IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 June 2024, accepted 22 July 2024, date of publication 1 August 2024, date of current version 15 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3436681

APPLIED RESEARCH

Comparative Evaluation of Deep Neural Network Performance for Point Cloud-Based IFC Object Classification

MAJID SEYDGAR, ÉRIK A. POIRIER, AND ALI MOTAMEDI
Department of Construction Engineering, École de Technologie Supérieure, Université du Québec, Montreal, QC H3C 1K3, Canada

Corresponding author: Ali Motamedi (ali.motamedi@etsmtl.ca)

This work was supported in part by MITACS under Grant IT27239.

ABSTRACT Point cloud-based deep neural networks (PC-DNNs) has seen growing interest in the construction domain due to their remarkable ability to enhance Building Information Modeling (BIM)-related tasks. Among these tasks, Industry Foundation Classes (IFC) object classification using PC-DNNs has become an active research topic. This focus aims to mitigate classification discrepancies that occur during the interoperability of BIM tools for information exchange. However, existing studies have not fully investigated the potential of the PC-DNN models for IFC object classification. This limitation is due to the reliance on a limited number of PC-DNN models trained on small, private datasets that are not openly accessible. To address this knowledge gap, this study evaluates diverse state-of-the-art PC-DNN models for IFC object classification. Our study provides a comprehensive analysis of how different PC-DNN components and loss functions affect IFC classification, utilizing two public IFC datasets: IFCNet and BIMGEOM. Experimental results offer a detailed comparison across metrics such as accuracy, learning progression, computation time, and model parameters.

INDEX TERMS Building information modeling (BIM), deep neural networks (DNN), industry foundation classes (IFC), metric learning, object classification, scan-to-BIM.

I. INTRODUCTION

The digital transformation of the built asset industry has been prompted in large part by the significant increase in the use of Building Information Modeling (BIM) [1]. This trend is in line with the growing interest in leveraging digital technologies to improve efficiency, accuracy, and collaboration in the design, construction, and maintenance of buildings and infrastructure [2]. Professionals in various sectors of the built asset industry use task-specific software products to address various requirements, such as safety planning and analysis [3], code compliance verification [4], and energy efficiency modeling [5]. In this regard, BIM has emerged as a crucial process for information exchange between stakeholders involved in the project and asset lifecycle, specifically by leveraging Industry Foundation Classes (IFC) as an open and international data standard [6].

The IFC schema is designed to meet a wide range of needs within the domain of the built industry [7], [8]. However, its detailed and comprehensive nature introduces a level of complexity that can lead to multiple interpretations. As a result, issues such as errors, omissions, and inaccuracies can occur in the data exchange process [9]. These issues are often due to the challenge of correctly mapping BIM elements to the extensive number of IFC classes that represent various building components and systems [9], [10]. For example, during the exporting phase of a BIM model, a Bearing Plate may be incorrectly mapped to IfcPlate instead of its corresponding IFC class, IfcBeam [11]. Furthermore, the use of massing tools or the complexity of identifying the right correspondence between elements and their IFC equivalents can result in the creation of generic elements for which no specific IFC classes are assigned by the authoring tool. These

The associate editor coordinating the review of this manuscript and approving it for publication was Bo Pu.

elements, such as IFCBuildingelementProxy, often cannot be classified under any specific predefined IFC entity due to mismatched property sets or manual errors [9], [12], [13], [14].

Semantic enrichment of BIM can be used to address the issues of inconsistent classification information that can occur during the import/export process of IFC across BIM-supporting tools [9], [12]. Traditional strategies within this domain are applied a series of predetermined rules to validate or refine the type information of elements [15]. However, these rule-based methods require extensive expert knowledge and a high number of assumptions. Recent studies have employed machine learning (ML) and deep neural networks (DNNs) as data-driven approaches to streamline the subjective IFC object classification process [12], [16]. In particular, several studies employed supervised DNNs to recognize and categorize BIM elements according to their geometric characteristics [14], [17].

The DNN models use different uniform data representation formats, such as projected (i.e., rendered) images and sampled point clouds, to process parametric data formats, including IFC, in a data-driven manner. Specifically, point cloud-based DNN (PC-DNN) models have gained prominence in 3D construction tasks within the built asset industry [18], [19], [20]. This prominence is due to the extensive pre-processing demand of multi-view image classification models, e.g., multi-view convolutional neural network (MV-CNN) [21], which makes them unsuitable for real-world applications [22]. The pre-processing includes the generation of multiple images per sample (e.g., 12 images), which leads to a high computational overhead. In contrast, PC-DNNs offer more manageable computational overhead since they typically require lightweight sampled point sets (e.g., 1K points per sample) to perform semantic enrichment (e.g., [14]). Furthermore, PC-DNNs can be applicable in constructing as-built or as-is BIM models from laser-scanned point cloud data by perform classification of scanned building components [23], [24], [25]. Given the above-mentioned advantages, this research focuses on PC-DNNs, which are more applicable for real-world applications that involve large dataset.

Numerous studies have been investigated PC-DNN models for different downstream tasks in built industry applications, including IFC object classification [14], [26]. PointNet [27] and PointNet ++ [28] were trained to perform IFC classification from point cloud and their results were compared with those of MVCNN in [17] and [22], respectively. Collins et al., presented a graph convolutional network (GCN) and compared its IFC classification results with those of DGCNN [10]. Emunds et al. introduced SpaRSE-BIM, a novel classifier for IFC objects from point clouds, and utilized DGCNN, as a PC-DNN competitor [14]; additionally, they compared SpaRSE-BIM's performance with multi-view and mesh-based approaches, specifically MVCNN and MeshNet [29], across two public IFC datasets (i.e., IFCNet [30], and BIMGEOM [31]). Similarly, DGCNN was used to compare its classification results with MVCNN and MeshNet [29], in [30] for IFC classification.

Despite the aforementioned efforts in comparative assessment of PC-DNNs for IFC classification, existing studies tend to report results based on a narrow range of DNN models and the datasets used in their experiments. Specifically, a majority of these studies have relied solely on DGCNN as the representative PC-DNN, comparing its performance against that of other rivals. However, recent research in other domains within the built asset industry, e.g., [18], [19], conducts extensive comparative analyses between a variety of PC-DNN. Consequently, the current body of research on IFC object classification falls short in demonstrating the full capabilities of PC-DNN, due to the limited scope of the model comparison selections.

This paper aims to bridge this gap by evaluating the performance of various PC-DNN models in the context of IFC object classification. Specifically, this study assesses the performance of the state-of-the-art PC-DNN models, each with different architectures, against two publicly available IFC datasets: IFCNet and BIMGEOM. The PC-DNNs selected for this study incorporate a variety of neural network components, including MLP, graph-based approaches, transformers, and residual learning techniques. This selection is intended to provide a comprehensive understanding of the impact these components have on IFC object classification. Additionally, this work offers a comparative analysis of PC-DNN model training using various loss functions to illustrate the influence of learning strategies on PC-DNN training for IFC object classification. The experimental results present a wide-ranging comparison of different PC-DNNs in terms of classification accuracy, learning progress, computational time, and the number of parameters.

II. RELATED WORKS

Although the IFC serves as a standard and open-source data exchange format, various BIM tools and platforms still rely on their unique internal formats. Consequently, converting data to and from IFC is an inevitable task for information exchange between BIM platforms, a process that may contain errors and omissions [12]. Additionally, volumetric models generated as a result of 3D indoor scanning or the use of massing tools for 3D modeling only contain geometric information, and semantic properties, such as class labels (i.e., element types) are mainly left unspecified. In this scenario, performing semantic enrichment through the process of object classification can update or refine the classification type information and avoid potential issues through the information modeling process. Specifically, the properties of BIM objects largely depend on their class (i.e., element type), which makes their classification vital for subsequent usage in various analysis tools.

In recent years, supervised DNN models have been developed to perform classification of IFC objects in a data-driven manner. These supervised DNN models are capable of automatically extracting feature descriptors, also known as embeddings, to represent the geometric characteristics of various IFC objects using annotated data (i.e., labeled data). Different data representation modalities have been employed by the DNN models to process the geometric information of IFC objects for classification purposes. These modalities include image projection [22], [32], point cloud sampling [14], [17], and voxel encoding [10], [33].

Projection-based models leverage a series of rendered images to represent 3D objects, as shown in the works of [21], [34], and [35], among others. A DNN extracts and aggregates features from these images, delivering the global feature descriptor to perform the classification task. Although these models show promising results, their pre-processing schemes often require extensive rendering task with significant computational overhead. In addition, the performance of the projection-based approaches heavily relies on additional modules for multi-view rendering and feature aggregation.

Grid-based (aka voxel-based) models address the challenges of irregular 3D input analysis through voxelization [36], [37]. Despite their computational and memory demands, these approaches have seen improvements in recent years by utilizing sparsity-aware representations models, such as unbalanced octrees [38], spare convolution neural network (CNN)-based processing [39], [40], and the use of variational autoencoders and radial basis functions to streamline voxelization [41]. However, the voxelization's quantization process of the voxel-based approach may result in a loss of semantic information inherited in the points (e.g., RGB data).

Point-based approaches, on the other hand, use models capable of directly processing the point clouds for various 3D tasks such as point cloud classification. An early example is PointNet [27], which maps irregular input data (i.e., point cloud) into a fully connected DNN. PointNet++ [28] enhances PointNet by recursively implementing it on a progressively divided input point cloud to develop hierarchical interpretations. PointMLP [42] introduces a residual-based MLP network that uses geometric affine transformation to extract shape descriptors. PointConv [43] introduces a convolution operation to group point samples and extract local geometry information. PointNeXt [44] augments PointNet++ with an inverted residual bottleneck blueprint and an array of enhanced training tactics for efficient and effective model scaling. Point Transformer [45] employed Transformer [46] used in various vision tasks (e.g., [47], [48]) to explore localize contextual information using self-attention mechanism.

In addition, several studies attempt to enhance the point-based processing approach by incorporating contextual information through graph-based networks and convolutions designed in continuous Euclidean space. RS-CNN [49] introduces relation-shape convolution to explicitly encode the geometric relationship of points. DGCNN [50] presents a local accumulation layer to connect a point with its surrounding neighbors to dynamically form a local graph, and obtain its geometric properties. PosPool [51] makes the local accumulation operator simpler and eliminates learnable weights. PointASNL [52] presents adaptive sampling and local-nonlocal modules to tackle noise and improve the incorporation of short- and long-term dependencies for feature learning task, respectively. CurveNet [53] further improves feature learning by introducing a two-staged module that generates curves through guided walks and then aggregates them to enhance point-wise features.RepSurf [54] expands PointNet++ by substituting the point location with surface representation in the stem layer. HGNet [55] learn point cloud representations by employing Transformer structure and geometrical aggregation modules.

These various models have recently been used to analyze point clouds in the construction domain. The projection-based model, multi view CNN (MV-CNN), for example, has been adopted in several studies (e.g., [14], [17], [22], [30]) to handle IFC object classification. In terms of point cloud-based models (i.e., PC-DNN models), PointNet and PointNet ++ has served as the backbone for multiple frameworks focused on point cloud segmentation (e.g., [56], [57]) and classification tasks (e.g., [17], [58]). For instance, PointNet was employed by Chen et al. [25] to facilitate 3D scene reconstruction in computer-aided design (CAD) format by classifying 3D point set based on building entities. Moreover, an enhanced version of PointNet ++ with residual connections was used in [59] to segment mechanical objects from a large-scale point cloud. SE-PseudoGrid [18], a modified variant of PseudoGrid [51], [60], was introduced for the classification of piping components from point clouds. The PseudoGrid was subsequently applied in a DNN-based framework to improve the efficiency of segmentation tasks. Despite the remarkable performance of PC-DNN models in IFC object classification, existing studies typically present restricted comparative evaluations, utilizing datasets that are not publicly available. This has resulted in a lack of comprehensive study on the efficacy of various PC-DNN architectures specifically for IFC object classification.

III. RESEARCH METHODOLOGY

To tackle the knowledge gap elaborated in the previous section and to assess the effectiveness of various PC-DNN architectures for IFC object classification, our study investigates and evaluates multiple state-of-the-art DNN models, each possessing distinct architectures and local feature extractors. Specifically, PointNet++, Point Transformer, CurveNet, and PointMLP models are assessed and their detailed properties will be discussed subsequently. The primary goal of our investigation is to determine which PC-DNN model, with its corresponding processing approach, can yield the most promising performance for IFC object classification. In this regard, the PC-DNN models are evaluated in terms of classification accuracy, computational requirements, and learning complexity. The details of the PC-DNN models are as follows.

- **PointNet++** [28] is one of the pioneering models that directly process the point cloud using MLP-based architecture. The PointNet++ contains hierarchical feature extraction and query ball grouping modules to extract and aggregate shape descriptors in different neighboring level for the classification task.
- **Point Transformer** [45] is one of the more recent PC-DNN model, which employed encoder-decoder architecture with self attention mechanisms [46] to extract a global shape descriptor.
- **CurveNet** [53] treats the input point set as an undirected graph and generates continues sequences of point segments. The generated sequences are then grouped using deterministic factors and used for geometry-aware feature learning.
- **PointMLP** [42] is a state-of-the-art (SOTA) PC-DNN model, which reported results with over 94% accuracy on well-known ModelNet40 dataset [61]. This model used affine transformation and residual connections to extract aggregated shape descriptors.

In order to fully explain the methodology used in this study, it is necessary to elaborate the mechanisms of feature extraction and classification applied by the experimental models (i.e., PointNet++, Point Transformer, CurveNet, and PointMLP).

Given the input set denoted as X, the feature extraction function of the model denoted as $\mathcal{F}$ maps X to a low dimensional feature set denoted as H, which represents the shape feature descriptor of the input point set X. It should be noted that X can contain feature set $F = \{f_i\}_{i=1}^{N}$ where $f_i \in \mathbb{R}^{d_e}$ along with point set $P = \{p_i\}_{i=1}^{N}$ where $p_i \in \mathbb{R}^{d}$. The shape feature descriptor is then passed through the classification module denoted as $\mathcal{C}$ to obtain the predicted label denoted as $\hat{y}$ for each input point set. The output of the DNN model can be defined using Eq. 1.

$$\hat{y} = \overbrace{\mathcal{C}_j\left(\mathcal{F}_j(X)\right)}^{\text{backbone}}, \tag{1}$$

where $\mathcal{C}_j$ and $\mathcal{F}_j$ correspond to the feature extraction and classification modules. During the training process of the DNN models, the output of the backbone $\hat{y}$ is used to calculate the model's error through the loss function. The loss function, which can be defined as $\mathcal{L} = (\hat{y}, y)$, calculates the error value of the input X using its true label value denoted as y. Since the improvements in training task have shown significant effects on the generalization capacity of PC-DNN models, this study examines several loss functions to assess the effect of different convergence strategies on the learning progress and optimization. The details of the examined loss functions are as follows.

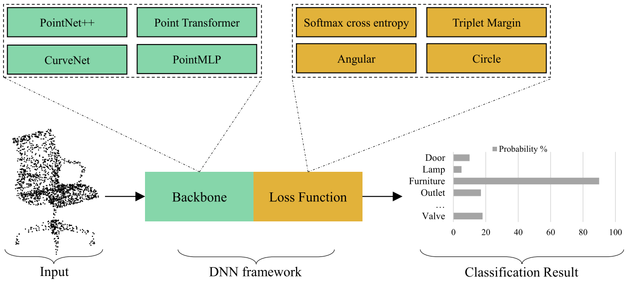


FIGURE 1. The summary of the comparison approach to assess the role of DNN models and loss functions for IFC object classification from point cloud.

1) SOFTMAX CROSS ENTROPY

This is the typical loss function used in the context of classification tasks where the loss is computed through the measurement of discrepancy between the probability distribution of $\hat{y}$ and y. The softmax cross entropy denoted as $\mathcal{L}_e(\cdot)$ can be defined as Eq. 2.

$$\mathcal{L}_e(\hat{y}, y) = -\sum_{j=1}^{K} y_j \log(\hat{y}_j), \tag{2}$$

where y_j and $\hat{y}_j$ are the corresponding j-th elements of the labeled and predicted samples respectively, and K is the total number of label classes.

2) TRIPLET MARGIN

The triplet margin computes the loss through the estimation of relative distance between samples to fulfill the term that an anchor sample should be closer to positive (similar) samples than to negative (dissimilar) samples by a certain margin. The Triplet margin loss can be defined as Eq. 3.

$$\mathcal{L}_t(a, p, n) = \max(d(a, p) - d(a, n) + m, 0), \tag{3}$$

where $d(\cdot, \cdot)$ is the L_p distance measure, a, p, n are anchor, positive, and negative samples respectively, and m is a predefined margin.

3) ANGULAR

Instead of distance-based computation among triplet points in the Triplet Margin, the Angular loss, denoted as $\mathcal{L}_a(\cdot)$, estimate the angles between the embedded features to enhance the optimization convergence in high dimensional solution space. By using the angles, the Angular loss attempts to ease the optimization of the triplet loss scheme while benefits from its relative similarity estimation technique. The Angular loss function can be defined as E.q. 4.

$$\mathcal{L}_a(X') = \frac{1}{N}\sum_{a\in X'} \left\{ \log\left[1 + \sum_{\substack{n\in X' \\ y_n \neq y_a, y_p}} \exp(s_{a,p,n}) \right] \right\}, \tag{4}$$

where X' is a batch of samples with N members, y_n, y_a, and y_p are the labels of a, p, and n samples, respectively. The definition of $s(\cdot)$ is brought in Eq. 5.

$$s_{a,p,n} = 4\tan^2(\beta)(a+p)^T n - 2(1+\tan^2(\beta))a^T p, \qquad (5)$$

where $\beta > 0$ is a predefined parameter.

4) CIRCLE

The circle loss, denoted as $\mathcal{L}_c(\cdot)$, provides an additional weighting module for calculating the loss value by estimating the importance of the obtained error rates of the samples according to a circular decision boundary. The aim of this loss function is to increase the discrimination among samples, while providing more flexibility through the optimization of the loss calculation. We refer the reader for more information to [62].

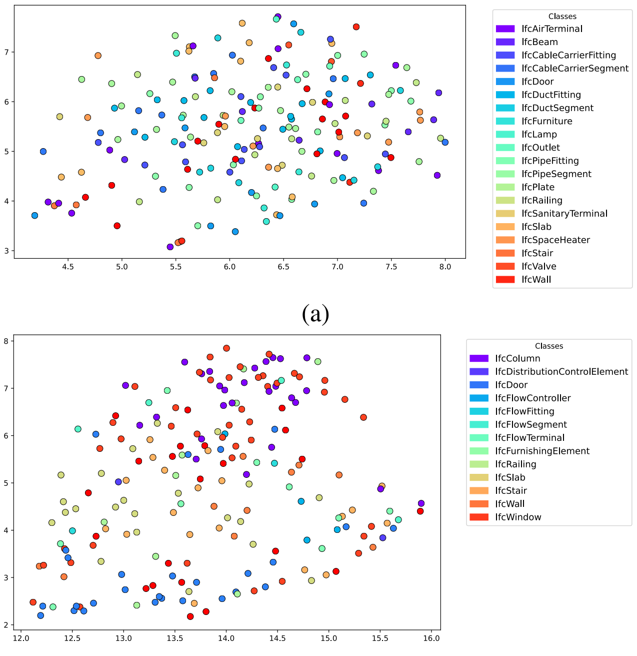
To investigate the effects of different feature extraction backbones and loss functions on the classification of IFC objects from point cloud data, this study proposes a unified framework that performs assessments in two distinct phases. In the first phase, the feature extraction backbone analysis, the assessment is conducted using similar input samples with a commonly used loss function, i.e., softmax cross-entropy. For the second phase, one of the feature extraction backbones is randomly selected to analyze the learning convergence and generalization capacity. This random selection is justified by the independence of the learning task from the backbone architecture and aims to efficiently manage computational resources. Fig. 1 provides a summary of the assessment framework to evaluate different DNN models and their loss functions in the context of IFC object classification from point cloud.

IV. EXPERIMENT

A. EXPERIMENTAL CONFIGURATION

For the evaluation of the selected DNN models on the task of IFC object classification, a robust experimental design is implemented, leveraging two synthetic and publicly available datasets, i.e., IFCNet [30] and BIMGEOM [31]. The IFCNet dataset includes 7,930 objects across 20 diverse IFC classes, and the BIMGEOM dataset, larger in comparison, comprised 10,146 objects containing 13 IFC classes. The IFC classes ranged from low texture element types, such as walls and cable carrier segments, to intricately structured objects, e.g., valves and furniture, offering a wide spread of classes. Fig. 2. provides an illustration of the distribution of the datasets' raw samples in the 2D space embedded using uniform manifold approximation and projection (UMAP) technique [63]. As can be seen in Fig. 2., both datasets' samples contain highly similar features with different class labels, which demonstrated their challenging nature with respect to the classification task.

Regarding the configuration of training parameters, we followed the commonly used optimization settings and employed the Stochastic Gradient Descent (SGD) optimizer with the momentum and weight decay set to 0.9 and 0.0001, respectively. We also used a cosine annealing schedule [64] to enhance the convergence. To maintain uniformity, all the DNN models, i.e., PointNet++, Point Transformer, CurveNet, and PointMLP, were trained for a consistent span of 150 epochs, and the highest performing test epoch during this training phase was recorded for each dataset based on the evaluation metrics: overall accuracy (OA) and mean average accuracy (mAcc). In addition, the computational efficiency and the learning progress analysis of the DNN models are assessed to provide a more comprehensive evaluation of each models' advantages and drawbacks. All models were tested on a Linux operating system using two GEFORCE 3090 RTX GPUs. It should be noted that we adhered to the predefined train and test splits as established in the standard configuration of the IFCNet and BIMGEOM datasets.



(a)

(b)

FIGURE 2. The 2D visualization of 200 random samples using UMAP of the datasets in question; a) IFCNet dataset b) BIMGEOM dataset.

B. COMPARISON RESULTS

Table. 1. tabulated the classification performance of the DNN models on the IFCNet datasets. Based on the results, the Point Transformer achieved superior OA performance. In addition, the PointNet++ and PointMLP show the relatively similar classification results with the highest mAcc accuracy. Conversely, the performance of CurveNet, despite its complex local feature extraction and aggregation modules, fell short of competitiveness.

Upon evaluation on the BIMGEOM dataset with the results tabulated in Table. 2., PointMLP outperformed its counterparts by a large margin, particularly in terms of mAcc. It is worth mentioning that by comparing the performance of

TABLE 1. The classification results of the DNN models for IFCNet dataset.

Model	Points	mAcc(%)	OA(%)
PointNet++	1K	**83.2**	84.9
Point Transformer	1K	83.1	**85.2**
CurveNet	1K	75.8	78.8
PointMLP	1K	**83.2**	84.1

TABLE 2. The classification results of the DNN models for BIMGEOM dataset.

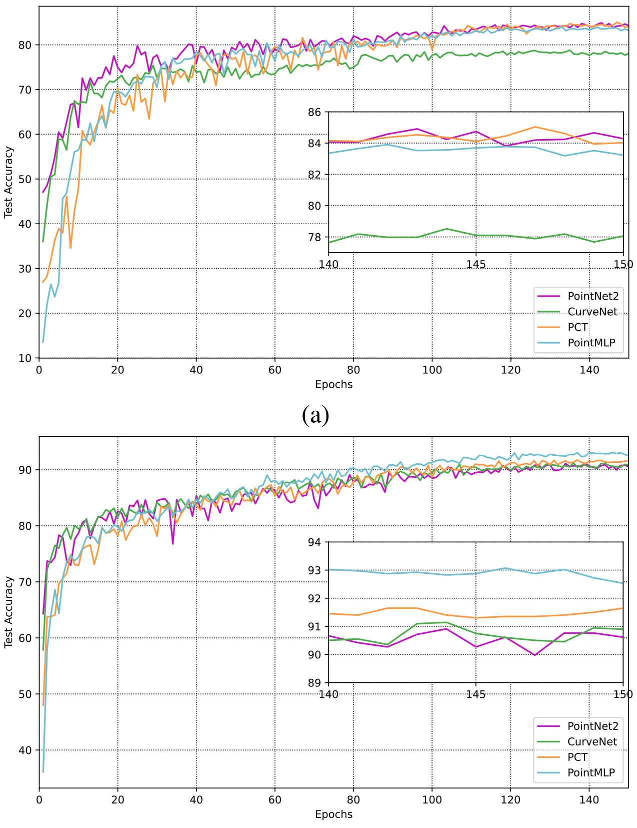
Model	Points	mAcc(%)	OA(%)
PointNet++	1K	86.7	91.2
Point Transformer	1K	88.1	91.8
CurveNet	1K	87.4	91.5
PointMLP	1K	**90.1**	**93.1**

CurveNet on the IFCNet and BIMGEOM datasets, it can be observed a notable improvement on the BIMGEOM, which contain more training samples compared to the IFCNet. Therefore, it can be inferred that CurveNet is more sensitive to the size of the training set compared to its rivals when the labeled data is more available. Regarding the learning progress analysis, Fig. 3. illustrates the OA results of test sets for the DNN models after training in each epoch.

As depicted in Fig. 3., all DNN models display performance progress over the duration of training on BIMGEOM, with PointMLP distinctly outperforming its rivals. The PointMLP's superior performance can be traced back to its impressive model's capacity, demonstrated by its over 13 million parameters (as shown in Table. 3.). It indicates that PointMLP can learn more effectively when it is trained on a larger dataset (i.e., BIMGEOM). According to the results from IFCNet, all models demonstrate limited learning progress, particularly after the 120th epoch. The results indicate that the DNN models exhibit moderate accuracy performance.

C. PERFORMANCE ANALYSIS

To study the impact of complex metric learning on improving the DNN learning for the IFC object classification, we selected Point Transformer due to its reasonable classification accuracy on both the IFCNet and BIMGEOM datasets according to Tables. 1- 2. The Point Transformer is trained using the selected loss functions, namely softmax cross entropy, triplet margin, Angular loss, and Circle loss. In order to represent how different optimization strategies navigate the loss landscape, we employed a loss landscape visualization [65]. As shown in Figs. 4 - 5., the conventional softmax function resulted in a smoother loss landscape, thereby facilitating more effective learning. Furthermore, an evaluation of the test loss (i.e., error rate) results for the test sets of the IFCNet and BIMGEOM datasets shown in Fig. 6. demonstrated that the softmax function yielded superior performance for both datasets.



(a)

(b)

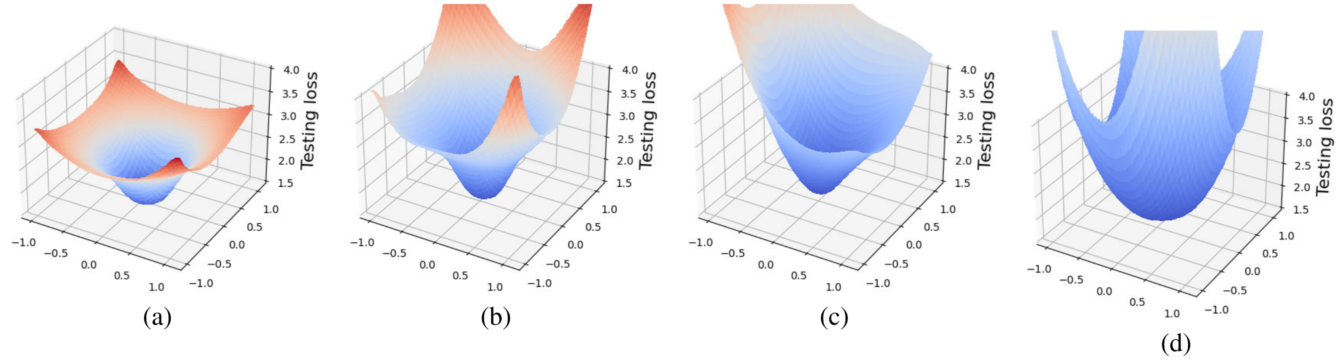
FIGURE 3. The learning progress comparison of the DNN models; a) IFCNet dataset b) BIMGEOM dataset.

In the context of computational cost, PointNet++ exhibited the highest efficiency, with shorter training time and faster test speed compared to the other models, as shown in Table. 3. In contrast, the CurveNet displayed high computational demand with the longest training time and slowest testing speed. It should be noted that despite its significant number of parameters, PointMLP exhibited moderate computational demand, which can be attributed to its use of residual connections to maintain the computational requirements alongside the high-level processing capacity.

V. DISCUSSION

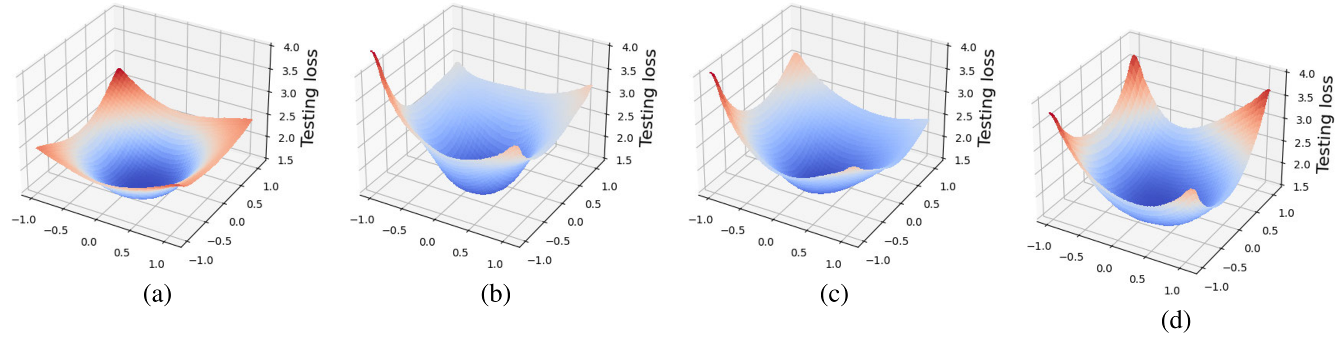
The experimental results reveal several significant observations regarding the utilization of advanced DNN models for the task of IFC object classification. From a practical perspective, this study emphasizes that the cutting edge DNN models do not always surpass their simpler alternatives when data availability is moderate or limited.

According to the experimental results, the similar performance between SOTA PointMLP and PointNet++ on IFCNet dataset highlight the fact that extra performance capacity and high computational overhead cannot guarantee high classification performance. The observation of similar performance from DNNs with varying models' capacity may be attributed to the under-training phenomenon that



(a) (b) (c) (d)

FIGURE 4. The loss landscape visualization for IFCNet dataset; a) Softmax cross entropy, b) Triplet margin c) Angular, and d) Circle.



(a) (b) (c) (d)

FIGURE 5. The loss landscape visualization for BIMGEOM dataset; a) Softmax cross entropy, b) Triplet margin c) Angular, and d) Circle.

TABLE 3. Comparison of Training Time (in hours), Test Speed (sample per second), and Number of Parameter (million) for Various DNN Models on IFCNet and BIMGEOM Datasets.

Model	Param (m)	IFCNet		BIMGEOM	
		Train Time (h)	Test Speed (S/s)	Train Time (h)	Test Speed (S/s)
PointNet++	**1.47**	**0.4**	**152**	**0.4**	**389**
Point Transformer	2.87	0.6	139	0.6	339
CurveNet	2.4	6.2	20	3.5	59
PointMLP	13.2	1.8	108	2.6	186

occurs when there is insufficient labeled data. The notable outperformance of PointMLP on the BIMGEOM dataset, which has relatively more labeled samples, further validates this observation. Consequently, it can be concluded that the use of elaborate DNN models for IFC object classification needs to be considered in conjunction with the curation of more comprehensive datasets. The dataset curation can be addressed using various methods, such as data collection, generative adversarial networks, and data augmentation methods.

Furthermore, this study highlights that the performance of the graph-based model, i.e., CurveNet, did not yield competitive results for IFC object classification, especially on the IFCNet dataset. From the theoretical view, this outcome can be attributed to factors such as the sensitivity of graph-based models to a limited number of training samples, as seen in the case of the DGCNN model in IFC object classification [66] and the characteristics of synthetic IFC samples, which may not provide sufficient local geometric details for CurveNet's feature extraction task. However, from the practical view, despite these observations, the potential of CurveNet should not be dismissed outright due to its demonstrated robustness on real and noisy datasets, such as ScanObjcetNN [67], as reported by [53]. Since applications such as BIModeling from point cloud are mostly dealing with high level of noise and occlusion, the utilization of CurveNet for IFC object classification from noisy point cloud should be addressed in future research.

Finally, the exploration of different loss functions revealed that the straightforward application of advanced loss functions may not always yield improved learning outcomes. As evidenced by the results from the loss landscapes, conventional loss functions, such as softmax cross-entropy, tend to result in smoother and flatter landscapes, thus

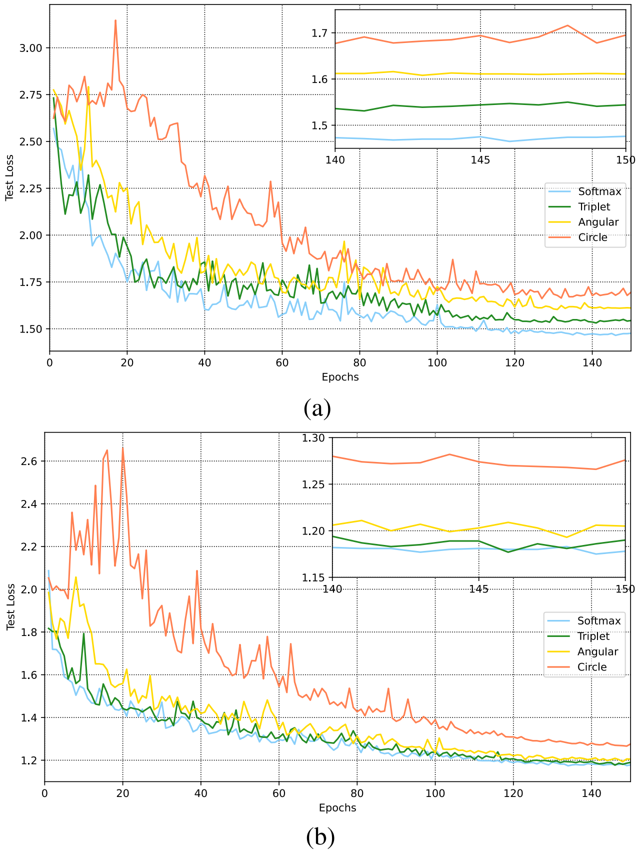


FIGURE 6. The comparison of the convergence speed of the loss functions in questions for the datasets; a) IFCNet dataset b) BIMGEOM dataset.

facilitating the training process. Future work intending to leverage advanced cost functions for IFC object classification may examine hybrid combination of loss function or novel approach to harness the discrimination strength of advance loss function through the training paradigm while avoid any unnecessary learning implications affecting the convergence rate and learning progress.

VI. CONCLUSION

Semantic enrichment of BIM has the potential to resolve inconsistencies in classification information that may arise during the import and export processes of IFC among various BIM authoring tools. Traditional methods for element type classification in this field depend on rigid rules that require expert knowledge and multiple assumptions. In contrast, the adoption of DNNs for IFC object classification has increased efficiency by enabling a data-driven approach that reduces the need for such constraints and assumptions. DNN models utilize various standardized data formats, such as rendered images and point clouds, to manage parametric data, e.g., IFC. Of the models designed for these formats, PC-DNNs (i.e., point cloud-based DNNs) have become particularly notable for 3D tasks in the built asset industry. This is due to the fact that image-based models, such as MV-CNN, often require heavy pre-processing, which limits their practical use in large-scale datasets.

A handful of studies have been employed PC-DNN models for IFC object classification and assessed their performance against various rivals. However, these research often presents findings from a limited selection of PC-DNN models and datasets. Most notably, several studies have exclusively used DGCNN as the standard PC-DNN to measure its effectiveness in comparison to competing DNN models. In contrast, research in other built industry applications has performed extensive comparisons across different PC-DNN models. This suggests that IFC classification research may not be fully exploring PC-DNN capabilities due to restricted model comparisons.

In this study, we aimed to bridge this knowledge gap by conducting an extensive comparison of various DNN model architectures and learning modules using public datasets with diverse properties. Experimental results demonstrated the performance of different feature extraction and processing architectures for IFC object classification from point clouds across metrics such as accuracy, learning progression, computation time, and model parameters. The results highlighted the existing challenges. In particular, our analysis revealed that employing elaborate DNN models for classifying publicly available IFC dataset samples does not necessarily ensure superior performance and may lead to computational resource wastage.

Our study is primarily focused on the adoption of PC-DNNs for IFC object classification. We provide an extensive comparative assessment of PC-DNNs according to several performance metrics, the learning progression, and the models' architecture. Our experimental findings suggest new research directions, including the hybrid application of complex loss functions and the utilization of PC-DNNs with graph-based models for noisy point cloud data. However, this study has not explored cutting-edge DNN models from other approaches, such as multi-view and voxel-based, as well as ensemble DNNs that combine multiple models. Future studies will be needed to assess the performance of state-of-the-art DNN models using various data processing approaches to provide a more comprehensive understanding of DNN performance in IFC object classification. Furthermore, the issue of limited training labels needs to be explored in more detail. Conducting sensitivity analysis to assess how DNN model performance scales with the size of the training samples within the datasets is recommended.

ACKNOWLEDGMENT

The authors wish to express their gratitude to the creators of IFCNet [30] and BIMGEOM [31] for making their datasets publicly accessible.

REFERENCES

[1] C. Boje, A. Guerriero, S. Kubicki, and Y. Rezgui, "Towards a semantic construction digital twin: Directions for future research," *Autom. Construct.*, vol. 114, Jun. 2020, Art. no. 103179.

[2] A. Zabin, V. A. González, Y. Zou, and R. Amor, "Applications of machine learning to BIM: A systematic literature review," *Adv. Eng. Informat.*, vol. 51, Jan. 2022, Art. no. 101474.

[3] C. K. I. Che Ibrahim, P. Manu, S. Belayutham, A.-M. Mahamadu, and M. F. Antwi-Afari, "Design for safety (DfS) practice in construction engineering and management research: A review of current trends and future directions," *J. Building Eng.*, vol. 52, Jul. 2022, Art. no. 104352.

[4] Z. Zheng, Y.-C. Zhou, X.-Z. Lu, and J.-R. Lin, "Knowledge-informed semantic alignment and rule interpretation for automated compliance checking," *Autom. Construct.*, vol. 142, Oct. 2022, Art. no. 104524.

[5] S. Seyedzadeh, F. P. Rahimian, S. Oliver, I. Glesk, and B. Kumar, "Data driven model improved by multi-objective optimisation for prediction of building energy loads," *Autom. Construct.*, vol. 116, Aug. 2020, Art. no. 103188.

[6] C. I. De Gaetani, M. Mert, and F. Migliaccio, "Interoperability analyses of BIM platforms for construction management," *Appl. Sci.*, vol. 10, no. 13, p. 4437, Jun. 2020.

[7] C. M. Eastman, Y.-S. Jeong, R. Sacks, and I. Kaner, "Exchange model and exchange object concepts for implementation of national BIM standards," *J. Comput. Civil Eng.*, vol. 24, no. 1, pp. 25–34, Jan. 2010.

[8] Building SMART Int. Ltd. (2024). *Industry Foundation Classes 4.3.2.0*. [Online]. Available: https://standards.buildingsmart.org/IFC/RELEASE/IFC4_3/

[9] M. Belsky, R. Sacks, and I. Brilakis, "Semantic enrichment for building information modeling," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 31, no. 4, pp. 261–274, Apr. 2016.

[10] F. C. Collins, M. Ringsquandl, A. Braun, D. M. Hall, and A. Borrmann, "Shape encoding for semantic healing of design models and knowledge transfer to scan-to-BIM," *Proc. Inst. Civil Eng. Smart Infrastruct. Construct.*, vol. 175, no. 4, pp. 160–180, Dec. 2022.

[11] B. Koo, S. La, N.-W. Cho, and Y. Yu, "Using support vector machines to classify building elements for checking the semantic integrity of building information models," *Autom. Construct.*, vol. 98, pp. 183–194, Feb. 2019.

[12] T. Bloch and R. Sacks, "Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models," *Autom. Construct.*, vol. 91, pp. 256–272, Jul. 2018.

[13] R. Sacks, M. Girolami, and I. Brilakis, "Building information modelling, artificial intelligence and construction tech," *Develop. Built Environ.*, vol. 4, Nov. 2020, Art. no. 100011.

[14] C. Emunds, N. Pauen, V. Richter, J. Frisch, and C. van Treeck, "SpaRSE-BIM: Classification of IFC-based geometry via sparse convolutional neural networks," *Adv. Eng. Informat.*, vol. 53, Aug. 2022, Art. no. 101641.

[15] L. Ma, R. Sacks, and K. Uri, "Building model object classification for semantic enrichment using geometric features and pairwise spatial relationships," in *Proc. Joint Conf. Comput. Construct.*, 2017, pp. 373–380.

[16] H. Luo, G. Gao, H. Huang, Z. Ke, C. Peng, and M. Gu, "A geometric-relational deep learning framework for BIM object classification," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 349–365.

[17] B. Koo, R. Jung, and Y. Yu, "Automatic classification of wall and door BIM element subtypes using 3D geometric deep neural networks," *Adv. Eng. Informat.*, vol. 47, Jan. 2021, Art. no. 101200.

[18] C. Yin, J. C. P. Cheng, B. Wang, and V. J. L. Gan, "Automated classification of piping components from 3D LiDAR point clouds using SE-PseudoGrid," *Autom. Construct.*, vol. 139, Jul. 2022, Art. no. 104300.

[19] C. Yin, B. Yang, J. C. P. Cheng, V. J. L. Gan, B. Wang, and J. Yang, "Label-efficient semantic segmentation of large-scale industrial point clouds using weakly supervised learning," *Autom. Construct.*, vol. 148, Apr. 2023, Art. no. 104757.

[20] D. Hu, V. J. L. Gan, and R. Zhai, "Automated BIM-to-scan point cloud semantic segmentation using a domain adaptation network with hybrid attention and whitening (DawNet)," *Autom. Construct.*, vol. 164, Aug. 2024, Art. no. 105473.

[21] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.

[22] B. Koo, R. Jung, Y. Yu, and I. Kim, "A geometric deep learning approach for checking element-to-entity mappings in infrastructure building information models," *J. Comput. Design Eng.*, vol. 8, no. 1, pp. 239–250, Jan. 2021.

[23] T. Czerniawski and F. Leite, "Automated digital modeling of existing buildings: A review of visual object recognition methods," *Autom. Construct.*, vol. 113, May 2020, Art. no. 103131.

[24] Y.-P. Zhao and P. A. Vela, "Scan2BrIM: IFC model generation of concrete bridges from point clouds," in *Proc. Comput. Civil Eng.*, Jun. 2019, pp. 455–463.

[25] J. Chen, Z. Kira, and Y. K. Cho, "Deep learning approach to point cloud scene understanding for automated scan to 3D reconstruction," *J. Comput. Civil Eng.*, vol. 33, no. 4, Jul. 2019, Art. no. 04019027.

[26] Y. S. Yu, S. H. Kim, W. B. Lee, and B. S. Koo, "Ensemble-based deep learning approach for performance improvement of BIM element classification," *KSCE J. Civil Eng.*, vol. 27, no. 5, pp. 1898–1915, May 2023.

[27] 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. (CVPR)*, 2017, pp. 652–660.

[28] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[29] Y. Feng, Y. Feng, H. You, X. Zhao, and Y. Gao, "MeshNet: Mesh neural network for 3D shape representation," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 8279–8286.

[30] C. Emunds, N. Pauen, V. Richter, J. Frisch, and C. van Treeck, "IFCNet: A benchmark dataset for IFC entity classification," in *Proc. EG-ICE Workshop Intell. Comput. Eng.* Berlin, Germany: Universitätsverlag der TU Berlin, 2021, pp. 166–175.

[31] F. C. Collins, A. Braun, M. Ringsquandl, D. M. Hall, and A. Borrmann, "Assessing IFC classes with means of geometric deep learning on different graph encodings," in *Proc. Eur. Conf. Comput. Construct.*, Jul. 2021.

[32] J. Kim, J. Song, and J.-K. Lee, "Recognizing and classifying unknown object in BIM using 2D CNN," in *Proc. 18th Int. Conf. Comput.-Aided Architectural Design Hello, Culture CAAD Futures*, Daejeon, Republic Korea. Cham, Switzerland: Springer, Jun. 2019, pp. 47–57.

[33] I. E. Evangelou, N. Vitsas, G. Papaioannou, M. Georgioudakis, and A. Chatzisymeon, "Shape classification of building information models using neural networks," in *Proc. 3DOR*, 2021, pp. 1–4.

[34] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "GVCNN: Group-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 264–272.

[35] T. Yu, J. Meng, and J. Yuan, "Multi-view harmonized bilinear network for 3D object recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 186–194.

[36] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[37] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.

[38] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6620–6629.

[39] B. Graham, M. Engelcke, and L. van der Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9224–9232.

[40] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal ConvNets: Minkowski convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3070–3079.

[41] H.-Y. Meng, L. Gao, Y.-K. Lai, and D. Manocha, "VV-Net: Voxel VAE net with group convolutions for point cloud segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8499–8507.

[42] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," 2022, *arXiv:2202.07123*.

[43] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9613–9622.

[44] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny, and B. Ghanem, "PointNeXt: Revisiting PointNet++ with improved training and scaling strategies," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 23192–23204.

[45] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, no. 2, pp. 187–199, 2021, doi: 10.1007/s41095-021-0229-5.

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[47] Y. Li, T. Yao, Y. Pan, and T. Mei, "Contextual transformer networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 1489–1500, Feb. 2023.

[48] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[49] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8887–8896.

[50] 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, Oct. 2019.

[51] Z. Liu, H. Hu, Y. Cao, Z. Zhang, and X. Tong, "A closer look at local aggregation operators in point cloud analysis," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K. Cham, Switzerland: Springer, Aug. 2020, pp. 326–342.

[52] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5588–5597.

[53] T. Xiang, C. Zhang, Y. Song, J. Yu, and W. Cai, "Walk in the cloud: Learning curves for point clouds shape analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 895–904.

[54] H. Ran, J. Liu, and C. Wang, "Surface representation for point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18920–18930.

[55] T. Yao, Y. Li, Y. Pan, and T. Mei, "HGNet: Learning hierarchical geometry from points, edges, and surfaces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 21846–21855.

[56] J. Balado, R. Sousa, L. Díaz-Vilariño, and P. Arias, "Transfer learning in urban object classification: Online images to recognize point clouds," *Autom. Construct.*, vol. 111, Mar. 2020, Art. no. 103058.

[57] E. Agapaki and I. Brilakis, "Instance segmentation of industrial point cloud data," *J. Comput. Civil Eng.*, vol. 35, no. 6, Nov. 2021, Art. no. 04021022.

[58] Y. Song, F. He, L. Fan, J. Dai, and Q. Guo, "DSACNN: Dynamically local self-attention CNN for 3D point cloud analysis," *Adv. Eng. Informat.*, vol. 54, Oct. 2022, Art. no. 101803.

[59] C. Yin, B. Wang, V. J. L. Gan, M. Wang, and J. C. P. Cheng, "Automated semantic segmentation of industrial point clouds using ResPointNet++," *Autom. Construct.*, vol. 130, Oct. 2021, Art. no. 103874.

[60] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6410–6419.

[61] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[62] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6397–6406.

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

[64] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.

[65] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 6389–6399.

[66] M. Seydgar, A. Motamedi, and É. Poirier, "Performance assessment of deep neural networks for classification of IFC objects from point cloud with limited labeled data," in *Proc. Transforming Construct. Reality Capture Technol.*, 2022, pp. 1–7.

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

MAJID SEYDGAR received the M.Sc. degree in remote sensing from the University of Isfahan, in 2018. He is currently pursuing the Ph.D. with the Engineering School (ETS), University of Quebec. His research interests include computer vision and machine learning.

ÉRIK A. POIRIER received the bachelor's degree in architecture from Laval University and the master's and Ph.D. degrees in construction engineering from École de Technologie Supérieure (ETS), Montreal. He is currently a Professor with the Department of Construction Engineering, ETS. He heads the Research Group for the Integration and Sustainable Development in the Built Environment [Groupe de recherche en intégration et développement durable en environnement bâti (GRIDD)]. He is the President of Groupe BIM du Québec. He completed his postdoctoral fellowship with the BIM TOPiCS Lab, The University of British Columbia. He has worked extensively in the construction industry, cumulating more than 20 years of experience across Canada. He is a member of the Board of Directors of buildingSMART Canada and the Chair of the ISO TC 59/SC13 Mirror Committee for the Standards Council of Canada.

ALI MOTAMEDI received the Ph.D. degree from the Interdisciplinary Engineering Program, Concordia University. From 2014 to 2016, he conducted research in the World-Renowned Laboratory, Osaka University, Japan. He is currently an Associate Professor with École de Technologie Supérieure (ETS). His research investigates the application of sensor networks, ambient and artificial intelligence, and visual analytics for the design, construction, and management of built facilities. His interests also reside in the cyber-physical interactions provided by "digital twins" of facilities through an integration of digital models and the Internet of Things (IoT). He is the Holder of several national and international scholarships, fellowships, and awards, including JSPS, FQRNT, NSERC, and the Merit Award. His research results have been published in several journals and presented in various conferences and won multiple best-paper awards. He has also many years of consulting/design experience in the area of facilities management and information technology and has participated in several large-scale IT projects.