LABEL-EFFICIENT DEEP LEARNING-BASED SEMANTIC SEGMENTATION OF BUILDING POINT CLOUDS AT LOD3 LEVEL

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 Point Cloud, Autoencoder, Label-efficient, LoD3 Building, Unsupervised Deep Learning

ABSTRACT:

In recent research, fully supervised Deep Learning (DL) techniques and large amounts of pointwise labels are employed to train a segmentation network to be applied to buildings' point clouds. However, fine-labelled buildings' point clouds are hard to find and manually annotating pointwise labels is time-consuming and expensive. Consequently, the application of fully supervised DL for semantic segmentation of buildings' point clouds at LoD3 level is severely limited. To address this issue, we propose a novel label-efficient DL network that obtains per-point semantic labels of LoD3 buildings' point clouds with limited supervision. In general, it consists of two steps. The first step (Autoencoder - AE) is composed of a Dynamic Graph Convolutional Neural Network-based encoder and a folding-based decoder, designed to extract discriminative global and local features from input point clouds by reconstructing them without any label. The second step is semantic segmentation. By supplying a small amount of task-specific supervision, a segmentation network is proposed for semantically segmenting the encoded features acquired from the pre-trained AE. Experimentally, we evaluate our approach based on the ArCH dataset. Compared to the fully supervised DL methods, we find that our model achieved state-of-the-art results on the unseen scenes, with only 10% of labelled training data from fully supervised methods as input.

1. INTRODUCTION

In recent years, 3D buildings' point cloud representation enables and promotes new applications in many fields such as Cultural Heritage preservation (Pierdicca et al., 2020), Construction Engineering (Ham, Golparvar-Fard, 2015), Emergency Decision-making (Fazeli et al., 2016), and Smart Cities (Hu et al., 2018). However, point clouds of buildings generally provide the representation of the entire building including only a few types of architectural elements with no semantic information, limiting the efficient exploitation in the abovementioned application domains (Czerniawski, Leite, 2020). Hence, it's essential to investigate the methods of extracting semantic information from 3D buildings' point clouds to acquire high Level-of-Details (LoDs) modelling, see Wang and Kim (2019).

LiDAR data sets have become available at an even growing resolution and accuracy. Inspired by the success of *deep neural networks* (DNNs) used in Computer Vision to accomplish subset tasks (i.e., classification, detection and semantic segmentation), Deep Learning (DL) approaches have appeared in the last few years for understanding 3D point clouds (Cao et al., 2020). In the buildings' point cloud domain, DL techniques also played an essential role in numerous applications, such as indoor (Wang et al., 2018), urban (Kumar et al., 2019) and buildings' scenes (Huang et al., 2019) analysis. Even though important results were achieved, the existing DL approaches for 3D building point clouds are strongly supervised, and these methods have substantial demands for finely labelled data, see Meng et al. (2020).

However, it is not feasible to create such an amount of labelled training data in many real-world problems. For example, to the best of our knowledge, only the Architectural Cultural Heritage (ArCH) Data Set (Matrone et al., 2020a) is publicly available to provide pointwise annotations and support for generating high-resolution LoD3 building models. Furthermore, billions of pointwise accurate labels are demanded to train a satisfactory segmentation network (Meng et al., 2020), which can be obtained from extremely time-consuming and expensive processes. For instance, only after the setup of the fine-labelled ArCH Data Set some studies addressed the application of DL into architectural semantic segmentation, see Matrone et al. (2020b) and Pierdicca et al. (2020).

In Computer Vision, the hunger for fine-labelled pointwise training data problems is often tackled by using unsupervised methods. However, these approaches are mostly designed for 2D images, which are fundamentally different from unordered point clouds. Unlike 2D images that are projective observations from the built environment, 3D point clouds provide a metric reconstruction of the scenes without scale ambiguity (Han, 2021). Furthermore, the application of label-efficient unsupervised learning to downstream tasks in the 3D field is still limited to classification and segmentation tasks of small-scale point clouds. From a scientific viewpoint, the unsupervised DL-based buildings' point clouds semantic segmentation is still an open issue, and current knowledge about it is deeply unsatisfactory.

For the abovementioned reasons, we decided to put our efforts in developing an unsupervised DL method for buildings' point clouds semantic segmentation. We explored the possibility of learning a point cloud segmentation network by only supplying limited task-specific labelled buildings' point cloud. We relied on the state-of-the-art Dynamic Graph Convolutional Neural Network (DGCNN) and FoldingNet to learn point cloud features (Wang et al., 2019; Yang et al., 2018). We chose the Autoencoder (AE) architecture to simultaneously learn reconstruction and

* Corresponding author

discriminative features of the input 3D buildings' point cloud. To achieve this, we leveraged the DGCNN as our *encoder* and *decoder* of FoldingNet as our decoder to acquire powerful embeddings without any labelled data. To this end, with limited labelled point cloud data, we designed three fully connected layers in the end-to-end segmentation network to achieve the downstream segmentation task.

In particular, our new contributions can be summarized as follows:

1. We propose an unsupervised AE network to learn powerful features from non-labelled building datasets; and
2. We train an end-to-end segmentation network for the buildings' segmentation task. The output of our model is a semantically enriched LoD3 3D building representation; and
3. We experimentally demonstrate how to exploit limited labelled point clouds and the features learned from pre-trained AE to segment the input data.

2. RELATED WORK

Laser scanning techniques are able to collect point clouds of buildings. At the same time, the massive amount of data requires a semantic interpretation at a high Level-of-Details (LoDs) in order to increase the exploitation of these data sets (Previtali et al., 2018; Griffiths, Böhm, 2019). While several types of fully supervised *Deep Neural Networks* (DNNs) are continuously developed and improved in 3D point clouds analysis tasks, fine-grained labels are always required in model training processes. These include pointwise labels, shape class labels for semantic segmentation task, and part-segmentation task. Thus, DNNs' application to LoD3 buildings' point clouds has been limited. This question has sparked the interest about this problem. Several unsupervised approaches have emerged to tackle the scarcity of labelled data.

In this section, the state-of-the-art methods to figure out the possibility of unsupervised DL methods applied to buildings' point clouds is dealt with. As there are only a handful of papers that focus on DL techniques on buildings' point cloud segmentation task, we will not limit our review on the application to this category of objects. We review these approaches from two aspects as follows:

1. Fully supervised methods on 3D point clouds; and
2. Label-efficient unsupervised methods on 3D point clouds.

2.1 Fully Supervised Methods on 3D Point Clouds

3D buildings' point clouds generally represent complex geometric structures, where semantic content is not directly included. Therefore, semantic segmentation of it is still a challenging task. In traditional *data-driven* approaches (Forlani et al., 2006; Verma et al., 2006), points with some notions of similarity are clustered together to map point clouds into classes by constructing feature descriptors (e.g., verticality, planarity and elevation). In conventional *model-fitting* approaches (Haala et al., 1998; Maas, Vosselman, 1999; Chen et al., 2014), some geometric models are sought to detect specific objects, such as houses, roofs, trees, etc. Despite the impressive performances from these traditional approaches, models or geometric descriptors cannot interpret the complexity of real data. Moreover, conventional semantic segmentation approaches heavily rely on hand-crafted features, making the generalization difficult. Their efficient application to obtain high LoD models still remains quite challenging.

Due to these reasons, the chance of using DNNs to effectively and automatically extract features in an end-to-end fashion, gives rise to the application of these promising methods for semantic segmentation of point clouds of 3D buildings. Based on DNN input data format, existing point clouds semantic segmentation methods can be grouped into *direct* and *indirect* methods. The latter usually first partition the 3D space into *regular image representations* (Su et al., 2015) or *voxels* (Wang et al., 2017) data structures, to take advantage of well-established 2D/3D DL networks for feature learning and semantic segmentation. However, due to point clouds' inherent nature, transfer to another intermediate representation would result in quantization error and inefficiency, see Qi et al. (2017a). In contrast, direct methods do not introduce explicit information loss, as reported in Guo et al. (2020). In 2017, the pioneering direct method PointNet (Qi et al., 2017a) was proposed. PointNet directly operates on point clouds, using the Multilayer Perceptron (MLP) to learn high-dimensional features for each point independently. Subsequently, pointwise features are stacked to a global feature. Since pointwise features are learned individually from each point in PointNet, the local context information between points is ignored. Dynamic Graph Convolutional Neural Network (DGCNN) improves the performance of segmentation by considering the relations between points in the local neighbourhood and by aggregating them into a global feature in the EdgeConv Module (Wang et al., 2019), which can be plugged into existing architectures. Since current state-of-the-art methods have shown that aggregating local and global information may increase the network's capabilities of capturing context information, our paper will exploit the feature extraction power of EdgeConv modules in our encoder, which directly consumes points and incorporates the local neighbour information obtained from the point cloud.

DL techniques for 3D point cloud segmentation have been successfully applied in recent years, while the development in the built environment domain has just started to be explored. A limited number of studies use DL methods to segment buildings' point clouds into the category to which each point belongs to. Compared to the improvement of DL methods in indoor scenes, the segmentation methods of high LoD buildings' point clouds are still at the initial stage of development. Most existing studies focus on LoD1 (Chen et al., 2014; Zhang, Zhang, 2017; Zhang, 2018; Griffiths, Böhm, 2019; Huang et al., 2019), LoD2 (Hensel, 2019; Jarząbek-Rychard, Borkowski, 2016), or one category of building's element (Axelsson et al., 2018).

Moreover, it is noteworthy to mention that only the ArCH Data Set (Matrone et al., 2020a) with pointwise annotations is made publicly available. The lack of semantic segmentation labels indicates the challenge for human beings to provide pointwise labels. Thus, there are also just a handful of DL-based research in higher LoD (e.g., LoD3 and superior) segmentation tasks. Pierdicca et al. (2020) proposed to employ DGCNN for the point cloud segmentation task applied on the ArCH Data Set. By adding radiometric (HSV value) and Normal features, they further improved the performance of segmentation. Their work showed the potential offered by DL techniques for the segmentation task. By fusing spectral information and hand-crafted geometric features, DGCNN-Mod+3Dfeat (Matrone et al., 2020b) combines the positive aspects and advantages of *machine learning* and DL for semantic segmentation of point clouds in the field of Cultural Heritage. But 3D features in

DGCNN-Mod+3Dfeat are hand-designed and extracted by machine learning methods, so this solution is out of our consideration.

Even though the intensive efforts to improve buildings' point clouds segmentation performance, (1) most existing algorithms are insufficient to model details and are associated with a heavy workload, which meets current requirements in the development phase; (2) existing methods mostly used both 3D coordinates and hand-crafted features (i.e., geometric features) as their input to enhance the performance; (3) most of the existing DL-based approaches for 3D point cloud analysis are strongly supervised and rely on massive amounts of labelled 3D data.

2.2 Label-Efficient Methods on 3D Point Clouds

Unsupervised learning refers to learning methods without using any human-annotated labels. Since the scarcity of fine-labelled point clouds data sets, unsupervised learning methods are designed to exploit the inherent and underlying information in large unlabelled data, which may dramatically decrease the need for labelled training data. Several unsupervised methods (e.g., GAN, AE) applied to 3D point clouds are reported in the literature, partly due to the common criticism that in a DNN a huge amount of labelled data is required for training. The research of unsupervised AE methods on 3D point cloud data is a relatively new research topic. Existing literature is mostly very recent, as far as we know, there is no unsupervised AE model for buildings' point clouds segmentation task to date. Therefore, our review of the label-efficient unsupervised AE approach will not be limited to the built environment domain in this section.

2.2.1 Autoencoder (AE): An AE was trained to learn a compressed representation by faithfully reconstructing input original image/point cloud. In FoldingNet (Yang et al., 2018), the authors adopted the idea of the folding-based decoder to deform a canonical 2D grid onto the underlying 3D object surface of a point cloud. Built upon the fully supervised PPFNet (Deng et al., 2018a) and FoldingNet, in PPF-FoldNet (Deng, 2018b) the authors improved their earlier solution by involving more features in their network in an unsupervised fashion. 3D-PointCapsNet (Zhao, 2019) employed a dynamic routing scheme to extract discriminative representation while considering the geometric relations between parts. BAE-NET (Chen et al., 2019) proposed a branched AE network trained with a shape co-segmentation task.

However, most of these existing 3D point cloud unsupervised AE methods are trained by using simple 3D objects, and none of them applied to the semantic segmentation of architectural objects. Overall, learning to automatically generate powerful representation from uneven point clouds, especially buildings' point clouds with complex geometric structures, still poses a challenge. To address these issues, in this paper we propose an improved AE approach to benefit from the ability to learn features without labelled data.

3. METHOD

In FoldingNet, an AE is utilized to reconstruct input point cloud, whilst discriminative representations are learned without any labelled data. Inspired by this, our label-efficient method aims to: (1) construct an AE network for extracting features without any labelled data; (2) with just a few labelled data, we train a segmentation network for the high-resolution LoD3 buildings' point cloud semantic segmentation. Specifically, we proposed an AE network that may learn representations without any label by a dynamically updated graph-based encoder and folding-based decoder. Thus, we may reduce the need for large amounts of labels. Instead of the encoder in FoldingNet, we employ the EdgeConv layers in DGCNN to exploit local geometric structures and generate discriminative representations. Then, we use the learned representations as input to our downstream task. In general, the proposed network architecture consists of three components: a DGCNN-based encoder, a folding-based decoder, and a segmentation network. The input of the encoder is N coordinates (x,y,z) of buildings' points, and outputs are discriminative features, which are also the input of both decoder and the segmentation network. The outcome is a matrix of size *(m, 3)* representing the reconstructed point cloud and per-point classification scores *(m, n_classes)* for decoder and segmentation network, respectively. The architecture of our improved AE is illustrated in Figure 1, and the segmentation network is shown in Figure 2.

3.1 DGCNN-based Encoder

Both encoders of FoldingNet and DGCNN use graph-based layers to extract the local geometric information in point's neighbourhood and a max-pooling layer to aggregate information. The local features of FoldingNet are computed as follows:

$$h_\Theta(x_i, x_j) = h_\Theta(x_j - x_i), \quad (1)$$

In this edge function, x_i is the central point belonging to Point Set $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^3$, x_j is the local neighbours around the central point x_i and h_Θ is implemented by a fully connected multi-perceptron layer, which includes learnable parameters. FoldingNet obtains the local information by encoding $x_j - x_i$ edge features. Then the learned local information aggregated by a local max-pooling operation on the constructed graphs $G = (V, E)$, where $V = \{1, \ldots, N\}$ and $E \subseteq V \times V$ are the vertices and the edges, respectively, and N is the number of vertices. On the other side, the operation on the constructed graph G of DGCNN is EdgeConv operation, which may extract both local geometric and global shape information from the constructed graph. Firstly, the EdgeConv layer computes an edge feature set of size k for each input point clouds through asymmetric edge function:

$$h_\Theta(x_i, x_j) = h_\Theta(x_i, x_j - x_i), \quad (2)$$

In this edge function, EdgeConv captures the global shape by encoding the coordinates of x_i, then obtains the local information by encoding $x_j - x_i$. The output feature is aggregated by edge features from each connected vertex and itself in the constructed graph. Thus, EdgeConv can explicitly combine the global shape structure information with local neighbourhoods' information.

Furthermore, in FoldingNet, they construct the graph by computing pairwise distances using initial input point coordinates. Hence their graph G is fixed. In contrast, we calculate the pairwise distance in feature space at each layer and choose the nearest k points per each central point, and then we dynamically construct $G^l = (V^l, E^l)$ at layer l. The receptive field becomes larger while such dynamic graph updates in each layer, and local information around central points and global information in different receptive fields are aggregated and stacked in the last layer before the max-pooling layer.

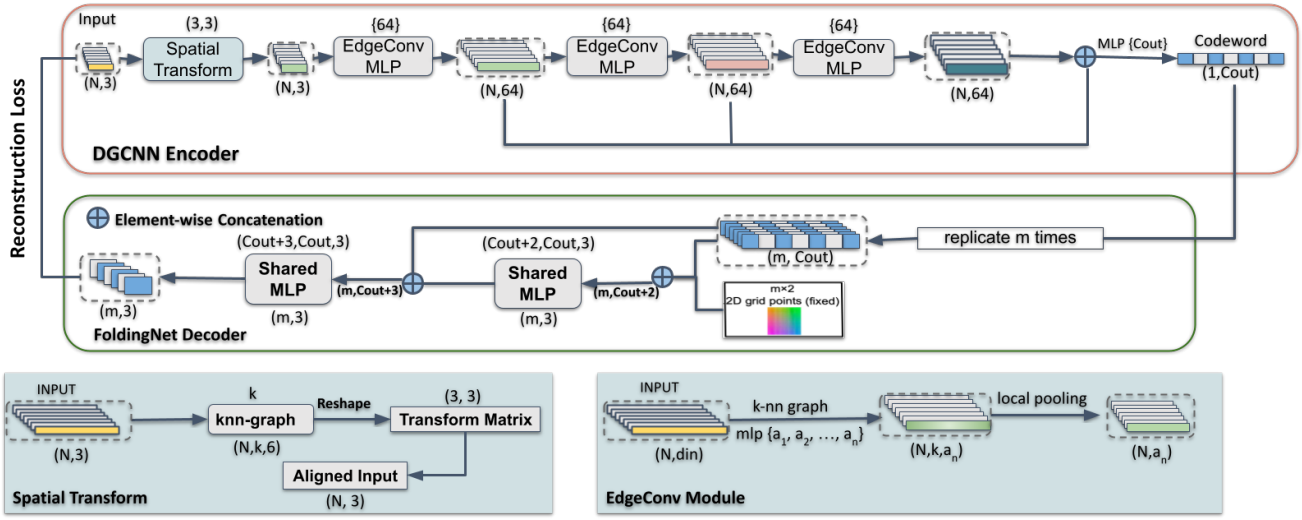

Figure 1. The architecture of our 3D Autoencoder (AE) network, consisting of a DGCNN-based encoder module (top) and a folding-based decoder module (bottom). The AE learns a discriminative representation "codeword" by reconstructing it to 3D surface and training with Chamfer Distance loss. The outputs of three EdgeConv layers and codeword from pre-trained AE will be further concatenated and used in our semantic segmentation network.

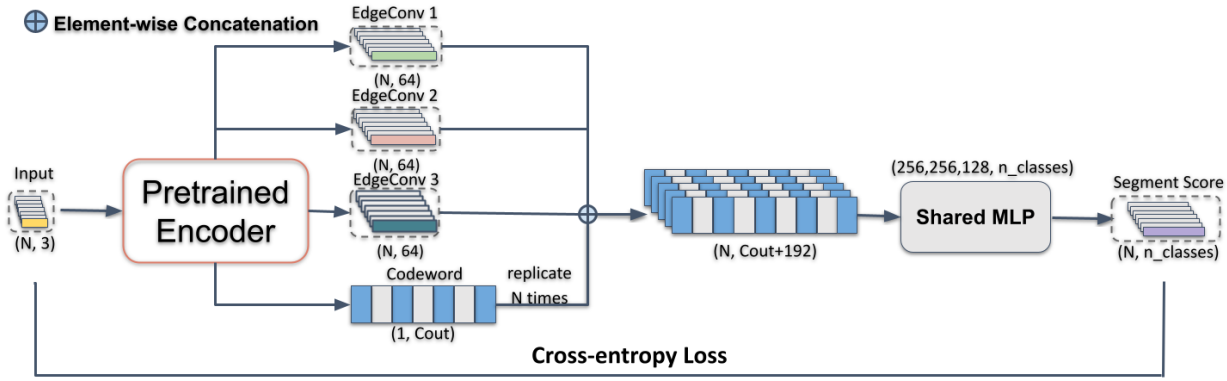

Figure 2. The architecture of the semantic segmentation network. The network takes the outputs of pre-trained AE as input. It is trained to output pointwise segmentation scores of building point clouds by simply shared Multilayer Perceptron (MLP).

The procedure for producing feature representations in DGCNN-based encoder is visualized in Figure 1. We remove the encoder of FoldingNet and replace it with three EdgeConv layers in DGCNN segmentation architecture. The outputs of the three EdgeConv layers are concatenated and then passed to a feature-wise max-pooling layer to produce a C_{out} -dimensional "codeword" θ. The outcomes of three EdgeConv layers and the "codeword" θ are stored in the pretrained AE model, which will be the basis for our segmentation network.

3.2 Folding-based Decoder

We use the "codeword" output from the DGCNN-based encoder and a 2D grid as input to our decoder. A folding-based decoder is then utilized to reconstruct input "codeword" with a 2D grid to 3D point clouds by two successive folding operations.

The folding-based decoder in our AE network is adopted from FoldingNet's decoder that contains two successive folding operations. The first one folds the 2D manifold into 3D space, and the second one operates inside the 3D space. As shown in Figure 1, we have modified the decoder of FoldingNet to make it usable with different sizes of input "codeword" θ (512-dimensional and 1024-dimensional) instead of a fixed size 512-dimensional representation in FoldingNet. Before feeding the "codeword" into the folding-based decoder, we replicate the "codeword" θ m times and concatenate the replicated (m, C_{out}) matrix with an (m, 2) matrix, which contains the m grid points (U) on a square centred at the origin. As each row of U is a 2D grid point, we define the i-th row of U as u_i. Thus, the i-th row of the input matrix to the folding operation is $[u_i, C]$ after above concatenation. The following two folding operations essentially form a universal 2D-to-3D mapping by two successive Multilayer Perceptron (MLP). The MLPs are applied in parallel to each row of the input matrix. We denote the i-th row of the output matrix as $f([u_i, C])$, where f is approximated by the MLPs which can be tuned by the input "codeword" and learn a "force" to reconstruct the input into arbitrary point cloud surfaces. During the training process, we use Chamfer distance (Fan et al., 2017) as our reconstruction loss, which measures the similarity of the reconstructed point cloud and the input point cloud. With the DGCNN-based encoder and folding-based decoder, we learn a set of powerful and separable features and pass these learned features into our downstream semantic segmentation task.

3.3 Semantic Segmentation Network Architecture

We created a semantic segmentation network to semantically segment buildings' point clouds. The goal here is to assign a semantic label to each of the points given an input point cloud. Hence, we treat this semantic segmentation as a per-point classification task. The output of the pre-trained encoder is a C_{out}-dimensional representation ("codeword") and three stacked edge features, which are learned from non-labelled buildings' point clouds. We replicate the codeword N times and concatenate it with the outputs of three EdgeConv layers in the pre-trained AE. A standard 3-layer shared MLP with a cross-entropy loss is then employed as our semantic segmentation classifier after the above concatenation. Considering the features obtained by the proposed AE are already distinctive, we chose this simplest MLP for segmentation of the building point cloud. This semantic segmentation network is trained independently from the proposed AE. The procedure for acquiring per-point classification scores in the semantic segmentation network is illustrated in Figure 2.

4. EXPERIMENTS

4.1 Dataset

We have qualitatively and quantitatively evaluated our method on the Architectural Cultural Heritage (ArCH) Data Set. In the state-of-the-art, the most used data sets to train unsupervised learning are: ModelNet40 (Wu et al., 2015) with more than 100k CAD models of objects from 40 different categories for classification tasks; ShapeNetPart (Chang et al., 2015) data set with 31,693 meshes classified into 16 common classes (i.e., plane, table, chair, etc.); each shape has 2-5 parts for part-segmentation tasks. However, none of them can be used for buildings' point cloud segmentation. Other outdoor data sets used in this task, such as Semantic3D (Hackel et al., 2017) and Oakland (Munoz et al., 2009) features LoD1 or LoD2 for the architectural elements.

A building in LoD3 has detailed surface structures such as walls, roof, and potentially openings (doors and windows). To date, there are still no published data sets focusing on urban buildings' point clouds with a sufficient level of details such as LoD3. The components of historical architectural heritage including detailed roof, façade structures, openings, and some unique structures such as vaults, which are similar to but more complicated than contemporary buildings. Networks trained on this kind of data sets are easy to generalize to other building scenes. So, we chose an immovable cultural assets data set with fine per-point labels, named ArCH Data Set, to evaluate our method. It consists of 15 indoor and outdoor labelled scenes, including churches, chapels, cloisters and porticoes.

Our primary motivation to study unsupervised classification problems is that the number of training data is limited. To test the performance when the number of unlabelled and labelled data is small, we select three small scenes (namely, "SMV_1", "SMV_24", "SMV_28") from the 15 labelled scenes as the training data in both unsupervised AE training and supervised segmentation training stages. The training data in our experiment is only 10% of state-of-the-art (Pierdicca et al., 2020), who use all scenes as training data. Then we follow the settings adopted by Pierdicca et al. (2020) that removed the "others" category, selected two scenes ("A_SMG_portico" - Scene_A and "B_SMV_ chapel_27to35" - Scene_B) as our test data. The scenes used in our experiments are acquired by both terrestrial laser scanners (i.e., a FARO Focus 3D X 120/130 and a Riegl VZ-400) and Structure-from-Motion photogrammetry (Barazzetti et al., 2009) based on drone images (Fugazza et al., 2018). A DJI Phantom UAV platform equipped with a SONY Ilce 5100L were used for data acquisition.

4.2 Implementation Details

We choose $1m \times 1m$ area as the block size for splitting each building scene into blocks to train. Prior to training, the input point clouds are aligned to a common reference frame. In addition, for training convenience, the points in each block are sampled into a uniform number of 8,192 points. During training, we have randomly sampled n (2,048 or 4,096) points in each block on-the-fly. To train our AE network, we have employed ADAM (Kingma, Ba, 2015) as an optimizer with an initial learning rate 0.001, batch size 16, and weight decay 10^{-6} with 250 epochs. The setting of hidden layers in our encoder is the same as DGCNN, but we have removed the layers after the max-pooling layer. The architecture of the encoder incorporates the following steps:

1. Three EdgeConv layers to extract local and global geometric features. The EdgeConv layers take a tensor of shape $n \times f$ as input, then acquire edge features for each point by applying an MLP with the number of layer neurons defined as $\{a_1, a_2, \ldots, a_n\}$. The number of nearest neighbours k is set as 20 at every EdgeConv layer; and
2. Features generated in three EdgeConv layers are concatenated to aggregate features in different receptive fields; and
3. The dimension of the MLP layer before the last max-pooling layer is set as C_{out} (512 or 1,024) to globally aggregate a 1D global descriptor "codeword" θ.

In our graph-based decoder, we have used two consecutive 3-layer MLPs to warp a fixed 2D grid into point cloud surfaces. Before feeding the "codeword" into the folding-based decoder, we have replicated the "codeword" m times and concatenated the replicated (m, C_{out}) matrix with an $(m, 2)$ matrix. According to the input point cloud size (2,048 or 4,096), m is set as 2,025 or 4,096. Then the sizes of two 3-layer Shared MLPs is (C_{out}+ 2, C_{out}, 3) and (C_{out}+3, C_{out}, 3), and implemented by six 1-D convolutional layers, each followed by a ReLU layer. The output is the reconstructed point cloud with size $(m, 3)$.

Similarly, in the semantic segmentation network, we have also used ADAM as our optimizer (learning rate 0.01, batch size 16, 250 training epochs). According to the dimension of C_{out}, our shared MLPs is $(C_{out} + 64 + 64 + 64, 512, 256, 128, n\_classes)$ with layer output sizes $(512, 256, 128, n\_classes)$ on each point. The evaluation metrics of overall point accuracy (OA) and mean Intersection-over-Union (mIoU) are calculated on the ArCH data set. The method has been implemented using PyTorch. All experiments have been conducted on an NVIDIA Tesla T4 GPU.

4.3 Results

If the features obtained by the proposed AE are already distinctive, the required number of labelled data in semantic segmentation network training process should be small. In this section, to demonstrate this intuitive statement, we report our experiment's results on the ArCH Data Set. We evaluate our model on two unseen scenes ("Scene_A" and "Scene_B") for testing. In Table 1, the overall performances are reported and

compared with state-of-the-art (SOTA) methods, which are retrieved from Pierdicca et al. (2020): PointNet (Qi et al., 2017a), PointNet++ (Qi et al., 2017b), PCNN (Atzmon et al., 2018) and DGCNN (Matrone et al., 2020b) with 10 scenes, and DGCNN (Pierdicca et al., 2020) with 15 scenes as training data.

Networks	Train Scenes	Test Scene	Evaluation Matrix	
			mIoU	OA
PointNet	10 scenes	Scene_B	0.114	0.307
PointNet++	10 scenes	Scene_B	0.121	0.441
PCNN	10 scenes	Scene_B	0.260	0.635
DGCNN	10 scenes	Scene_B	0.290	0.74
DGCNN	15 scenes	Scene_B	0.353	**0.752**
DGCNN	15 scenes	Scene_A	0.376	**0.784**
DGCNN	3 scenes	Scene_B	0.163	0.362
DGCNN	3 scenes	Scene_A	0.243	0.499
Ours	3 scenes	Scene_B	**0.408**	0.666
Ours	3 scenes	Scene_A	**0.463**	0.773

Table 1. Our results *vs* prior works on Architectural Cultural Heritage (ArCH) Data Set. OA and mIoU denote overall accuracy and mean Intersection-over-Union, respectively. Our method performs the best on mIoU with only 3 scenes (about 10% of 10 scenes).

With only about 10% of training data with respect to SOTA methods in both AE and segmentation network training stages, our model achieves the best results on the ArCH Data Set with the same training strategy (only input x, y, z coordinates). In particular, the test mIoU on Scene A is 0.463, which overcomes the previous SOTA (0.376). The mIoU on Scene B is 0.408, which also outperforms the 0.353 of SOTA. The overall point accuracy (OA) is the ratio between the amount of properly classified points and the total number of points in the two scenes. mIoU takes the false alarms and different categories into consideration, which is more informative. We also compare our results on Scene_A and Scene_B with respect to SOTA methods when training on the same subset (3 scenes). As shown in Table 1, the results of "ours" method also outperform the SOTA methods.

The results of per-class quantitative evaluations on Scene_B of ArCH Data Set are provided in Table 2. We can see that the proposed model performs better than other competitive methods in many classes. The semantic segmentation qualitative result is shown in Figure 3. Our network is able to output smooth predictions.

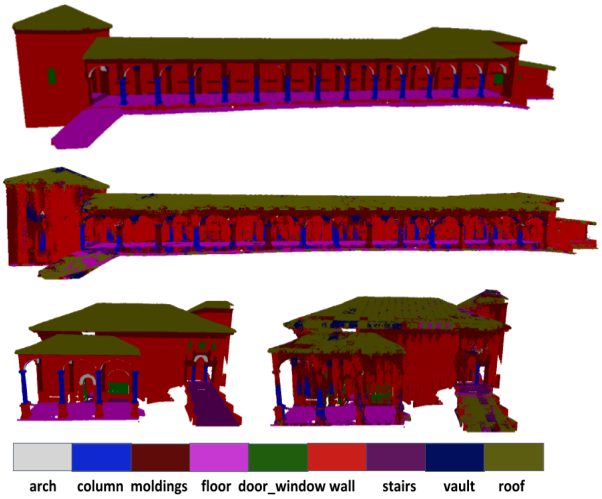

Figure 3. Qualitative results for semantic segmentation. The top row is the ground truth, and the second row is the prediction result of "Scene_A". The bottom row is the ground truth and output semantic segmentation result of "Scene_B". Same scenes are displayed in the same camera viewpoint.

4.4 Ablation Study

In this section, we validate our design choices by control experiments. We also show the effects of choices of our network's hyperparameters.

4.4.1 Effectiveness of DGCNN-based Encoder: In Table 3, we show the positive effects of our proposed DGCNN-based encoder. Compared to FoldingNet, the performance of DGCNN-based encoder has a 7% boost while using one scene in the training stage, and a 14% improvement while three scenes in the training stage, testing on Scene_B.

Setting	Training Scene	OA
FoldingNet	1 scene	0.425
FoldingNet	3 scenes	0.420
DGCNN	1 scene	0.493
DGCNN	3 scenes	0.561

Table 3. The varying encoders tested on ArCH Data Set Scene_B. OA denotes overall accuracy. 1 scene ("SMV_24") and 3 scenes ("SMV_1", "SMV_24", "SMV_28") in 'Training Scenes' column mean we use only 1 scene or 3 scenes in both AE training stage and segmentation network training stage (training without using any data augmentation).

Method	mIoU	Arch	Column	Molding	Floor	Door-Window	Wall	Stair	Vault	Roof
PointNet	0.114	0.000	0.000	0.001	0.294	0.000	0.411	0.000	0.337	0.094
PointNet++	0.121	0.000	0.000	0.002	0.009	0.000	0.514	0.000	0.074	0.608
PCNN	0.260	0.072	0.062	0.198	0.482	0.004	0.581	0.082	0.468	0.658
DGCNN	0.290	0.060	0.064	0.142	0.470	0.006	0.603	0.290	0.520	0.845
Ours	**0.408**	**0.880**	**0.243**	0.117	0.471	0.005	**0.676**	0.035	**0.577**	0.659

Table 2. Our per-class IoU and class-averaged mIoU results on Scene_B *vs* prior works on ArCH data set. Each column represents the IoU of the category it belongs. Our method performs better than others in many classes with only 3 scenes (about 10% of 10 scenes).

4.4.2 Comparison with Different Dimensions of Codeword: We use 512-dimensional codeword and 1,024-dimensional codeword as control experiments to compare the segmentation result of buildings' point clouds. In Table 4, we show the effects of different codeword dimensions, test on Scene_A. The input point cloud size is fixed in 2,048 in the AE training stage, and we set two input point size (2,048 and 4,096) in the segmentation training stage.

As shown in Table 4, the model performs better when the dimension of the codeword is 512, regardless of whether the input point size is 2,048 or 4,096 in the segmentation network.

Dims	seg_n_points	OA
512	2,048	0.747
1,024	2,048	0.722
512	4,096	0.773
1,024	4,096	0.694

Table 4. The varying number of dimensions for output codeword learned from AE. 512 and 1,024 in column "Dims" denotes 512-dimensional codeword and 1,024-dimensional codeword, respectively. "seg_n_points" denotes the input point size in the segmentation network training stage. Input point size in the AE training stage is 2,048.

5. CONCLUSIONS

In this study, we have presented an effective label-efficient unsupervised network for LoD3 buildings' point clouds semantic segmentation. The results of our experiments provide support that our proposed Autoencoder architecture may learn powerful representations from unlabelled data, and these representations can be further used in downstream tasks. Furthermore, our network supplies a unified approach for the segmentation task of building point clouds while obtaining on equal or better results w.r.t. the state of the arts on the basis of only 10% training data from the ArCH dataset. We experimentally demonstrated the effectiveness of the DGCNN-based encoder. We also provided detailed ablation studies to validate our design choices.

However, our network has the following limitations: (1) In the data pre-processing stage, the block size and the sampling number of points in each block is fixed. Thus, network was training on a small region of building scenes, and the performance would be degraded, resulting in wrong segmentation; (2) some details of our implementation could also be extended to improve efficiency, e.g., incorporating RGB, laser intensity (if available – see Scaioni et al. 2018) and normal information on one side, and by increasing the unlabelled training data in the unsupervised AE training stage on the other.

In future work, we plan to improve the model's performance by breaking through the input point size and incorporating more features of the input buildings' point cloud.

ACKNOWLEDGEMENTS

Financial support from the program of China Scholarships Council (Grant No. 201906860014) is acknowledged. We thank Dr.Matrone et al. for the ArCH dataset. We thank the two anonymous reviewers for their constructive comments, their comments greatly improved this paper.

REFERENCES

Atzmon, M., Maron, H., Lipman, Y., 2018. Point convolutional neural networks by extension operators. *ACM Trans. Graph.* 37(4), pp. 0730-0301.

Axelsson, M., Soderman, U., Berg, A., Lithen, T., 2018. Roof Type Classification Using Deep Convolutional Neural Networks on Low Resolution Photogrammetric Point Clouds from Aerial Imagery. In IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Apr, Calgary (AB - Canada), pp. 1293-1297.

Barazzetti L., Remondino, F., Scaioni, M., 2009. Combined use of Photogrammetric and Computer Vision techniques for fully automated and accurate 3D modeling of terrestrial objects. Int. Conf. "Videometrics, Range Imaging, and Applications X", 2-3 Aug, San Diego (CA - USA), Proc. of SPIE, Vol. 7447, Paper No. 74470M, 12 pages.

Cao, Y., Previtali, M., Scaioni, M., 2020. Understanding 3D Point Cloud Deep Neural Networks by Visualization Techniques. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLIII-B2-2020, 651–657.

Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., et al., 2015. Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012.

Chen, D., Zhang, L., Mathiopoulos, P.T., Huang, X., 2014. A methodology for automated segmentation and reconstruction of urban 3-D buildings from ALS point clouds. *IEEE J. Select. Topics App. Earth Observ. Remote Sens.* 7 (10), 4199-4217.

Czerniawski, T., Leite, F., 2020. Automated digital modeling of existing buildings: A review of visual object recognition methods. *Autom. Constr.*, 113, 103-131.

Deng, H., Birdal, T., Ilic, S., 2018a. Ppfnet: Global context aware local features for robust 3d point matching. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 18-23 Jun, Salt Lake City, 195-205.

Deng, H., Birdal, T., Ilic, S., 2018b. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. Europ. Conf. on Computer Vision,Sep, Munich, Germany, pp. 602-618.

Fan, H., Su, H., Guibas, L.J., 2017. A point set generation network for 3d object reconstruction from a single image. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 21-26 July, Honolulu (HI - USA), pp. 605-613.

Fazeli, H., Samadzadegan, F. Dadrasjavan, F., 2016. Evaluating the potential of RTK-UAV for automatic point cloud generation in 3D rapid mapping. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci*., XLI-B6, pp. 221-226.

Forlani, G., Nardinocchi, C., Scaioni, M., Zingaretti, P., 2006. Complete classification of raw LIDAR data and 3D reconstruction of buildings. *Pattern Anal. Appl.*, 8(4), 357-374.

Fugazza, D., Scaioni, M., Corti, M., D'Agata, C., Azzoni, R.S., Cernuschi, M., Smiraglia, C., Diolaiuti, G.A., 2018. Combination of UAV and terrestrial photogrammetry to assess rapid glacier evolution and map glacier hazards. *Nat. Haz. Earth Syst Sci.*, 18, 1055-1071.

Griffiths, D., Boehm, J., 2019. A review on deep learning techniques for 3D sensed data classification. *Remote Sens.*, 11(12), 1499.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L. Bennamoun, M., 2020. Deep learning for 3d point clouds: A survey. *IEEE TPAMI.* 10.1109/TPAMI.2020.3005434.

Haala, N., Brenner, C. and Anders, K.H., 1998. 3D urban GIS from laser altimeter and 2D map data. *Int. Arch. Photogramm. Remote Sens.* 32, 339-346.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M., 2017. Semantic3d. net: A new large-scale point cloud classification benchmark. *Int. Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, 4, 91-98.

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(4), B4014009.

Han, X.; Laga, H.; Bennamoun, M., 2021. Image-Based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43, 1578-1604.

Hu, P., Yang, B., Dong, Z., Yuan, P., Huang, R., Fan, H. and Sun, X., 2018. Towards reconstructing 3D buildings from ALS data based on gestalt laws. *Remote Sens.*, 10(7), 1127.

Huang, J., Zhang, X., Xin, Q., Sun, Y. and Zhang, P., 2019. Automatic building extraction from high-resolution aerial images and LiDAR data using gated residual refinement network. *ISPRS J. Photogramm. Remote Sens.*, 151, 91-105.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. In ICLR (Poster), 1 Jan, San Diego (CA -USA).

Kumar, B., Lohani, B., Pandey, G., 2019. Development of deep learning architecture for automatic classification of outdoor mobile LiDAR data. *Int. J. Remote Sens,* 40(9), 3543-3554.

Maas, H.G. and Vosselman, G., 1999. Two algorithms for extracting building models from raw laser altimetry data. *ISPRS J. Photogramm. Remote Sens.*, 54(2-3), 153-163.

Matrone, F., Lingua, A., Pierdicca, R., Malinverni, E.S., Paolanti, M., Grilli, E., et al., 2020a. A benchmark for large-scale heritage point cloud semantic segmentation. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci*., XLIII-B2, 1419–1426.

Matrone, F., Grilli, E., Martini, M., Paolanti, M., Pierdicca, R., Remondino, F., 2020b. Comparing machine and deep learning methods for large 3D heritage semantic segmentation. *ISPRS Int. J. Geo-Inf.* 9 (9), 535.

Meng, Q., Wang, W., Zhou, T., Shen, J., Van Gool, L. and Dai, D., 2020, August. Weakly Supervised 3D Object Detection from Lidar Point Cloud. In Europ. Conf. on Computer Vision (ECCV), 23-28 Aug, SEC, Glasgow, 515-531.

Munoz, D., Bagnell, J.A., Vandapel, N. and Hebert, M., 2009, June. Contextual classification with functional max-margin markov networks. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 20-25 Jun, Miami (FL – USA), 975-982.

Pierdicca, R., Paolanti, M., Matrone, F., Martini, M., Morbidoni, C., Malinverni, E.S., Frontoni, E. and Lingua, A.M., 2020. Point cloud semantic segmentation using a deep learning framework for cultural heritage. *Remote Sens*, 12(6), 1005.

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.

Qi, C.R., Su, H., Mo, K. and Guibas, L.J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conf. on computer vision and pattern recognition (CVPR), 21-26 Jul, Honolulu (HI – USA), 652-660.

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.

Scaioni, M., Höfle, B., Baungarten Kersting, A.P., Barazzetti, L., Previtali, M., Wujanz, D., 2018. Methods for Information Extraction from Lidar Intensity Data and Multispectral Lidar Technology. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLII-3, 1503-1510.

Su, H., Maji, S., Kalogerakis, E. and Learned-Miller, E., 2015. Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE Int. Conf. on computer vision (CVPR), 7-13 Dec, Santiago, Chile, 945-953.

Verma, V., Kumar, R. and Hsu, S., 2006. 3d building detection and modeling from aerial lidar data. In IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR), 17-22 Jun, New York (NY-USA), 2213-2220.

Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y. and Tong, X., 2017. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Trans. Graph.*, 36(4), 1-11.

Wang, C., Hou, S., Wen, C., Gong, Z., Li, Q., Sun, X. and Li, J., 2018. Semantic line framework-based indoor building modeling using backpacked laser scanning point cloud. *ISPRS J. Photogramm. Remote Sens.*, 143, 150-166.

Wang, Q., Kim, M.K., 2019. Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. *Adv. Eng. Inform.,* 39, 306-319.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M. and Solomon, J.M., 2019. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.*, 38(5), 1-12.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. and Xiao, J., 2015. 3d shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conf. on computer vision and pattern recognition (CVPR), 7-12 Jun, Boston, (MA - USA), 1912-1920.

Yang, Y., Feng, C., Shen, Y. and Tian, D., 2018. Foldingnet: Point cloud auto-encoder via deep grid deformation. In Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 19-23 Jun, Salt Lake City (UT – USA), 206-215.