LEARNING FROM THE PAST: CROWD-DRIVEN ACTIVE TRANSFER LEARNING FOR SEMANTIC SEGMENTATION OF MULTI-TEMPORAL 3D POINT CLOUDS

M. Kölle*, V. Walter, U. Soergel

Institute for Photogrammetry, University of Stuttgart, Germany - (michael.koelle, volker.walter, uwe.soergel)@ifp.uni-stuttgart.de

Commission II, WG II/6

KEY WORDS: Active Learning, Transfer Learning, Domain Adaptation, Crowdsourcing, Multi-Temporality, 3D Point Clouds, Semantic Segmentation.

ABSTRACT:

The main bottleneck of machine learning systems, such as convolutional neural networks, is the availability of labeled training data. Hence, much effort (and thus cost) is caused by setting up proper training data sets. However, models trained on specific data sets often perform unsatisfactorily when used to derive predictions for another (yet related) data set. We aim to overcome this problem by employing active learning to iteratively adapt an existing classifier to another domain. Precisely, we are concerned with semantic segmentation of 3D point clouds of multiple epochs. We first establish a Random Forest classifier for the first epoch of our data set and adapt it for successful prediction to two more temporally disjoint point clouds of the same but extended area. The point clouds, which are part of the newly introduced Hessigheim 3D benchmark data set, incorporate different characteristics with respect to the acquisition date and sensor configuration. We demonstrate that our workflow for domain adaptation is designed in such a way that it i) offers the possibility to greatly reduce labeling effort compared to a passive learning baseline or to an active learning baseline trained from scratch, if the domain gap is small enough and ii) at least does not cause more expenses (compared to a newly initialized active learning loop), if the domain gap is severe. The latter is especially beneficial in scenarios where the similarity of two different domains is hard to assess.

1. INTRODUCTION

Although machine learning systems, such as convolutional neural networks, have seen significant improvement in past years and reach top results for various benchmark challenges (Niemeyer et al., 2014; Kölle et al., 2021a), two main issues remain: i) the availability of suitable training sets and ii) the transfer of models learned on a specific data set to another similar data set (transfer learning). To efficiently overcome the lack of appropriate training sets, active learning (AL) can be used, which focuses labeling effort on most informative samples only (Settles, 2009). This technique was recently successfully applied in context of the semantic segmentation of 3D point clouds (Luo et al., 2018; Li and Pfeifer, 2019; Lin et al., 2020; Kölle et al., 2021b). When AL is employed alongside with crowdsourcing, where crowdworkers are asked to label queried instances, an automated hybrid intelligence system (Vaughan, 2018) can be formed where no expert is required for labeling (Vijayanarasimhan and Grauman, 2011; Kölle et al., 2021c). This is especially appealing when not only the labeling procedure is outsourced, but if all crowd management (e.g., hiring & paying crowdworkers, posting campaigns) is automatized by leveraging respective platforms such as *Amazon Mechanical Turk* (Buhrmester et al., 2011) or *microWorkers* (Hirth et al., 2011) as described by Kölle et al. (2021c).

But no matter if a model is learned through AL or passively by a pre-defined training set (i.e., passive learning, PL) on a given source domain D_S, learned models often fail to derive a satisfactory predictional quality for a target domain D_T of different characteristics (Penatti et al., 2015). This is caused by a so-called domain gap, which can have various reasons. Either D_S was inadequately sampled (due to insufficient feature space exploration by the machine in AL or a biased selection of samples by a human operator in PL), so that it is not representative for D_T. This problem is referred to as *covariate shift* problem (Shimodaira, 2000). More precisely, when considering that both D_S and D_T can be described by respective distributions $p(\mathbf{x}, y) = p(\mathbf{x})p(y|\mathbf{x})$ (where $\mathbf{x}$ corresponds to the feature vector and y to the respective true class), the *covariate shift* can be translated to $p^S(\mathbf{x}) \neq p^T(\mathbf{x})$ while $p^S(y|\mathbf{x}) \approx p^T(y|\mathbf{x})$. A more severe kind of domain gap is related to the more general *sample selection bias* (Heckman, 1979). In this case, not only the marginal distributions but also the conditional probabilities are different, i.e., $p^S(\mathbf{x}) \neq p^T(\mathbf{x})$ and $p^S(y|\mathbf{x}) \neq p^T(y|\mathbf{x})$. This can be interpreted in such a way that the representation of the same classes in feature space is different in the two domains.

In case of remote sensing data, we expect to deal with *sample selection bias*, where the domain gap can be caused by different sensor configurations (sensor and/or flight planning), different atmospheric conditions (illumination and humidity) and changes of the scene itself due to phenological phenomena (i.e., seasonal effects). For bridging this gap, various approaches for domain adaptation (DA) were discussed by Tuia et al. (2016), which focus on generating invariant or joint features or aim to adapt the classifier itself. If a (limited) budget is available for retrieving true labels of D_T, DA of the classifier can be accomplished by including this additional training data. For this, the PL approach *TrAdaBoost* is presented by Dai et al. (2007), where the classifier is iteratively adapted through boosting and whereby samples of D_S and D_T are treated differently in weighting. Rajan et al. (2008) proposed a first framework to actively sample points in D_T based on expectation maximization. Following the same idea of only labeling most informative samples of D_T, Matasci et al. (2012) embedded the *TrAdaBoost* framework into a classic AL loop and, in contrast

*Corresponding author

to the work of Rajan et al. (2008), adaptively weight samples of D_S and D_T. Jun and Ghosh (2008) also demonstrated that reweighting samples is capable to outperform AL loops for DA where no individual weights are applied.

Tuia et al. (2011) explored the capabilities of AL in DA to include new classes by first deriving a clustering of feature space of D_T to sample from those clusters formed instead of relying on conventional sampling strategies such as described by Settles (2009). This strategy is related to guaranteeing diversity in batches in batch mode AL and aims to boost the convergence of the classifier (Zhdanov, 2019). The same goal is also pursued by Persello and Bruzzone (2012), but transferred to DA AL loops where samples are drawn from D_T based on both uncertainty and diversity. At the same time, conflicting samples of D_S, which are not longer representative for D_T and would thus harm the classification performance, are discarded. A similar formulation of this strategy is discussed by Persello (2013). In order not to fully discard the information such samples might carry, the authors do not only remove inconsistent samples, but also apply different weights for the remaining samples of D_S in accordance to their agreement with D_T. As an additional measure to minimize labeling cost, Rai et al. (2010) argue to avoid samples of D_T which are situated close to the decision borders but similar to samples of D_S (i.e., avoiding quasi-duplicates from different domains).

Our contributions within this work can be summarized as follows: i) we present a comprehensive framework which allows to cost-efficiently build models for the semantic segmentation of 3D point clouds by making use of both AL and DA techniques, ii) we explore whether this approach is suitable to run in a truly automatized manner, where an expert oracle is replaced by a crowd oracle and iii) this work is the first one to perform inter-domain semantic segmentation on the newly introduced H3D benchmark data set (Kölle et al., 2021a).

2. METHODOLOGY

The main purpose of our approach is to cost-efficiently train models for semantic segmentation by minimizing required labeling effort. This incorporates a conventional AL loop, which defines the basic structure of our pipeline (section 2.1). The loop is enhanced to be suitable for DA by exploiting previously generated labels (section 2.2). The automation of this workflow by means of paid crowdsourcing is discussed in section 2.3.

2.1 AL Loop for Semantic Segmentation of Point Clouds

Our means to minimize labeling costs is in the first place AL to actively sample only the subset of points worth labeling by human operators. To initialize our pipeline visualized in Figure 1, we first present a completely unlabeled point cloud to the crowd and ask crowdworkers to indicate one point for each of our desired classes. Received points are then checked by a second group of crowdworkers to form the initialization training set. This can then be used for training a first machine learning model, which is a Random Forest (RF) (Breiman et al., 1984) in our case (cf. Kölle et al. (2021b) for discussion of alternatively employing a CNN classifier). This initial model is then utilized for inference on the remaining unlabeled data points to determine most informative samples, which can then be presented to the crowd oracle $\mathcal{O}_C$ for labeling. Based on the enhanced training pool, a new RF model is trained and the loop is repeated for n_i iteration steps. Identification of most informative points is achieved by entropy of a posteriori probabilities that point $\mathbf{x}$ belongs to class c:

$$x^+ = \underset{\mathbf{x}}{\operatorname{argmax}} \left(-\sum_c p(c|\mathbf{x}) \cdot \log p(c|\mathbf{x}) \right) \tag{1}$$

To account for class imbalanced data sets, we enrich the entropy definition by adding a weighting function, which is computed based on the total number of points n_T currently present in the training data set and the number of instances of each class n_c at iteration step i: $w_c(i) = n_T(i)/n_c(i)$. In order to boost the convergence of the iteration (in batch-mode AL), we aim to avoid sampling of points which are similar in terms of their representation in feature space. To derive a diverse training set, we apply a k-means clustering of feature space and sample one point per cluster formed (Zhdanov, 2019). Consequently, the number of clusters k equals the number of points n^+ to be sampled in each iteration step.

Since we aim to outsource labeling to real crowdworkers, we apply the *Reducing Interpretation Uncertainty* (*RIU*) strategy proposed by Kölle et al. (2021b) to ease labeling. Instead of presenting the point with highest sampling score to the crowd, this point is only considered as seed point. An alternative point within a distance of d_{RIU} (in object space) having the lowest sampling score but still being informative due to its adjacency to the seed point is used. The rationale of this approach is to increase interpretability of points by avoidance of point sampling directly on class borders, where even experts might struggle to determine the correct class.

2.2 Enhancement of the AL loop for DA

The previous section implied that no samples from a source domain are available, so that the iteration needs to be performed from scratch. Since we expect that in many applications such a data set exists, we would like it to contribute for learning our model. AL is well suitable for such DA tasks by design (if labels can be obtained in D_T) by simply changing the pool of instances where the query function operates from D_S to D_T as suggested by Matasci et al. (2012) and Persello (2013), for example (please note that merging D_S and D_T into a common pool is theoretically also possible, which would likely result in a more general classifier, but at least convergence might be delayed and/or performance on D_T might suffer). The comprehensive training pool is then obtained by merging the training set sampled in D_S with the newly sampled and labeled points of D_T. We will refer to this procedure as active transfer learning (ATL). This DA technique will work well when D_S and D_T are sufficiently similar, but might also result in even more labeling effort when the domain gap is considerable.

If source domain samples used for training the RF model are not representative for D_T, the selection of points in D_T is suboptimal since the assumed class borders (in the vicinity of which samples are drawn) differ from the ones truly inherent in D_T. Consequently, more iteration steps might be necessary to adapt the initial model to the new distribution and starting the iteration from scratch would be more cost-efficient. However, it is often difficult to decide whether two domains are similar enough for adaptation, i.e., whether re-using samples of D_S is beneficial or harmful. Hence, an additional procedure is required where samples of D_S can be included to optimally boost convergence or at least ensuring that the effect of conflicting samples is neutralized in case of a too large domain gap.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2022-259-2022 | © Author(s) 2022. CC BY 4.0 License.

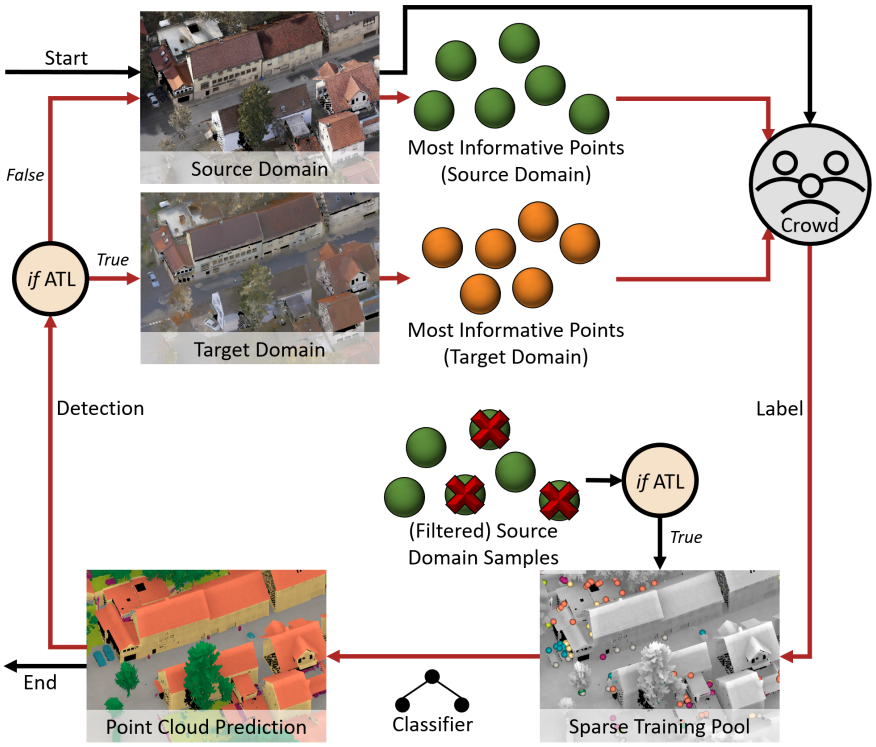


Figure 1. Crowd-powered pipeline for DA to adapt a classifier learned on a source domain to a related target domain. The A(T)L loop is indicated by *red* arrows.

For this to achieve, we follow the suggestion of Persello and Bruzzone (2012) and remove n^- samples of D_S which are most inconsistent with D_T by measuring the disagreement between a model trained by samples of D_S only and another model trained by the combined training set from D_S and D_T at iteration step i. Precisely, the a posteriori probabilities of each sample $\mathbf{x}$ of D_S are compared according to:

$$x^- = \underset{\mathbf{x}}{\operatorname{argmax}} \left(\left| p^S(c|\mathbf{x}) - p^{S \cup T}(c|\mathbf{x}) \right| \right) \tag{2}$$

Recalling our workflow (cf. Figure 1), this means that we check which samples of D_S disagree most with samples of D_T (implicitly compared through learned decision rules of respective RF models) and remove those before re-training our current classifier on samples from both D_S and D_T. We refer to this procedure as ATL$_{FSD}$. Controlling the impact of D_S is therefore accomplished through n^- or rather the ratio $q = n^-/n^+$.

2.3 Employing the Crowd Oracle and Automation

To shift labeling effort from an expert to the crowd, we rely on the *microWorkers* crowdsourcing marketplace. In order to avoid any overhead due to the management of the crowd, we integrate ATL into a tool similar to the recently introduced CATEGORISE framework (Kölle et al., 2021c). Actual labeling is carried out by the crowdworkers using webtools as described by Kölle et al. (2021c).

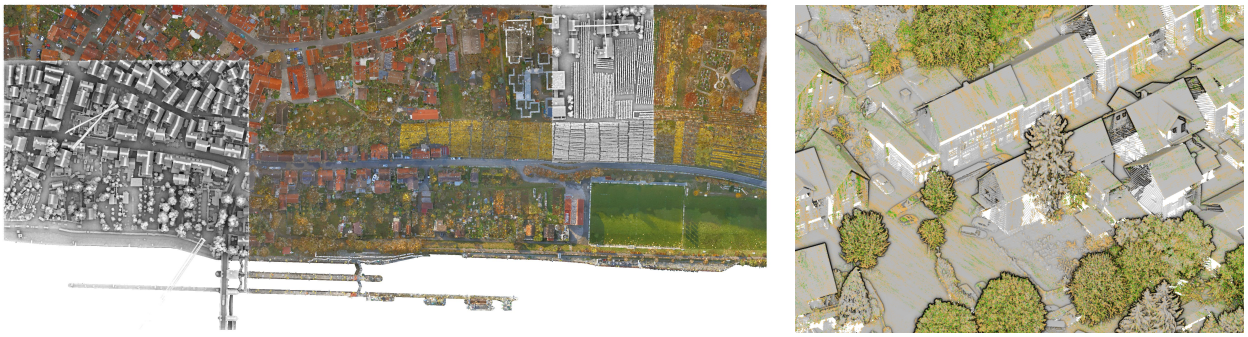
A prerequisite of automated A(T)L runs (and a measure to minimize labeling effort) is the definition of an effective stopping criterion without the need to compare predictions to ground truth labels (which are not available in real-world applications). We apply an approach similar to the one presented by Bloodgood and Vijay-Shanker (2009), where the congruence of predictions in successive iteration steps is evaluated. For each point we check whether it carries the same label as in the iteration step d_{stop} steps before to get an overall congruence measure C_o. To account for smaller classes, for each class we filter points predicted as this specific class and check again whether the same label was predicted in the iteration step d_{stop} steps before. These classwise congruence measures are then averaged to receive a class-sensitive overall measure C_{ac} in company of C_o. Afterwards we can evaluate the standard deviation of congruence values and stop the iteration if a certain threshold is reached.

3. DATA SETS AND EXPERIMENTAL SETUP

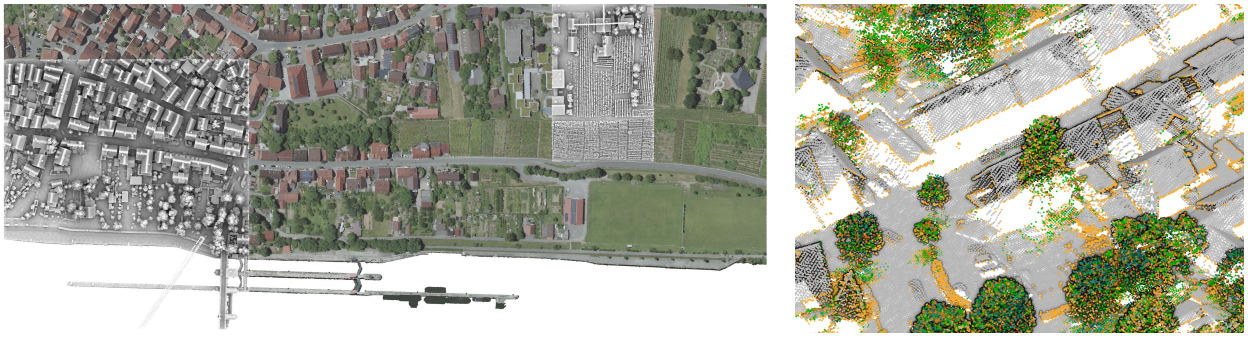
The DA problem we aim to study concerns temporally disjoint data sets. This means that the same area was captured at multiple epochs under different atmospheric conditions and sensor configurations. We apply our approach to the multi-temporal H3D data set. As visualized in Figure 2, we focus on three different epochs. Two of them (epochs March & November 2018) are high-resolution UAV LiDAR point clouds having mean point densities of about $800\,\mathrm{pts/m^2}$. They were acquired by a *Riegl VUX-1LR* scanner at a height above ground of $50\,\mathrm{m}$ and a field of view of the scanner of $\pm 70\,^\circ$. Thus, vertical surfaces such as façades are well depicted (cf. Figure 2 a & b). These data sets differ in terms of i) the spatial extent (epoch November 2018 depicts an area more than two times larger than epoch March 2018), ii) phenological changes and iii) the atmospheric conditions what mainly affects radiometric properties. Apart from different illumination, thick fog was present in November 2018 severely affecting RGB colorization from concurrently captured images of *Sony Alpha 6000* cameras. This can be clearly seen in Figure 1, where *source domain* is a subset of epoch March 2018 and *target domain* of epoch November

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2022-259-2022 | © Author(s) 2022. CC BY 4.0 License.

(a) H3D - Epoch March 2018



(b) H3D - Epoch November 2018



(c) H3D - Epoch March 2016

Figure 2. Compilation of our data sets/epochs used for testing our approach. Each visualization depicts an overview of the complete data set, where the training set is colored according to available RGB data while the test set is presented as shading (*left*). Respective close-ups are colorized according to the number of returns of an emitted laser pulse (*right*).

2018. Presence of fog also resulted in recording multiple echos for streets, façades and roofs due to additional returns caused by aerosols (compare right side of Figure 2 a & b). The third data set, epoch March 2016, incorporates features of a typical airborne laser scanning point cloud of national mapping agencies. The data set was captured by a *Riegl LMS-Q780* scanner at a height above ground of $600\,\mathrm{m}$ and a field of view of $\pm 20\,^{\circ}$. The mean point density is about $20\,\mathrm{pts/m^2}$. This configuration leads to scarce depiction of vertical surfaces due to the nadir-like measuring perspective (see few façades in Figure 2 c, *right*). Since no concurrently captured imagery is available for this epoch, we both i) interpolated colors from epoch November 2018 via nearest neighbor transfer and ii) used the point cloud colorization obtained by orthogonal projection of an orthophoto as available in the H3D benchmark (visualized in Figure 2 c *left*). The orthophoto was acquired by a *DMC II 140* camera configured to achieve a ground sampling distance of $20\,\mathrm{cm}$ and was taken on June 11, 2017.

Using these data sets, we formulate the scenarios for applying our workflow:

- **Scenario I:** DA of epoch March 2018 to epoch November 2018, i.e., DA between two data sets having similar geometric but deviating radiometric properties due to phenological and atmospheric changes.
- **Scenario II:** DA of epoch March 2018 to epoch March 2016 colored by RGB values from epoch November 2018 (epoch March 2018 could be used as well but would ease the DA problem), i.e., DA between two data sets having deviating geometric and deviating radiometric properties.
- **Scenario III:** Same as scenario II but colorization of epoch March 2016 by using the orthophoto. Scenario III is designed to assess the impact of different colorization methods on DA.

To ease the complexity of point labeling for crowdworkers, we merge some classes to form our class catalog comprising *Urban*

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2022-259-2022 | © Author(s) 2022. CC BY 4.0 License.

Furniture, *Low Vegetation* (including *Soil/Gravel*), *Impervious Surface*, *Vehicle*, *Roof*, *Façade* (including *Vertical Surface*) and *Vegetation* (*Shrub* & *Tree*). Please note that all epochs incorporate this same set of classes.

Feature computation for all epochs is conducted as proposed by Kölle et al. (2021a). Both geometric and radiometric features were derived for each point using spherical neighborhoods of $0.125, 0.25, 0.5, 075, 1, 2, 3$ and $5\,\mathrm{m}$ for scenario I and $1, 2, 3$ and $5\,\mathrm{m}$ in scenarios II and III due to the point sparsity in epoch March 2016. Please note that we do not explicitly choose our features to be domain-invariant. However, geometric features such as the ones based on the structural tensor or roughness are well generalizable. Radiometric features (especially reflectance/intensity of laser returns and color features), on the other hand, are expected to be prone to domain changes.

4. RESULTS

To prove the effectiveness of our methodology, we first analyze its performance on all three scenarios presented in section 3 by conducting AL runs with simulated oracles (section 4.1) and afterwards by exemplarily running an ATL loop with a real crowd oracle (section 4.2).

4.1 Performance of ATL with Simulated Oracles

For each scenario presented in section 3, we derive both a set of baseline solutions and actual ATL runs:

- **PL** is a passive learning approach on D_T relying on the completely labeled training set of D_T.
- Passive transfer learning, **PTL**, is the most naive transfer learning setting where the classifier is trained in a passive manner on the completely labeled training set of D_S.
- A classic **AL** run is conducted from scratch on D_T. The initialization data set is received by asking a total of 100 crowdworkers to present one point for each class. In a second step, every point is checked by three different crowdworkers (following the recommendation of Kölle et al. (2021c)). If the majority of the crowdworkers of the second group disagree that the label is correct, the point is discarded. Payment for crowdworkers is \$0.10 for the first job and \$0.10 for the second job with an option for a bonus of \$0.05 if crowdworkers perform well (evaluated on four check points in each task, cf. Kölle et al. (2021c)). We set $n^+ = k = 300$.
- **ATL** is an AL run where we use the acquired training set from D_S after 10 iteration steps (assumed to be sufficiently representative for D_S, since also obtained through AL) as initialization data set (both with and without label noise, cf. Figure 3) and continue this iteration already begun on D_T (i.e., $n^- = 0$).
- Finally, **ATL**$_{FSD}$ is the same as ATL, but with deletion of samples of D_S as discussed in section 2.2.

In each case, we form the RF ensemble to be composed of 100 binary decision trees with a maximum depth of 18. All AL-based runs are derived both for an omniscient oracle $\mathcal{O}_O$ always returning ground truth labels and a simulated crowd oracle $\mathcal{O}_{CS}$. Following the results of Kölle et al. (2021b), we can assume that relying on *RIU* leads to random label noise only (no systematic errors). Similarly, we select $d_{RIU} = 0.75\,\mathrm{m}$ and add $10\,\%$ of random label noise to received ground truth labels. We will analyze whether this assumption holds true for a real crowd oracle $\mathcal{O}_C$ in section 4.2. The performance of all these methods with respect to semantic segmentation is reported in Figure 3 both in terms of the overall accuracy (OA) and the mean F1-score (mF1).

The need for adapting an already available model to a new domain becomes evident when comparing the results of PL and PTL for all scenarios. Blindly (and passively) transferring a classifier from one domain to another leads to a loss in mF1/OA of $6.41/4.46$ and $12.30/5.17$ percentage points for scenario I and II respectively and even $26.59/20.13$ percentage points for scenario III. These results already give an impression of the inherent domain gaps in the different scenarios and are according to expectation recalling their design. PTL and PL can be considered as lower and upper accuracy bound respectively. We expect that our methods for DA (i.e., ATL) surpass a solution where no DA at all was performed (i.e., PTL). However, it is challenging to surpass one with a completely labeled training set of D_T (i.e., PL) when using a sparse training set (3000 labeled samples of D_T after 10 iteration steps in our case). Yet, the main advantage of ATL is the significant reduction of labeling effort (only $0.02/0.02/1.07\,‰$ points of D_T have to be labeled for each scenario respectively).

Regarding scenario I, we can observe that the performance of the AL loops running from scratch is characterized by a steep increase in accuracy converging from about iteration step 6 (cf. Figure 3). Here the loop using a simulated crowd oracle (i.e., AL($\mathcal{O}_{CS}$)) performs worse in terms of mF1, which is mainly due to a higher level of confusion between classes *Vehicle* and *Urban Furniture*. For these classes, noisy labels are especially harmful due to typically similar features (e.g., consider trash bins or containers vs. cars). Compared to AL, our ATL runs initialize on a much higher level of accuracy for respective oracle types (about 10 percentage points better both in terms of mF1 and OA). To be able to account for inconsistent samples of D_S, we set $q = 1.1$ for ATL$_{FSD}$ (i.e., we remove 330 samples of D_S in every iteration step). In case of omniscient oracles, this removal seems unnecessary and even marginally harms the accuracy in D_T, since the domain gap is small. However, in case of a simulated noisy crowd oracle ($\mathcal{O}_{CS}$), removing samples greatly helps improve accuracy of ATL$_{FSD}$($\mathcal{O}_{CS}$) compared to ATL($\mathcal{O}_{CS}$), which performs even worse than the conventional AL($\mathcal{O}_{CS}$) run. Again, this is mainly due to confusion of classes *Urban Furniture* and *Vehicle*, where the inherent overlap in feature space is even further amplified by mixing samples from two domains.

As previously stated, our main motivation is reducing manual labeling effort and thus costs. Our pure AL runs cause costs for initialization of 100 jobs·\$0.10+100·3·\$0.15 = \$55, which can be avoided relying on ATL. Additional expenses can be saved depending on the required quality of semantic segmentation. The more limited the labeling budget, the more beneficial ATL becomes. In terms of OA, ATL$_{FSD}$($\mathcal{O}_{CS}$) is about three iteration steps ahead of AL($\mathcal{O}_{CS}$) (e.g., AL($\mathcal{O}_{CS}$) in iteration step 4 vs. ATL$_{FSD}$($\mathcal{O}_{CS}$) in iteration step 1). With greater number of iteration steps, accuracies of AL and ATL runs converge to an OA of about $86\,\%$.

In case of the second scenario, where geometries are rather different between the domains (cf. section 3), the general trend of all runs is similar. Due to an increased domain gap, the difference between AL runs and ATL approaches is consequently smaller. But still, ATL runs are beneficial in case of limited labeling budget scenarios. In terms of mF1, ATL$_{FSD}$($\mathcal{O}_{CS}$)

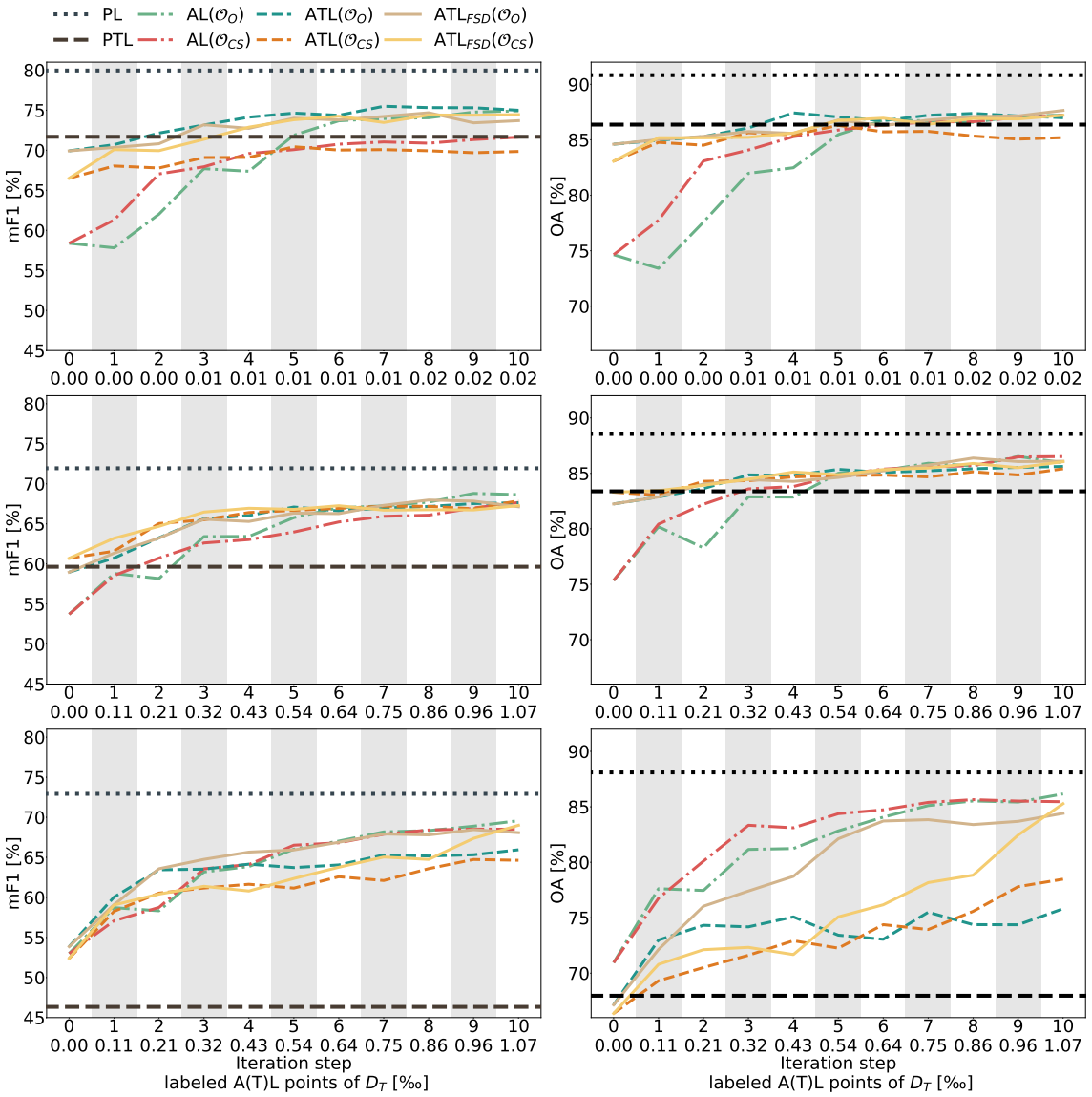


Figure 3. Results of our simulated runs for scenario I-III (*top* to *bottom*) evaluated in terms of mF1 and OA (number of initialization points is not included in the relative proportion of labeled points in case of pure AL runs).

reaches a close to final accuracy level already in the fourth iteration step, while AL($\mathcal{O}_{CS}$) runs for nine iteration steps to achieve the same level of accuracy. It is noteworthy that despite different geometries of the two domains, source domain samples positively contribute to the training of the model. This means that geometric features generalize well.

For scenario III, we can observe that conventional AL runs perform best and that ATL is not appropriate due to a too large domain gap. Please note that this gap was solely introduced by the altered colorization (2D colorization) compared to scenario II (3D colorization, just like for D_S). In contrast to scenarios I and II, where removing source samples could rather be neglected, ATL$_{FSD}$ runs perform significantly better compared to their ATL counterparts. They almost reach the accuracy levels of pure AL runs in the tenth iteration step, but without the requirement of an initial training set from D_T. Hence, costs can still be saved. This means that the operator can include samples of a source domain if available without leading to loops that cause more costs than pure AL runs. We would like to stress that for the deletion/sampling factor q, we used 1.1 for all scenarios underlining the robustness of this parameter.

Also noteworthy is the much more desirable behavior of ATL$_{FSD}$($\mathcal{O}_O$) compared to ATL$_{FSD}$($\mathcal{O}_{CS}$), which is due to the frequently occurring misprediction of class *Low Vegetation* as *Impervious Surface* for the latter (e.g., 40 % of *Low Vegetation* in iteration step 5). This can be explained by Figure 4. In case of ATL$_{FSD}$($\mathcal{O}_O$) 20 % of deleted points belong to class *Impervious Surface* in iteration steps 2 to 4, meaning this class is impacted by a domain shift. Actually, one would assume that the representation of samples of this class is not subject of domain changes. However, this is due to the scarce depiction of façades in D_T (cf. Figure 2 c vs. 2 a). Since challenging points for the classifier lie among others on the class borders between *Façade* and *Impervious Surface*, such points are sampled by AL on D_S (March 2018). But the lack of façades in D_T (March 2016) means an altered representation of points situated in such spots, so that the description of D_S does not hold true anymore. Thus, such points of D_S are to be discarded. For ATL$_{FSD}$($\mathcal{O}_{CS}$), on the other hand, *Urban Furniture* is the main subject of deletion in early iteration steps. In presence of

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2022-259-2022 | © Author(s) 2022. CC BY 4.0 License.

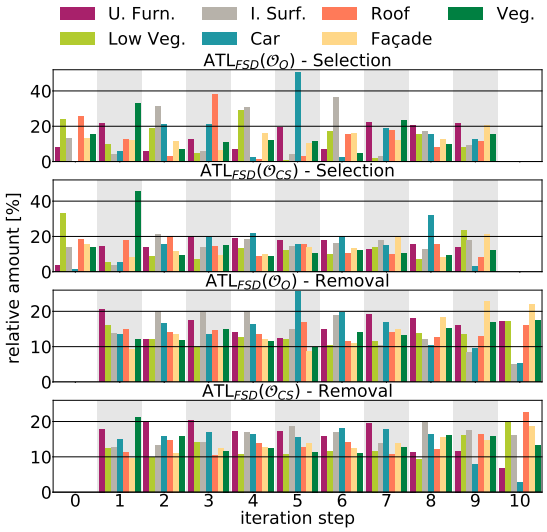


Figure 4. Sampled points of D_T and deleted points of D_S for scenario III (at iteration step 0 no points can be deleted since no samples of D_T are present, hence the offset by 1).

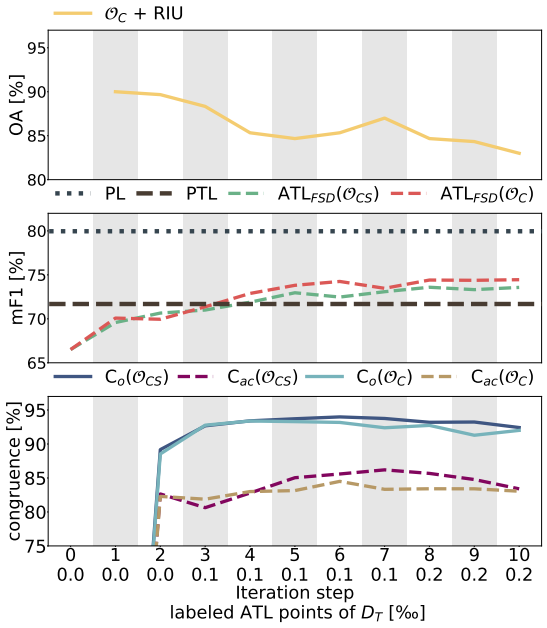


Figure 5. Results obtained for scenario I when relying on a real crowd oracle. We report the overall labeling accuracy of the crowd (*top*) and evaluate the progress of respective AL loops in terms of the mean F1-score (*middle*) and the congruence of predictions (*bottom*).

label noise this quasi-class *Other* becomes even more confusing for a classifier. Hence, it kind of distracts the classifier. For $\mathcal{O}_O$, it is worth highlighting that in the fifth iteration step a high number of *Car* samples from D_S are removed and replaced by respective samples of D_T.

Generally, when comparing the class distributions of sampling and deletion, we can observe that for sampling the classifier often focuses on specific classes in each iteration step (especially in case of ATL$_{FSD}$($\mathcal{O}_O$), which also explains its faster convergence compared to ATL$_{FSD}$($\mathcal{O}_{CS}$); cf. Figure 3). Histograms for deletion, on the other hand, resemble equal distributions, which means that there are no specific classes which have completely changed, but rather there are inconsistent points spread over all classes. This is a result of the discrepancies between the high-resolution data set (March 2018) and the much more scarcely sampled cloud (March 2016).

From computed RF feature relevance, there is also a trend of more universal geometric features gradually loosing in importance while more case specific color features gain in importance meaning that they can be more efficiently used

4.2 Performance of ATL with a Real Crowd Oracle

Since we aim to not only minimize labeling effort but to outsource it to the crowd, we actually replace the simulated crowd oracle $\mathcal{O}_{CS}$ by a real one $\mathcal{O}_C$ for scenario I (precisely, we repeat ATL$_{FSD}$ under real world conditions). From Figure 5 *top*, we can observe that the performance of the crowd decreases with the number of iteration steps. This can be explained by sampling of points which become more and more demanding for the classifier, but also for crowdworkers to interpret them. This behavior is in accordance with the findings of Kölle et al. (2021c). Mean OA for crowd labels over all iteration steps is 86.23 %. Hence, the assumption of a noisy crowd oracle with 10 % of random noise is justified. This is also reflected in the accuracy of ATL$_{FSD}$, powered by $\mathcal{O}_C$ instead of $\mathcal{O}_{CS}$, since mF1 only differs marginally (cf. Figure 5 *middle*).

To monitor training progress, we rely on our intrinsic measure (cf. section 2.3). We set $d_{stop} = 2$ to emphasize differences in course of the loops (hence, we cannot derive congruence values for iteration steps 1 and 2). Generally, we would like our congruence curves to behave the same way as the accuracy curves with respect to convergence. In case of the overall measure C_o, this does not seem to hold true with regard to mF1, but is reasonable when compared to the OA (i.e., ATL$_{FSD}$($\mathcal{O}_{CS}$) in Figure 3). This demonstrates that an overall measure is not sufficient if high classification accuracies for all classes are desired. Please note that for congruence curves mainly changes rather than absolute values are decisive. If the same congruence level is obtained in multiple successive iteration steps, the iteration can be aborted since predictions do not change anymore. If we consider the standard deviations of the last three values respectively, the iteration can probably be aborted (depending on a user-defined threshold) after iteration step 7 for both cases ($\mathcal{O}_{CS}$ & $\mathcal{O}_C$). Please note that most of our congruence curves have a negative trend towards the end of the iteration. This might be an effect of removing too many samples of D_S so that the decision borders start to alter again. However, we argue that this behavior would only come into effect when the iteration was already aborted.

5. CONCLUSION AND OUTLOOK

Within this paper, we have demonstrated that AL can be efficiently employed for DA purposes for the semantic segmentation of 3D point clouds. This is especially advantageous in limited budget scenarios where only few iteration steps can be conducted. If domains are sufficiently similar, ATL accuracies are at least three iteration steps ahead of AL runs starting from scratch. We have observed that our classifiers perform rather robustly when facing atmospheric and phenological changes or even a different geometric depiction (i.e., point density). But they are prone to significant changes in colorization (see scenario III). Even in presence of a severe domain gap, ATL can at least avoid the necessity of an initial training set, which can be

costly to collect. The approach discussed can be used for updating the LiDAR map archive of national mapping agencies, where identical areas are periodically surveyed by the same (or at least a similar) sensor configuration. It is also applicable for multi-temporal deformation analyses, where it is desirable to filter dynamic objects in a first step (Haala et al., 2020).

ACKNOWLEDGEMENT

The authors would like to show their gratitude to the State Office for Spatial Information and Land Development Baden-Wuerttemberg for providing the ALS point cloud and the respective orthophoto imagery of the village of Hessigheim.

REFERENCES

Bloodgood, M., Vijay-Shanker, K., 2009. A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping. *CoNLL-2009*, Association for Computational Linguistics, Boulder, Colorado, 39–47.

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J., 1984. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

Buhrmester, M., Kwang, T., Gosling, S. D., 2011. Amazon's Mechanical Turk: A New Source of Inexpensive,Yet High-Quality, Data? *Perspectives on Psychological Science*, 6(1), 3–5.

Dai, W., Yang, Q., Xue, G.-R., Yu, Y., 2007. Boosting for transfer learning. *ICML '07*, Association for Computing Machinery, New York, NY, USA, 193–200.

Haala, N., Kölle, M., Cramer, M., Laupheimer, D., Mandlburger, G., Glira, P., 2020. Hybrid Georeferencing, Enhancement and Classification of Ultra-High Resolution UAV LIDAR and Image Point Clouds for Monitoring Applications. *ISPRS Annals*, V-2-2020, 727–734.

Heckman, J. J., 1979. Sample Selection Bias as a Specification Error. *Econometrica*, 47(1), 153–161.

Hirth, M., Hoßfeld, T., Tran-Gia, P., 2011. Anatomy of a Crowdsourcing Platform - Using the Example of Microworkers.com. *IMIS 2011*, IEEE Computer Society, Washington, DC, USA, 322–329.

Jun, G., Ghosh, J., 2008. An efficient active learning algorithm with knowledge transfer for hyperspectral data analysis. *IGARSS 2008*, 1, I–52–I–55.

Kölle, M., Laupheimer, D., Schmohl, S., Haala, N., Rottensteiner, F., Wegner, J. D., Ledoux, H., 2021a. The Hessigheim 3D (H3D) benchmark on semantic segmentation of high-resolution 3D point clouds and textured meshes from UAV LiDAR and Multi-View-Stereo. *ISPRS Open Journal*, 1.

Kölle, M., Walter, V., Schmohl, S., Soergel, U., 2021b. Remembering both the machine and the crowd when sampling points: Active learning for semantic segmentation of als point clouds. *ICPR International Workshops and Challenges*, Springer International Publishing, Cham, 505–520.

Kölle, M., Walter, V., Shiller, I., Soergel, U., 2021c. Categorise: An automated framework for utilizing the workforce of the crowd for semantic segmentation of 3d point clouds. *Pattern Recognition*, Springer International Publishing, Cham, 633–648.

Li, N., Pfeifer, N., 2019. Active Learning to Extend Training Data for Large Area Airborne LiDAR Classification. *ISPRS Archives*, XLII-2/W13, 1033-1037.

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

Luo, H., Wang, C., Wen, C., Chen, Z., Zai, D., Yu, Y., Li, J., 2018. Semantic Labeling of Mobile LiDAR Point Clouds via Active Learning and Higher Order MRF. *IEEE TGRS*, 56(7), 3631-3644.

Matasci, G., Tuia, D., Kanevski, M., 2012. SVM-Based Boosting of Active Learning Strategies for Efficient Domain Adaptation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(5), 1335-1343.

Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. *ISPRS Journal*, 87, 152–165.

Penatti, O. A. B., Nogueira, K., dos Santos, J. A., 2015. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? *2015 IEEE CVPRW*, 44–51.

Persello, C., 2013. Interactive Domain Adaptation for the Classification of Remote Sensing Images Using Active Learning. *IEEE Geoscience and Remote Sensing Letters*, 10(4), 736-740.

Persello, C., Bruzzone, L., 2012. Active learning for domain adaptation in the supervised classification of remote sensing images. *IEEE TGRS*, 50(11), 4468–4483.

Rai, P., Saha, A., Daumé, Iii, H., Venkatasubramanian, S., 2010. Domain adaptation meets active learning. *HLT-NAACL 2010*.

Rajan, S., Ghosh, J., Crawford, M. M., 2008. An Active Learning Approach to Hyperspectral Data Classification. *IEEE TGRS*, 46(4), 1231-1242.

Settles, B., 2009. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Shimodaira, H., 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90, 227-244.

Tuia, D., Pasolli, E., Emery, W. J., 2011. Using active learning to adapt remote sensing image classifiers. *Remote Sensing of Environment*, 115, 2232-2242.

Tuia, D., Persello, C., Bruzzone, L., 2016. Domain Adaptation for the Classification of Remote Sensing Data: An Overview of Recent Advances. *IEEE Geoscience and Remote Sensing Magazine*, 4(2), 41-57.

Vaughan, J. W., 2018. Making Better Use of the Crowd: How Crowdsourcing Can Advance Machine Learning Research. *Journ. Mach. Learn. Res.*, 18(193), 1-46.

Vijayanarasimhan, S., Grauman, K., 2011. Large-scale live active learning: Training object detectors with crawled data and crowds. *CVPR 2011*, 1449–1456.

Zhdanov, F., 2019. Diverse mini-batch Active Learning. *CoRR*, abs/1901.05954. http://arxiv.org/abs/1901.05954.