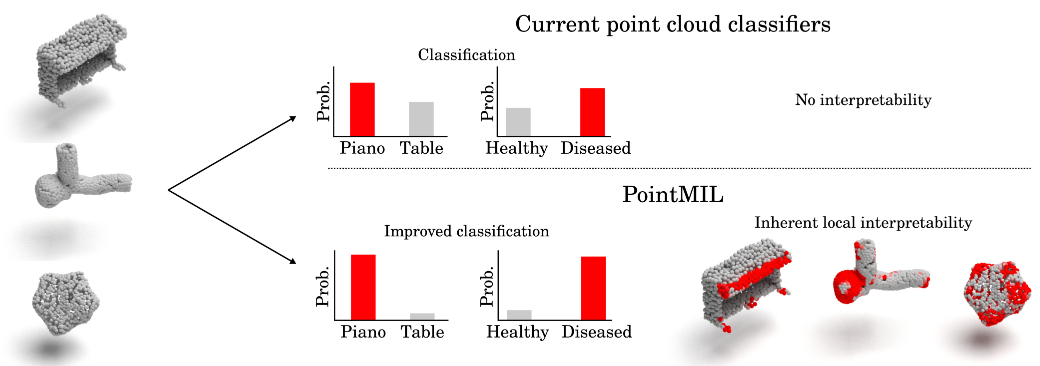

Figure 1: Current point cloud classifiers usually only provide predictive probabilities. We propose POINTMIL to inherently incorporate interpretability and improve predictive performance into point-based architectures.

1 INTRODUCTION

Three-dimensional (3D) imaging data is prevalent in various fields, including autonomous driving, augmented reality, robotics, and biology. In autonomous driving, 3D point clouds enable vehicles to perceive and navigate their surroundings safely, identifying obstacles and road features. In biology, the 3D shape of cells has provided insight into the underlying cell state (Viana et al., 2023), enabling advances in diagnostics (Song et al., 2024) and drug discovery.

Significant progress has been made in the processing of point clouds representations of 3D shapes for classification and segmentation tasks (Guo et al., 2020). However, most methods do not explain their decision-making, which limits adoption in real world scenarios due to concerns about safety and trustworthiness (Rudin, 2019; Rudin et al., 2022). Despite significant advancements in the interpretability of machine learning models in 2D image analysis (Zhang et al., 2021; Wang et al., 2023; Hu et al., 2024; Paul et al., 2024), there has been a lack of research on the interpretability of 3D point cloud models. More so, of those proposed, the majority are either *post-hoc*, meaning that an extra modelling step is required to obtain interpretations, or they are *globally* interpretable, meaning that they lack the ability to offer fine-grained, point-specific explanations.

To address these challenges and elucidate the model's decision-making process, we propose POINTMIL, an inherently interpretable classification framework for point clouds that offers fine-grained, *local* and class-specific interpretations using Multiple Instance Learning (MIL; Dietterich et al. (1997)). Given its ability to handle data organised into bags of instances, MIL is well suited for point cloud analysis, especially in bioimaging domains, where each point in a point cloud is assigned the same label, but only certain points are discriminatory (Yang et al., 2020). Building on this foundation, we present a model that leverages the strengths of MIL to offer robust performance and interpretability in point cloud classification. Furthermore, we introduce a contextual attention mechanism, which incorporates neighbourhood information into the attention calculation, addressing the sparsity of traditional attention methods and enabling smoother, more coherent attention distributions. This adaptation ensures that the model can better capture local geometric relationships within the point cloud, improving both classification performance and interpretability. Our main contributions are as follows:

1. We propose POINTMIL, a point-based classification pipeline based on MIL, to offer inherent *local* interpretability and enhanced classification performance to existing point-based feature extractors.
2. We adapt and introduce a new transformer-based model to extract high-quality point-specific features from a point cloud.
3. We incorporate contextual attention to address sparsity in attention weights, improving interpretability and classification performance by leveraging local neighbourhood information.
4. We show the generality of POINTMIL on *de-facto* public benchmarks (ModelNet40 (Wu et al., 2015) and ShapeNetPart (Yi et al., 2016)) and biomedical imaging datasets, achieving the state-of-the-art (SOTA) on IntrA (Yang et al., 2020).

2 RELATED WORK

Point cloud analysis: One of the first methods that used unordered point clouds directly for classification and segmentation was PointNet (Qi et al., 2017a). PointNet, however, ignored local relationships between points. Subsequently, PointNet++ (Qi et al., 2017b) introduced hierarchical feature learning to capture locality recursively. Many modern algorithms are built on the design philosophy of PointNet++, including convolutional kernel-based (Li et al., 2018b; Thomas et al., 2019; Wu et al., 2019), graph-based (Wang et al., 2019a;b; Xu et al., 2020), MLP-based (Choe et al., 2022; Ma et al., 2022), and transformer-based methods (Zhang et al., 2020; Zhao et al., 2021; Guo et al., 2021; Yu et al., 2021; Cheng et al., 2022; Akwensi et al., 2024). Although significant progress has been made in advancing classification and segmentation accuracy, little work has focused on interpretability.

Interpretability on point clouds: Interpretability methods can be classified along two key dimensions: (1) the stage at which interpretability is introduced and (2) the scope of the explanations provided. Regarding the stage, methods are either *post-hoc* or *inherently interpretable*. *Post-hoc* methods generate explanations after the model has made its predictions, often through additional analysis, approximation techniques, or assessing gradients with respect to the input (Zhou et al., 2016). In contrast, *inherently interpretable* methods are designed to integrate interpretability into the model itself, producing explanations as part of the prediction process. With respect to scope, methods are categorised as either *local* or *global*. *Local* approaches focus on explaining individual predictions, offering insights specific to a single input. *Global* approaches aim to provide a holistic understanding of the model's behaviour across all inputs. Since PointNet ++ (Qi et al., 2017b), many point-based models have used some form of sampling and grouping (Guo et al., 2021; Zhao et al., 2021; Xiang et al., 2021; Ma et al., 2022), thus losing point-level information in the classification stage. Therefore, most *local* interpretability methods for point cloud classification are *post-hoc*, including gradient-based (Zhang et al., 2019; Huang et al., 2020) and surrogate models (Tan & Kotthaus, 2022) based on LIME (Ribeiro et al., 2016). Zhang et al. (2019) and Huang et al. (2020) developed explainability methods for PointNet using global average pooling (GAP) and class activation maps. Taghanaki et al. (2020) introduced a module into point set encoders that masked points with negligible contributions, leaving only informative points in the classification layer. Similarly, Zheng et al. (2019) obtained saliency maps by shifting points to the object centroid and calculating the corresponding loss gradient with respect to the shifted points. However, *post hoc* methods have been shown to be deceptive and often troublesome (Laugel et al., 2019; Rudin et al., 2021; Feng et al., 2024). For example, the interpretations of *post hoc* methods can differ depending on the interpretability methods (Li et al., 2018a), leading to convincing but conflicting interpretations for the same classification. *Post-hoc* methods also involve an additional modelling step, raising further concerns about the precision of their interpretations Fan et al. (2021). Few inherently interpretable methods for point cloud classifications have been proposed, and of these, most are *global*. Arnold et al. (2023) developed XPCC, a prototype-based interpretable model that used point cloud representation distributions to learn class-specific prototypes. Similarly, Feng et al. (2024) developed Interpretable3D, a prototype-based global interpretability model that can be used in conjunction with other model architectures for classification and segmentation. However, none of these inherently interpretable methods offers local interpretations on a point-level basis. While global interpretability provides valuable insights into the overall behaviour of a model, local methods can be especially beneficial when understanding specific, individual predictions is crucial, offering more granular and context-sensitive explanations. To our knowledge, no one has yet offered an inherently *locally* interpretable model for point cloud classification. PointMIL utilises MIL to offer an inherently *locally* interpretable model.

Multiple instance learning: In the typical binary MIL problem, a bag is labelled positive if and only if at least one of its instances is labelled positive (Dietterich et al., 1997); however, there is no access to individual instances during training. MIL algorithms then attempt to classify entire bags of instances and often pinpoint important or class conditional discriminatory instances as interpretability output. Many MIL methods have been proposed for drug activity prediction (Dietterich et al., 1997), video image analysis (Ali & Shah, 2010), and cancer detection and sub-typing (Ilse et al., 2018; Shao et al., 2021; Lu et al., 2021; Fourkioti et al., 2024). Recently, Early et al. (2024) extended MIL to time series classification in an interpretable plug-and-play framework. However, to our knowledge, no one has used MIL for interpretable point cloud classification.

3 Methods

Given a point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3} = \{\mathbf{p}_i | i = 1, \ldots, N\}$, consisting of N points in Cartesian space (x, y, z), and their associated d-dimensional point features (often point normals, however, these can be the point coordinates if no point-level features exist) $\mathbf{F} \in \mathbb{R}^{N\times d_{in}} = \{\mathbf{f}_i | i = 1, \ldots, N\}$, traditional point-based methods use a point-based encoder f_{enc} to learn a global representation $\mathbf{z} \in \mathbb{R}^d$ for $\mathbf{P}$ by aggregating the points with equal weighting (often through adaptive pooling), followed by a classification head f_{clf}.

We propose a new approach by learning a representation $\mathbf{z}_i \in \mathbb{R}^d$ for each point $\mathbf{p}_i$ for $i \in \{1, \ldots, N\}$, and then applying MIL pooling for simultaneous classification and interpretability. Our framework consists of a point-based feature extractor f_{enc} and a MIL pooling module f_{MIL}.

3.1 Feature extractor

To develop a point-level feature extractor, we follow much of the Transformer block from Yu et al. (2021). However, unlike Yu et al. (2021), we did not use point sampling strategies. Furthermore, we did not use their multi-graph reasoning. This feature extractor aimed to incorporate contextual information into the point cloud features by: (1) grouping points with k-Nearest Neighbours (k-NN), (2) including relative positional embeddings, and (3) refining point-level features through an attention mechanism. These are detailed in Appendix A.

We also presented analysis on PointNet (Qi et al., 2017a), DGCNN (Wang et al., 2019b), CurveNet (Xiang et al., 2021), PointMLP (Ma et al., 2022), and PointNeXt (Qian et al., 2022) feature extractors. For PointNet and DGCNN we replaced the classification heads of these architectures with MIL pooling described in Section 3.2. CurveNet and PointMLP downsample the original point cloud. In order to retain point-level features for every point, we slightly adapted these architectures to remove point sampling. We show the affect of this adaptation on classification results so that any difference in performance can then be attributed to the MIL pooling instead of this adaptation. We used PointMLPElite for our analysis. For PointNeXt-S, we slightly adapted the architecture such that point-level features from the first layer were concatenated with global features from the last layer before input into our MIL pooling. These adaptations are discussed further in Appendix A. Each feature extractor produced d-dimensional point-level features $\mathbf{Z} \in \mathbb{R}^{N \times d} = f_{enc}(\mathbf{P})$, for N points which were fed into different MIL pooling algorithms.

3.2 MIL POOLING

After obtaining feature representations $\mathbf{z}_i$ for each point $\mathbf{p}_i$, we evaluated four MIL pooling methods that offer inherent interpretability, `Instance` (Wang et al., 2018), `Attention` (Ilse et al., 2018), `Additive` (Javed et al., 2022), and `Conjunctive` (Early et al., 2024).

`Instance` pooling predicts the label of each point through an instance classifier and then pools the predictions by taking the mean:

$$\hat{\mathbf{y}}_i \in \mathbb{R}^c = f_{clf}\left(\mathbf{z}_i\right); \qquad \hat{\mathbf{Y}} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{\mathbf{y}}_i\right), \tag{1}$$

where c is the number of classes.

`Attention` pooling calculates the attention weights of the point features through an MLP, calculates a weighted average feature representation for the point cloud using those weights and then classifies that features using an MLP:

$$a_i \in [0,1] = f_{attn}\left(\mathbf{z}_i\right); \qquad \hat{\mathbf{Y}} = f_{clf}\left(\frac{1}{N}\sum_{i=1}^{N} a_i \mathbf{z}_i\right). \tag{2}$$

`Additive` pooling calculates attention weights for each point feature, then classifies each point according to its weighted feature vector, and finally produces a bag classification from the mean of all weighted instance classifications:

$$a_i \in [0,1] = f_{attn}\left(\mathbf{z}_i\right); \quad \hat{\mathbf{y}}_i = f_{clf}\left(a_i \mathbf{z}_i\right); \quad \hat{\mathbf{Y}} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{\mathbf{y}}_i\right). \tag{3}$$

`Conjunctive` pooling trains the point attention and point classification heads independently so that attention weights and point predictions are computed on the features alone. The final point cloud classification is given by the weighted sum of the point classifications weighted by the attention weights:

$$a_i \in [0,1] = f_{attn}\left(\mathbf{z}_i\right); \quad \hat{\mathbf{y}}_i = f_{clf}\left(\mathbf{z}_i\right); \quad \hat{\mathbf{Y}} = \frac{1}{N}\sum_{i=1}^{N}\left(a_i \hat{\mathbf{y}}_i\right). \tag{4}$$

3.3 CONTEXTUAL ATTENTION

As Early et al. (2024) showed that these pooling operations often produced sparse explanations which occasionally did not cover the entire discriminatory regions, we propose injecting a contextual prior into our calculation of attention, following ideas similar to Fourkioti et al. (2024). For attention-based pooling methods, `Attention`, `Additive`, and `Conjunctive`, attention weights for each point are calculated as:

$$a_i \in [0,1] = f_{attn}(\mathbf{z}_i), \tag{5}$$

where f_{attn} is an MLP and $\mathbf{z}_i$ is a feature vector for each point $\mathbf{p}_i$. We propose updating these attention weights according to the attention weights of the nearest neighbours of each point i, such that:

$$a_i^{\text{new}} \in [0,1] = \frac{1}{k}\sum_{j \in \mathcal{N}(\mathbf{p}_i)} a_j, \tag{6}$$

Table 1: Interpretability results in terms of AOPCR and NDCG@n (AOPCR/NDCG@n) on IntrA. The best results are given for each method in **bold**.

	PointNet	DGCNN	CurveNet	PointNeXt	Transformer
PSM	0.579/0.243	0.916/0.248	1.371/0.218	0.092/0.272	6.518/0.320
CLAIM	0.967/0.187	**6.033**/0.480	1.363/0.252	0.226/0.294	14.023/0.593
`Add.`	0.792/**0.254**	4.486/**0.482**	0.615/**0.266**	1.259/0.300	**18.162**/**0.613**
`Att.`	0.005/0.222	−0.031/0.223	1.520/0.260	0.044/0.235	14.541/0.539
`Conj.`	0.741/0.208	4.828/0.467	**2.660**/0.207	1.531/**0.310**	16.305/0.610
`Inst.`	**0.973**/0.225	5.212/0.462	1.709/0.236	**2.160**/0.285	16.166/0.587

where $\mathcal{N}(\mathbf{p}_i)$ represents the set of points in the neighbourhood of $\mathbf{p}_i$. This update mechanism smooths the attention weights by incorporating the information from the local neighbourhood, thus addressing the sparsity of the original attention mechanism and providing a more context-aware attention distribution across the point cloud.

3.4 Interpretability

Interpretations were derived through MIL pooling. The `Instance` pooling strategy classifies each point individually before pooling, yielding point-level predictions: $\{\hat{\mathbf{y}}_i | i = 1, \ldots, N\}$. `Additive` and `Conjunctive` also make point-level predictions; however, the interpretations are scaled by attention weights: $\{a_i\hat{\mathbf{y}}_i | i = 1, \ldots, N\}$. For each of these pooling algorithms, we applied a softmax operation over the class dimension and took the index of the class for which we wished to obtain interpretations, so that we obtained a scalar for each point in the point cloud. For the `Attention` pooling strategy, we used the attention weights: $\mathbf{a} \in \mathbb{R}^{1\times N} = \{a_i | i = 1, \ldots, N\}$, which were interpreted as a measure of general importance for each point in the point cloud and were not class-specific.

4 Experiments

We compared the interpretability of PointMIL with other *locally* interpretable point cloud classification methods including class attentive interpretable mapping (CLAIM; Huang et al. (2020)), and point cloud saliency maps (PSM; Zheng et al. (2019)). Similarly to class activation maps (CAM; Zhou et al. (2016)), CLAIM uses global average pooling (GAP) after point-level feature extractors (the original paper focused on PointNet) and projects the weights of the classifier after GAP on the features of each point to obtain interpretations for each point. PSM assigns scores to each point based on its contribution to the classification loss. This is done by shifting the points towards the centroid of the point cloud and then calculating the gradient of the loss with respect to each point

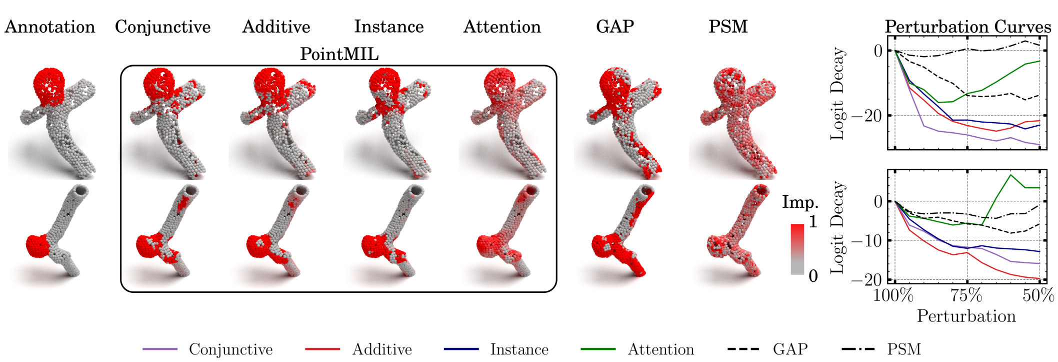

Figure 2: PointMIL, CLAIM and PSM interpretability visualisations and corresponding perturbation curves using the Transformer backbonfor example cells from the IntrA dataset.

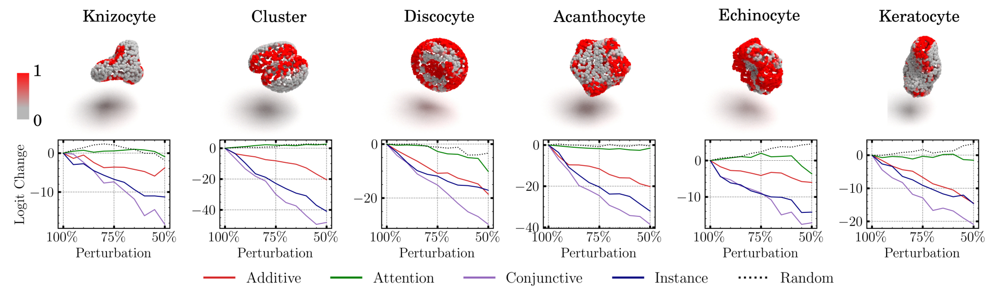

Figure 3: Interpretability visualisation (top row) and corresponding perturbation (bottom row) curves for different RBC shapes.

in spherical coordinates. We then compared POINTMIL to several other point-based architectures in terms of classification performance and assessed how the MIL pooling affected the results of the original backbones in segmentation tasks.

4.1 EVALUATION METRICS

We used the area over the perturbation curve to random (AOPCR; Samek et al. (2017)) and normalised discounted cumulative gain at n (NDCG@n) to quantitatively evaluate interpretability (Early et al., 2022; 2024). Please see Appendix B for more details. For classification, we used the overall accuracy (oACC), mean class accuracy per class (mACC), and the F1 score. For segmentation, we used the average class intersection of union (IoU) and the instance IoU.

4.2 DATASETS

We evaluated POINTMIL on several open source datasets, including two real-world datasets of 3D cell shapes (IntrA (Yang et al., 2020) and 3D red blood cell (RBC) dataset (Simionato et al., 2021)) and two of everyday objects (ModelNet40 (Wu et al., 2015) and ShapeNetPart (Yi et al., 2016)). See Appendix C for more details.

5 RESULTS

5.1 INTERPRETABILITY

Table 1 shows the interpretability results on the IntrA dataset for PointNet, DGCNN, CurveNet, PointNeXt and the Transformer backbone. POINTMIL provided better interpretability performance than both PSM and CLAIM, overall. Across backbones, POINTMIL had the highest AOPCR and NDCG@n. The only exception was CLAIM that had a higher AOPCR for the DGCNN backbone. Among the interpretability methods, the Transformer produced the highest AOPCR and NDCG@n results. This could be due to the attention mechanisms within the Transformer block that already enabled the model to focus on informative points, which is further exacerbated by the MIL pooling. Among all backbones, PointNet performed the worst, suggesting that PointNet is not adequate in

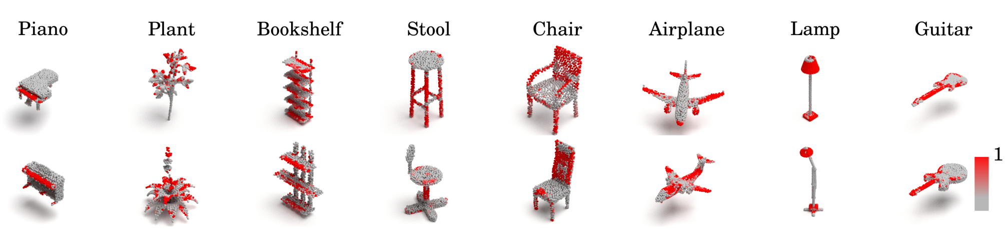

Figure 4: Interpretability outputs of PointMIL for different shape classes from ModelNet40

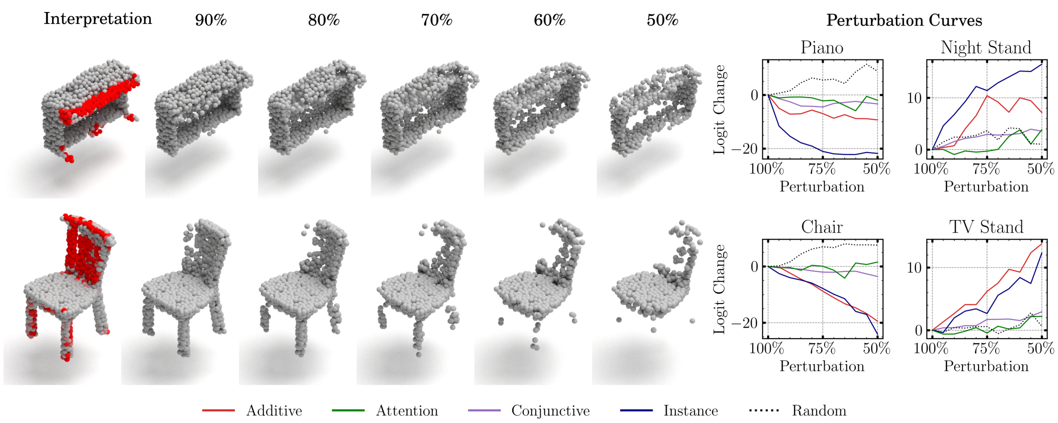

Figure 5: Interpretability outputs and perturbation curves of POINTMIL with the Transformer backbone for different shape classes from ModelNet40

capturing discriminative morphological cues. For PointNeXt, although the PointMIL versions outperformed PSM and CLAIM, the lower values when compared to DGCNN and the Transformer could be attributed to the concatenation of local with global features before the MIL pooling.

Visualisations of the interpretability for each pooling method on the annotated *Aneurysm* class using the Transformer backbone are shown in Figure 2. The red points indicate areas deemed significant by the model for that specific class. *Aneurysm's* are presented by the abnormal bulging or ballooning of blood vessels. The first column in Figure 2 shows local annotations of *Aneurysms*, with each other column presenting interpretations for the *Aneurysm* class using the different methods. The last columns show the perturbation curves. These show the decay in the logit of the predicted class after removing the most important points. A larger decay suggests that those points are indeed discriminative for the class. POINTMIL is clearly able to localise on informative regions better than other methods as seen by the visualisation as well as a larger decay in logits shown by the perturbation curve.

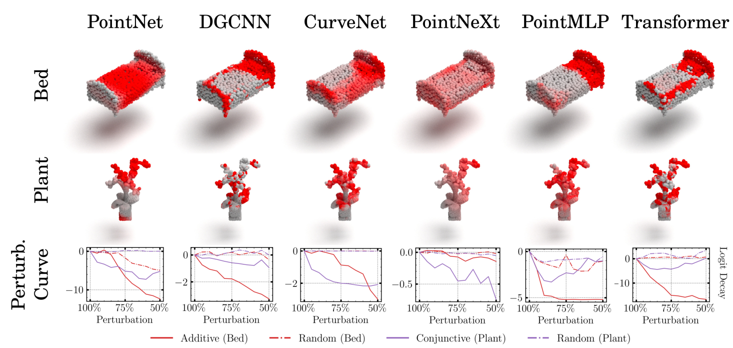

Figure 6: Interpretability of POINTMIL with different backbones on an example *Bed* (top row) and *Plant* (midle row) from ModelNet40. Perturbation curves are shown in the bottom row.

Among all MIL pooling methods, `Additive` and `Conjunctive` performed best on the IntrA dataset. This superior performance of `Additive` and `Conjunctive` pooling can be attributed to their ability to better aggregate point-level importance scores. Additive pooling scales point features with their importance weights, preserving detailed information while focusing on relevant points before being passed into a point-level classifier. Conjunctive pooling further enhances this by independently computing attention weights and class-specific contributions, explicitly aligning the model's focus with the predicted class. In contrast, `Instance` pooling lacks this importance weighting, and `Attention` pooling does not offer class-specific explanations and rather provides a general measure of importance across classes, which limits their interpretability.

We also present local interpretations for other datasets lacking ground truth annotations. Figure 3 illustrates the visual interpretations of POINTMIL with the Transformer backone for six of the nine classes of RBC with their corresponding perturbation curves. This demonstrates that PointMIL successfully localises on biologically relevant structural areas. For example, *Discocytes* are characterised by their biconcave shapes, with interpretations for this class focussing on regions identified around the central concavity. In the case of *Acanthocytes*, which exhibit several spicules of varying sizes that project from their surfaces at irregular intervals, PointMIL similarly focused on these projections for identifying this class. For *Knizocytes*, which have a triangular morphology, the model highlighted the areas where the lobes converge. Additionally, PointMIL pinpointed the spiky projections of *Echinocytes* and *Keratocytes*, as well as the interaction zones where two cells meet in *Cell Clusters*.

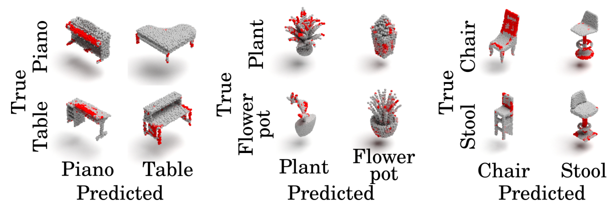

Figure 7: Interpretability visualisations of incorrect classifications from PointMIL with Transformer backbone on ModelNet40.

PointMIL is a versatile tool that is not limited to specific domains, making it suitable for a wide range of 3D shape classification tasks. Figure 4 presents the visual interpretations of PointMIL applied to the ModelNet40 dataset, showcasing a subset of classes. For instance, when classifying a *Piano*, the model focused primarily on the keys, while the it emphasised on the branches and foliage of a *Plant*. The *Bookshelf* displayed red points along the shelves. Similarly, for the *Chair*, crucial features included the seat and legs, while the wings and fuselage were highlighted for *Airplane*. More examples are given in Appendix E.

Figure 5 shows the effect of removing the top 10% to 50% of important points on a *Piano* and *Chair* on the logits of those classes. The perturbation curves illustrate that when the points identified as most informative for classifying a *Piano* are removed, PointMIL misclassifies the object as a Night Stand. Similarly, when the points identified as the most informative for classifying a *Chair* are removed, PointMIL misclassifies the object as a TV stand. These interpretations reveal how PointMIL effectively identified and localised relevant features across various object categories, enhancing our understanding of the model's decision-making process. Figure 6 presents the interpretability results for different backbones when classifying a *Bed* with `Additive` pooling (top row) and a *Plant* with `Conjunctive` pooling (middle row) from the ModelNet40 dataset. The perturbation curves are shown in the bottom row. Interestingly, DGCNN, CurveNet, PointMLP, and Transformer backbones consistently highlight similar regions of importance on the *Bed*, particularly focusing on the frame and headboard of the bed, which are key features distinguishing it from other objects. All backbones focussed on the leaves in the *Plant* as opposed to the pot. This consistency across backbones demonstrates the robustness of PointMIL in identifying informative regions. Additionally, the agreement among backbones suggests that PointMIL effectively leverages the feature representations generated by each model, ensuring the interpretability results are meaningful and aligned with the task. Finally, we demonstrated how PointMIL could be used to assess where the model went wrong. For example, Figure 7 shows example confusion plots in which the attention of the predicted class is shown in red. Interestingly, for classifying plants, the model only focused on the plant, although when classifying flower pots, the model focused on both the flower and the pot.

5.2 Classification

Interpretability should promote classification accuracy and not hinder it. To showcase this, we performed classification on three separate datasets, two 3D biological cell-shape datasets, IntrA , and RBC, and the 3D shape classification benchmark ModelNet40. The results are shown in Table 2. PointMIL outperformed all methods on IntrA and RBC in terms of mACC and F1 score by a considerable margin of at least 4.5% and 3.3% respectively. PointMIL achieved SOTA on IntrA with an mACC of 97.3% and an F1 of 97.5% using `Conjunctive` pooling with the Transformer backbone. Importantly, PointMIL increased the performance of all backbones on all datasets by up to 11.3% in terms of mACC on RBC (shown in violet in Table 2). While PointMIL was outperformed by recent SOTA methods like PointMLP (Ma et al., 2022), the original CurveNet (Xiang

Table 2: Classification results on IntrA, RBC, and ModelNet40. All results are shown without voting strategy on 1024 points. The highest results are shown in **bold**. Differences between backbones and POINTMIL are shown in violet. Adapted architectures without farthest point sampling results are shown with a †.

	IntrA		RBC		ModelNet40	
Method	mACC(↑)	F1(↑)	mACC(↑)	F1(↑)	mACC (↑)	oACC(↑)
PointNet(Qi et al., 2017a)	81.8	82.4	67.7	67.1	86.2	89.2
PointNet++(Qi et al., 2017b)	92.7	94.2	86.2	87.1	-	91.9
PointConv(Wu et al., 2019)	83.0	82.1	68.1	67.9	-	92.5
DGCNNWang et al. (2019b)	90.6	91.8	84.8	85.1	90.2	92.9
PCT(Guo et al., 2021)	69.2	68.9	68.7	69.2	-	93.2
CurveNet(Xiang et al., 2021)	88.3	89.8	88.3	87.8	-	93.8
CurveNet†	87.8	87.8	85.8	85.7	90.6	93.4
PointMLP(Ma et al., 2022)	88.4	88.8	91.8	**92.2**	**91.3**	**94.1**
PointMLPElite	-	-	-	-	90.9	93.6
PointMLPElite†	-	-	-	-	90.1	92.6
PointNeXt(Qian et al., 2022)	91.8	94.7	86.1	87.1	90.8	93.2
3DMedPT(Yu et al., 2021)	92.2	93.3	81.3	83.2	-	93.4
POINTMIL(PointNet)	82.0 +0.2	82.4 +0.0	69.0 +1.3	69.1 +2.0	87.1 +0.9	90.7 +1.5
POINTMIL(DGCNN)	95.2 +3.2	94.6 +2.8	92.4 +7.6	92.4 +7.3	90.8 +0.6	93.1 +0.2
POINTMIL(CurveNet†)	91.3 +3.5	89.9 +2.1	91.2 +5.4	90.5 +4.8	91.0 +0.4	93.5 +0.1
POINTMIL(PointMLPElite†)	-	-	-	-	90.5 +0.4	93.5 +0.9
POINTMIL(PointNeXt)	94.6 +2.8	96.2 +1.5	87.6 +1.5	88.2 +0.4	90.5 −0.3	93.3 +0.1
POINTMIL(Trans.)	**97.3** +5.1	**97.5** +4.2	**92.6** +11.3	**92.2** +9.0	89.0	92.7 −0.7

et al., 2021) and PCT (Guo et al., 2021) on Modelnet40, POINTMIL outperformed these methods by considerable margins on IntrA and RBC. POINTMIL offered interpretability without harming and often improving classification performance.

5.3 ABLATION STUDIES

We evaluated the effect of including contextual attention in our attention-based pooling mechanisms: `Additive`, `Attention`, and `Conjunctive` and the impact of varying the value of k (Figure 8). A value of $k = 0$ represented no contextual attention. Including contextual attention consistently offered advantages across all pooling methods and metrics compared to not using it. In terms of F1 and mACC contextual attention led to improved performance, particularly with the `Conjunctive` and `Attention` mechanisms, which consistently outperformed the `Additive` method as k increased. All pooling methods produced F1 and mACC scores of $> 97\%$ after including contextual attention. For AOPCR, contextual attention was found to be most beneficial when using a value of $k = 12$. Lastly, considering NDCG@n, increasing k provided the most benefit to `Attention` pooling, while offering slight improvements to `Additive` and `Conjunctive`. `Additive` and `Conjunctive pooling` outperformed `Attention` pooling across interpretability metrics, whether or not contextual attention was used. Although contextual pooling improved classification and interpretation methods, there is a trade-off in computation since the time complexity for

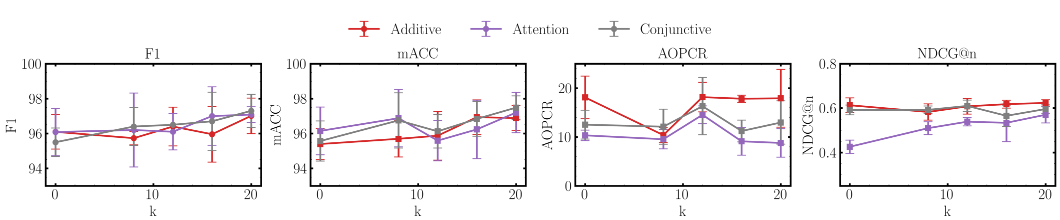

Figure 8: Ablation studies on the value of k in our contextual attention on F1, mACC, AOPCR, and NDCG@n using the transformer backbone.

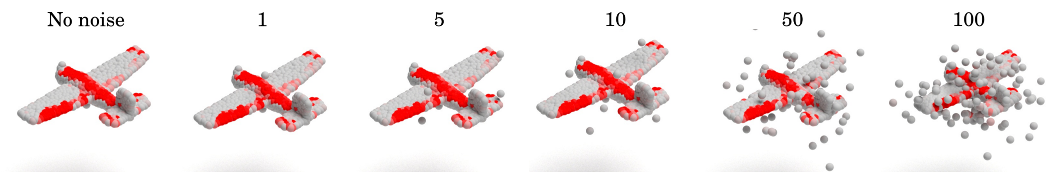

Figure 9: Interpretability visualisations of POINTMIL on a *Airlane* from ModelNet40 after adding a number (shown on the heading) of noisy points. POINTMIL is able to still focus on salient shape motifs ignoring noise.

k-NN graph search is $O(N^2)$ for the N number of points. The graph construction time complexity is also $O(Nk)$, therefore, as k increases, this process takes longer. We additionally demonstrate POINTMIL's robustness to noise. Figure 9 shows how, even when noisy points are added to objects, POINTMIL is still able to focus on salient 3D shape motifs. Further analysis is shown in Appendix F

5.4 SEGMENTATION

We evaluated POINTMIL for part segmentation on IntrA and ShapeNetPart using three of the five backbones. For IntrA, only the *Aneurysm* class contains annotations, therefore, we only reported metrics on this class. We followed the same settings as from Qi et al. (2017a) for segmentation on ShapeNetPart. The class-specific point-level interpretations were used as segmentation predictions. We assessed the `Conjunctive` and `Additive` MIL pooling as `Instance` was the equivalent to the original model's segmentation algorithms and `Attention` does not produce class-specific point-level classification as interpretations. Interestingly, the segmentation results did not deteriorate and sometimes improved when using POINTMIL on both datasets. The only exception was 3DMedPT on ShapeNetPart, where the original 3DMedPT outperformed POINTMIL with the transformer backbone by a relatively larger margin.

Table 3: Segmentation results on IntrA and ShapeNetPart in terms of Class (Cls.) and Instance (Inst.) mIoU. The highest metrics are shown in **bold**.

	IntrA	ShapeNetPart	
Method	IoU(↑)	Cls. IoU(↑)	Inst. IoU(↑)
PointNet	72.2	81.7	84.2
DGCNN	76.4	83.6	85.2
3DMedPT	82.4	**84.3**	-
POINTMIL(PointNet)	72.3	81.5	84.0
POINTMIL(DGCNN)	79.7	84.2	**85.6**
POINTMIL(Trans)	**84.0**	82.0	82.1

6 CONCLUSION

In this work, we introduced POINTMIL, the first framework to apply MIL to point cloud classification, providing fine-grained point-specific interpretability without *post-hoc* techniques. We also introduced a contextual attention mechanism to adapt attention-based MIL to point clouds, accounting for the spatial and structural relationships inherent in 3D data. Using MIL, our approach improved both interpretability and classification performance on multiple backbones and datasets. POINTMIL achieved SOTA F1 and mACC by a significant margin. Future work could extend POINTMIL to consider using segmentation versions of other point-based models as backbones, as they provide point-specific features. Furthermore, analysis on more datasets that include point-specific ground-truth interpretation would help to better evaluate interpretability. The choice of pooling method should be guided by the specific requirements of the task and dataset characteristics. For tasks prioritising interpretability, `Conjunctive` pooling with contextual attention is recommended due to its class-specific focus. For applications prioritising simplicity, Instance pooling offers computational efficiency. An exploration of MIL pooling techniques specific to point cloud data could also enhance this work further. In conclusion, POINTMIL is a novel approach that effectively improved classification performance while providing inherent local interpretability, making it a valuable tool for 3D point cloud analysis in real-world applications.

Reproducibility Statement

The code for this work was implemented in Python 3.10, with PyTorch and Lightning as the main machine learning libraries. The anonymous code is available at `https://anonymous.4open.science/r/PointMIL_ICLR-98B2/`. Model training was performed using an NVIDIA Tesla V100 GPU with 32GB of VRAM and CUDA v12.0 to enable GPU support.

References

Perpetual Hope Akwensi, Ruisheng Wang, and Bo Guo. Preformer: A memory-efficient transformer for point cloud semantic segmentation. *International Journal of Applied Earth Observation and Geoinformation*, 128:103730, 2024. ISSN 1569-8432. doi: https://doi.org/10.1016/j.jag.2024.103730. URL `https://www.sciencedirect.com/science/article/pii/S1569843224000840`.

Saad Ali and Mubarak Shah. Human action recognition in videos using kinematic features and multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(2):288–303, feb 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2008.284. URL `https://doi.org/10.1109/TPAMI.2008.284`.

Nicholas I. Arnold, Plamen Angelov, and Peter M. Atkinson. An improved explainable point cloud classifier (xpcc). *IEEE Transactions on Artificial Intelligence*, 4(1):71–80, 2023. doi: 10.1109/TAI.2022.3150647.

Le Cheng, Cuijuan An, Yu Gao, Yinfeng Gao, and Dawei Ding. Point mlp-former: Combining local and global receptive fields in point cloud classification. In *2022 China Automation Congress (CAC)*, pp. 4895–4900, 2022. doi: 10.1109/CAC57257.2022.10055719.

Jaesung Choe, Chunghyun Park, Francois Rameau, Jaesik Park, and In So Kweon. Pointmixer: Mlp-mixer for point cloud understanding. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 620–640, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19812-0.

Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1):31–71, 1997. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(96)00034-3. URL `https://www.sciencedirect.com/science/article/pii/S0004370296000343`.

Joseph Early, Christine Evers, and SArvapali Ramchurn. Model agnostic interpretability for multiple instance learning. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=KSSfF5lMIAg`.

Joseph Early, Gavin Cheung, Kurt Cutajar, Hanting Xie, Jas Kandola, and Niall Twomey. Inherently interpretable time series classification via multiple instance learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=xriGRsoAza`.

Feng-Lei Fan, Jinjun Xiong, Mengzhou Li, and Ge Wang. On interpretability of artificial neural networks: A survey. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(6):741–760, 2021. doi: 10.1109/TRPMS.2021.3066428.

Tuo Feng, Ruijie Quan, Xiaohan Wang, Wenguan Wang, and Yi Yang. Interpretable3d: An ad-hoc interpretable classifier for 3d point clouds. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(2):1761–1769, Mar. 2024. doi: 10.1609/aaai.v38i2.27944. URL `https://ojs.aaai.org/index.php/AAAI/article/view/27944`.

Olga Fourkioti, Matt De Vries, and Chris Bakal. CAMIL: Context-aware multiple instance learning for cancer detection and subtyping in whole slide images. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=rzBskAEmoc`.

Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, Jun 2021. ISSN 2096-0662. doi: 10.1007/s41095-021-0229-5. URL https://doi.org/10.1007/s41095-021-0229-5.

Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

Brian Hu, Paul Tunison, Brandon RichardWebster, and Anthony Hoogs. Xaitk-saliency: An open source explainable ai toolkit for saliency. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(13):15760–15766, Jul. 2024. doi: 10.1609/aaai.v37i13.26871. URL https://ojs.aaai.org/index.php/AAAI/article/view/26871.

Shikun Huang, Binbin Zhang, Wen Shen, and Zhihua Wei. A claim approach to understanding the pointnet. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, ACAI '19, pp. 97–103, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450372619. doi: 10.1145/3377713.3377740. URL https://doi.org/10.1145/3377713.3377740.

Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2127–2136. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/ilse18a.html.

Syed Ashar Javed, Dinkar Juyal, Harshith Padigela, Amaro Taylor-Weiner, Limin Yu, and aaditya prakash. Additive MIL: Intrinsically interpretable multiple instance learning for pathology. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=5dHQyEcYDgA.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. The dangers of post-hoc interpretability: unjustified counterfactual explanations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, pp. 2801–2807. AAAI Press, 2019. ISBN 9780999241141.

Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: a neural network that explains its predictions. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018a. ISBN 978-1-57735-800-8.

Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018b. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/f5f8590cd58a54e94377e6ae2eded4d9-Paper.pdf.

Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Richard J Chen, Matteo Barbieri, and Faisal Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature Biomedical Engineering*, 5(6):555–570, 2021.

Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=3Pbra-_u76D.

Dipanjyoti Paul, Arpita Chowdhury, Xinqi Xiong, Feng-Ju Chang, David Edward Carlyn, Samuel Stevens, Kaiya L Provost, Anuj Karpatne, Bryan Carstens, Daniel Rubenstein, Charles Stewart, Tanya Berger-Wolf, Yu Su, and Wei-Lun Chao. A simple interpretable transformer for fine-grained image classification and analysis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=bkdWThqE6q.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017b. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf.

Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 23192–23204. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9318763d049edf9a1f2779b2a59911d3-Paper-Conference.pdf.

Marco Tulio Ribeiro, Sameer Singh, and Carlos 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*, KDD '16, pp. 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, May 2019. ISSN 2522-5839. doi: 10.1038/s42256-019-0048-x. URL https://doi.org/10.1038/s42256-019-0048-x.

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *CoRR*, abs/2103.11251, 2021. URL https://arxiv.org/abs/2103.11251.

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistic Surveys*, 16:1–85, 2022.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 2017.

Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, and Yongbing Zhang. TransMIL: Transformer based correlated multiple instance learning for whole slide image classification. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=LKUfuWxajHc.

Greta Simionato, Konrad Hinkelmann, Revaz Chachanidze, Paola Bianchi, Elisa Fermo, Richard van Wijk, Marc Leonetti, Christian Wagner, Lars Kaestner, and Stephan Quint. Red blood cell phenotyping from 3d confocal images using artificial neural networks. *PLOS Computational Biology*, 17(5):1–17, 05 2021. doi: 10.1371/journal.pcbi.1008934. URL https://doi.org/10.1371/journal.pcbi.1008934.

Andrew H. Song, Mane Williams, Drew F.K. Williamson, Sarah S.L. Chow, Guillaume Jaume, Gan Gao, Andrew Zhang, Bowen Chen, Alexander S. Baras, Robert Serafin, Richard Colling, Michelle R. Downes, Xavier Farré, Peter Humphrey, Clare Verrill, Lawrence D. True, Anil V. Parwani, Jonathan T.C. Liu, and Faisal Mahmood. Analysis of 3d pathology samples using weakly supervised ai. *Cell*, 187(10):2502–2520.e17, May 2024. ISSN 0092-8674. doi: 10.1016/j.cell.2024.03.035. URL https://doi.org/10.1016/j.cell.2024.03.035.

Saeid Asgari Taghanaki, Kaveh Hassani, Pradeep Kumar Jayaraman, Amir Hosein Khasahmadi, and Tonya Custis. Pointmask: Towards interpretable and bias-resilient point cloud processing. *arXiv preprint arXiv:2007.04525*, 2020.

Hanxiao Tan and Helena Kotthaus. Surrogate model-based explainability methods for point cloud nns. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2927–2936, 2022. doi: 10.1109/WACV51458.2022.00298.

Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6410–6419, 2019. doi: 10.1109/ICCV.2019.00651.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

Matheus P. Viana, Jianxu Chen, Theo A. Knijnenburg, Ritvik Vasan, Calysta Yan, Joy E. Arakaki, Matte Bailey, Ben Berry, Antoine Borensztejn, Eva M. Brown, Sara Carlson, Julie A. Cass, Basudev Chaudhuri, Kimberly R. Cordes Metzler, Mackenzie E. Coston, Zach J. Crabtree, Steve Davidson, Colette M. DeLizo, Shailja Dhaka, Stephanie Q. Dinh, Thao P. Do, Justin Domingus, Rory M. Donovan-Maiye, Alexandra J. Ferrante, Tyler J. Foster, Christopher L. Frick, Griffin Fujioka, Margaret A. Fuqua, Jamie L. Gehring, Kaytlyn A. Gerbin, Tanya Grancharova, Benjamin W. Gregor, Lisa J. Harrylock, Amanda Haupt, Melissa C. Hendershott, Caroline Hookway, Alan R. Horwitz, H. Christopher Hughes, Eric J. Isaac, Gregory R. Johnson, Brian Kim, Andrew N. Leonard, Winnie W. Leung, Jordan J. Lucas, Susan A. Ludmann, Blair M. Lyons, Haseeb Malik, Ryan McGregor, Gabe E. Medrash, Sean L. Meharry, Kevin Mitcham, Irina A. Mueller, Timothy L. Murphy-Stevens, Aditya Nath, Angelique M. Nelson, Sandra A. Oluoch, Luana Paleologu, T. Alexander Popiel, Megan M. Riel-Mehan, Brock Roberts, Lisa M. Schaefbauer, Magdalena Schwarzl, Jamie Sherman, Sylvain Slaton, M. Filip Sluzewski, Jacqueline E. Smith, Youngmee Sul, Madison J. Swain-Bowden, W. Joyce Tang, Derek J. Thirstrup, Daniel M. Toloudis, Andrew P. Tucker, Veronica Valencia, Winfried Wiegraebe, Thushara Wijeratna, Ruian Yang, Rebecca J. Zaunbrecher, Ramon Lorenzo D. Labitigan, Adrian L. Sanborn, Graham T. Johnson, Ruwanthi N. Gunawardane, Nathalie Gaudreault, Julie A. Theriot, and Susanne M. Rafelski. Integrated intracellular organization and its variations in human iPS cells. *Nature*, 613(7943):345–354, January 2023. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-022-05563-7. URL `https://www.nature.com/articles/s41586-022-05563-7`.

Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10288–10297, 2019a. doi: 10.1109/CVPR.2019.01054.

Wenguan Wang, Cheng Han, Tianfei Zhou, and Dongfang Liu. Visual recognition with deep nearest centroids. In *International Conference on Learning Representations (ICLR)*, 2023.

Xinggang Wang, Yongluan Yan, Peng Tang, Xiang Bai, and Wenyu Liu. Revisiting multiple instance neural networks. *Pattern Recognition*, 74:15–24, 2018. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2017.08.026. URL `https://www.sciencedirect.com/science/article/pii/S0031320317303382`.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019b.

Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9613–9622, 2019. doi: 10.1109/CVPR.2019.00985.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 915–924, October 2021.

Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5660–5669, 2020. doi: 10.1109/CVPR42600.2020.00570.

Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Xi Yang, Ding Xia, Taichi Kin, and Takeo Igarashi. Intra: 3d intracranial aneurysm dataset for deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Trans. Graph.*, 35(6), dec 2016. ISSN 0730-0301. doi: 10.1145/2980179.2980238. URL `https://doi.org/10.1145/2980179.2980238`.

Jianhui Yu, Chaoyi Zhang, Heng Wang, Dingxin Zhang, Yang Song, Tiange Xiang, Dongnan Liu, and Weidong Cai. 3d medical point transformer: Introducing convolution to attention networks for medical point cloud analysis, 2021. URL `https://arxiv.org/abs/2112.04863`.

Binbin Zhang, Shikun Huang, Wen Shen, and Zhihua Wei. Explaining the pointnet: What has been learned inside the pointnet? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

Gege Zhang, Qinghua Ma, Licheng Jiao, Fang Liu, and Qigong Sun. Attan: Attention adversarial networks for 3d point cloud semantic segmentation. In Christian Bessiere (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 789–796. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/110. URL `https://doi.org/10.24963/ijcai.2020/110`. Main track.

Qinglong Zhang, Lu Rao, and Yubin Yang. A novel visual interpretability for deep neural networks by optimizing activation maps with perturbation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):3377–3384, May 2021. doi: 10.1609/aaai.v35i4.16450. URL `https://ojs.aaai.org/index.php/AAAI/article/view/16450`.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16239–16248, 2021. doi: 10.1109/ICCV48922.2021.01595.

T. Zheng, C. Chen, J. Yuan, B. Li, and K. Ren. Pointcloud saliency maps. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1598–1606, Los Alamitos, CA, USA, nov 2019. IEEE Computer Society. doi: 10.1109/ICCV.2019.00168. URL `https://doi.ieeecomputersociety.org/10.1109/ICCV.2019.00168`.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization . In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, Los Alamitos, CA, USA, June 2016. IEEE Computer Society. doi: 10.1109/CVPR.2016.319. URL `https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.319`.

A MODEL DETAILS

A.1 TRANSFORMER BLOCK FEATURE EXTRACTOR

A.1.1 GROUP FEATURES THROUGH k-NEAREST NEIGHBOURS:

Formally, we constructed a k-NN graph on $\mathbf{P}$ with the graph including a self-loop to point-level features:

$$\begin{gathered}\mathcal{N}(\mathbf{p}_i) = \mathrm{KNN}(\mathbf{P}, ||\mathbf{p}_i - \mathbf{p}_j||_2^2), \mathbf{p}_i, \mathbf{p}_j \in \mathbf{P}, \\ \mathbf{f}_i' = [(\mathbf{f}_j - \mathbf{f}_i), \mathbf{f}_i]_{j \in \mathcal{N}(\mathbf{p}_i)} \in \mathbb{R}^{k \times 2d_{in}},\end{gathered} \tag{7}$$

where KNN$(\cdot)$ is the k-NN function, $[\cdot,\cdot]$ is concatenation, k is the hyperparameter of the k-NN graph, $\mathcal{N}(\mathbf{p}_i)$ is the set of neighbours of $\mathbf{p}_i$, and $\mathbf{f}_i'$ is the point feature augmented with local contextual information.

A.1.2 LEARNED RELATIVE POSITIONAL ENCODING:

To encode spatial configurations per point-cloud neighbourhood we incorporated positional embeddings, $\mathbf{h}_i$ such that:

$$\mathbf{h}_i \in \mathbb{R}^{k \times d_h} = \phi_{pos}([\mathbf{p}_i - \mathbf{p}_j]_{j \in \mathcal{N}(\mathbf{p}_i)}), \tag{8}$$

where ϕ_{pos} is an MLP and d_h is the output channel dimension of ϕ_{pos}. The features were then further augmented with this positional encoding to give:

$$\mathbf{f}_i'' = [\mathbf{f}_i', \mathbf{h}_i]. \tag{9}$$

Thus, we obtained a new feature set $\mathbf{F}'' \in \mathbb{R}^{N \times k \times (2d_{in} + d_h)} = \{\mathbf{f}_i''\}_{i=1}^N$. This is then passed

A.1.3 ATTENTION ON THE AUGMENTED FEATURES:

The resulting features, $\mathbf{F}''$, were then fed into a transformer with EdgeConv as the query operation. Recall that EdgeConv (Wang et al., 2019b) computes graph features for each point using the equation:

$$\mathbf{e}_i \in \mathbb{R}^{d_e} = \max_{j \in \mathcal{N}(\mathbf{p}_i)}(\phi_{edge}(\mathbf{p}_i, \mathbf{p_j} - \mathbf{p}_i)), \tag{10}$$

where ϕ_{edge} is an MLP with output dimension d_e. The $\mathbf{F}''$ were then transformed using attention Vaswani et al. (2017):

$$\begin{aligned}\mathbf{Q} \in \mathbb{R}^{N \times d_k} &= \mathrm{EdgeConv}\left(\mathbf{F}''\right) W_q \\ \mathbf{K} \in \mathbb{R}^{(N \times k) \times d_k} &= \mathrm{Flatten}\left(\mathbf{F}''\right) W_k \\ \mathbf{V} \in \mathbb{R}^{(N \times k) \times d_v} &= \mathrm{Flatten}\left(\mathbf{F}''\right) W_v,\end{aligned} \tag{11}$$

where $\mathbf{W}_q \in \mathbb{R}^{d_e \times d_k}$, $\mathbf{W}_k \in \mathbb{R}^{(2d_{in}+d_h) \times d_k}$ and $\mathbf{W}_v \in \mathbf{R}^{(2d_{in}+d_h) \times d_v}$ are learnable weight matrices. Our final point-level output features from the transformer block was then given by:

$$\mathbf{z}_i \in \mathbb{R}^{N \times d_v} = \mathbf{q}_i(\mathrm{softmax}(\mathbf{k}_i)^\top \mathbf{v}_i). \tag{12}$$

For all experiments, we used two transformer layers such that the final feature vector for each point was of size 256.

A.2 CURVENET ADAPTATION

CurveNet uses sampling and grouping. Our only adaptation to CurveNet was use the same number of input points as input into the farthest point sampling algorithm. We kept everything else the same as the original paper. We replaced the original adaptive max, adaptive mean pooling, and the classification head with MIL pooling. The final feature vector for each point was of size 1024.

A.3 POINTNEXT ADAPTATION

PointNeXt uses sampling and grouping. To adapt PointNeXt to POINTMIL, we did not modifying the architecture itself. Instead, we concatenated the point-level features from the first layer of the encoder with global features from the final layer of the encoder. This resulted in a final feature vector for each point of size 544.

A.4 MIL POOLING

A.4.1 CLASSIFICATION HEAD

We tested several different classification heads for each dataset. The final classification heads for each dataset are summarised in Table 4.

Table 4: Classification head architecture

Type	Layer	Input	Output
IntrA/RBC	Linear	$b \times 1 \times N \times d$ (feature dimension)	$b \times 1 \times N \times c$
MN40	Linear + ReLU	$b \times 1 \times N \times d$	$b \times 1 \times N \times d//2$
	Linear + ReLU	$b \times 1 \times N \times d//2$	$b \times 1 \times N \times d//4$
	Linear	$b \times 1 \times N \times d//4$	$b \times 1 \times N \times c$ (Point Pred)

A.4.2 ATTENTION HEAD

Table 5: Attention head architecture

Process	Layer	Input	Output
Attention	Linear + tanh	$b \times 1 \times N \times d$	$b \times 1 \times N \times 8$
	Linear + sigmoid	$b \times 1 \times N \times 8$	$b \times 1 \times N \times 1$ (Attn. Scores)

We used the same attention head for all attention-based pooling. This is summarised in Table 5.

B INTERPRETABILITY METRICS

AOPCR does not require instance labels, whereas NDCG@n does. AOPCR works by removing the most important instances in sequence and observing the impact on prediction accuracy. The faster the prediction declines, the better the ordering, as the most influential instances are removed earlier. When point clouds are annotated, NDCG@n evaluates how closely the model's interpretability ranking matches the true order. It rewards rankings that prioritise relevant instances, with higher scores indicating better alignment and interpretability.

C DATASETS

C.1 INTRA

IntrA is an open source dataset of 3D intracranial aneurysm (Yang et al., 2020). The task is to classify blood vessels as healthy and aneurysm. There is a total of 1909 blood vessel segments, including 1694 healthy vessel segments and 215 aneurysm segments for diagnosis. 116 of the aneurysm segments are expertly annotated. We use IntrA to evaluate interpretability, classification, and segmentation.

C.2 RED BLOOD CELL

We used another dataset of 3D red blood cells (RBC; Simionato et al. (2021)) for classification. This dataset includes 825 3D red blood cells imaged using confocal microscopy grouped into 9 expertly annotated shape classes. Blood samples were collected from healthy donors and patients using finger prick blood sampling. For inducing RBC shape transitions, blood from 5 healthy donors was treated with NaCl solutions of varying concentrations to create different RBC shapes. Specific shape classes were expertly annotated according to particular motifs. Thus, similar to IntrA, RBC was suitable for evaluating interpretability by the ability to identify these motifs. Segmentation masks are publicly available. We converted the segmentation to mesh objects using marching cubes with Laplacian smoothing, and then sampled points from the vertices of these mesh objects.

C.3 ModelNet40

ModelNet40 (Wu et al., 2015) is the *de-facto* benchmark for point cloud classification containing 9,843 training and 2,468 testing meshed CAD models belonging to 40 different object classes.

C.4 ShapeNetPart

ShapeNetPart (Yi et al., 2016) consists of 16,881 shapes with 16 classes belonging to 50 parts labels. We use ShapeNetPart for segmentation.

C.5 Trainig splits

For IntrA and RBC, we used a five-fold cross-validation and reported the average test metrics across folds. For ModelNet40 and ShapeNetPart, we used the provided train and test splits and reported the test results.

D Additional results

This section contains additional results of individual pooling methods.

D.1 Interpretability

Tables 6, 7, and 8 show the IntrA interpretability results for each of the pooling methods using the Transformer, PointNet, and DGCNN backbones, respectively. The mean and standard deviations on the test sets across the five folds are shown.

Table 6: Additional PointMIL interpretability results on IntrA using the transformer backbone. We also show the effect of the best contextual attention for each attention-based method.

Model	NDCG@n	AOPCR
`Additive`	$0.613_{0.033}$	$18.108_{4.374}$
`Additive` + context 12	$0.608_{0.035}$	$18.162_{3.013}$
`Attention`	$0.426_{0.030}$	$10.336_{1.065}$
`Attention` + context 12	$0.539_{0.019}$	$14.541_{1.821}$
`Conjunctive`	$0.592_{0.018}$	$12.526_{2.960}$
`Conjunctive` + context 12	$0.610_{0.024}$	$16.305_{5.859}$
`Instance`	$0.587_{0.022}$	$16.166_{3.794}$

Table 7: Additional interpretability results on IntrA using PointMIL with the PointNet backbone

Model	NDCG@n	AOPCR
`Additive`	$0.254_{0.064}$	$0.792_{0.298}$
`Attention`	$0.222_{0.027}$	$0.005_{0.035}$
`Instance`	$0.225_{0.072}$	$0.973_{0.212}$
`Conjunctive`	$0.208_{0.067}$	$0.741_{0.140}$

Table 8: Additional interpretability results on IntrA using POINTMIL with the DGCNN backbone

Model	NDCG@n	AOPCR
`Additive`	$0.482_{0.009}$	$4.486_{0.550}$
`Attention`	$0.223_{0.002}$	$-0.031_{0.070}$
`Conjunctive`	$0.467_{0.008}$	$4.828_{0.617}$
`Instance`	$0.462_{0.022}$	$5.212_{0.547}$

E VISUAL INTERPRETATION EXAMPLES

Figure 10 shows additional interpretability visualisations on ModelNet40.

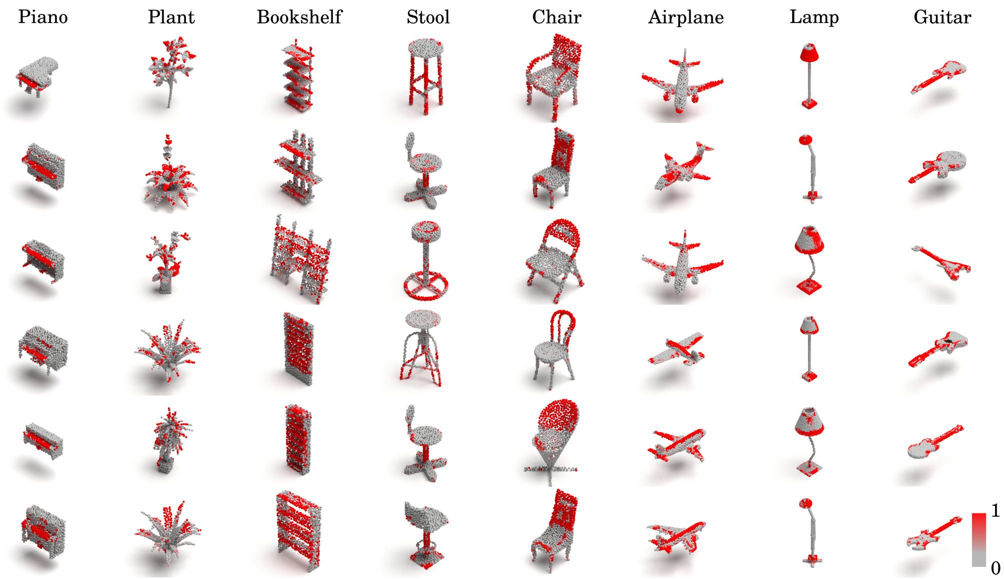

Figure 10: Examples of POINTMIL interpretations for correctly classified shapes from ModelNet40.

F Robustness to noise

Similar to the methods described by Xiang et al. (2021) and Yan et al. (2020), we assessed the robustness of PointMIL to noisy inputs. Specifically, we measured the F1 score of models trained on clean (raw) inputs when subjected to noisy inputs during inference. This approach allowed us to evaluate the model's ability to maintain performance in the presence of input perturbations. The F1 score (left) and the mACC (right) is plotted against the number of noisy points introduced during inference for different PointMIL methods with the DGCNN backbone and the original DGCNN model in Figure 11. PointMIL methods demonstrate higher robustness to noise compared to baseline models, with `Additive` and `Conjunctive` maintaining consistently higher F1 and mACC scores than the original DGCNN without MIL.

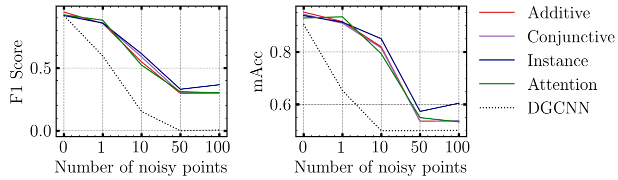

Figure 11: Robustness evaluation of models to noisy inputs.

G Segmentation

Figure 12 presents segmentation results for PointMIL with the Transformer backbone in the IntrA dataset. Clearly, PointMIL is able to accurately Aneurysm regions with a 3D shape of a diseased blood vessel.

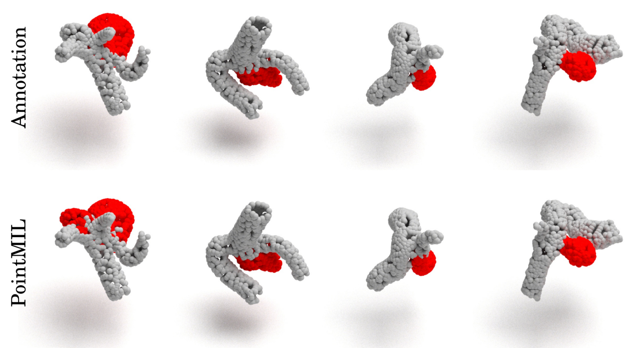

Figure 12: Segmentation examples for PointMIL with the Transformer backbone on the IntrA dataset.

H Rendering

All renderings of point clouds were made with Mitsuba2.