A PRE-TRAINING METHOD FOR 3D BUILDING POINT CLOUD SEMANTIC SEGMENTATION

Yuwei Cao*, Marco Scaioni

Department of Architecture, Built Environment and Construction Engineering, Politecnico di Milano
via Ponzio 31, 20133 Milano, Italy - emails: {yuwei.cao, marco.scaioni}@polimi.it

Commission II, WG II/6

KEY WORDS: 3D Building Point Cloud, Deep Learning, Fine-tune, Pre-training, Transfer Learning

ABSTRACT:

As a result of the success of Deep Learning (DL) techniques, DL-based approaches for extracting information from 3D building point clouds have evolved in recent years. Despite noteworthy progress in existing methods for interpreting point clouds, the excessive cost of annotating 3D data has resulted in DL-based 3D point cloud understanding tasks still lagging those for 2D images. The notion that pre-training a network on a large source dataset may help enhance performance after it is fine-tuned on the target task and dataset has proved vital in numerous tasks in the Natural Language Processing (NLP) domain. This paper proposes a straightforward but effective pre-training method for 3D building point clouds that learns from a large source dataset. Specifically, it first learns the ability of semantic segmentation by pre-training on a cross-domain source Stanford 3D Indoor Scene Dataset. It then initialises the downstream networks with the pre-trained weights. Finally, the models are fine-tuned with the target building scenes obtained from the ArCH benchmarking dataset. Our paper evaluates the proposed method by employing four fully supervised networks as backbones. The results of two pipelines are compared between training from scratch and pre-training. The results illustrate that pre-training on the source dataset can consistently improve the performance of the target dataset with an average gain of 3.9%.

1. INTRODUCTION

The digital representation of the building point clouds enables and facilitates new applications in a variety of subsequent tasks, including simulation (Hong et al., 2018), smart city (Ruohomäki et al., 2018), monitoring (Begić et al., 2021), reconstruction (Previtali et al., 2018), and geographic information systems (GIS) update (Dukai et al., 2020). In these applications, the automatic classification of high Level-of-Detail (LoD) buildings is a fundamental and critical task. Inspired by the success of Artificial Intelligence (AI) approaches applied to subset tasks in Computer Vision (CV) (e.g., classification, object detection, and semantic segmentation), Deep Learning (DL) has been used in the last few years to extract information from 3D building point clouds in various applications , such as building modelling (Czerniawski et al., 2020), energy estimation (Ham et al., 2015), and cultural heritage (Sánchez-Aparicio et al., 2016; Pierdicca et al., 2020; Teruggi et al., 2020).

Existing DL-based approaches for analysing and interpreting point clouds have been developed, such as techniques for registration (Aoki et al., 2019; Lu et al., 2019), classification (Joseph-Rivlin et al., 2019; Thabet et al., 2020), and semantic segmentation (Qi et al., 2017a, 2017b; Landrieu et al., 2018). However, applying DL methods to the semantic segmentation task of 3D building point clouds remains a challenging task, as DL-based methods heavily rely on large-scale labelled datasets. For instance, ImageNet (Deng et al., 2009) in the 2D CV domain contains more than 14 million images classified into 2000 categories. In contrast, one of the most prominent 3D indoor point cloud datasets, the Stanford 3D Indoor Scene Dataset (S3DIS), only contains 270 rooms from 6 areas (Armeni et al., 2016). Moreover, for the 3D building point cloud field, the existing dataset with high LoDs collected by Matrone et al. (2020), the Architecture Cultural Heritage Dataset (ArCH), comprises only fifteen training scenes and two testing scenes. The data magnitude mismatch between the 3D and 2D domains may limit the capability of DL models trained on 3D point cloud datasets.

The challenges associated with labelling hundreds of million points in building scenes restrict the implementation of DL-based point cloud models into building-related applications where annotated data is extremely scarce. There is no doubt that transfer learning is one of the most fruitful fields of deep learning research. The insight that pre-training a network on a typically large source dataset can help enhance performance once it has been transferred to downstream tasks and fine-tuned on target datasets where labels are scarce has proven essential in developing numerous tasks in Natural Language Processing (NLP) and 2D Vision. However, training from scratch on the target data remains the dominant approach in 3D point cloud fields (Xie et al., 2020), indicating that in all 3D scene understanding tasks, DL-based 3D point cloud understanding continues to lag compared to their 2D counterparts.

Informed by the recent advances in pre-training (Xie et al., 2020; Zhang, 2021), this paper proposes the use of a straightforward but effective pre-training approach. To improve the performance on the label-scarce target 3D building dataset, we first learn the semantic segmentation capacity on the source dataset and then transfer the learned capability to the target dataset. Rather than establishing pretext tasks to capture prior knowledge in *self-supervised learning* networks (Xie et al., 2020; Zhang et al., 2021), we ask: can the ability of semantic segmentation acquired from other datasets be transferred to target datasets? To test this hypothesis, this work employs *fully supervised learning* networks to learn the semantic segmentation capability from a

* Corresponding author

cross-domain dataset. Since we observe that prior knowledge regarding distinguishing different objects can be transferred from a cross-domain dataset via pre-training, we refer to our method as "learn-to-distinguish".

The proposed pre-training method is composed of three simple steps:

1. Training DL models on a large indoor source dataset, i.e., the Stanford 3D Indoor Scene Dataset (S3DIS – Armeni et al., 2016);
2. The pre-trained weights are used as the initialisation for the downstream point cloud semantic segmentation task; and
3. Fine-tuning models using the smaller Architecture Cultural Heritage Dataset (ArCH – Matrone et al, 2020).

The proposed approach is evaluated by comparing the results of two pipelines: (1) training from scratch (without pre-training) and (2) training with pre-training. To provide more convincing results, four networks - PointNet (Qi et al., 2017a), PointNet++ (Qi et al., 2017b), DGCNN (Wang et al., 2019), and KPConv (Thomas et al., 2019) - are trained in each pipeline using varied proportions of training data (4%, 10%, 100%). The results show that transferring prior knowledge of pre-trained networks, even if it is acquired from a cross-domain dataset, can consistently improve the performance of the target building dataset.

Our contributions can be summarised as follows:

1. We present a straightforward but effective pre-training method to transfer the prior knowledge of semantic segmentation learned from a cross-domain dataset to the target dataset; and
2. In this study, empirical experiments were conducted to validate the effectiveness and practicality of the proposed method.

2. RELATED WORKS

Laser scanning techniques can collect point clouds of buildings. At the same time, the massive amount of data requires semantic interpretation at a high Level-of-Detail (LoD) to maximise the exploitation of these datasets. While Deep Learning (DL) algorithms for 3D point cloud analysis are constantly being developed and refined, labelled building point cloud datasets are rare. As a result, Deep Neural Networks (DNNs) have been limited in their applicability to architectural point clouds. In this section, the state-of-the-art of DL methods and pre-training approaches is discussed to determine the feasibility of using DL and pre-training methods applied to architectural point clouds.

2.1 DL-based Methods

Inspired by the diverse types of DNNs that are continuously being developed and improved in 2D image analysis, the ability to extract features effectively and automatically from DNNs enables the application of these promising algorithms for semantic segmentation of 3D point clouds. According to the architecture of the DNNs, existing methods for 3D point cloud semantic segmentation can be divided into *multilayer perceptron* (MLP) networks, *convolutional neural network* (CNN) networks, and *graph convolutional networks* (GCN).

Recently, the ground-breaking approach PointNet (Qi et al., 2017a) that can operate directly on point clouds was proposed. PointNet uses the MLP to learn pointwise high-dimensional features, and the max-pooling operation is used to address the disorderly inherent nature of 3D point clouds. However, since pointwise features are extracted and updated individually from each point in PointNet, the local context information between points in spatial space is neglected. This work has now been extended in a variety of ways to allow for the extraction of local and neighbouring information inside a point cloud. For instance, PointNet++ (Qi et al., 2017b) constructs a sampling and grouping scheme to learn hierarchical features at increasing spatial scales. Because point clouds are irregular and points in point clouds are continuous in space, making fixed-grid convolution in 2D cannot be used directly on point clouds. To address this, KPConv (Thomas et al., 2019) proposes using kernel points as convolutional filters and operating on points without transformation. The weights of convolution are learned from kernel points and their neighbours in the Euclidean space. On the other hand, GCNs can naturally extract geometric information from their surroundings. For example, EdgeConv (Wang et al., 2019) enhances semantic segmentation performance via graph convolution, which performs a convolution operation on the edges that connect points and their neighbours. Thus, the correlation between points and their K-nearest neighbours can be computed by operating the convolution on the edges of the constructed local graph. The features of input point clouds are then updated by aggregating edge features together through a local max-pooling layer.

In regard to the building domain, the *deep competition* network (Khoshboresh-Masouleh et al., 2019) uses encoder-decoder blocks to extract robust super-pixel representation from multiple building scenes. Then five blocks are stacked together to extract the building footprint from the building's LiDAR point clouds. Pierdicca et al. (2020) proposes employing DGCNN (Wang et al., 2019) to perform the point cloud semantic segmentation task applied to an architectural cultural heritage dataset, the ArCH Dataset (Matrone et al., 2020). In 3DLEB-Net (Cao and Scaioni, 2021a), they proposed a two-step label-efficient DL-based network to obtain per-point semantic labels for the point clouds of LoD3 buildings. Specifically, 3DLEB-Net first utilises an Autoencoder (AE) to learn discriminative representations by reconstructing the input unlabelled point clouds. Then, the learned representations are used as the inputs of the classifier in the second step, thus decreasing the demand for a large amount of labelled data in conventional DL methods.

2.2 Pre-training Methods

Several techniques have been developed to address the lack of data and finely-annotated label problems in the 3D domain. In the 2D image (He et al., 2019) and natural language processing (NLP) (Devlin et al., 2018) domains, pre-training is one of the critical components in today's innovative approaches. Pre-training is defined as training on a typically large-source dataset and then transferring the acquired prior knowledge to a downstream task and fine-tuning it on a smaller target dataset to improve the performance of the network. This notion motivated the idea of using pre-training methods to improve the performance of DNNs in architectural domains where data and labels are scarce.

Existing pre-training methods attempt to extract robust features from enormous amounts of unlabelled data by designing unsupervised or self-supervised pretext tasks. For example, OcCo (Wang et al., 2021) trained a point cloud completion model to reconstruct the occluded point clouds from the given portion of the observed data. Using the contrastive loss to capture point-level correspondences, both PointContrast (Xie et al., 2020) and DepthContrast (Zhang et al., 2021) demonstrate that joint pre-

training can improve the performance of DNNs. On the LoD3 buildings, 3DLEB-Net (Cao and Scaioni, 2021a, 2021b) utilised an Autoencoder (AE) based unsupervised learning method. It consists of two steps. The initial step is to create an AE-based point cloud reconstruction task that will extract discriminative features from unlabelled input point clouds. The weights of the pre-trained AE are then used as initialisation for the semantic segmentation network in the second step. In this way, these unsupervised learning methods can learn a pre-trained model able to identify the visual constraints inherent in real-world point clouds.

The findings of these methods indicate that the pre-training method is one of the potential solutions to the lack of labels issue. However, the approaches outlined above require the creation of pretext tasks and the corresponding datasets (e.g., half point clouds, augmented point clouds), and then the learned knowledge or model weights are used for downstream tasks. With this in mind, this paper presents a "learn-to-distinguish" method that attempts to directly transfer the capabilities of semantic segmentation acquired by training fully supervised learning methods. To be more specific, we first train on a cross-domain but existing and annotated source dataset to learn the ability of semantic segmentation using four different fully supervised learning methods. Then, we transfer the prior knowledge obtained in the first step to our target dataset by using the pre-trained weights as the initialisation of the downstream network. Finally, fine-tuning is performed on the labelled target dataset.

3. METHOD

3.1 Overview

In this paper, we propose applying a pre-training method to enhance the semantic segmentation performance of existing fully supervised methods when labels are insufficient. Deep learning networks (DLNs) can learn prior knowledge about the ability to distinguish between different objects from a source dataset. We believe this can be transferred to the target dataset. For example, we used an indoor dataset as the source dataset and an outdoor dataset as the target dataset in this study. To validate such a hypothesis, this paper compares the results obtained from two pipelines with and without the pre-training method (training from scratch). As shown in Figure 1a, we use the ArCH Dataset directly in the first pipeline to train different fully supervised learning methods. Our pre-training pipeline is straightforward but effective, requiring only three steps, as illustrated in Figure 1b. We begin by training four fully supervised methods on the source indoor dataset (S3DIS). And then we use the pre-trained networks as an initialisation of the downstream network. Finally, fine-tuning is performed on the target ArCH Dataset.

3.2 Backbone DL Methods

Diverse types of state-of-the-art DLNs as our backbones are selected to assess the effectiveness of the proposed method and its adaptability to different DNN architectures, including two MLP-based networks – PointNet (Qi et al., 2017a) and PointNet++ (Qi et al., 2017b), a CNN-based network – KPConv (Thomas et al., 2019), and a GCN network– DGCNN (Wang et al., 2019). For more information, please refer to the source papers.

3.2.1 PointNet and PointNet++: Both PointNet and PointNet++ are proposed by Qi et al. (2017a, 2017b). As illustrated in Figure 2a, to directly manipulate unordered point clouds, PointNet proposes to leverage the MLP to learn high-dimensional features for each point individually. Afterwards, pointwise features are aggregated into a global feature through a symmetric function called max pooling, which solves the unordered inherent characteristic of point clouds. Due to the fact that the features of each point in the point cloud are learned independently in PointNet, the local context information between points is neglected. Taking this into account, PointNet++ adds relations between points in local regions. Specifically, it stacked several Set Abstraction (illustrated in Figure 2b) blocks to hierarchically sample, group, and extract fine geometric structures from the neighbours of each point through the sampling layers, grouping layers, and PointNet layers.

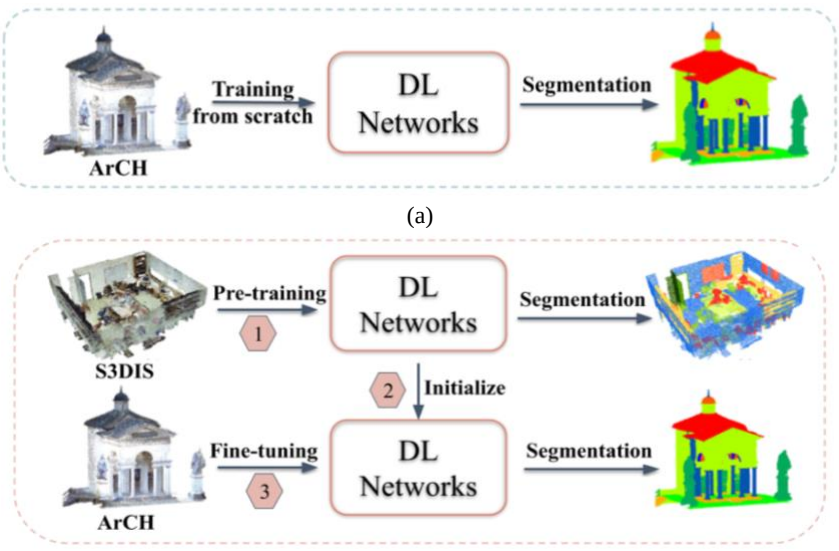


Figure 1. An illustration of the proposed pre-training method for building point cloud semantic segmentation. Our approach compares two pipelines: the top row represents the training from the scratch pipeline, while the bottom row represents the three steps of training with the proposed pre-training method pipeline: Pre-training with the Stanford 3D Indoor Scene Dataset (S3DIS – Armeni et al., 2016); using the pre-trained network to initialise the downstream task; and fine-tuning with the target Architecture Cultural Heritage Dataset (ArCH – Matrone et al, 2020).

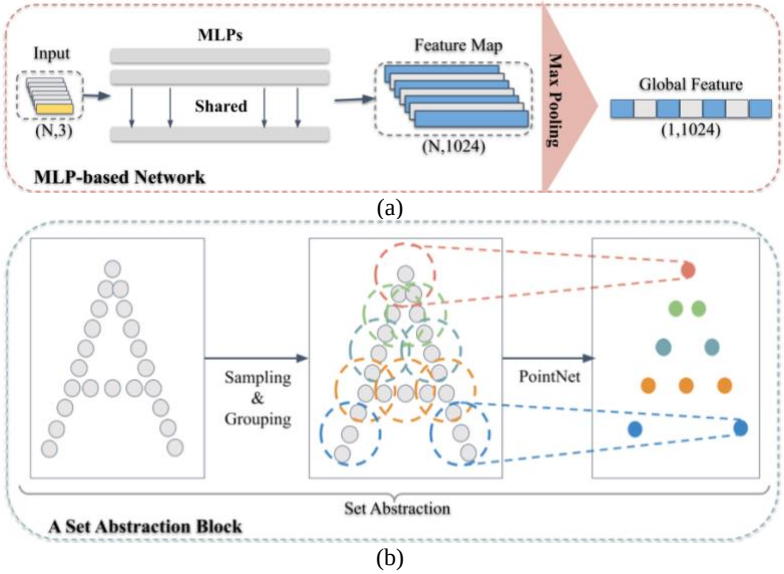


Figure 2. An illustration of PointNet (Qi et al., 2017a) and PointNet++ (Qi et al., 2017b). Top: a demonstration of how to use multilayer perceptron (MLP) layers to learn pointwise features and utilise a max pooling layer to learn and aggregate the global features in PointNet. Bottom: a 2D explanation of how to use a Set Abstract block in PointNet++ to sample, group, and extract local geometric information from input points and their neighbours.

3.2.2 KPConv: CNNs are utilised in the majority of state-of-the-art 2D DLNs for the purpose of understanding 2D images. As seen in Figure 3(a), an image is a 2D grid (5×5) of pixels in which each pixel is allocated three channels representing three colour values (RGB). Furthermore, the distance between adjacent pixels is always constant, so that the 2D convolution kernel can be designed to fit fixed-size grids (3×3). 3D point clouds, on the other hand, are unstructured since each point in a point cloud is sampled independently, and its distance to its neighbours may vary. This makes designing the convolutional kernels for 3D point clouds difficult and prevents us from using the kernels of 2D CNNs directly on 3D point clouds.

KPConv (Thomas, 2019) overcomes these challenges by defining convolutions in continuous 3D space, where the weights for neighbouring points are proportional to the spatial distribution of the centroid point in the Euclidean space. Each pixel in 2D CNN is represented by a list of values referred to as RGB channels and is processed through k filters. As seen in Figure 3(a), the new representation of the pixel is the dot product of the channels of adjacent pixels by the filters. Similarly, each point in a KPConv layer has f_i features (e.g., coordinates xyz, RGB values, and intensity) that correspond to channels in 2D images and is multiplied by k kernel points (similar to filters), as seen in Figure 3b. The new representation of the point equals the sum of all the kernel values multiplied by the features of its neighbours and itself. The neighbourhood of a point is defined by all the points that are located in a sphere with a fixed radius around that point.

3.2.3 DGCNN: The GCN treats each point in a point cloud as a node of a graph, with edges formed by the relations between the neighbours of each point. As illustrated in Figure 4a, a typical graph-based network initially constructs a graph from input points and connected edges. We use DGCNN (Wang et al., 2019) as one of our backbones, which is one of the state-of-the-art GCNs. As depicted in Figure 4b, the EdgeConv layers in DGCNN are used as the representation learning functions for each graph edge. The edge function of EdgeConv captures the global shape by encoding the coordinates of p_i and then obtains the local information by encoding $p_j - p_i$. The output feature inside each local region is aggregated by a local max-pooling operation on the edge features from each connected vertex and itself in the constructed graph. This enables DGCNN to capture local geometric features and global information inside each local region. Furthermore, by dynamically constructing a graph in each layer and stacking several EdgeConv layers, the receptive field becomes larger, and information is aggregated in different receptive fields. Thus, DGCNN can combine the global shape structure information with hierarchical local neighbourhood information.

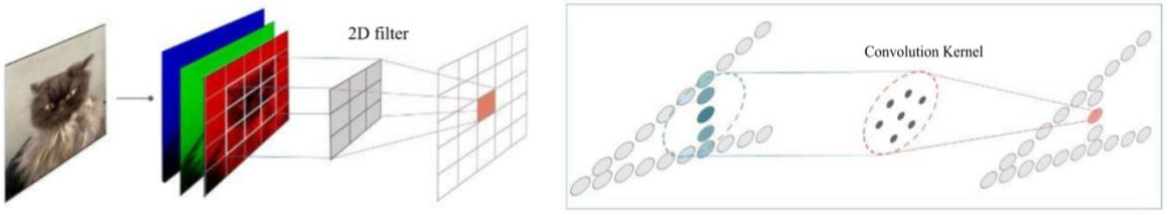
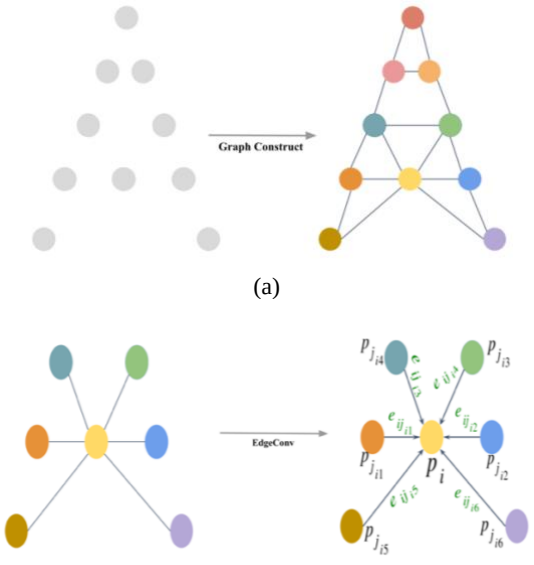


Figure 3. The simplified 2D illustration of convolution. Left: the use of a 2D filter on an input image. Right: using a 3D rigid convolution kernel on the input point cloud, where the weights for neighbouring points are proportional to the spatial distribution of the centroid point in Euclidean space, shown in the gradient colours in the neighbour points of the centroid.



(a)

(b)

Figure 4. An illustration of the graph construction process and edge features' learning process in a local region on the contrasted graph.

4. EXPERIMENT

4.1 Datasets

Pre-training approaches can use either in-domain or cross-domain datasets as the source and target data. We use the Stanford 3D Indoor Scene Dataset (S3DIS) as the source dataset and the Architectural Cultural Heritage Dataset (ArCH) as the target dataset to prove our hypothesis that the semantic segmentation capabilities learned from pre-training can be transferred to the target dataset while using a rich source dataset as input to the pre-trained network, even if both datasets are cross-domain. We will introduce the aforementioned datasets in this subsection.

4.1.1 S3DIS Dataset: This dataset consists of a large-scale indoor environment including six indoor areas with 271 rooms for a total of 695 million points. These rooms cover office areas, educational and exhibition spaces, conference rooms, personal offices, restrooms, open spaces, lobbies, stairways, and hallways. Each point in the scene point cloud is annotated with one of the 13 semantic categories, which are structural elements (ceiling, floor, wall, beam, column, window, and door), furniture (table, chair, sofa, bookcase, and board) and clutter for all other elements.

4.1.2 ArCH Dataset: This dataset consists of seventeen classified indoor and outdoor scenes, fifteen for training and two for testing purposes, respectively. It includes a variety of building scenes, including churches, chapels, cloisters, pavilions, squares, and porticoes. Each point in the scene point cloud is labelled with one of the 10 semantic categories with a level of semantic detail at LoD3, including the "arch," "column," "mouldings," "floor," "door_window," "wall," "stairs," "vault," "roof," and "others" categories for remaining elements.

The target ArCH Dataset is relatively small and contains both outdoor and indoor point clouds representing cultural heritage building. The content of this dataset, to be used as target, is significantly different from the source dataset (S3DIS). Therefore, we compare the discrepancies in Table 1. As we can see, the number of points ("N. of Points") in the S3DIS Dataset overcome the one of the target ArCH Dataset by a factor of five. Secondly, while both datasets are in the architectural domain, the S3DIS Dataset is a large-scale dataset for the indoor environment. In contrast, the ArCH Dataset primarily contains the exteriors of cultural heritage buildings, resulting in considerable differences in the semantic labelling ("Categories") of the points in their point clouds. Finally, the acquisition methods are different in the two datasets, which may produce domain gaps.

Dataset	ArCH	S3DIS
Type	Outdoor & Indoor	Indoor
N. of Points	136,138,423	695,878,620
Categories	"arch", "column", "mouldings", "floor", "door_window", "wall", "stairs", "vault", "roof", and "others"	"ceiling", "floor," "wall", "beam", "column", "window", "door", "table", "chair", "sofa", "bookcase", "board", and "cluster"
Acquisition method	TLS + UAV + Terrestrial photogrammetry	Matterport Camera

Table 1. Comparison of two datasets: The ArCH Dataset and the S3DIS Dataset. The "N. of Points" column denotes the total number of points in each dataset, except for the points classified as "others" in the ArCH Dataset.

4.2 Experiment Settings

The 3D scenes in the ArCH Dataset are too large to be used as input to the network, and therefore need to be segmented before training. Specifically, we chose a block size of 1 × 1 square-metre area for splitting each building scene into the horizontal blocks for PointNet, PointNet++, and DGCNN. In addition, the points in each block are subsampled into a uniform number of 2,048 points. In particular, the point clouds are segmented in KPConv (Thomas et al., 2019) using spheres (the sphere radius is chosen to be 50 × 4 cm) in accordance with the original processing of the data. During training, spheres are randomly chosen from the scene. During testing, the spheres are picked regularly in the testing point cloud to ensure that each point is tested multiple times at different sphere locations. For the S3DIS Dataset, this paper follows the same configuration as the original networks, see the papers on PointNet, PointNet++, KPConv, and DGCNN for details (Qi et al., 2017a; Qi et al., 2017b; Thomas et al., 2019; Wang et al., 2019).

In the experiments, different proportions of the ArCH Dataset are used as training data to evaluate the model's performance when labelled data is scarce. Specifically, we select one scene ("SMV_chapel_28"), three scenes (namely, "SMV_chapel_1", "SMV_chapel_24", and "SMV_chapel_28"), and all scenes from the fifteen labelled scenes, representing 4%, 10%, and 100% of the total number of points, respectively, as training data. We use one unseen scene ("B_SMV_chapel_27to35") as our test data. As shown in Figure 5, the test data represents a complicated and asymmetrical building that includes both in the indoor and outdoor (Matrone et al., 2020).

While training the S3DIS Dataset and the ArCH Dataset from scratch, the data augmentation strategies, the hidden layer sizes, and training parameters are kept constant across different

networks according to the original respective settings. During fine-tuning, we also maintain the hyper-parameters like learning rate and optimizing strategy unchanged. We set the training time for fine-tuning to 100 epochs. There are only two distinctions during training:

- The batch sizes in different backbones are accordingly altered to 4 to fit our computational resources, which may limit the performance;
- Since the linear function of the last layer in the network takes the possibility of each point being classified into each category as output, the last layer's size corresponds to the number of categories contained in the dataset. The categories of S3DIS and ArCH are different (13 *vs.* 10), and therefore the size of this layer is different in the pre-trained networks and the downstream networks. Due to this reason, the pre-trained weights are not transferred in this layer.

The Mean Intersection-over-Union (mIoU) evaluation matrix is calculated on the ArCH Dataset, which first computes the ratio between the intersection of the pointwise classification results with the ground truth to their union for each semantic class, and then computes the average over all classes.

(a) (b)

Figure 5. Pictures of the test scene from ArCH Dataset. Left: "B_SMV_chapel_27to35 south side". Right: "B_SMV_chapel_27to35 indoor part" Copyright © 2022 ArCH Dataset).

4.3 Results

We train four DL-based methods using different subsets of training data and compared the results of the two pipelines described above on the ArCH Dataset. Firstly, we train four DL-based methods from scratch on one selected scene, three selected scenes, and all scenes in the ArCH Dataset. Then, the pre-training method is applied to four DL-based methods using the labelled S3DIS Dataset. Finally, the pre-trained models are fine-tuned using different portions of the ArCH Dataset, which is the same as the training from the scratch pipeline.

Table 2 summarises the mIoU performances of the two pipelines on the ArCH Dataset. By comparison, we find that the pre-training method always performs better. The results show that prior knowledge learned from the cross-domain S3DIS Dataset can be used to reliably improve the performance of the target architectural dataset, with an average gain of 3.9%. Furthermore, we observe that in the case of training DL-based models using only three scenes (approx. 10% of the total number of points of all scenes) and utilising the proposed pre-training approach, the models obtain comparable results to those training from scratch using all scenes in the Arch Dataset (0.395 *vs.* 0.325).

The qualitative result of the DGCNN's semantic segmentation is represented in Figure 6 - 8 when different pipelines and different sections of training data are used. As can be seen, our proposed pre-training approach produces more accurate results than training from scratch.

N. of Scenes	Models	Scratch (mIoU)	Pre-training (mIoU)	Gain (%)
One Scene	PointNet	0.165	0.224	5.867
	PointNet2	0.207	0.246	3.949
	KPConv	0.430	0.456	2.589
	DGCNN	0.290	0.307	1.757
Three Scenes	PointNet	0.226	0.235	0.909
	PointNet2	0.246	0.288	4.175
	KPConv	0.589	0.642	5.300
	DGCNN	0.331	0.395	6.389
All Scenes	PointNet	0.267	0.309	4.148
	PointNet2	0.269	0.237	-3.204
	KPConv	0.611	0.645	3.411
	DGCNN	0.357	0.472	11.492

Table 2. Comparing two pipelines while utilising distinct deep learning (DL) pipelines and various subsets of training data, using the ArCH Dataset as a benchmark. "N. of Scenes" denotes the used scenes in the DL training stage.

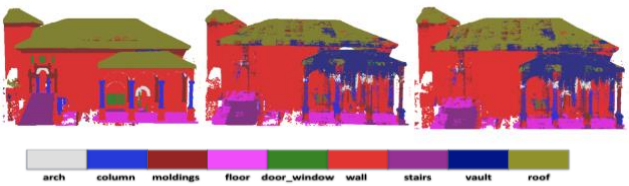


Figure 6. Qualitative outcomes for semantic segmentation of "B_SMV_chapel_27to35" while training with a single scene from the ArCH Dataset. From left to right: ground truth (a), the prediction of training from the scratch pipeline (b), and the result obtained by the pre-training pipeline (c).

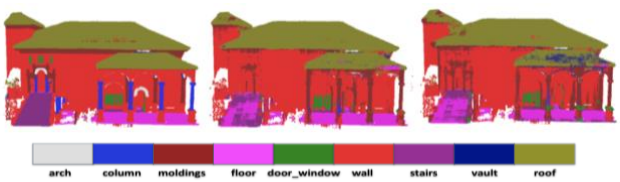


Figure 7. Qualitative outcomes for semantic segmentation of "B SMV chapel 27to35" while training with three scenes from the ArCH Dataset. The picture in each column represents the same meaning as each column of Figure 6.

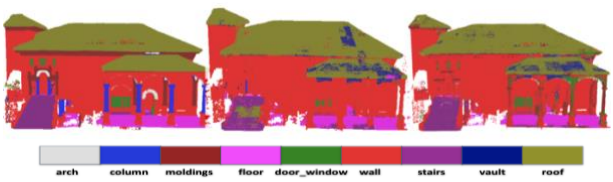


Figure 8. Qualitative outcomes for semantic segmentation of "B SMV chapel 27to35" while training with all scenes from the ArCH Dataset. The picture in each column represents the same meaning as each column of Figure 6.

	mIoU	arch	column	mouldings	floor	door_window	wall	stairs	vault	roof
Scratch	58.9	19.0	75.0	26.5	81.3	42.5	75.1	70.4	59.4	81.0
Pre-training	64.2	30.1	85.5	29.1	87.5	40.6	81.0	76.2	63.0	84.8
Gain	5.3	11.1	10.5	2.6	6.2	-1.9	5.9	5.8	3.6	3.8

Table 3. Comparing the per-category prediction results of two pipelines utilising KPConv as the backbone and three scenes from the ArCH Dataset as training data.

5. DISCUSSION

The source S3DIS Dataset is an indoor dataset. In contrast, the target ArCH Dataset is primarily the exterior of a cultural heritage building with three overlapping objects in their point clouds – "columns", "floors", and "walls" - while the other categories do not overlap. Table 3 provides the semantic segmentation results for the training from scratch and pre-training pipelines in each category, using KPConv as the backbone and the three scenes as training data. We can see that the improvement in the overlapping categories is more significant than the average improvement. Also, for the non-overlapping categories, our performance still improves. The result supports our hypothesis that pre-training can learn from distinct categories. It is worth noting that the drop in performance for the "door_window" category (-1.9%) may be explained by the fact that in the source dataset, both categories were separately labelled. However, they are combined in the target dataset, considering the small number of point clouds in both of them (Matrone et al., 2020). This conflict between labels significantly interferes with the predictions, with a resulting drop in performance.

Compared to the source dataset, the target dataset is smaller and has a different structure. Consequently, learning rates should be adjusted. We keep the learning rate the same because: (1) all backbones use adaptive optimizers; and (2) we use a smaller batch size to fit the memory size and provide implicit regularization for the models by adding noise to convergence.

6. CONCLUSION AND FUTURE DEVELOPMENT

This paper presents a straightforward and effective pre-training approach for 3D building point cloud semantic segmentation. To be more specific, our paper conducted solid experiments to evaluate the proposed method by comparing the semantic segmentation results of a 3D building point cloud from two pipelines: training from scratch *vs.* fine-tuning. The results show that pre-training a network on a large source dataset might consistently enhance performance when it is fine-tuned on a typically much smaller target dataset. We also observed that using three scenes from the ArCH dataset to fine-tune the pre-trained network has gained comparable results to training from scratch using all scenes, which suggests that pre-training is also beneficial to the data and label-scarce 3D building point cloud domain.

In future work, we will extend our work in several aspects: (1) different source datasets like ScanNet (Dai et al., 2017) and different amounts of training data in the pre-training stage could be involved to distinguish the impact of using different source datasets; (2) exploring hyperparameter settings such as learning rate, augmentation strategies, and layers for fine-tuning using control experiments; and (3) comparing the approaches that learn from pretext tasks (e.g., contrastive learning) with the proposed pre-training method.

ACKNOWLEDGEMENTS

Financial support from the programme of the China Scholarships Council (Grant No. 201906860014) is acknowledged. We thank the two anonymous reviewers for their constructive comments, which helped improve this paper. We thank Dr. F. Matrone et al. for the ArCH Dataset and Dr. I. Armeni et al. for the S3DIS Dataset.

REFERENCES

Aoki, Y., Goforth, H., Srivatsan, R.A., Lucey, S., 2019. Pointnetlk: Robust & Efficient Point Cloud Registration Using Pointnet. In 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 15-20 June, Long Beach (CA - USA), pp. 7163-7172. doi: 10.1109/CVPR.2019.00733.

Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S., 2016. 3d Semantic Parsing of Large-Scale Indoor Spaces. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 27-30 June, Las Vegas (NV, USA), pp. 1534-1543. doi: 10.1109/CVPR.2016.170.

Begić, H., Galić, M., 2021. A Systematic Review of Construction 4.0 in the Context of the BIM 4.0 Premise. *Buildings*., 11(8), 337. doi: 10.3390/buildings11080337.

Cao, Y., Scaioni, M., 2021a. 3DLEB-Net: Label-Efficient Deep Learning-Based Semantic Segmentation of Building Point Clouds at LoD3 Level. *Appl. Sci*., 11, 8996. doi: 10.3390/app11198996.

Cao, Y., Scaioni, M., 2021b. Label-Efficient Deep Learning-Based Semantic Segmentation of Building Point Clouds at LoD3 Level. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLIII-B2-2021, 449–456. doi: 10.5194/isprs-archives-XLIII-B2-2021-449-2021.

Czerniawski, T., Leite, F. 2020. Automated Digital Modelling of Existing Buildings: A Review of Visual Object Recognition Methods. *Autom. Constr*., 113, 103-131. doi: 10.1016/j.autcon.2020.103131.

Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T. and Nießner, M., 2017. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 21-26 July, Honolulu (HI, USA), pp. 5828-5839. doi: 10.1109/CVPR.2017.261.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L., 2009. ImageNet: A Large-Scale Hierarchical Image Database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 20-25 June, pp. 248-255. doi: 10.1109/CVPR.2009.5206848.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

Dukai, B., Peters, R., Wu, T., Commandeur, T., Ledoux, H., Baving, T., Post, M., van Altena, V., van Hinsbergh, W., Stoter, J., 2020. Generating, Storing, Updating and Disseminating a Countrywide 3D Model. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci*., XLIV-4/W1-2020, 27–32. doi: 10.5194/isprs-archives-XLIV-4-W1-2020-27-2020.

Ham, Y., Golparvar-Fard, M., 2015. Three-Dimensional Thermography-Based Method for Cost-Benefit Analysis of Energy Efficiency Building Envelope Retrofits. *J. Comput. Civ. Eng.*, 29, B4014009. doi: 10.1061/(ASCE)CP.1943-5487.0000406.

He, K., Girshick, R., Dollár, P., 2019. Rethinking ImageNet Pre-Training. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 27 Oct. – 2 Nov., Seoul, Korea (South), pp. 4917-4926. doi: 10.1109/ICCV.2019.00502.

Hong, T., Langevin, J., Sun, K., 2018. Building Simulation: Ten Challenges. *Build. Simul*., 11, 871–898. doi: 10.1007/s12273-018-0444-x.

Joseph-Rivlin, M., Zvirin, A., & Kimmel, R. (2019). Momen^et: Flavour the Moments in Learning to Classify Shapes. In 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 27-28 Oct., Seoul, Korea (South), pp. 4085-4094. doi: 10.1109/ICCVW.2019.00503.

Khoshboresh-Masouleh, M., Saradjian, M. R., 2019. Robust Building Footprint Extraction from Big Multi-Sensor Data Using Deep Competition Network. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci*., XLII-4/W18, 615–621. doi: 10.5194/isprs-archives-XLII-4-W18-615-2019.

Landrieu, L., Simonovsky, M., 2018. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. In 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 18-23 June, Salt Lake City (UT, USA), pp. 4558-4567. doi: 10.1109/CVPR.2018.00479.

Lu, W., Wan, G., Zhou, Y., Fu, X., Yuan, P., Song, S., 2019. Deepvcp: An End-to-end Deep Neural Network for Point Cloud Registration. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 27 Oct. – 2 Nov., Seoul, Korea (South), pp. 12-21. doi: 10.1109/ICCV.2019.00010.

Matrone, F., Lingua, A., Pierdicca, R., Malinverni, E.S., Paolanti, M., Grilli, E., Remondino, F., Murtiyoso, A., Landes, T., 2020. A Benchmark for Large-scale Heritage Point Cloud Semantic Segmentation. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLIII-B2-2020, 1419–1426. doi: 10.5194/isprs-archives-XLIII-B2-2020-1419-2020.

Pierdicca, R., Paolanti, M., Matrone, F., Martini, M., Morbidoni, C., Malinverni, E.S., Frontoni, E., Lingua, A.M., 2020. Point Cloud Semantic Segmentation Using a Deep Learning Framework for Cultural Heritage. *Remote Sens*., 12, 1005. doi: 10.3390/rs12061005.

Previtali, M., Díaz-Vilariño, L., Scaioni, M., 2018. Indoor Building Reconstruction from Occluded Point Clouds Using Graph-cut and Ray-tracing. *Appl. Sci.*, 8 (9), 1529. doi: 10.3390/app8091529.

Qi, C.R., Su, H., Mo, K., Gibus, L.J., 2017a. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 21-26 July, Honolulu (HI, USA), pp. 77-85. doi: 10.1109/CVPR.2017.16.

Qi, C.R., Yi, L., Su, H. and Guibas, L.J., 2017b. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In Proceedings of the 31st Int. Conf. on Neural Information Processing Systems (NIPS), Dec, Long Beach (CA - USA), 5105-5114.

Ruohomäki T., Airaksinen E., Huuska P., Kesäniemi O., Martikka M., and Suomisto J., 2018. Smart City Platform Enabling Digital Twin. In 2018 International Conference on Intelligent Systems (IS), 25-27 Sept., Funchal Portugal, pp. 155-161. doi: 10.1109/IS.2018.8710517.

Sánchez-Aparicio, L.J., Del Pozo, S., Ramos, L.F., Arce, A., Fernandes, F., 2018. Heritage Site Preservation with Combined Radiometric and Geometric Analysis of TLS data. *Autom. Constr*., 85, 24–39. doi: 10.1016/j.autcon.2017.09.023.

Teruggi, S., Grilli, E., Russo, M., Fassi, F. and Remondino, F., 2020. A Hierarchical Machine Learning Approach for Multi-Level and Multi-Resolution 3D Point Cloud Classification. *Remote Sens*, 12(16), 2598. doi: 10.3390/rs12162598.

Thabet A., Alwassel H., Ghanem B., 2020. Self-Supervised Learning of Local Features in 3D Point Clouds. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 14-19 June, Seattle (WA, USA), pp. 4048-4052. doi: 10.1109/CVPRW50498.2020.00477.

Thomas, H., Qi, C. R., Deschaud, J. E., Marcotegui, B., Goulette, F., Guibas, L. J., 2019. Kpconv: Flexible and Deformable Convolution for Point Clouds. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 27 Oct. – 2 Nov., Seoul, Korea (South), pp. 6411-6420. doi: 10.1109/ICCV.2019.00651.

Wang, H., Liu, Q., Yue, X., Lasenby, J., Kusner, M. J., 2021. OcCo: Unsupervised Point Cloud Pre-Training via View-Point Occlusion Completion. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 11-17 Oct., Montreal, Canada, pp. 9782-9792. doi: 10.1109/ICCV48922.2021.00964.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019. Dynamic Graph CNN for Learning on Point Clouds. *Acm Trans. Graph. Tog*., 38, 1–12. doi:10.1145/3326362.

Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L. and Litany, O., 2020. PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding. In 2020 European Conference on Computer Vision (ECCV), 23–28 Aug., Glasgow, UK, pp. 574-591. doi: 10.1007/978-3-030-58580-8_34.

Zhang, Z., Girdhar, R., Joulin, A. and Misra, I., 2021. Self-supervised Pretraining of 3D Features on Any Point Cloud. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 11-17 Oct., Montreal, Canada, pp. 10252-10263. doi: 10.1109/IC.