Contrastive Maximum Mean Discrepancy for Unsupervised Domain Adaptation Applied to Large Scale 3D LiDAR Semantic Segmentation

Lamiae El Mendili[1]
lamiae.el-mendili.1@ulaval.ca
Sylvie Daniel[1]
sylvie.daniel@scg.ulaval.ca
Thierry Badard[1]
thierry.badard@scg.ulaval.ca
Patrick Dallaire[2]
patrick.dallaire@ift.ulaval.ca

[1] Department of Geomatics
Université Laval,
QC, Canada

[2] Department of Computer Science and
Software Engineering
Université Laval,
QC, Canada

Abstract

Semantic segmentation of 3D LiDAR point clouds is very important for applications like autonomous driving and digital twins of cities. However, current deep learning models suffer from a significant generalization gap. Unsupervised Domain Adaptation (UDA) methods have recently emerged to tackle this issue. While domain invariant feature learning using maximum mean discrepancy (MMD) has shown promise in image domains due to its simplicity, its application remains unexplored in large-scale outdoor point clouds. Moreover, previous methods don't consider the class information, which can lead to suboptimal adaptation performance. In response, we propose a new approach—Contrastive Maximum Mean Discrepancy (CMMD)— to maximize intra-class domain alignment and minimize inter-class domain discrepancy. We integrate CMMD into a 3D semantic segmentation model for LiDAR point clouds. The evaluation of our method with large-scale UDA datasets shows that it surpasses several state-of-the-art UDA approaches for 3D LiDAR point clouds while being competitive with the current best-performing approach. CMMD is a promising UDA approach with strong potential for point cloud semantic segmentation.

1 Introduction

Semantic segmentation of large-scale outdoor point clouds is a very important task for numerous applications like autonomous driving and digital twins of cities. Although the performance of supervised deep learning-based methods has increased in recent years, they require large amounts of annotated data to train which is complex and expensive to obtain. This prompted researchers to focus more on other avenues. One such avenue is transfer learning from synthetic datasets, for which labels are readily available. However, semantic segmentation models have been shown to suffer from a significant generalization gap. This has been demonstrated in some recent work [1, 5, 11]. These models are static as they are obtained from a set of data present at the time of training and are unable to adapt to changes in the data. These changes can be due to differences in LiDAR sensors that influence the density, noise distribution, and geometry of the point clouds, geographical variations that affect the appearance of a scene, and class variations that are more pronounced in urban point clouds. Unsupervised Domain Adaptation (UDA) methods have emerged to bridge the generalization gap. The majority of the proposed methods are targeted toward images [7, 8, 22] while 3D point clouds have not been sufficiently explored. This can be attributed to their sparsity, irregularity, and larger data volume, which makes it challenging to apply existing techniques. UDA approaches can be divided broadly into two categories: domain mapping and domain invariant feature learning. Domain mapping approaches map the source point cloud to the target point cloud using conditional generative adversarial networks (GANs) [14, 15, 32] or data augmentation approaches [1, 16]. In this case, either 3D point clouds are used directly as input data [16, 32] or projected images of LiDAR scans [1, 14, 15] are used instead. Domain invariant feature learning approaches propose to learn a shared feature representation by minimizing a discrepancy measure between source and target features [25] or through adversarial training [9, 11, 23]. Compared to adversarial training, discrepancy-based approaches are simpler to train and integrate into existing models. Furthermore, since they explicitly measure domain discrepancy, they can provide insights into the differences between the source and target domains. Domain discrepancy can be measured by Maximum Mean Discrepancy (MMD) [10], which has been extensively used to obtain domain-invariant features for images. However, this line of work hasn't been explored for 3D point clouds. Moreover, most of the existing UDA methods for point clouds consider the domain discrepancy at the domain level independent of the class information, even though samples from two domains should be aligned according to their semantic labels to ensure the discriminability of the features. In this paper, we propose a new contrastive MMD (CMMD), which can perform class-wise domain alignment (Figure 1). Inspired by the N-pair contrastive loss [18], our CMMD performs contrastive learning on probability distributions using the MMD as a similarity metric. Since this approach requires target domain labels, we adopt confidence-based filtering and entropy minimization to produce accurate pseudo-labels. Our method has been validated on the large-scale segmentation benchmarks SynLiDAR [26] (source dataset) and SemanticPOSS [12] (target dataset). Our contributions are as follows : (1) We present an MMD-based UDA approach for large-scale LiDAR point cloud semantic segmentation. (2) We propose a new contrastive MMD to conduct UDA. (3) We present a class-wise alignment module for UDA designed for LiDAR point cloud semantic segmentation.

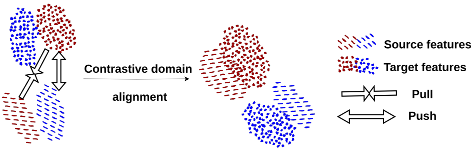

Figure 1: Our proposed contrastive domain alignment performs class-wise UDA across domains by pulling together similar class distributions and pushing away dissimilar ones. The resulting feature distributions are aligned class-wise.

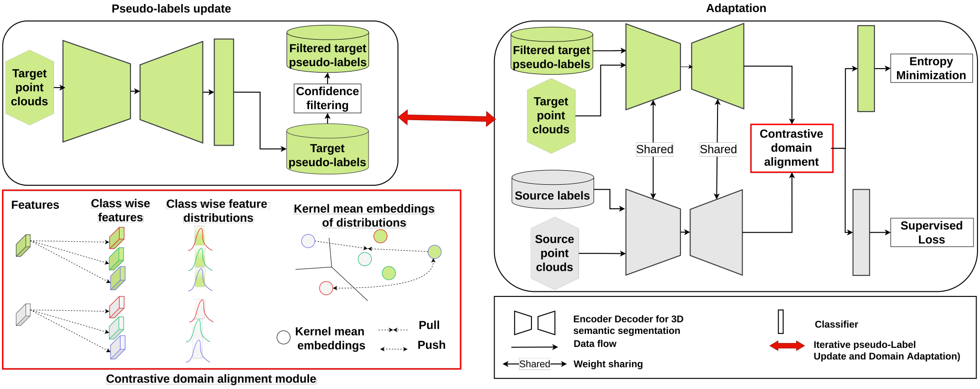
Pseudo-labels update
Target point clouds
Filtered target pseudo-labels
Confidence filtering
Target pseudo-labels
Adaptation
Filtered target pseudo-labels
Target point clouds
Shared
Shared
Contrastive domain alignment
Entropy Minimization
Source labels
Source point clouds
Supervised Loss
Features
Class wise features
Class wise feature distributions
Kernel mean embeddings of distributions
Kernel mean embeddings
Pull
Push
Contrastive domain alignment module
Encoder Decoder for 3D semantic segmentation
Data flow
Shared
Weight sharing
Classifier
Iterative pseudo-Label Update and Domain Adaptation)

Figure 2: Class-wise UDA: Source and target point clouds are fed to a pre-trained model. Then contrastive alignment is conducted on the task-specific layers from the decoder. Entropy minimization and confidence filtering are used to increase the accuracy of pseudo-labels.

2 Unsupervised domain adaptation for point cloud semantic segmentation

In recent years, several UDA techniques have been proposed that are based on deep neural networks. The underlying assumption of UDA is that we have access to label-rich samples in the source domain and label-scarce samples in the target domain. They can be generally grouped into domain mapping methods and domain-invariant feature learning methods. Compared with image UDA, point clouds have not been sufficiently explored.

Domain mapping methods aim at transferring the appearance of the target data into the annotated source data. Then, a model is trained and applied to the target domain. This is usually done through GANs. Numerous works apply existing GANs to projected images of the LiDAR points. For example, CycleGAN [33] can be used for translating real Bird-Eye-View (BEV) into synthetic BEVs obtained from synthetic point clouds [14, 15]. Zhao *et al.* [32] use a sensor-view image instead of a BEV. Then a GAN is used to simulate LiDAR dropout noise on real data and is applied to the synthetic data. Other lines of work attempt to conduct non-adversarial domain mapping. For example, Alonso *et al.* [1] use local augmentation operations to achieve alignment on the input space while minimizing the Kullback Leibler divergence between the source and target label distributions. Xiao *et al.* [26] decompose the synthetic-to-real gap into an appearance component and a sparsity component and use a 3D GAN to align the synthetic and real feature distribution at the input level and feature level. Recently, Yi *et al.* [28] propose an approach where UDA is formulated as a 3D surface completion task to transfer knowledge between different LiDAR sensors.

Domain-invariant feature learning methods aim at learning a feature representation that is shared between the source and target domains. Hence, if the model performs well on the source domain using this shared representation, then it will generalize well to the target domain. One way to achieve this is by minimizing a discrepancy measure between the source and target features. Wu *et al.* [25] use the geodesic distance between the output distributions of source and target, while updating batch normalization statistics based on the target domain. They rely on the squeezeSeg network [24] that performs a spherical projection of the 3D point clouds to obtain a projected image which can result in a loss of geometric information. MMD [6] is another popular discrepancy used to measure the distance between two probability distributions. By minimizing the MMD between source and target features, we can obtain domain invariant features. This has been used in [10, 30] for image classification by minimizing the MMD in the last layers of a convolutional neural network. Yan *et al.* [27] further introduce class weights to account for class weight bias in domain adaptation. For image semantic segmentation, Erkent and Laugier [4] adapt to varying weather using the MMD between source and target encoder features. The MMD has been applied in the context of point cloud classification of CAD objects to learn domain-invariant features both locally and globally [13]. However, works applying the MMD to 3D semantic segmentation of large-scale point clouds are lacking which can be attributed to the computational cost of these massive datasets. Another way to achieve domain invariant features is through adversarial training, either using projected LiDAR scans [9, 23] or 3D points clouds [11]. Domain invariant feature learning methods are based on a strongly developed theory and thus have better theoretical guarantees for learning from different domains [2]. Furthermore, since they operate in the feature space or the output space, they are by design related to the target segmentation task and can be combined with various deep learning architectures. Domain mapping approaches can be more challenging to train and are more susceptible to mapping distortions, which makes them less suitable for 3D point clouds. On the other hand, while the above-mentioned methods reduce the generalization gap of 3D semantic segmentation models, they conduct marginal feature alignment and don't consider the semantic information which has been shown crucial for domain adaptation of semantic segmentation tasks [31]. Some recent works begin addressing this problem. For example, Saltori *et al.* [16] propose a semantic mixing strategy combined with data augmentation to reduce the domain shift. In our work, we propose class-conditioned domain alignment to address the domain difference.

3 Proposed methodology

Our approach implements a UDA method for point cloud semantic segmentation that performs class-wise domain alignment. We exploit the MMD as a distance metric for class conditional feature distribution to bring closer distributions for the same classes regardless of the domain. Formally, we consider a set of source domain samples $S = \{(f_i^s, y_i^s)\}_{i=1}^{n_s}$ and target domain samples $T = \{(f_i^t, y_i^t)\}_{i=1}^{n_t}$, $f_i^s \sim \mathbb{P}_s$ are source features and $f_i^t \sim \mathbb{P}_t$ are target features such as those obtained through a deep neural network. $y_i^s = \{1, 2, ..., K\}$ are source labels for K classes. n_s and n_t represent the number of the source and target features respectively. We assume the target domain shares the same classes as the source domain and that target labels y_i^t are unknown. The goal of UDA is to train a model using the labeled source data that generalizes well to the target data. Figure 2 shows our approach. The source and target point clouds are fed to a 3D semantic segmentation network. The task-specific layers from the decoder are used to conduct the contrastive alignment. Specifically, the distribution of class-wise features from source and target are embedded into a reproducing kernel Hilbert space (RKHS) to obtain a kernel mean embedding per distribution. Then, using contrastive learning, the mean embeddings from the same class are pulled closer together using the MMD while those from different classes are pushed apart. Since class-wise contrastive alignment needs target labels, we use the network output for the target data as pseudo-labels.

Entropy minimization and confidence filtering are used to increase their accuracy.
In the rest of this section, we briefly review the concept of MMD as a distance metric for probability distributions in section 3.1. Then we introduce a new contrastive domain alignment metric in section 3.2. Finally in section 3.3, we discuss the training procedure using the proposed approach to conduct UDA for semantic segmentation of 3D point clouds.

3.1 Kernel mean embeddings of probability distributions

The empirical kernel mean embedding of the source and target feature distributions in the RKHS induced by kernel k are the elements $\mu_s = \frac{1}{n_s}\sum_{i=1}^{n_s} k(f_i^s,.)$ and $\mu_t = \frac{1}{n_t}\sum_{i=1}^{n_t} k(f_i^t,.)$ respectively. A useful property of the kernel mean embedding is that it captures all characteristics of probability distributions for a good choice of kernel. When the kernel is characteristic (like the RBF kernel), the mapping from probability distributions to kernel mean embeddings is injective and there is no information loss when mapping the distribution. Hence, the kernel choice controls how much information about the distribution is retained.
When the kernel mean embedding is unique, it can be used to define a metric for probability distributions. The maximum mean discrepancy is then defined as the distance between the kernel mean embeddings and is 0 if and only if the probability distributions are the same.:

$$MMD(\mathbb{P}_s, \mathbb{P}_t) = ||\mu_s - \mu_t||_H^2 \tag{1}$$

Using the kernel trick, the empirical estimate of the MMD is given by :

$$\begin{aligned} MMD(\mathbb{P}_s, \mathbb{P}_t) = \frac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s} k(f(x_i^s), f(x_j^s)) + \frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t} k(f(x_i^t), f(x_j^t)) \\ - \frac{2}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t} k(f(x_i^s), f(x_j^t)) \end{aligned} \tag{2}$$

3.2 Contrastive domain alignment

In contrastive learning, a contrastive loss aims to maximize the similarity between positive pairs and minimize the similarity between negative pairs. This will encourage the model to learn discriminative features that capture the underlying similarities within each domain while emphasizing the differences. Furthermore, utilizing probability distributions can incorporate higher-level information about the data like their moments. Therefore, we seek to apply this paradigm to feature distributions instead of feature instances. In order to learn from distributions, kernel mean embeddings provide an efficient representation that also preserves important information. In particular, a characteristic kernel enables a unique mapping and thus no loss of information. We consider the class conditional feature distribution of source and target $\mathbb{P}_s^c$ and $\mathbb{P}_t^c$ respectively. They can be embedded into kernel mean embeddings using a characteristic kernel. Let μ_s^c be the kernel mean embedding of the source distribution of class c. We use μ_t^+ and μ_t^- to denote positive and negative kernel mean embeddings, i.e. μ_s^c and μ_t^+ are from the same class and μ_t^- is a different class. Based on this, we can define an intra-class maximum mean discrepancy and an inter-class maximum mean discrepancy. To conduct class-wise unsupervised domain adaptation, the intra-class distance should be minimized while the inter-class distance should be maximized. This will make samples from a class compact regardless of the domain while ensuring samples from different classes have different distributions. Similar to metric learning, we would like to have a joint comparison among more than one negative kernel mean embedding. Inspired by the multi-class N pair contrastive loss [18], we formalize our proposed method of class-wise domain alignment: consider the following kernel mean embeddings: $\{\mu_s^c, \mu_t^+, \mu_{t1}^-, ..., \mu_{tK-1}^-\}$ where $\{\mu_{t1}^-, ..., \mu_{tK-1}^-\}$ are negatives. The contrastive domain alignment loss is defined as follows :

$$CMMD(\{\mu_s^c, \mu_t^+, \mu_{t1}^-, ..., \mu_{tK-1}^-\}) = -\log \frac{\exp^{-MMD(\mu_s^c, \mu_t^+)}}{\exp^{-MMD(\mu_s^c, \mu_t^+)} + \sum_j^{K-1} \exp^{-MMD(\mu_s^c, \mu_{tj}^-)}} \tag{3}$$

The contrastive domain alignment loss learns to identify a positive kernel mean embedding from multiple negative ones. It is worth noting that CMMD can be used for other interesting use cases such as label denoising of noisy target annotations. Exploiting the source domain knowledge can then help improve target domain labels.

3.3 Unsupervised Contrastive domain adaptation for semantic segmentation

The contrastive domain alignment module requires target labels for optimization. To estimate these labels, we use the network's predictions as pseudo-labels. Since the estimation can be noisy, we employ two strategies to improve the estimation quality: We perform confidence-based filtering on the estimated pseudo-labels. This will ensure that unconfident classes are not contributing to the domain alignment. The filtering is performed according to equation 4. p_i^c is the softmax output containing the class probabilities of point i and β is a confidence threshold to filter uncertain classes. We include entropy minimization [21] on the target predictions to incorporate target points that weren't included during domain alignment. This technique is commonly used in semi-supervised learning and clustering and is good for promoting the formation of compact clusters. Entropy minimization is performed using the loss described in equation 5. f_θ is the classifier layer of the model.

$$\hat{y}_i^{(c)} = \begin{cases} 1, \; if \; c = argmax \; p_i^c \;, p_i^c > \beta \\ 0, \; otherwise \end{cases} . \tag{4}$$

$$L^{ent} = -\frac{1}{n_t} \sum_{n_t} f_\theta(x_t) log(f_\theta(x_t)) \tag{5}$$

Although deep neural networks are able to learn more transferable features, deep features slowly transition from general to task-specific through the last layers of the network [29]. Therefore, the transferability gap becomes particularly large when transferring the higher layers. In line with previous work based on the MMD for image classification [10], we incorporate the domain alignment module over the activations of the last task-specific layer before the classification head. As a result, the overall objective minimizes the contrastive domain alignment loss over the activations of the last task-specific layer, the cross-entropy loss over the labeled source data, and the entropy loss over the unlabeled target data. Therefore, the overall objective is:

$$L = \lambda_1 L^{ce} + \lambda_2 CMMD + \lambda_3 L^{ent} \tag{6}$$

where L^{ce} is the supervised cross-entropy loss and λ_1, λ_2 and λ_3 are the weights of the supervised loss, the CMMD objective, and the entropy loss respectively.

4 Experiments

4.1 Datasets and baselines

To evaluate our method, we use the SynLiDAR dataset [26] as a source domain and SemanticPOSS as a real target domain [12]. SynLiDAR [26] is a large-scale synthetic dataset produced with the Unreal Engine. SynLiDAR is composed of 13 sequences containing a total of 198396 LiDAR frames annotated into 32 semantic classes. We randomly selected 19840 frames for training and 1976 for validation. SemanticPOSS [12] is a real-world dataset collected using the Pandora sensor. It contains 5 sequences and 2988 frames classified into 14 semantic classes. Sequence 03 is used for validation and the remaining sequences for training [12]. We follow Xiao *et al.* [26] in mapping SynLiDAR labels into 13 segmentation classes for the purposes of UDA. The segmentation performance is evaluated using the per-class Intersection over Union (IoU) and the mean Intersection over Union (mIoU) over all semantic classes before and after UDA.

We compare our method with the following state-of-the-art UDA approaches that have been applied to 3D LiDAR point clouds. ADDA [20] is based on adversarial learning, ST [34] and Ent-Min [21] represent the self-training paradigm. These three methods belong to the domain invariant feature learning category. PCT [26] belongs to the domain mapping category. Additionally, ST-PCT [26] is also included in the comparison which is a hybrid approach involving both a domain mapping and feature alignment component. These approaches are evaluated with respect to the same source model where the semantic segmentation network is trained on the source dataset and evaluated on the target. This constitutes the before adaptation performance. Furthermore, we also compare our method with CoSMIX, the current best-performing method for our dataset[16]. For a fair and realistic comparison, we report their source model performance that is used as a starting point for adaptation.

4.2 Implementation details

We used the same point cloud semantic segmentation network as previous methods, namely MinkUNet32 [3]. The network was pre-trained on the source domain to get the source model in Table 1. To do so, we used mini-batch stochastic gradient descent (SGD) with a learning rate of 0.01 and momentum of 0.9 for 50 epochs starting from randomly initialized weights. We fixed the batch size to 4 and the voxel size to 0.05. The source model in Table 2 is provided by CosMix [16] and was trained on the source domain for 10 epochs with Dice loss [19] and a batch size of 12. For adaptation, we initialized the model weights using the source model. We fixed $\lambda_1 = 2$, $\lambda 2 = 1$ and $\lambda 3 = 0.001$. We set the confidence threshold for pseudo-label filtering to $\beta = 0.85$. SGD was used with a learning rate of 0.001 and a momentum of 0.9. We fixed the batch size to 4 for both source and target and the voxel size to 0.05. For adaptation, the network parameters were shared between the source and target domain except for the batch normalization layers. For the CMMD kernel, we used the Gaussian RBF kernel $\mathbf{k}(\mathbf{x},\mathbf{x}') = \mathbf{exp}(\frac{-||\mathbf{x}-\mathbf{x}'||^2}{2\sigma^2})$ which is characteristic. The bandwidth parameter was chosen via the median heuristic: $\sigma^2 = \mathbf{median}\{||\mathbf{x_i} - \mathbf{x_j}||^2\}$. The MMD was computed using its unbiased estimate [6] which can be computed with linear complexity. At each iteration, we chose the set of classes that were shared between both source and target batches to compute CMMD. We also discarded any class for which the number of samples in either source or target was less than 10 samples. This helped stabilize the training process.

Table 1: Adaptation results on SynLiDAR → SemanticPOSS. The source corresponds to the model trained on the source dataset. Results are reported in terms of mean Intersection over the Union (mIoU).

Model	**pers**	**rider**	**car**	**trunk**	**plants**	**traf.**	**pole**	**garb.**	**buil.**	**cone.**	**fence**	**bike**	**grou.**	**mIoU**
Source	3.7	25.1	12.0	10.8	53.4	0.0	19.4	12.9	49.1	3.1	20.3	0.0	59.6	20.7
ADDA [20]	27.5	35.1	18.8	12.4	53.4	2.8	27.0	12.2	**64.7**	1.3	6.3	**6.8**	55.3	24.9
Ent-Min [21]	24.2	32.2	21.4	**18.9**	61.0	2.5	36.3	8.3	56.7	3.1	5.3	4.8	57.1	25.5
ST [34]	23.5	31.8	22.0	**18.9**	63.2	1.9	**41.6**	13.5	58.2	1.0	9.1	**6.8**	60.3	27.1
PCT [26]	13.0	**35.4**	13.7	10.2	53.1	1.4	23.8	12.7	52.9	0.8	13.7	1.1	66.2	22.9
ST-PCT [26]	28.9	34.8	**27.8**	18.6	**63.7**	**4.9**	41.0	**16.6**	64.1	1.6	12.1	6.3	63.9	29.6
CMMD (Ours)	**29.2**	28.1	24.1	13.1	63.4	2.2	33.2	9.1	61.3	**16.5**	**23.1**	2.0	**74.3**	29.2

Table 2: Adaptation results on SynLiDAR → SemanticPOSS. The source corresponds to the model provided by Saltori *et al.* [16]. Results are reported in terms of mean Intersection over the Union (mIoU)

.

Model	**pers**	**rider**	**car**	**trunk**	**plants**	**traf.**	**pole**	**garb.**	**buil.**	**cone.**	**fence**	**bike**	**grou.**	**mIoU**
Source	46.4	39.4	35.4	15.6	67.3	3.5	37.5	29.1	61.1	17.4	27.1	6.9	79.3	35.9
CoSMix [16]	**55.8**	**51.4**	**36.2**	**23.5**	**71.3**	**22.5**	34.2	28.9	**66.2**	20.4	24.9	**10.6**	**78.7**	40.4
CMMD (Ours)	54.3	46.8	**36.2**	18.5	68.0	4.3	**38.9**	**31.1**	64.0	**23.3**	**27.5**	4.9	63.8	37.1

4.3 Results

Table 1 summarizes the performance before and after adaptation starting from the same pre-trained model. As expected, the source-only performance is worse due to the domain gap. CMMD improves over the baseline source model mIoU by 9.5%. We can observe that PCT which is a domain mapping method performs worse than the domain invariant feature learning methods. This is expected as large-scale point clouds display both low-level and high-level domain differences. We can also observe that CMMD performs better than adversarial training and self-training-based methods in terms of mIoU. For some of the minority classes such as fence, cone, and person, the performance increase is more significant (+11% IoU for fence, +14.9% IoU for cone, and +0.3% IoU for person) . CMMD also attains a comparable performance to ST-PCT, which is a mixed approach relying jointly on GANs for input-level adaptation and self-training without the challenges of adversarial learning like instability and mode collapse [17]. We also notice instances of negative transfer, which describes the case when transferring knowledge from the source has a negative impact on the target performance. All methods suffer from this problem for the minority classes cone and fence except CMMD. This is important as the rare classes are more prone to poor generalization. For the latter, this only occurs for the class garbage can. Table 2 shows the performance before and after adaptation starting from the pre-trained model provided by Saltori *et al.* [16]. It is worth noting that the source model's performance already outperforms all the approaches in Table 1. We are therefore already starting from a well-performing model. CMMD improves the source model by 1.2% mIoU vs. 4.5% for CoSMix. Furthermore, CMMD reaches new state-of-the-art performance for the minority classes fence, cone, garbage-can and car. Although CoSMix has a better performance overall in terms of mIoU, we argue that CMMD is more robust against negative transfer considering the drop in performance of CosMix for the classes pole, garbage-can, fence, and ground. CMMD on the other hand shows a drop for the class ground and bike. Overall, CMMD has a competitive performance, especially since it could be complemented by the teacher-student learning scheme used by CosMix to generate pseudo-labels while being computationally less expensive. We also show in Figure

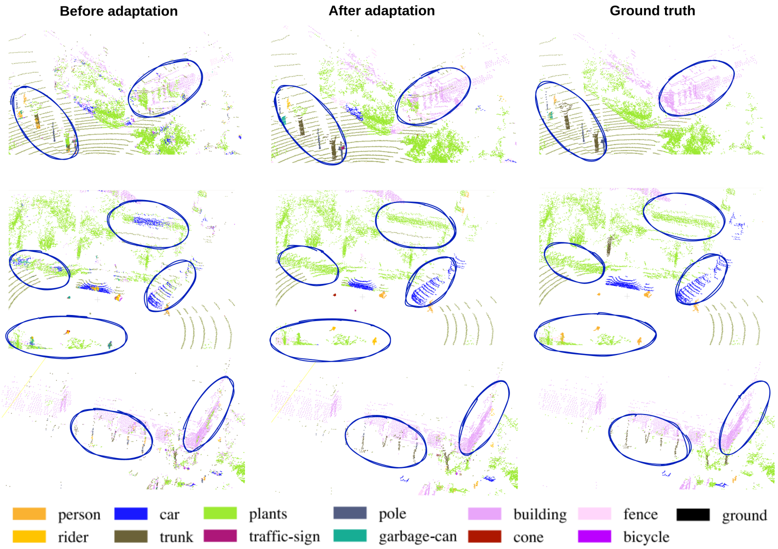
Before adaptation
After adaptation
Ground truth

Figure 3: Qualitative results on SemanticPOSS before and after adaptation. The regions where the adaptation is most visible are circled in blue.

3 the qualitative results of target point cloud predictions before and after adaptation using CMMD and the corresponding ground truth. These results were generated using the CMMD model reported in Table 1. We can see that some objects have less uniform predictions before adaptation, especially for the class trunk, buildings and cars. The adaptation step allows for better segmentation.
We observe promising results with CMMD's capacity to learn from a less efficient source model for UDA. However, further testing across datasets is essential for a conclusive evaluation, especially considering challenges on benchmarks like SynLiDAR-SemanticKITTI. Negative transfer remains a challenge, prompting future exploration of progressive adaptation strategies and class-weight integration into CMMD to mitigate majority class bias.

5 Conclusion

In this work, we proposed a novel contrastive maximum mean discrepancy (CMMD) to conduct unsupervised domain adaptation for semantic segmentation of large-scale LiDAR point clouds. Our proposed approach performs contrastive class-wise domain alignment by bringing closer same-class feature distributions while pushing apart different ones. We evaluated our method using large-scale real and synthetic LiDAR sequential point clouds. Experiments show that our approach outperforms the usual domain adaptation approaches for 3D point clouds and is competitive with the current best-performing technique while achieving new state-of-the-art performance for 4 minority classes and being more robust to negative transfer. To our knowledge, this is the first application of MMD for UDA for semantic segmentation of large-scale LiDAR point clouds.

References

[1] Inigo Alonso, Luis Riazuelo, Luis Montesano, and Ana C. Murillo. Domain adaptation in lidar semantic segmentation by aligning class distributions. *arXiv:2010.12239 [cs]*, 12 2021. URL http://arxiv.org/abs/2010.12239. arXiv: 2010.12239.

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Vaughan. A theory of learning from different domains. *Machine Learning*, 79:151–175, 05 2010. doi: 10.1007/s10994-009-5152-4.

[3] Christopher B. Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. *CoRR*, abs/1904.08755, 2019. URL http://arxiv.org/abs/1904.08755.

[4] Özgür Erkent and Christian Laugier. Semantic segmentation with unsupervised domain adaptation under varying weather conditions for autonomous vehicles. *IEEE Robotics and Automation Letters*, 5(2):3580–3587, 2020. doi: 10.1109/LRA.2020.2978666.

[5] Biao Gao, Yancheng Pan, Chengkun Li, Sibo Geng, and Huijing Zhao. Are we hungry for 3d lidar data for semantic segmentation? a survey and experimental study. *arXiv:2006.04307 [cs]*, 11 2020. URL http://arxiv.org/abs/2006.04307. arXiv: 2006.04307.

[6] Arthur Gretton, Karsten Borgwardt, Malte J. Rasch, Bernhard Scholkopf, and Alexander J. Smola. A kernel method for the two-sample problem, 2008.

[7] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation, 2022.

[8] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation, 2022.

[9] Peng Jiang and Srikanth Saripalli. Lidarnet: A boundary-aware domain adaptation model for lidar point cloud semantic segmentation. *CoRR*, abs/2003.01174, 2020. URL https://arxiv.org/abs/2003.01174.

[10] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks, 2015.

[11] Haifeng Luo, Kourosh Khoshelham, Lina Fang, and Chongcheng Chen. Unsupervised scene adaptation for semantic segmentation of urban mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169:253–267, 11 2020. ISSN 09242716. doi: 10.1016/j.isprsjprs.2020.10.002.

[12] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. *CoRR*, abs/2002.09147, 2020. URL https://arxiv.org/abs/2002.09147.

[13] Can Qin, Haoxuan You, Lichen Wang, C. C. Jay Kuo, and Yun Fu. Pointdan: A multiscale 3d domain adaption network for point cloud representation, 2019.

[14] Khaled Saleh, Ahmed Abobakr, Mohammed Attia, Julie Iskander, Darius Nahavandi, and Mohammed Hossny. Domain adaptation for vehicle detection from bird's eye view lidar point cloud data. *arXiv:1905.08955 [cs]*, 5 2019. URL http://arxiv.org/abs/1905.08955. arXiv: 1905.08955.

[15] Ahmad El Sallab, Ibrahim Sobh, Mohamed Zahran, and Nader Essam. Lidar sensor modeling and data augmentation with gans for autonomous driving. *CoRR*, abs/1905.07290, 2019. URL http://arxiv.org/abs/1905.07290.

[16] Cristiano Saltori, Fabio Galasso, Giuseppe Fiameni, Nicu Sebe, Elisa Ricci, and Fabio Poiesi. Cosmix: Compositional semantic mix for domain adaptation in 3d lidar segmentation. 7 2022. URL http://arxiv.org/abs/2207.09778. arXiv:2207.09778 [cs].

[17] Divya Saxena and Jiannong Cao. Generative adversarial networks (gans survey): Challenges, solutions, and future directions, 2023.

[18] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 1857–1865, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

[19] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In M. Jorge Cardoso, Tal Arbel, Gustavo Carneiro, Tanveer Syeda-Mahmood, João Manuel R.S. Tavares, Mehdi Moradi, Andrew Bradley, Hayit Greenspan, João Paulo Papa, Anant Madabhushi, Jacinto C. Nascimento, Jaime S. Cardoso, Vasileios Belagiannis, and Zhi Lu, editors, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248, Cham, 2017. Springer International Publishing.

[20] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017. doi: 10.1109/CVPR.2017.316.

[21] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2512–2521, 2019. doi: 10.1109/CVPR.2019.00262.

[22] Yuxi Wang, Junran Peng, and Zhaoxiang Zhang. Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9072–9081, 2021. doi: 10.1109/ICCV48922.2021.00896.

[23] Ze Wang, Sihao Ding, Ying Li, Minming Zhao, Sohini Roychowdhury, Andreas Wallin, Guillermo Sapiro, and Qiang Qiu. Range adaptation for 3d object detection in lidar. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2320–2328, 2019. doi: 10.1109/ICCVW.2019.00285.

[24] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. 10 2017. URL http://arxiv.org/abs/1710.07368. arXiv:1710.07368 [cs].

[25] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. *arXiv:1809.08495 [cs]*, 9 2018. URL http://arxiv.org/abs/1809.08495. arXiv: 1809.08495.

[26] Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Transfer learning from synthetic to real lidar point cloud for semantic segmentation, 2021.

[27] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation, 2017.

[28] Li Yi, Boqing Gong, and Thomas Funkhouser. Complete label: A domain adaptation approach to semantic segmentation of lidar point clouds. pages 15358–15368, Nashville, TN, USA, 6 2021. IEEE. ISBN 978-1-66544-509-2. doi: 10.1109/CVPR46437.2021.01511. URL https://ieeexplore.ieee.org/document/9578920/. [Online; accessed 2022-01-16].

[29] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks?, 2014.

[30] Wen Zhang and Dongrui Wu. Discriminative joint probability maximum mean discrepancy (djp-mmd) for domain adaptation, 2020.

[31] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Dong Liu, and Tao Mei. Transferring and regularizing prediction for semantic segmentation. pages 9618–9627, Seattle, WA, USA, 6 2020. IEEE. ISBN 978-1-72817-168-5. doi: 10.1109/CVPR42600.2020.00964. URL https://ieeexplore.ieee.org/document/9156873/. [Online; accessed 2022-03-12].

[32] Sicheng Zhao, Yezhen Wang, Bo Li, Bichen Wu, Yang Gao, Pengfei Xu, Trevor Darrell, and Kurt Keutzer. epointda: An end-to-end simulation-to-real domain adaptation framework for lidar point cloud segmentation. page 10, 2021.

[33] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. pages 2242–2251, Venice, 10 2017. IEEE. ISBN 978-1-5386-1032-9. doi: 10.1109/ICCV.2017.244. URL http://ieeexplore.ieee.org/document/8237506/. [Online; accessed 2022-03-12].

[34] Yang Zou, Zhiding Yu, Xiaofeng Liu, B. V. K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. *CoRR*, abs/1908.09822, 2019. URL http://arxiv.org/abs/1908.09822.