EXPLORING CROSS-CITY SEMANTIC SEGMENTATION OF ALS POINT CLOUDS

Yuxing Xie[1,2*], Konrad Schindler[3], Jiaojiao Tian[1], Xiao Xiang Zhu[1,2]

[1]Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Oberpfaffenhofen, Germany
{yuxing.xie, jiaojiao.tian, xiaoxiang.zhu}@dlr.de
[2]Data Science in Earth Observation, Technical University of Munich (TUM), Munich, Germany
[3]Photogrammetry and Remote Sensing, ETH Zürich, Switzerland
schindler@ethz.ch

Commission II

KEY WORDS: Point Clouds, Semantic Segmentation, Deep Learning, Transfer Learning, Domain Adaptation

ABSTRACT:

Deep learning models achieve excellent semantic segmentation results for airborne laser scanning (ALS) point clouds, if sufficient training data are provided. Increasing amounts of annotated data are becoming publicly available thanks to contributors from all over the world. However, models trained on a specific dataset typically exhibit poor performance on other datasets. I.e., there are significant *domain shifts*, as data captured in different environments or by distinct sensors have different distributions. In this work, we study this domain shift and potential strategies to mitigate it, using two popular ALS datasets: the ISPRS Vaihingen benchmark from Germany and the LASDU benchmark from China. We compare different training strategies for cross-city ALS point cloud semantic segmentation. In our experiments, we analyse three factors that may lead to domain shift and affect the learning: point cloud density, LiDAR intensity, and the role of data augmentation. Moreover, we evaluate a well-known standard method of domain adaptation, deep CORAL (Sun and Saenko, 2016). In our experiments, adapting the point cloud density and appropriate data augmentation both help to reduce the domain gap and improve segmentation accuracy. On the contrary, intensity features can bring an improvement within a dataset, but deteriorate the generalisation across datasets. Deep CORAL does not further improve the accuracy over the simple adaptation of density and data augmentation, although it can mitigate the impact of improperly chosen point density, intensity features, and further dataset biases like lack of diversity.

1. INTRODUCTION

Unordered point clouds in 3D space have become a standard representation of spatial data, used across a wide range of applications like digital mapping, building information modelling and transportation planning. An important task for many such applications is semantic segmentation, i.e., assigning a semantic class label to every point. As manual labelling is time-consuming and expensive, researchers have for a long time sought to automate that task. Thanks to deep neural networks the accuracy of supervised semantic segmentation has improved significantly in recent years. But deep learning relies on large quantities of annotated reference data. Labelling a sufficiently large and diverse training set for every location and/or every sensor still presents a significant workload and is not scalable. E.g., labelling 2km^2 of ALS data from Dublin (Ireland) into 13 hierarchical multi-level classes took >2,500 person-hours (Zolanvari et al., 2019). More and more annotated ALS data is available in public datasets and benchmarks, labelled according to various nomenclatures. If models trained from such public data (source scenes) could be transferred to other target scenes, per-project annotation would become obsolete. However, in practice almost every project (including the public datasets) is different in terms of source and target environment. Machine learning models, in particular deep learning models, will tend to overfit to the source data and therefore deliver poor results when naively applied to new, previously unseen target data.

Many studies have explored strategies to mitigate domain shift and overfitting (from here on simply termed "training strategies"), so as to employ machine learning when the source and target data follow different distributions. One natural approach, often used for point clouds, is data augmentation to artificially increase the diversity of the training data. Besides, there are also more formal methods for so-called *unsupervised domain adaptation*, meaning statistically inspired strategies to adapt to a new target distributions for which only data, but no ground truth annotations, are available. Unsupervised domain adaptation has recently shown promise in 2D image processing (Wilson and Cook, 2020, Wang and Deng, 2018). Recently, some authors have also started to adopt it for 3D point cloud interpretation (Wu et al., 2019, Luo et al., 2020, Jaritz et al., 2020).

Here, we investigate a number of elementary training strategies for semantic segmentation of ALS point clouds across different cities. To that end, we work with two public ALS datasets from Germany and China, and transfer models between them. In terms of semantic segmentation model, we construct a residual U-net style convolution architecture and employ KPConv (Thomas et al., 2019) as the backbone, due to its proven performance on ALS point clouds (Varney et al., 2020). In our experiments, we analyse three factors that may affect generalisation across cities: *(i)* the point density that is fed into the network; *(ii)* the augmentation method employed to synthetically increase data diversity; and *(iii)* the influence of intensity features (on top of pure point coordinates). Furthermore, inspired by the success of unsupervised, statistical domain adaptation in image processing, we also evaluate the effectiveness of a widely known method, deep CORAL (Sun and Saenko, 2016). We find that elementary measures, like setting a suitable

point density and augmentation, significantly benefit cross-city generalisation, whereas deep CORAL does not further improve over them. On the contrary, intensity complicates generalisation and might best be discarded when the generality of the model is desirable.

2. RELATED WORK

This section reviews recent developments of point cloud semantic segmentation, and associated training strategies aimed at improving generalisation.

2.1 Semantic Segmentation Techniques for Point Clouds

Point cloud semantic segmentation is a supervised classification task. Shallow machine learning classifiers with manually designed features have been the traditional way to address the problem, including for example support vector machines (Zhang et al., 2013), random forests (Weinmann et al., 2015, Hackel et al., 2016), and Adaboost (Wang et al., 2014). The crucial step in this setting is feature extraction. For point clouds, the most common features are basic geometric properties, height-based features if the gravity direction is known, and features based on eigenvalues of the local point distribution (Weinmann et al., 2015, Hackel et al., 2016, Xu et al., 2019). Besides, graph-based neighborhood models such as conditional random fields have been utilised as a post-processing step to smooth the per-point labels (Landrieu et al., 2017).

In recent years, deep learning has become the dominant approach for point cloud analysis. It requires no feature engineering and achieves better performance for many tasks including semantic segmentation. Deep learning-based methods can be sorted into three main categories: image-based, voxel-based, and point-based (Xie et al., 2020). Image-based methods project point clouds to image-like 2D representations, then apply 2D convolutions to them (Boulch et al., 2018, Yang et al., 2017). Their main shortcoming is that they do not fully exploit the 3D geometry. Another solution is to discretise the point cloud to a regular, ordered voxel grid and then use regular 3D convolutions (Tchapmi et al., 2017). Voxel-based deep learning is time-consuming and memory-hungry, so most methods now exploit the sparsity of the voxel space and employ sparse convolutions (Choy et al., 2019, Graham et al., 2018) that only operate on non-empty voxels. Point-based methods include different techniques that make it possible to operate directly on the point cloud. They mainly differ by the way they define the kernels. The pioneering PointNet (Qi et al., 2017a) simply replaces convolution with a more general multi-layer perceptron (MLP). However, PointNet only learns global features, but not local ones. To overcome this limitation, PointNet++ was proposed, which captures local features via an image pyramid-like hierarchical aggregation (Qi et al., 2017b). Several recent works instead design explicit convolution kernels for point clouds. Among them, KPConv (Thomas et al., 2019) has demonstrated high efficiency and good performance for point cloud semantic segmentation, notably for large, mobile-mapping type outdoor scenarios.

Also for ALS point clouds, deep learning is increasingly being the method of choice. PointNet/PointNet++ has been widely utilised as network backbone, since it appeared earlier (Yousefhussien et al., 2018, Lin et al., 2020, Huang et al., 2020). More recently, PointCNN (Arief et al., 2019), graph convolutions (Wen et al., 2021), spatially sparse convolution (Schmohl and Sörgel, 2019), and KPConv (Varney et al., 2020, Lin et al., 2021) have also been adopted and have achieved good results on ALS data.

2.2 Training Strategies for Point Clouds

Data augmentation is an elementary training strategy for deep learning tasks (Shorten and Khoshgoftaar, 2019). Rotation, scaling, symmetry, random noise, and randomly removing points are common augmentation operations for point clouds (Chaton et al., 2020, Thomas et al., 2019). By synthetically increasing the diversity of patterns in the data, they can help to prevent overfitting when training data is limited. Recently, it has also been proposed to learn the data augmentation (Chen et al., 2020, Li et al., 2020).

While, at first glance, deep learning continues to set the state of the art on many public benchmarks, the situation in reality is more complex. The excellent performance is achieved only when trained on data from the same dataset, i.e., recorded in the same (or a very similar) environment with the same sensor setup. Effectively, the semantic segmentation easily overfits to the unique, specific conditions, so that domain shifts exist even between seemingly similar datasets. From this extreme specialisation, due to the high capacity of deep networks, arises a need for domain adaptation. This was first observed for 2D images (Wilson and Cook, 2020, Wang and Deng, 2018), but more recently also explored for various 3D point cloud analysis tasks. In the setting of self-driving scenarios, (Langer et al., 2020, Wu et al., 2019) first project LiDAR point clouds to images and then apply imaged-based domain adaptation on them to aid semantic segmentation. For the important, point cloud-specific domain shift of density differences, (Yi et al., 2020) formulate domain adaptation as a complete-and-label problem. A voxel completion network is proposed to fill in gaps between the source and target data, so they have similar density. xMUDA (Jaritz et al., 2020) utilises cross-modal learning with images to address the domain shift between point clouds in road scenes. Mutual information from cross-modal features is shown to improve semantic segmentation. These works are aimed at point cloud semantic segmentation, but the domain adaptation strategies do not directly operate on uni-modal point clouds. Towards direct point cloud domain adaptation, (Luo et al., 2020) propose a framework that jointly aligns data and feature distributions of MLS point clouds, with a small network to refine the elevation of target data and an adversarial network to align the features. (Peng et al., 2020) also address domain adaptation with adversarial learning and demonstrate their method for two similar ALS datasets (captured in the same region) and for an ALS dataset and a MLS dataset.

3. METHODOLOGY

We are not aware of any systematic comparison of different domain adaptation strategies for point clouds. In this work, we set up a state-of-the-art semantic segmentation pipeline, with KPConv as the backbone, and compare several basic and practical training strategies. We run experiments under different conditions in terms of input point cloud density, data augmentation, and the use of intensity features. Beyond these "hand-designed" manipulations of the input data, we also test a classical, well-established domain adaptation algorithm, deep CORAL (Sun and Saenko, 2016).

3.1 Semantic Segmentation by KPConv

KPConv (Thomas et al., 2019) is a direct point cloud convolution operator, based on the idea to approximate the continuous convolution operator in a local, spherical 3D neighbourhood. Let p_i and f_i be points from a point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$ and their corresponding features from $\mathcal{F} \in \mathbb{R}^{N\times D}$. The point convolution at a point $p \in \mathbb{R}^{N\times 3}$ is denoted as follows:

$$(\mathcal{F} * g)(p) = \sum_{p_i \in \mathcal{N}_p} g(p_i - p) f_i \ , \tag{1}$$

where g is the kernel function of KPConv. $\mathcal{N}_p = \{p_i \in \mathcal{P} \mid \|p_i - p\| \leq r\}$ represent neighbour points of p within a fixed radius $r \in \mathbb{R}$. In KPConv, g takes the those neighbours centered on p as input to the convolution. The domain g is defined as a 3D sphere:

$$\mathcal{B}_r^3 = \{q \in \mathbb{R}^3 \mid \|q\| \leq r\} \ , \tag{2}$$

where $q_i = p_i - p$.

KPConv provides two kernel versions, a rigid and a deformable one. In the former, the kernel points are distributed in a fixed layout within the sphere, whereas the deformable one allows for learned shifts of their positions. In practice, deformable KPConv does not outperform the rigid version on scenes lacking diversity such as ALS point clouds (Thomas et al., 2019, Lin et al., 2021) but requires more GPU memory and run time. Hence, we use rigid KPConv in this work.

In our experiments we use the authors' original PyTorch-based implementation (`https://github.com/HuguesTHOMAS/KPConv-PyTorch`). Our semantic segmentation embeds KPConv in a U-net architecture (Ronneberger et al., 2015), following ResNet block design (He et al., 2016) in the encoder. Each convolution layer in this network is followed by batch normalization (BN) (Ioffe and Szegedy, 2015) and a Leaky ReLU activation (Maas et al., 2013). Grid sampling is employed as the sub-sampling strategy to reduce the density and increase the context along the layers. Hence, the data in each layer are the center points of regularly spaced grid cells. The convolution sphere radius r_i for the i-th layer is adjusted by a corresponding factor α, i.e.,

$$r_i = \alpha l_i \ , \tag{3}$$

where l_i and r_i denote the grid size and convolution radius in the i-th layer. Due to limited GPU RAM, the size of the input sphere, and thus the size of the receptive field in the network, are dependant on the grid spacing of first sub-sampling: wider spacing causes stronger down-sampling (with potential loss of information), but on the other hand allows for a larger receptive field (with more context).

3.2 Domain Adaptation by Deep CORAL

Correlation alignment is a popular, representative statistical algorithm for unsupervised domain adaptation. It tries to minimise the domain shift by aligning the second-order statistics of source and target feature distributions, which can be done without any labels for the target domain. We adopt a deep version of correlation alignment, named deep CORAL (Sun and Saenko, 2016), which can be directly integrated into any neural network architecture.

Deep CORAL imposes the correlation alignment as a soft constraint, via the loss function. The CORAL loss is defined as the distance between the second-order statistics in the source and target feature matrices:

$$\mathcal{L}_{CORAL} = \frac{1}{4d^2} \|C_S - C_T\|^2 \ , \tag{4}$$

with C_S and C_T the source and target feature covariance matrices. During training, $\mathcal{L}_{CORAL}$ is minimised with mini-batches from the training set of the source domain and the target domain. The intuition behind deep CORAL is to "deform" the source and target feature distributions such that they match up to second-order statistics, assuming that the class-conditional distributions will then match better, too.

Multiple CORAL loss functions over different activation layers within the network can be combined, and added to the semantic segmentation loss $\mathcal{L}_{seg}$, to obtain a joint loss function:

$$\mathcal{L}_{total} = \mathcal{L}_{seg} + \sum_{i=1}^{t} \lambda^{(i)} \mathcal{L}_{CORAL}^{(i)} \ , \tag{5}$$

where t is the number of layer-wise CORAL losses and $\lambda^{(i)}$ is the weight coefficient of the i-th CORAL loss.

We train our KPConv-based residual U-net with standard cross-entropy loss for $\mathcal{L}_{seg}$. Empirically, CORAL terms for lower layers did not have much influence, so we only align the feature maps of the last activation layer with a single CORAL loss $\mathcal{L}_{CORAL}$. The network architecture is depicted in Figure 1. x_{Si} and x_{Ti} represent input source and target samples, respectively. y_{Si} is the set of input labels (given only for the input source data).

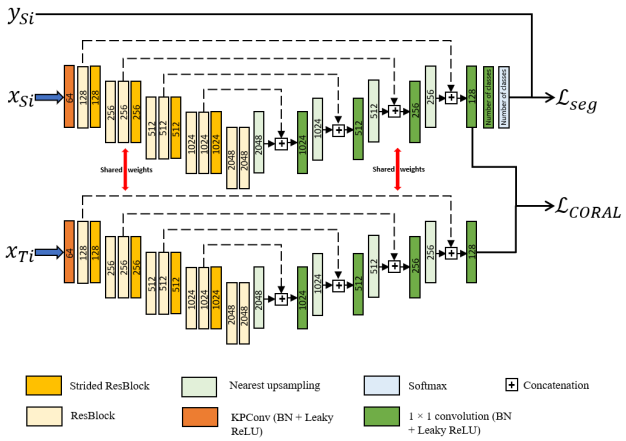


Figure 1. Illustration of the network structure.

4. EXPERIMENTS

4.1 Datasets

Two ALS point cloud benchmark datasets are adopted for the evaluation: the ISPRS Vaihingen benchmark (Cramer, 2010, Rottensteiner et al., 2012) and LASDU (Ye et al., 2020). ISPRS Vaihingen was captured with a Leica ALS50 system from an average flying height of ≈500m in Vaihingen, Germany; while LASDU was captured with a Leica ALS70 system at an average flying height of ≈1200m in a town of northwest China, which is a part of the HiWATER (Heihe Watershed Allied Telemetry

Experimental Research) project (Li et al., 2013). The original labels of LASDU (6 classes, including a rejection class "unclassified") and ISPRS Vaihingen (9 classes) are different. However, for evaluation purposes the label sets of the two domains should match. Hence, we map the 9 classes of ISPRS Vaihingen to the 6 classes of LASDU, following the classification rule of LASDU. The "powerline" class of ISPRS Vaihingen is mapped to "others", as no powerlines are labelled in LASDU. Points with label "others" are used for training, but are ignored in the quantitative evaluation. Table 1 shows the mapping between the two label sets.

Classes of ISPRS Vaihingen	Mapped classes	Classes of LASDU
Impervious surfaces	Ground	Ground
Roof, facade	Building	Building
Tree	Tree	Tree
Low vegetation, bushes	Low vegetation	Low vegetation
Car, fence/hedge	Artifact	Artifact
Powerline	Others (Ignored)	Unclassified

Table 1. Class mapping for ISPRS Vaihingen and LASDU point cloud datasets.

The ISPRS Vaihingen benchmark contains defined training and test portions. LASDU consists of four portions. It is recommended to use files 2 and 3 as the training data, and 1 and 4 as the test set for semantic labelling (Ye et al., 2020). Table 2 shows the number of points in each class for both datasets.

Mapped classes	ISPRS Vaihingen		LASDU	
	Training	Test	Training	Test
Ground	193,723	101,986	704,425	637,257
Building	179,295	120,272	508,479	395,109
Tree	135,173	54,226	204,775	108,466
Low vegetation	228,455	123,508	210,495	192,051
Artifact	16,684	11,130	66,738	53,061
Others (ignored)	546	600	33,206	12,659

Table 2. Point distributions in ISPRS Vaihingen and LASDU datasets.

4.2 Experiment Setup and Evaluation Metrics

Three experiments have been run. Each experiment includes six cases, obtained by using IPSRS Vaihingen (VH) or LASDU (LS) as source and target datasets and switching on (w DC) or off (w/o DC) correlation alignment with deep CORAL. Writing $A \rightarrow B$ to denote training on dataset A and testing on B, the six cases are: $VH \rightarrow VH$, $LS \rightarrow VH$ (w/o DC), $LS \rightarrow VH$ (w DC), $LS \rightarrow LS$, $VH \rightarrow LS$ (w/o DC), and $VH \rightarrow LS$ (w DC). In section 4.3, the influence of input grid size, i.e., point density is investigated. In section 4.4, the role of data augmentation is assessed. Section 4.5 explores how intensity features affect accuracy and generalisation.

In all experiments, the batch size is set to 8 and α in equation 3 is set to 2.5. Training with stochastic gradient descent (SGD) is run for 60,000 iterations, at which point the loss function has always converged. The initial learning rate is set to 0.01 and decays at a rate of 0.1 every 7,500 iterations when training on VH, respectively decays at the same rate every 12,500 iterations when training on the larger LS. Data augmentation, including random rotation around the z-axis, random scaling, random symmetry about the x-axis, and Gaussian noise, is always applied except for the dedicated experiments without data augmentation in Section 4.4. The scaling factor is randomized within [0.8, 1.2]. The standard deviation σ of Gaussian noise is set to 5cm. When using correlation alignment the weight coefficient $\lambda = 1.0$. Since KPconv can only operate on limited (spherical) subsets of a large point cloud, we adopt the authors' voting strategy during testing and average the estimated class probabilities of each point, obtained from at least 20 different sphere samples. Training and testing were performed on a Geforce RTX 2080 Ti GPU with 11GB RAM.

Following the ISPRS Vaihingen benchmark, all results are evaluated in terms of overall accuracy (*OA*) and $F1$ score.

$$F1_i = \frac{2TP_i}{2TP_i + FP_i + FN_i} \,, \tag{6}$$

$$OA = \sum_{i=1}^{n} \left(\frac{TP_i}{TP_i + TN_i + FP_i + FN_i}\right) \,, \tag{7}$$

where i is the class index and *TP* refers to the number of true positives, *FP* the false positives, *TN* the true negatives, *FN* the false negatives.

4.3 Experiment I: Evaluation of Input Point Density

As explained in Section 3.1, an important hyper-parameter of KPConv is the input point cloud density, i.e., in our setup defined via the grid size. A trade-off has to be found between input density and receptive field size. We test three settings for the grid spacing l: 0.25m, 0.5m and 0.8m. These correspond to input context spheres of radius 13m, 25m and 40m, respectively. When l is bigger, the network can see a larger region, but with sparser sampling and thus less geometric details and less information on small objects. Data augmentation as described in Section 4.2 is used in all runs. LiDAR intensities are not used.

Table 3 shows that the best generalisation results, for both $VH \rightarrow LS$ and $LS \rightarrow VH$, are achieved when $l = 0.5m$. Still, a clear domain shift exists. The *OA* of $LS \rightarrow VH$ (cross-city) is 9 percent points (pp) lower than that of $VH \rightarrow VH$ (within-city). Similarly, the *OA* of $VH \rightarrow LS$ is 10.5 pp lower than that of $LS \rightarrow LS$. The $F1$ scores are also lower across all classes. Deep CORAL has a *negative* impact on the overall accuracy, due to more mistakes on the large ground and vegetation classes. We note that in case of non-optimal sample density l deep CORAL has a mild positive effect. In $VH \rightarrow LS$ ($l = 0.25$m), $LS \rightarrow VH$ ($l = 0.25$m) and $LS \rightarrow VH$ ($l = 0.8$m), the *OA* increases between 0.7 and 2.2 pp. It seems that aligning the feature distributions somehow mitigates the domain difference for features derived from points sampled at sub-optimal density. An example for $LS \rightarrow VH$ is shown in Figure 2. Note that at overly coarse sampling (0.8m) large areas of ground points along roads are misclassified as low vegetation, and deep CORAL corrects these errors. However, at proper sampling density of $l = 0.5$m, the mistakes do not happen in the first place, so there is no room for improvement.

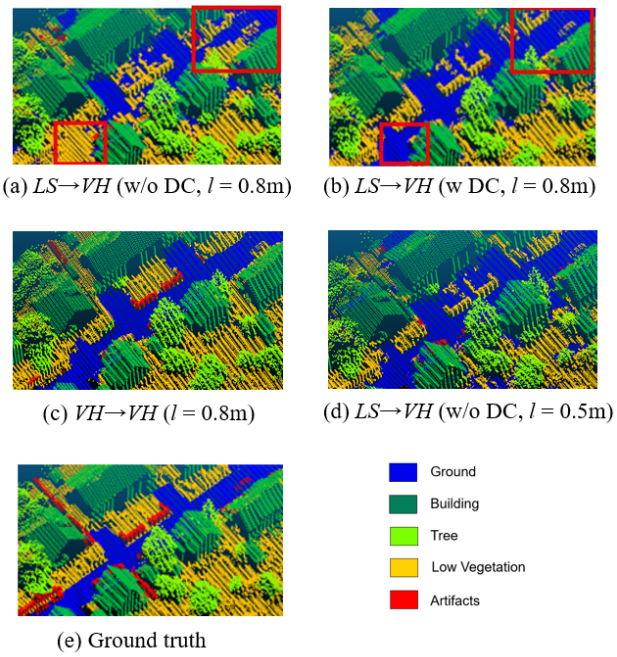
4.4 Experiment II: Evaluation of Data Augmentation

Deep neural networks require large training sets to avoid overfitting. Data augmentation is a technique to synthetically increase the sample size by manipulating existing samples in plausible ways. Table 4 presents results for the same settings as above for grid spacing $l = 0.5m$, but without data augmentation, compared to Table 3. Again, LiDAR intensity values are not used.

Comparing the numbers in Tables 4 and 3, we see that cross-city generalisation suffers if data augmentation is disabled. The

Input grid size	Settings	*OA*	Ground	Building	Tree	Low vegetation	Artifact
$l = 0.25$m	$VH \rightarrow VH$	0.8421	0.8427	0.9362	0.8272	0.7830	0.4306
	$LS \rightarrow VH$ (w/o DC)	0.7433	0.7279	0.8683	0.7475	0.6765	0.0145
	$LS \rightarrow VH$ (w DC)	0.7536	0.7862	0.8722	0.7986	0.5792	0.1002
	$LS \rightarrow LS$	0.8575	0.8892	0.9679	0.8670	0.5979	0.4025
	$VH \rightarrow LS$ (w/o DC)	0.6809	0.7696	0.8099	0.8458	0.3677	0.0904
	$VH \rightarrow LS$ (w DC)	0.6882	0.7616	0.8323	0.8627	0.3980	0.1172
$l = 0.5$m	$VH \rightarrow VH$	0.8724	0.9021	0.9504	0.8376	0.8165	0.4735
	$LS \rightarrow VH$ (w/o DC)	0.7812	0.8017	0.9038	0.7859	0.6705	0.3163
	$LS \rightarrow VH$ (w DC)	0.7290	0.7348	0.9209	0.8244	0.4509	0.2469
	$LS \rightarrow LS$	0.8683	0.9044	0.9642	0.8652	0.6382	0.4744
	$VH \rightarrow LS$ (w/o DC)	0.7638	0.8188	0.8892	0.8653	0.3935	0.1246
	$VH \rightarrow LS$ (w DC)	0.7181	0.7576	0.9087	0.8585	0.4086	0.0844
$l = 0.8$m	$VH \rightarrow VH$	0.8646	0.8902	0.9456	0.8322	0.8069	0.4730
	$LS \rightarrow VH$ (w/o DC)	0.7422	0.7782	0.8508	0.7368	0.6145	0.3079
	$LS \rightarrow VH$ (w DC)	0.7641	0.8026	0.8717	0.7548	0.8717	0.1581
	$LS \rightarrow LS$	0.8537	0.8860	0.9664	0.8665	0.5361	0.4306
	$VH \rightarrow LS$ (w/o DC)	0.7409	0.8088	0.8696	0.8699	0.3981	0.1232
	$VH \rightarrow LS$ (w DC)	0.6462	0.6348	0.9329	0.8573	0.3200	0.1294

Table 3. Results with different input grid spacing. Data augmentation was carried out but intensity features were not used.



(a) *LS→VH* (w/o DC, $l = 0.8$m) (b) *LS→VH* (w DC, $l = 0.8$m)

(c) *VH→VH* ($l = 0.8$m) (d) *LS→VH* (w/o DC, $l = 0.5$m)

(e) Ground truth

Figure 2. Example visualisation (experiment I).

OA decreased by 8 pp and 6 pp for $LS \rightarrow VH$ and $VH \rightarrow LS$, respectively. Deep CORAL manages to mitigate that performance drop, improving *OA* by 2.8 pp and 4.2 pp, correspondingly. However, its impact is again class-specific and the $F1$ scores of several classes are decreased significantly. In particular, in the $VH \rightarrow LS$ test the $F1$ scores for trees and buildings are lower than before, and the score for artifacts even drops to 0.

4.5 Experiment III: Evaluation of Intensity

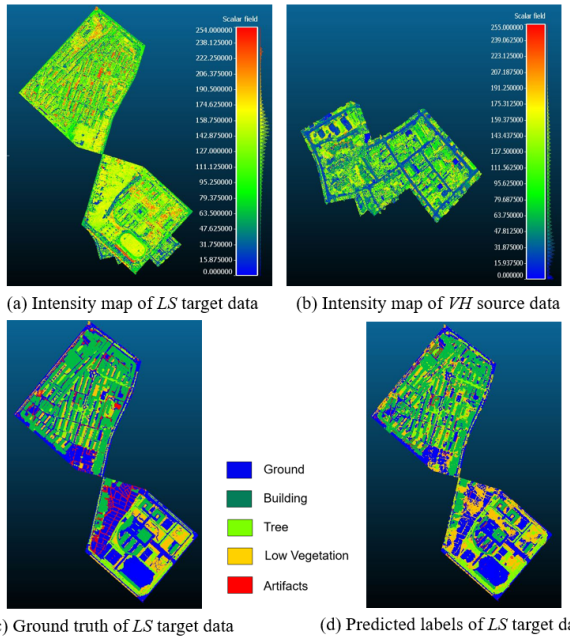
This experiment additionally assesses the role of intensity features, which one might expect to also influence cross-city generalisation. In the original ISPRS Vaihingen and LASDU datasets, the LiDAR return intensities have already been scaled to [0, 255], so we directly concatenate them with the 3D point coordinates and feed the resulting 4D points to the network. Table 5 shows segmentation results with added intensities, at density $l = 0.5$m, with data augmentation, compared to Table 3. Intensities do improve the within-city results slightly for $VH \rightarrow VH$ and more significantly for $LS \rightarrow LS$, especially the separation of ground and low vegetation. This makes sense, as the two classes are difficult to distinguish based only on geometric features – they both share low height and mostly horizontal, planar layout. However, the performance for both $LS \rightarrow VH$ and $VH \rightarrow LS$ drops significantly when using also intensity. *OA* decreases by 5 pp for $LS \rightarrow VH$, and even by 27 pp for $VH \rightarrow LS$. As can be seen in Figure 3, the classifier trained on VH misclassifies large regions of ground in LS as low vegetation. One can see that the intensity distributions of the two datasets differ significantly (Figure 3a and 3b), with almost all ground points in VH having low intensities <45, while in LS the intensities concentrate in the range [90, 150]. Thus, involving intensities widens the domain gap. As in Section 4.4, domain adaptation slightly mitigates the drop in OA, but does not resolve the main issues.

Settings	*OA*	Ground	Building	Tree	Low vegetation	Artifact
$VH \rightarrow VH$	0.8329	0.8537	0.9275	0.8029	0.7665	0.4360
$LS \rightarrow VH$ (w/o DC)	0.7005	0.7885	0.7453	0.6361	0.6468	0.2725
$LS \rightarrow VH$ (w DC)	0.7281	0.7612	0.8800	0.7673	0.5564	0.2012
$LS \rightarrow LS$	0.8608	0.8921	0.9692	0.8701	0.5750	0.4578
$VH \rightarrow LS$ (w/o DC)	0.7032	0.7749	0.8370	0.8371	0.3758	0.0886
$VH \rightarrow LS$ (w DC)	0.7455	0.8537	0.7626	0.6184	0.7499	0

Table 4. Results without data augmentation ($l = 0.5$m). Intensity features were not used.

Settings	*OA*	Ground	Building	Tree	Low vegetation	Artifact
$VH \rightarrow VH$	0.8775	0.9153	0.9454	0.8291	0.8247	0.5595
$LS \rightarrow VH$ (w/o DC)	0.7302	0.7749	0.8549	0.7499	0.5363	0.2891
$LS \rightarrow VH$ (w DC)	0.7433	0.7571	0.9269	0.7934	0.5091	0.3874
$LS \rightarrow LS$	0.8939	0.9225	0.9686	0.8736	0.7461	0.4517
$VH \rightarrow LS$ (w/o DC)	0.4897	0.2649	0.8748	0.8608	0.3241	0.0981
$VH \rightarrow LS$ (w DC)	0.5106	0.3634	0.8434	0.8217	0.3471	0

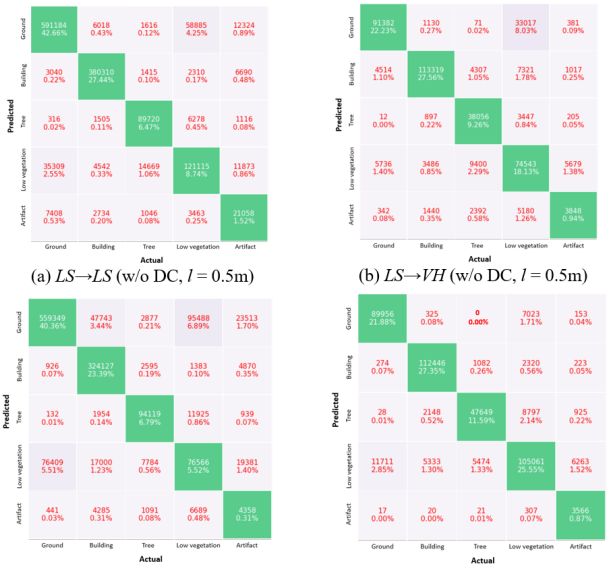
Table 5. Results with intensity ($l = 0.5$m). Data augmentation was carried out.



(a) Intensity map of *LS* target data (b) Intensity map of *VH* source data

(c) Ground truth of *LS* target data (d) Predicted labels of *LS* target data

Figure 3. The inference result visualisation (experiment III).



(a) *LS*→*LS* (w/o DC, $l = 0.5$m) (b) *LS*→*VH* (w/o DC, $l = 0.5$m)

(c) *VH*→*LS* (w/o DC, $l = 0.5$m) (d) *VH*→*VH* (w/o DC, $l = 0.5$m)

Figure 4. Confusion matrices.

4.6 Further Analysis

Confusion matrix. To analyse the class-wise effect of the domain shift, we inspect confusion matrices of $VH \rightarrow LS$, $LS \rightarrow LS$, $LS \rightarrow VH$, and $VH \rightarrow VH$, for the best-performing spacing $l = 0.5$m. In Figure 4 it can be seen that an obvious issue in both the $VH \rightarrow LS$ and $LS \rightarrow VH$ results is indeed the confusion between low vegetation and ground, due to their similar geometric characteristics. Within one dataset intensity can to some degree compensate the mistake, but it even extends the gap between the two point clouds. Arguably, the two large classes associated with the ground are the main challenge for domain adaptation between VH and LS.

Minority class. Minority classes, in our case especially artifacts, appear to be negatively affected by domain adaptation

with deep CORAL. In extreme cases, e.g., $VH \rightarrow LS$ (w DC) in Tables 4 and 5, the $F1$ score even drops to 0. There are multiple potential reasons for this behaviour. On the one hand, the deep CORAL loss is calculated without taking into account the classes (which are unknown for the target distribution). Rare classes will therefore have almost no influence on the adaptation. And the resulting warping of the feature space, optimised to accommodate the dominant classes, can be counter-productive. On the other hand, an aggravating factor might also be that the artifact class contains a too large variety of objects in LASDU, including walls, fences, light poles, vehicles, etc. The corresponding, wide and diffuse set of features each valid only for few examples might lead to a complicated feature distribution not sufficiently characterised by the second-order statistics.

5. CONCLUSION

We have empirically investigated cross-city learning of semantic segmentation for ALS point clouds, using example datasets from Germany and China. Three factors were considered that all affect the results, and a representative, generic domain adaptation strategy was evaluated. Our experiments indicate that data augmentation and proper choice of the input density play an important role and can significantly boost generalisation performance. On the contrary, LiDAR intensities exhibited stronger differences between datasets and might better be avoided, as they negatively impact performance across datasets. As for unsupervised, statistical domain adaptation with deep CORAL, we found that when training conditions are not optimal (e.g., intensities present or point density not well chosen) it brings a mild improvement, however it did not resolve the important problems and affected different classes rather unevenly. Elementary design choices, like choosing the right input density and using data augmentation, were much more important to achieve acceptable generalisation. Surprisingly, when those were set to support generalisation in the best possible way, correlation alignment even deteriorated the result by reinforcing class-dependent biases.

In future work we would like to develop domain adaptation methods that are better suited for semantic segmentation of ALS point clouds. Another important step for future research will be to introduce datasets with larger class nomenclatures, to analyse and tackle domain adaptation for application scenarios with more fine-grained semantics.

ACKNOWLEDGEMENTS

We thank the the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF), Germany and the Institute of Tibetan Plateau Research, Chinese Academy of Sciences, China, for providing the datasets. We thank Hugues Thomas for providing the open source code of KPConv.

REFERENCES

Arief, H. A., Indahl, U. G., Strand, G.-H., Tveite, H., 2019. Addressing overfitting on point cloud classification using Atrous XCRF. *ISPRS Journal of Photogrammetry and Remote Sensing*, 155, 90–101.

Boulch, A., Guerry, J., Le Saux, B., Audebert, N., 2018. SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Computers & Graphics*, 71, 189–198.

Chaton, T., Chaulet, N., Horache, S., Landrieu, L., 2020. Torchpoints3d: A modular multi-task framework for reproducible deep learning on 3d point clouds. *2020 International Conference on 3D Vision (3DV)*, 190–199.

Chen, Y., Hu, V. T., Gavves, E., Mensink, T., Mettes, P., Yang, P., Snoek, C. G. M., 2020. Pointmixup: Augmentation for point clouds. *European Conference on Computer Vision*, Springer, 330–345.

Choy, C., Gwak, J., Savarese, S., 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3075–3084.

Cramer, M., 2010. The DGPF-test on digital airborne camera evaluation overview and test design. *PFG Photogrammetrie, Fernerkundung, Geoinformation*, 73–82.

Graham, B., Engelcke, M., Van Der Maaten, L., 2018. 3d semantic segmentation with submanifold sparse convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9224–9232.

Hackel, T., Wegner, J. D., Schindler, K., 2016. Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 3, 177–184.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, R., Xu, Y., Hong, D., Yao, W., Ghamisi, P., Stilla, U., 2020. Deep point embedding for urban classification using ALS point clouds: A new perspective from local to global. *ISPRS Journal of Photogrammetry and Remote Sensing*, 163, 62–81.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, PMLR, 448–456.

Jaritz, M., Vu, T.-H., Charette, R. d., Wirbel, E., Pérez, P., 2020. xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12605–12614.

Landrieu, L., Raguet, H., Vallet, B., Mallet, C., Weinmann, M., 2017. A structured regularization framework for spatially smoothing semantic labelings of 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 132, 102–118.

Langer, F., Milioto, A., Haag, A., Behley, J., Stachniss, C., 2020. Domain Transfer for Semantic Segmentation of LiDAR Data using Deep Neural Networks. *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and System (IROS)*.

Li, R., Li, X., Heng, P.-A., Fu, C.-W., 2020. Pointaugment: an auto-augmentation framework for point cloud classification. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6378–6387.

Li, X., Cheng, G., Liu, S., Xiao, Q., Ma, M., Jin, R., Che, T., Liu, Q., Wang, W., Qi, Y. et al., 2013. Heihe watershed allied telemetry experimental research (HiWATER): Scientific objectives and experimental design. *Bulletin of the American Meteorological Society*, 94(8), 1145–1160.

Lin, Y., Vosselman, G., Cao, Y., Yang, M. Y., 2020. Active and incremental learning for semantic ALS point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169, 73–92.

Lin, Y., Vosselman, G., Cao, Y., Yang, M. Y., 2021. Local and global encoder network for semantic segmentation of Airborne laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 176, 151–168.

Luo, H., Khoshelham, K., Fang, L., Chen, C., 2020. Unsupervised scene adaptation for semantic segmentation of urban mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169, 253–267.

Maas, A. L., Hannun, A. Y., Ng, A. Y., 2013. Rectifier nonlinearities improve neural network acoustic models. *Proceedings of International Conference on Machine Learning*.

Peng, S., Xi, X., Wang, C., Xie, R., Wang, P., Tan, H., 2020. Point-Based Multilevel Domain Adaptation for Point Cloud Segmentation. *IEEE Geoscience and Remote Sensing Letters*.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017b. Pointnet++ deep hierarchical feature learning on point sets in a metric space. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 5105–5114.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, Springer, 234–241.

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., Breitkopf, U., 2012. The ISPRS benchmark on urban object classification and 3D building reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1*, 1(1), 293–298.

Schmohl, S., Sörgel, U., 2019. Submanifold Sparse Convolutional Networks for Semantic Segmentation of Large-Scale ALS Point Clouds. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, 77–84.

Shorten, C., Khoshgoftaar, T. M., 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 1–48.

Sun, B., Saenko, K., 2016. Deep coral: Correlation alignment for deep domain adaptation. *European conference on computer vision*, Springer, 443–450.

Tchapmi, L., Choy, C., Armeni, I., Gwak, J., Savarese, S., 2017. Segcloud: Semantic segmentation of 3d point clouds. *2017 international conference on 3D vision (3DV)*, IEEE, 537–547.

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L. J., 2019. Kpconv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6411–6420.

Varney, N., Asari, V. K., Graehling, Q., 2020. Dales: a large-scale aerial lidar data set for semantic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 186–187.

Wang, M., Deng, W., 2018. Deep visual domain adaptation: A survey. *Neurocomputing*, 312, 135–153.

Wang, Z., Zhang, L., Fang, T., Mathiopoulos, P. T., Tong, X., Qu, H., Xiao, Z., Li, F., Chen, D., 2014. A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5), 2409–2425.

Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105, 286–304.

Wen, C., Li, X., Yao, X., Peng, L., Chi, T., 2021. Airborne LiDAR point cloud classification with global-local graph attention convolution neural network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173, 181–194.

Wilson, G., Cook, D. J., 2020. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5), 1–46.

Wu, B., Zhou, X., Zhao, S., Yue, X., Keutzer, K., 2019. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 4376–4382.

Xie, Y., Tian, J., Zhu, X. X., 2020. Linking points with labels in 3D: A review of point cloud semantic segmentation. *IEEE Geoscience and Remote Sensing Magazine*, 8(4), 38–59.

Xu, Y., Ye, Z., Yao, W., Huang, R., Tong, X., Hoegner, L., Stilla, U., 2019. Classification of LiDAR point clouds using supervoxel-based detrended feature and perception-weighted graphical model. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 72–88.

Yang, Z., Jiang, W., Xu, B., Zhu, Q., Jiang, S., Huang, W., 2017. A convolutional neural network-based 3D semantic labeling method for ALS point clouds. *Remote Sensing*, 9(9), 936.

Ye, Z., Xu, Y., Huang, R., Tong, X., Li, X., Liu, X., Luan, K., Hoegner, L., Stilla, U., 2020. LASDU: A Large-Scale Aerial LiDAR Dataset for Semantic Labeling in Dense Urban Areas. *ISPRS International Journal of Geo-Information*, 9(7), 450.

Yi, L., Gong, B., Funkhouser, T., 2020. Complete & label: A domain adaptation approach to semantic segmentation of LiDAR point clouds. *arXiv preprint arXiv:2007.08488*.

Yousefhussien, M., Kelbe, D. J., Ientilucci, E. J., Salvaggio, C., 2018. A multi-scale fully convolutional network for semantic labeling of 3D point clouds. *ISPRS journal of photogrammetry and remote sensing*, 143, 191–204.

Zhang, J., Lin, X., Ning, X., 2013. SVM-based classification of segmented airborne LiDAR point clouds in urban areas. *Remote Sensing*, 5(8), 3749–3775.

Zolanvari, S. M. I., Ruano, S., Rana, A., Cummins, A., da Silva, R. E., Rahbar, M., Smolic, A., 2019. Dublincity: Annotated lidar point cloud and its applications. *BMVC*.