# Label-efficient Deep Learning-based Semantic Segmentation of Building Point Clouds at LoD3 Level

Yuwei Cao[1], Marco Scaioni[1]

[1] Dept. of Architecture, Built Environment and Construction Engineering, Politecnico di Milano, via Ponzio 31, 20133 Milano, Italy, e-mail: {yuwei.cao, marco.scaioni}@polimi.it

**Abstract.** In recent years, Deep Learning (DL) techniques and large amounts of pointwise labels are employed to segment point clouds of the built environment. However, annotating pointwise labels is a time-consuming task. To address this issue, we propose a label-efficient DL network that obtains per-point semantic labels of LoD3 (Level-of-Detail) building point clouds with limited supervision. Experimentally, we compared our approach to the fully supervised DL methods, and we find our approach achieved comparable results on the ArCH Data Set, with only 10% of labelled training data obtained from fully supervised methods as input.

**Keywords:** 3D Building, Deep Learning, Label-efficient, Point Cloud, Semantic Segmentation.

## 1 Introduction

In recent years, 3D buildings' point cloud representation enables and promotes new applications in many fields such as Cultural Heritage preservation [1-2], Construction Engineering [3-4], Emergency Decision-making [5], and Smart Cities [6]. Extracting semantic information from 3D buildings' point clouds to acquire high Level-of-Details (LoDs) modelling is an essential task [7].

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 (CV) to accomplish subset tasks (i.e., classification, detection and semantic segmentation), in recent research, *fully-supervised* Deep Learning (DL) techniques and large amounts of pointwise labels have been employed to train a segmentation network to be applied to buildings' point clouds. However, fine-labelled point clouds of the built environment are hard to find and manually annotating pointwise labels is a time-consuming and expensive task. The application of fully supervised learning for semantic segmentation of buildings' point clouds at LoD3 level is severely limited.

In CV, the hunger for fine-labelled pointwise training data is often tackled by using *unsupervised* methods. However, these approaches are mostly designed for 2D images, which are fundamentally different from unordered 3D point clouds. 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 semantic segmentation of buildings' point clouds is still an open issue, and current knowledge about it is deeply unsatisfactory.

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 main steps. The first step, named Autoencoder, is composed of a Dynamic Graph Convolutional Neural Network-based [8] *encoder* and a folding-based *decoder*. It is designed to extract discriminative global and local features from input point clouds by reconstructing them without any label. The second step is the semantic segmentation network. 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 Autoencoder.

## 2 Related Work

*Unsupervised learning* refers to learning methods without using any human-annotated labels. Since the scarcity of fine-labeled point cloud datasets, unsupervised learning methods have become popular alternatives to fully supervised learning to exploit the inherent and underlying information in large unlabeled datasets, which may dramatically decrease the need for labeled training data. Following the impressive results that have been achieved with unsupervised learning in the 2D image field, previous efforts to perform unsupervised learning on point clouds have been derived from tailoring these methods. Several unsupervised methods (e.g., Generative Adversarial Networks, Autoencoder) applied to 3D point clouds are reported in the literature, partly due to the common criticism that a huge amount of labeled data is required for training in a DNN. We provide a quick overview of both types of methods.

### 2.1 Generative Adversarial Networks

Typically, Generative Adversarial Networks (GANs) consist of a generator that learns how to map from a latent space to a data distribution of interest. A discriminator distinguishes generated point cloud produced by the generator from the true data distribution. For example, Achlioptas [9] investigated and compared GAN-based method for generating point clouds in raw data space and latent space of a pre-trained autoencoder. Li [10] proposed a "sandwiching" reconstruction method that combines a modification of Wasserstein GAN [11] loss with Earth Mover's Distance (EMD). AtlasNet [12] introduces a shape generation framework that represents a 3D shape as a collection of parametric surface elements by locally mapping a set of squares to the target surface of a 3D shape. Although impressive results were achieved, GAN-based methods more focus on generative models of point clouds, which aims to generate point clouds or complete shapes of point clouds.

### 2.2 Autoencoders (AEs)

An Autoencoder (AE) is trained to learn a compressed representation by faithfully reconstructing input original image/point cloud [13]. In FoldingNet [14], 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, in which the learned representation achieves high linear SVM classification accuracy on ModelNet40 dataset. Built on the fully supervised PPFNet [15] and FoldingNet, in PPF-FoldNet [16] the authors improve their earlier solution by involving more features in their network in an unsupervised fashion. PPF-FoldNet achieves better reconstruction performance at rotations and different point densities, but their research focuses on reconstruction rather than downstream tasks. BAE-NET [17] proposed a branched AE network which trains with a collection of objects from the ShapeNetPart dataset trained with a shape co-segmentation task.

Existing methods achieve state-of-the-art in their downstream tasks (i.e., classification, part-segmentation and co-segmentation). However, most of these existing unsupervised AE methods for 3D point clouds are: 1) trained and tested using simple 3D objects; 2) designed for low-level tasks such as reconstruction, denoising and completion that are not designed for high-level downstream semantic segmentation task, resulting in downstream tasks of these AE methods that have not been applied to high-level semantic segmentation tasks either.

## 3 Method

In FoldingNet, an Autoencoder (AE) is utilized to reconstruct input point clouds, whilst discriminative representations were 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 Dynamic Graph Convolutional Neural Network (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 (see Fig. 1) consists of two components: an AE and a segmentation network.

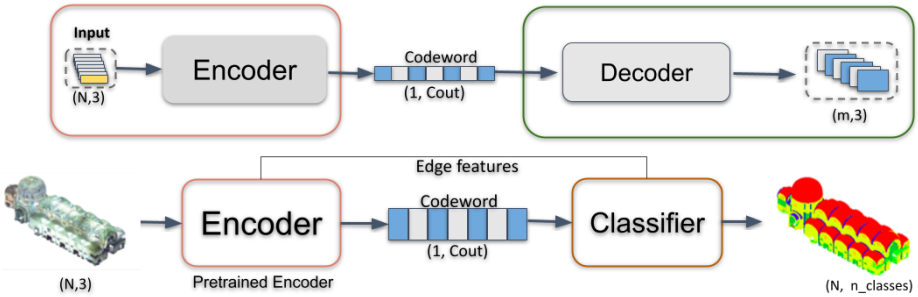


**Fig. 1.** The architecture of our Autoencoder-based building point cloud semantic segmentation network. Our approach works in two steps: on the top row is the Autoencoder step, and on the bottom row is the downstream segmentation step.

### 3.1 Autoencoder

The input of the AE is given by the *N* coordinates *(x, y, z)* of buildings' points, and intermedia outputs are discriminative features, which are also the input of both decoder of AE and the segmentation network. The final outcome is a matrix of size *(m, 3)* representing the reconstructed point cloud. We use graph-based layers to extract the local geometric information around points and a max-pooling layer to aggregate information. The edge features are computed as follows:

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

In this edge function, $x_i$ is the central point belonging to Point Set $\{X = x_i, \ldots, x_n\} \subseteq \mathbb{R}^3$, $x_j$ is the local neighbors around the central point $x_i$ and $h_\Theta$ is implemented by a fully connected multi-perceptron layer, which includes learnable parameters. EdgeConv captures the global shape by encoding the coordinates of $x_i$, then obtains the local information by encoding $x_j - x_i$. 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 and the edges respectively and $N$ is the number of vertices.

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 folding operation folds the 2D manifold into 3D space, and the second one operates inside the 3D space.

### 3.2 Semantic Segmentation Network

To semantically segment buildings' point clouds, we created a segmentation network. 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 AE is a *Cout*-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 Multi-Layer Perceptron (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 the segmentation of the point cloud. This semantic segmentation network is trained independently from the proposed AE. The final output is per-point classification scores *(m, n_classes)* for the segmentation network.

# 4 Experiment

## 4.1 Implementation Details

Experimentally, we evaluate our approach based on the ArCH Data Set [18], which is acquired by both terrestrial laser scanners (i.e., a FARO Focus 3D X 130 and 120 a Riegl VZ-400) and Structure-from-Motion Photogrammetry based on images collected by a DJI Phantom UAV platform equipped with a SONY Ilce 5100L camera.

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 (SMV_1, SMV_24, SMV_28) scenes from the 15 labelled scenes as the training data in both unsupervised AE training and supervised segmentation training stage. The training data in our experiment is only 10% of state-of-the-art [2], where 10 scenes are used as training data. Then we follow the settings in state-of-the-art [2] that remove the "others" category, select two unseen scenes: "A_SMG_portico" (Scene_A) and "B_SMV_chapel_27to35" (Scene_B) as our test data. We choose *1m×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. At training time, we randomly sample *n* (2,048 or 4,096) points in each block on-the-fly. To train our AE network, we employ ADAM as an optimizer with an initial learning rate 0.001, batch size 16, and weight decay 1e−6, during 250 epochs. The setting of hidden layers in our encoder is the same as DGCNN, but we remove the layers after the max-pooling layer. Similarly, in the semantic segmentation network, we also use ADAM as our optimizer (learning rate 0.01, batch size 16, 250 training epochs). According to the dimension of *Cout*, our shared MLPs is *(Cout+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 accuracy* (OA) and *mean Intersection-over-*

*Union* (mIoU) are calculated on the ArCH Data Set. The method is implemented using PyTorch. All experiments are conducted on an NVIDIA Tesla T4 GPU.

### 4.2 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 an unseen scene (Scene_B) for testing. In Table 1, the overall performances are reported and compared with respect to state-of-the-art methods, which are retrieved from Pierdicca [2]: PointNet [19], PointNet++ [20], DGCNN [8] with 10 scenes, and DGCNN with 15 scenes [2] as training data.

Overall, with only about 10% of training data of state-of-the-art (SOTA) methods in both AE and segmentation network training stages, our model achieves the best results on the ArCH dataset with the same training strategy (only input x, y, z coordinates), as shown in Table 1. The mIoU on Scene_B is 0.408, which also outperforms the 0.353 of SOTA. The semantic segmentation qualitative results Scene_B are shown in Fig. 2, respectively. Our network is able to output smooth predictions.

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

| Networks | Train Scenes | mIoU |
|---|---|---|
| PointNet [17] | 10 scenes | 0.114 |
| PointNet++ [18] | 10 scenes | 0.121 |
| DGCNN [8] | 10 scenes | 0.29 |
| DGCNN [2] | 15 scenes | 0.353 |
| Ours | 3 scenes | 0.408 |

### 4.3 Comparison with Different Training Data Size

To evaluate the impact of training data size (both labeled and unlabeled), we further provided more solid experiments on another unseen Scene_A from four aspects:

- Add one scene ("4_CA_church") as unlabeled training data in AE training stage;
- Add one scene ("4_CA_church") as labeled training data in segmentation network training stage;
- Add one scene ("4_CA_church") both in AE and segmentation network training stage; and
- Decrease the labeled training data size, we just keep one scene ("7_SMV_chapel_24") in the segmentation network training stage.

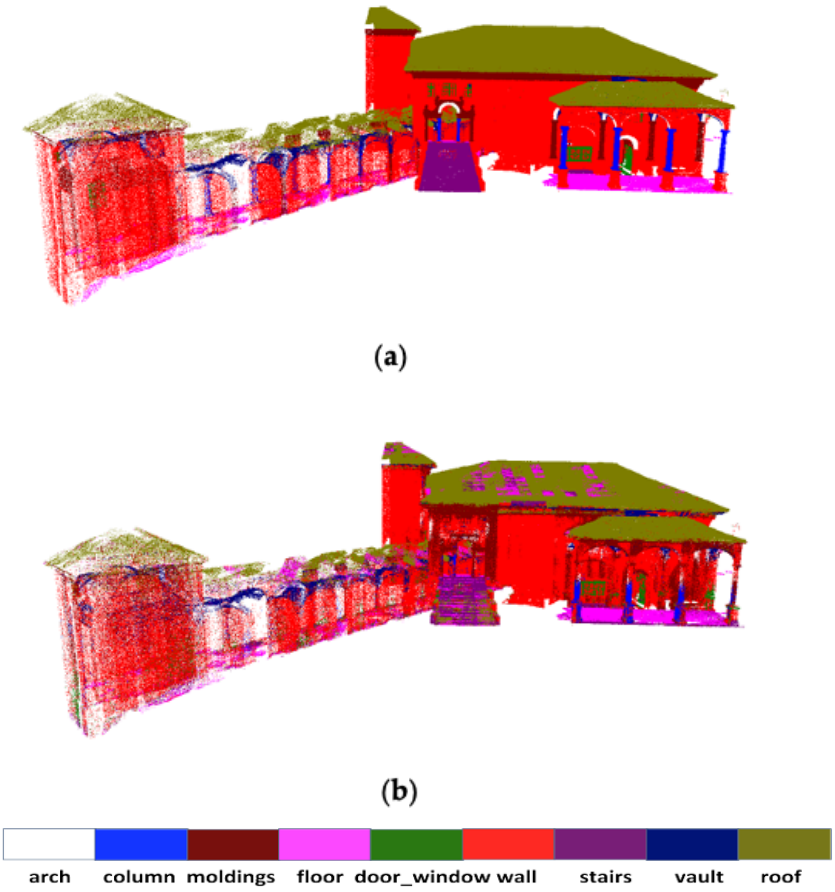


**Fig. 2.** Qualitative results of Scene_B for semantic segmentation. The ground truth (a), and the prediction result (b) of Scene_B on north side. Different colors dente different categories. Scenes from same row are displayed in the same camera viewpoint.

The result of the segmentation result on Scene_A is shown in Table 2. The result here suggested that if we add labeled training data in segmentation network training stage will further improve our performance. For instance, when we add one labeled scene in the segmentation network training stage, our performance will increase by 1% and 3% on the AE pre-trained on three scenes and four scenes, respectively. Furthermore, no increase was detected when we tried to add the unlabeled training data, which infer through training AE from three scenes, we have already been learned a good representation. More importantly, we can further prove our network is label efficient. As even the labeled data was decreased to just one scene (4% of overall labeled data in the supervised method), our overall accuracy still remains at 0.695.

**Table 2.** Analysis of the varying size of training data in the AE training stage and segmentation network training stage. "AE_training_scene" and "Seg_training_scene" denote the number of scenes of ArCH Data Set used in our AE and segmentation training phases, respectively.

| AE_training_scene | Seg_training_scene | OA_Scene_A |
|---|---|---|
| 3_scene | 1_scene | 0.695 |
| 3_scene | 3_scene | 0.747 |
| 3_scene | 4_scene | 0.76 |
| 4_scene | 3_scene | 0.743 |
| 4_scene | 4_scene | 0.772 |

## 5 Conclusions

In this study, we have presented an effective label-efficient unsupervised network for LoD3 buildings' point cloud semantic segmentation. The result in our experiment provide support that our proposed Autoencoder architecture may learn powerful representations from unlabeled data, and these representations can be further used in downstream tasks. Furthermore, the segmentation task of building point clouds obtaining equal or better results with respect to the state of the arts on the basis of only 10% training data from the ArCH Dataset.

In future work, it might be possible to improve the performance by breaking through the input block size and incorporating more features (if available – see [21]) of the input point cloud of buildings while using the very limited amount of labeled training data.

**Acknowledgments:** Financial support from the program of China Scholarships Council (grant number: 201906860014) is acknowledged. We thank Dr. F. Matrone et al. for the ArCH dataset.

## References

1. Brunetaud, X.; Luca, L.D.; Janvier-Badosa, S.; Beck, K.; Al-Mukhtar, M. Application of Digital Techniques in Monument Preservation. *Eur. J. Environ. Civ. Eng.* **2012**, *16*, 543–556.
2. Pierdicca, R.; Paolanti, M.; Matrone, F.; Martini, M.; Morbidoni, C.; Malinverni, E.S.; Frontoni, E.; Lingua, A.M. Point Cloud Semantic Segmentation Using a Deep Learning Framework for Cultural Heritage. *Remote Sens.* **2020**, *12*, 1005.
3. Bosché, F.; Guenet, E. Automating Surface Flatness Control Using Terrestrial Laser Scanning and Building Information Models. *Autom. Constr.* **2014**, *44*, 212–226.
4. Ham, Y.; Golparvar-Fard, M. Three-Dimensional Thermography-Based Method for Cost-Benefit Analysis of Energy Efficiency Building Envelope Retrofits. *J. Comput. Civ. Eng.* **2015**, *29*, B4014009.
5. Fazeli, H.; Samadzadegan, F.; Dadrasjavan, F. Evaluating the Potential of RTK-UAV for Automatic Point Cloud Generation in 3D Rapid Mapping. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XLI-B6*, 221–226.

6. Hu, P.; Yang, B.; Dong, Z.; Yuan, P.; Huang, R.; Fan, H.; Sun, X. Towards Reconstructing 3D Buildings from ALS Data Based on Gestalt Laws. *Remote Sens.* **2018**, *10*.
7. Wang, Q.; Kim, M.-K. Applications of 3D Point Cloud Data in the Construction Industry: A Fifteen-Year Review from 2004 to 2018. *Adv. Eng. Inform.* **2019**, *39*, 306–319.
8. Wang, C.; Hou, S.; Wen, C.; Gong, Z.; Li, Q.; Sun, X.; Li, J. Semantic Line Framework-Based Indoor Building Modeling Using Backpacked Laser Scanning Point Cloud. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 150–166.
9. Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; Guibas, L. Learning Representations and Generative Models for 3D Point Clouds. In Proceedings of the 35th International Conference on Machine Learning; PMLR.org, July 10 2018; Vol. 80, pp. 40–49.
10. Li, C.-L.; Zaheer, M.; Zhang, Y.; Poczos, B.; Salakhutdinov, R. Point Cloud Gan. *ArXiv Prepr. ArXiv181005795* **2018**.
11. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. In Proceedings of the 34th International Conference on Machine Learning; PMLR.org: Sydney, NSW, Australia, August 2017; Vol. 70, pp. 214–223.
12. Groueix, T.; Fisher, M.; Kim, V.G.; Russell, B.C.; Aubry, M. A Papier-Mâché Approach to Learning 3d Surface Generation. In Proceedings of the IEEE conference on computer vision and pattern recognition; IEEE: Salt Lake City, UT, USA, June 18 2018; pp. 216–224.
13. Sauder, J.; Sievers, B. Self-Supervised Deep Learning on Point Clouds by Reconstructing Space. In Proceedings of the Advances in Neural Information Processing Systems; Curran Associates, Inc.: Vancouver, Canada, December 8 2019; Vol. 32, pp. 12962–12972.
14. Yang, Y.; Feng, C.; Shen, Y.; Tian, D. FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation. *2018 IEEECVF Conf. Comput. Vis. PATTERN Recognit. CVPR* 2018, 206–215.
15. Deng, H.; Birdal, T.; Ilic, S. Ppfnet: Global Context Aware Local Features for Robust 3d Point Matching. In Proceedings of the 31st IEEE Conference on Computer Vision and Pattern Recognition; IEEE: Salt Lake City, UT, June 18 2018; pp. 195–205.
16. Deng, H.; Birdal, T.; Ilic, S. Ppf-Foldnet: Unsupervised Learning of Rotation Invariant 3d Local Descriptors. In Proceedings of the 2018 European Conference on Computer Vision (ECCV); SPRINGER INTERNATIONAL PUBLISHING AG: Munich, GERMANY, September 8 2018; Vol. 11209, pp. 602–618.
17. Chen, Z.; Yin, K.; Fisher, M.; Chaudhuri, S.; Zhang, H. Bae-Net: Branched Autoencoder for Shape Co-Segmentation. In Proceedings of the IEEE International Conference on Computer Vision; IEEE: Seoul, SOUTH KOREA, November 27 2019; pp. 8490–8499.
18. 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-B2-2020, 1419–1426.
19. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep Learning on Point Sets for 3d Classification and Segmentation. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition; IEEE: Honolulu, HI, USA, July 21 2017; pp. 652–660.
20. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In Proceedings of the Advances in Neural Information Proceding Systems 30 (NIPS 2017); NIPS: Long Beach, CA, December 4 2017; Vol. 30, pp. 5105–5114.
21. 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.