2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

# LiOn-XA: Unsupervised Domain Adaptation via LiDAR-Only Cross-Modal Adversarial Training

Thomas Kreutz[1], Jens Lemke[1], Max Mühlhäuser[1] and Alejandro Sanchez Guinea[1]

***Abstract*— In this paper, we propose LiOn-XA, an unsupervised domain adaptation (UDA) approach that combines LiDAR-Only Cross-Modal (X) learning with Adversarial training for 3D LiDAR point cloud semantic segmentation to bridge the domain gap arising from environmental and sensor setup changes. Unlike existing works that exploit multiple data modalities like point clouds and RGB image data, we address UDA in scenarios where RGB images might not be available and show that two distinct LiDAR data representations can learn from each other for UDA. More specifically, we leverage 3D voxelized point clouds to preserve important geometric structure in combination with 2D projection-based range images that provide information such as object orientations or surfaces. To further align the feature space between both domains, we apply adversarial training using both features and predictions of both 2D and 3D neural networks. Our experiments on 3 real-to-real adaptation scenarios demonstrate the effectiveness of our approach, achieving new state-of-the-art performance when compared to previous uni- and multi-model UDA methods. Our source code is publicly available at `https://github.com/JensLe97/lion-xa`.**

## I. Introduction

Supervised deep learning models for LiDAR semantic segmentation trained on one dataset usually encounter domain shifts when being tested on a different dataset, which often arises from changes to geographic regions or the type of LiDAR sensor (e.g., [1]). In settings where no labels from the target domain are available, various unsupervised domain adaptation (UDA) methods have been proposed to enhance model robustness and prevent performance degradation caused by these shifts. Common UDA methods may target (i) domain-invariant data representations for deep neural networks (e.g., [1], [2], [3]) or (ii) domain-invariant feature representations using cross-modal learning strategies (e.g., [4], [5]), which can be combined with adversarial training (e.g., [6], [7]) or contrastive learning (e.g., [8]). Recent advances in learning domain-invariant feature representations from a combination of RGB and LiDAR data have been shown to be effective for UDA. In spite of this, and especially for robotic applications, there are relevant scenarios in which RGB data might not be available or in which its usage might pose strong privacy concerns (e.g., [9], [10]).

For scenarios where only LiDAR data is available and no target domain labels can be obtained, we propose LiOn-XA, a UDA approach that combines **LiDAR-Only** Cross-Modal (**X**) learning from two different LiDAR data representation with **A**dversarial training. We motivate LiOn-XA from recent advances in LiDAR semantic segmentation, where fusing multiple LiDAR data representations helps to mitigate inherent issues of these representations (e.g., [11], [12], [13]). For instance, voxelization introduces quantization loss and computation can grow cubicly with increasing resolution (limiting the receptive field), while range image-based representations distort physical dimensions (e.g., [13]). However, voxelization preserves important geometric structure that benefits segmenting *thing* classes, while range images can be processed with a much larger receptive field.

LiOn-XA combines two LiDAR data representations with a cross-modal mimicking task (e.g., [4]) to obtain predictions that are more robust to a domain shift. We leverage a) voxelized point clouds to preserve the geometric structure, and b) corresponding range images for information about, for instance, the orientation and surface of objects. In addition, we align the feature spaces of the source and target domain with adversarial training. Unlike prior works that either leverage 2D features or 3D predictions (e.g., [6], [7]), LiOn-XA leverages both to learn better domain-invariant feature representations. That is, during training, LiOn-XA aligns both 2D features as well as 2D and 3D predictions across domains.

We evaluate the effectiveness of our approach on 3 real-to-real adaptation scenarios from different urban environments and sensor characteristics. Our experiments show that LiOn-XA outperforms state-of-the-art uni- and multi-modal strategies. Our main contributions are as follows:

- A new method for UDA that is based on cross-modal learning between two representations of the same LiDAR data, i.e., between voxelized LiDAR point clouds and their range image projections.
- We propose a novel UDA approach called LiOn-XA, which combines LiDAR-only cross-modal learning with adversarial training to bridge country-to-country as well as sensor-to-sensor domain gaps.

## II. Related Work

### A. Domain Mapping

Domain mapping approaches seek to obtain a common input data representation in terms of visual appearance of both the source and target domain. For instance, the works in [2], [14] address cross-sensor UDA by transforming source domain LiDAR data with a high beam size to the low resolution of the target domain. A different approach, unpaired mask transfer (UMT) [15] operates on range images

[1]The authors are with the Telecooperation Lab at the Technical University Darmstadt, Germany, {<kreutz, lemke, sanchez>@tk, <max>@informatik}.tu-darmstadt.de

and maps source data to the same sparsity of the target domain. Recent work in [3] proposes a self-training LiDAR UDA approach that generates reliable pseudo labels for the target domain using cross-frame ensembling to aggregate predictions from multiple frames on a common input data representation of source and target domain.

### B. Domain-Invariant Data Representation

Domain-invariant data representation methods aim to learn a common input representation among both domains, which does not necessarily yields a similar visual appearance that is based on one domain. For instance, Complete&Label [1] address cross-sensor UDA with a shared voxel completion network, which transforms both source and target data into a common LiDAR representation. While addressing differences in sensor characteristics, such alignment strategies are not necessarily designed to bridge domain gaps for geographic-to-geographic scenarios.

### C. Domain-Invariant Feature Representation

Learning domain-invariant feature representations is usually addressed by forcing the backbones to learn a distribution over the extracted feature representation that is similar across both source and target domain. Different works proposed to learn such a feature representation by, for instance, cross-modal learning between RGB images and LiDAR point clouds (e.g., [4], [5]) or a combination of both cross-modal learning with adversarial training (e.g., [6], [7]). Recent work in [8] also proposed a cross-modal contrastive learning approach, which, analogous to adversarial training, aims to directly align corresponding pixel and point features across both domains. However, the assumption that RGB images synchronized with LiDAR data in both source and target domains are always available may not always hold, which limits the practicality of such methods (e.g., [16]).

### D. Multi-Modal LiDAR Semantic Segmentation

Recent state-of-the-art approaches for semantic segmentation, such as SPVNAS [12], $(AF)^2$-S3Net [11], RPVNet [13], 2DPASS [17] show that fusion-based methods that leverage LiDAR-only representations (i.e., Point Cloud, Voxels, Range Image) or LiDAR and RGB images in a cross-modal learning setting, complement each other and help mitigating issues of individual LiDAR representations to improve the overall performance.

Compared to recent state-of-the-art LiDAR UDA methods that depend on RGB and LiDAR data, our approach especially accounts for sensor-to-sensor and country-to-country UDA scenarios where only LiDAR point cloud data is available. Different from previous works, we propose to combine cross-modal learning (e.g., [4]) between two different LiDAR representations with adversarial training (e.g., [6], [7]) to learn domain-invariant feature representations for LiDAR-only UDA and only depend on LiDAR data.

## III. Approach

Typically, cross-modal learning methods (e.g., [4]), treat image data provided by cameras and 3D LiDAR point clouds as two distinct modalities. Under the assumption that 2D and 3D LiDAR data representations can learn from each other, we focus on a cross-modal learning setting for scenarios where only 3D LiDAR point cloud data is available. In this setting, we propose LiOn-XA, **Li**DAR-**On**ly Cross-Modal (**X**) learning with **A**dversarial training, where we use voxelized 3D LiDAR point clouds as a first modality and their corresponding 2D range images as a second modality.

### A. Overview

Figure 1 depicts an overview of LiOn-XA, which consists of a source, target, and discriminator module. The source and target domain modules process the respective point clouds and perform cross-modal learning. The voxelized point clouds are processed by a 3D network, while a different 2D network processes the corresponding range-image representations. Both networks do not share features.

During training, cross-modal learning in both source and target modules enforces consistency between both network outputs, which in return improves the capabilities of both segmentation networks. However, the learning of both modalities in the source and target modules are mutually independent. We follow the idea in [7] and use adversarial training to enforce an alignment between the features and predictions from both domains and modalities. The 2D and 3D segmentation networks serve as generators, and we include three discriminator networks inside the discriminator module for adversarial training.

### B. Supervised Training

We train our approach on a source domain dataset $\mathcal{S}$ that consists of 3D point clouds $\boldsymbol{x}_s^{3D} \in \mathbb{R}^{N\times 3}$ with their corresponding 3D segmentation labels $\boldsymbol{y}_s^{3D} \in [\![1, C]\!]^N$.

Concerning sensor-to-sensor UDA, training with additional target-like data (e.g., [2]) improves the performance of our approach (see Section IV-D) by compensating for different sensor configurations regarding beam size. In this case, we consider an additional target-like domain dataset $\mathcal{T}_\ell$ consisting of point clouds $\boldsymbol{x}_{tl}^{3D}$ with the associated ground truth $\boldsymbol{y}_{tl}^{3D}$. More specifically, following the approach in [2], the target-like point cloud data $\mathcal{T}_\ell$ is obtained by transforming $\boldsymbol{x}_s^{3D} \in \mathbb{R}^{N\times 3}$, so that its beam size aligns with the beam size of the target domain. We denote the corresponding range images of the source and target-like dataset as $\boldsymbol{x}_s^{2D}$ and $\boldsymbol{x}_{tl}^{2D}$.

*Pixel- and Point-wise Segmentation Loss:* With provided labels, we train the 3D segmentation network in a supervised manner using the well-known cross-entropy loss:

$$\mathcal{L}_{\text{seg}}^{3D}(\boldsymbol{x}, \boldsymbol{y}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} w_c \boldsymbol{y}^{(n,c)} \log \mathbf{P}_{\boldsymbol{x}}^{(n,c)} \qquad (1)$$

while the 2D segmentation network is trained with:

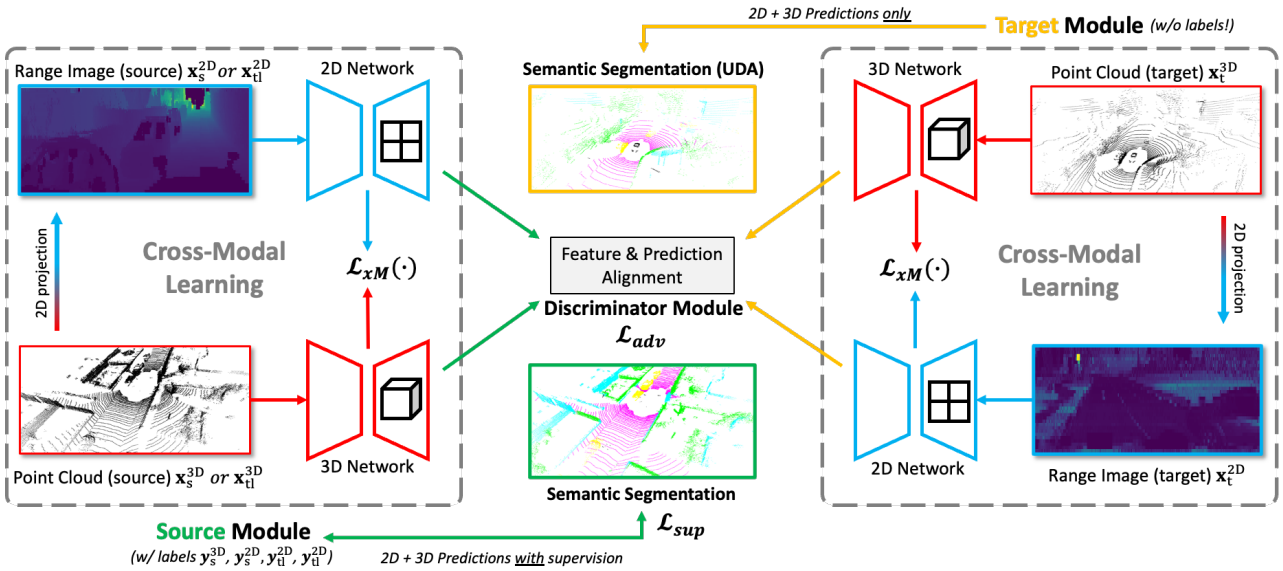



Fig. 1: LiOn-XA consists of a source and target module. The source module optimizes the 2D and 3D networks with a supervised segmentation loss $\mathcal{L}_{sup}$ on the source domain data as well as target-like data. The target module contains unlabelled data only from the target domain. Both modules optimize both networks with a respective cross-modal loss $\mathcal{L}_{xM}(\cdot)$. Finally, the discriminator module further connects source and target representations for unsupervised domain adaptation using an adversarial loss on the feature representations $\mathcal{L}_{adv}$ to enforce feature alignment between both domains.

$$\mathcal{L}_{\text{seg}}^{2D}(\boldsymbol{x}, \boldsymbol{y}) = -\frac{1}{H \cdot W} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{c=1}^{C} \Big( w_c \boldsymbol{y}^{(h,w,c)} \log \mathbf{P}_{\boldsymbol{x}}^{(h,w,c)} \Big) \quad (2)$$

where $\boldsymbol{x}$ is either $\boldsymbol{x}_s^{2D}$, $\boldsymbol{x}_s^{3D}$, $\boldsymbol{x}_{tl}^{2D}$, or $\boldsymbol{x}_{tl}^{3D}$, and $\boldsymbol{y}$ is equal to to the annotations $\boldsymbol{y}_s^{2D}$, $\boldsymbol{x}_s^{3D}$, $\boldsymbol{y}_{tl}^{2D}$ or $\boldsymbol{y}_{tl}^{3D}$. Similar to [4], we compensate for class imbalances by weighting each point by its log-smoothed classed weight $w_c$.

Overall, the objective function for the 2D and 3D stream from the source and target-like domain reads:

$$\mathcal{L}_{\text{sup}} = \mathcal{L}_{\text{seg}}^{3D}(\mathcal{S}) + \lambda_p \mathcal{L}_{\text{seg}}^{2D}(\mathcal{S}) + \mathcal{L}_{\text{seg}}^{3D}(\mathcal{T}_\ell) + \lambda_p \mathcal{L}_{\text{seg}}^{2D}(\mathcal{T}_\ell) \quad (3)$$

where $\mathcal{L}_{\text{seg}}^{2D}(\cdot)$ and $\mathcal{L}_{\text{seg}}^{3D}(\cdot)$ denote the loss over the whole respective datasets $\mathcal{S}$ and $\mathcal{T}_\ell$. We choose $\lambda_p < 1$ as a weighting hyperparameter for $\mathcal{L}_{\text{seg}}^{2D}$ to limit the impact on the objective function for labels obtained by a projection in which information is lost.

### C. LiDAR-Only Cross-Modal Training

The goal of cross-modal learning is to enforce consistent predictions between two modalities. Intuitively, we aim to transfer knowledge from one LiDAR data modality to the other by implementing a mimicking task. This is based on the assumption that one modality could be more sensitive to a domain shift than the other (e.g., [4]). For instance, point clouds lack connectivity information of an underlying surface, which is introduced by a range image projection. In this case, the more robust range image acts as a teacher to guide the more sensitive point cloud.

*a) Learning from Range Images and Point Clouds:* To complement the structural information of LiDAR point clouds, we combine them with 2D range images that we construct from the point cloud data. We hypothesize that this combination allows to extract further relevant information from the data that is more robust against domain shifts. Figure 2 shows a range image that has five channels, displayed as three separate images. From top to bottom, Figure 2 consists of a range map, remission map, and normal map. By transferring the unstructured and sparse point cloud into a structured and dense range map, object contours become more visible compared to 3D point clouds. Remission values can be a major cue for the recognition of objects. For example, the high reflection of shiny *thing* classes, such as cars or traffic signs, highlighted in white in the middle of Figure 2, facilitates the learning of these categories.

To benefit from the last component of the range image, we include normal maps as 3 separate channels for the coordinates $x$, $y$, and $z$. In contrast to other projection-based approaches (e.g., [14], [15], [18]), we do not add the 3D coordinates directly into the images. This information is already included in the point cloud data for the 3D stream. Further, point clouds from different LiDAR sensors have differently oriented coordinate systems, which may negatively affect the adaptation process. Since the normal map values are in a fixed range and provide meaningful information about orientation, we use them as a type of canonical domain, comparable to [1]. The model benefits from both the information related to edges and from the orientation that resembles a surface in 3D space. Point

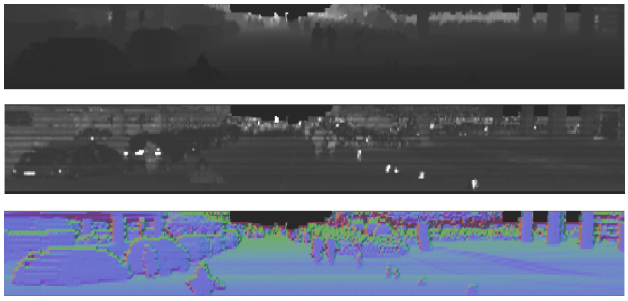


Fig. 2: Range image consisting of a range, remission and normal map.

clouds lack this connectivity information of an underlying mesh [19].

*b) LiDAR-Only Cross-Modal Loss:* To learn a mapping from 2D to 3D and vice versa, we use a cross-modal loss proposed by [4], which can be formulated as:

$$\mathcal{L}_{\mathrm{xM}}\left(\boldsymbol{x}\right) = D_{\mathrm{KL}}\Big(\mathbf{P}_{\boldsymbol{x}}^{(n,C)} \Big\| \mathbf{Q}_{\boldsymbol{x}}^{(n,C)}\Big) \tag{4}$$

$$= -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C}\mathbf{P}_{\boldsymbol{x}}^{(n,c)}\log\frac{\mathbf{P}_{\boldsymbol{x}}^{(n,c)}}{\mathbf{Q}_{\boldsymbol{x}}^{(n,c)}} \tag{5}$$

where $D_{\mathrm{KL}}(\cdot)$ denotes the Kullback-Leibler divergence. In Equation 4 and Equation 5, $\mathbf{P}_{\boldsymbol{x}}$ is the distribution of the main prediction that either originates from the 2D or 3D stream and $\mathbf{Q}_{\boldsymbol{x}}$ is the mimicry prediction that should be learned. We strive for consistency between both mimicry predictions $\mathbf{P}^{2D\to 3D}$ and $\mathbf{P}^{3D\to 2D}$.

Since the cross-modal loss is unsupervised, we apply this function to all available datasets, namely the source, target-like, and target domain. In total, the objective function from Equation 3 extends to:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{sup}} + \lambda_s\mathcal{L}_{\mathrm{xM}}(\mathcal{S}) + \lambda_{tl}\mathcal{L}_{\mathrm{xM}}(\mathcal{T}_\ell) + \lambda_t\mathcal{L}_{\mathrm{xM}}(\mathcal{T}) \tag{6}$$

where $\mathcal{L}_{\mathrm{xM}}(\cdot)$ denotes the loss over the whole respective datasets $\mathcal{S}$, $\mathcal{T}_\ell$ or $\mathcal{T}$, with $\lambda_s$, $\lambda_{tl}$ and $\lambda_t$ being hyperparameters that weight the influence of the cross-modal loss $\mathcal{L}_{\mathrm{xM}}$. The target domain dataset is denoted as $\mathcal{T}$ and only contains point clouds $\boldsymbol{x}_t^{3D}$ without annotations.

### D. Discriminator Module and Adversarial Training

We train 3 discriminator networks to realize the adversarial domain adaptation procedure. The first discriminator $D_{s^{2D}\Leftrightarrow t^{2D}}$ distinguishes between 2D source and 2D target features and outputs the probability that the input belongs to the target domain. In this setting, the 2D network as the generator is trained and updated to fool the discriminator. Likewise, the other two discriminator networks take predictions from mixed modalities and domains as input and predict the corresponding domain, i.e., $D_{s^{2D}\Leftrightarrow t^{3D}}$ and $D_{s^{3D}\Leftrightarrow t^{2D}}$.

*a) Discriminator Loss:* We label the source domain as 0 and the target domain as 1, and the discriminator is trained to predict the corresponding source or target label. We let $\mathcal{L}_{\mathrm{BCE}}$ be the binary cross-entropy loss for prediction $x$ and label $y$. For the feature maps $\bar{\mathbf{F}}_s^{2D}$ and $\bar{\mathbf{F}}_t^{2D}$ from the 2D network, the discriminator loss is:

$$\min_{\theta_D}\Bigg[\frac{1}{|\mathcal{S}|}\sum_{\boldsymbol{x}_s\in\mathcal{S}}\lambda_{D_{sf}^{2D}}\mathcal{L}_{\mathrm{BCE}}\big(D\big(\bar{\mathbf{F}}_s^{2D}\big),0\big) + \frac{1}{|\mathcal{T}|}\sum_{\boldsymbol{x}_t\in\mathcal{T}}\lambda_{D_{tf}^{2D}}\mathcal{L}_{\mathrm{BCE}}\big(D\big(\bar{\mathbf{F}}_t^{2D}\big),1\big)\Bigg] \tag{7}$$

where $\lambda_{D_{sf}^{2D}}$ and $\lambda_{D_{tf}^{2D}}$ are weighting hyperparameters for the source and target features $sf$ and $tf$, and $\theta_D$ are the parameters for discriminator $D$.

*b) Adversarial Objective:* We update the 2D network $\theta_{2D}$ by training with the adversarial objective:

$$\min_{\theta_{2D}}\frac{1}{|\mathcal{T}|}\sum_{x_t\in\mathcal{T}}\lambda_{G_{tf}^{2D}}\mathcal{L}_{\mathrm{BCE}}\left(D\big(\bar{\mathbf{F}}_t^{2D}\big),0\right) \tag{8}$$

where $\lambda_{G_{tf}^{2D}}$ is a weighting hyperparameter for the loss of generator $G$. If we set the label of $\mathcal{L}_{\mathrm{BCE}}$ to 0, the resulting term is $-\log\big(1-D\big(\bar{\mathbf{F}}_t^{2D}\big)\big)$. The generator tries to minimize this loss, which forces the discriminator to output the wrong label for a given target domain input.

The same principle applies to the other two discriminator networks that take source and target predictions as input. The only difference is that both discriminators have the goal to align the distribution of the respective source and target domain predictions $\mathbf{P}_t^{2D} \Leftrightarrow \mathbf{P}_s^{3D}$ and $\mathbf{P}_s^{2D} \Leftrightarrow \mathbf{P}_t^{3D}$.

### E. Training Details

In total, we optimize a three-fold objective loss function in the same iteration over one batch. First, we train the segmentation networks with the supervised and cross-modal loss from Equation 6. Second, we update the segmentation networks according to Equation 8. Finally, we update the discriminators on both the source and target data using Equation 7. We weigh all three generator and six discriminator loss terms using hyperparameters that have been determined empirically based on prior works, which are summarized in the supplementary material.

## IV. Experiments

We first outline our experimental setup and design. Afterward, we evaluate our approach quantitatively and qualitatively against state-of-the-art approaches on three different real-to-real domain adaptation scenarios. Furthermore, we perform an ablation study that highlights the influence of different components of our approach.

### A. Datasets

We evaluate our approach on 4 publicly available large-scale datasets that are designed for autonomous driving. More specifically, we follow related work (e.g., [4], [6], [7], [20]) and use nuScenes [21], nuScenes-Lidarseg [22], SemanticKITTI [23], and SemanticPOSS [24] to construct country-to-country, dataset-to-dataset, and sensor-to-sensor domain adaptation scenarios. For the first scenario, we split the nuScenes dataset into LiDAR scans recorded in USA and

| Method | nuScenes: USA → SG | | | nS-Lidarseg: USA → SG | | | SemanticKITTI → nS-Lidarseg | | |
|---|---|---|---|---|---|---|---|---|---|
| | 2D | 3D | 2D+3D | 2D | 3D | 2D+3D | 2D | 3D | 2D+3D |
| Baseline (source only) | 53.4 | 46.5 | 61.3 | 58.4 | 62.8 | 68.2 | 47.6 | 54.9 | 61.5 |
| xMUDA [4], [5] | 59.3 | 52.0 | 62.7 | 64.4 | 63.2 | 69.4 | **57.6** | 57.7 | 63.2 |
| AUDA [6] | 59.8 | 52.0 | 63.1 | – | – | – | – | – | – |
| DsCML + CMAL [7] | **63.4** | 55.6 | 64.8 | 65.6 | 56.2 | 66.1 | – | – | – |
| CM-CL+PL [8] | 63.3 | 57.1 | 66.7 | – | – | – | – | – | – |
| LiOn-XA (ours) | 58.0 | **63.4** | **68.9** | **67.2** | **69.7** | **72.8** | 51.9 | **70.7** | **71.3** |
| Oracle (target only) | 66.4 | 63.8 | 71.6 | 75.4 | 76.0 | 79.6 | 75.4 | 76.0 | 79.6 |
| Unsupervised advantage | 4.6 | 17.3 | 7.6 | 8.8 | 6.9 | 4.6 | 4.3 | 15.8 | 9.8 |
| Domain gap | 13.0 | 17.5 | 10.3 | 17.0 | 13.2 | 11.4 | 27.8 | 21.1 | 18.1 |
| Closed gap | 35.4 | 97.7 | 73.8 | 51.8 | 52.3 | 40.4 | 15.5 | 74.9 | 54.1 |

TABLE I: Quantitative results on the nuScenes, nuScenes-Lidarseg (nS-Lidarseg) and SemanticKITTI dataset from the USA to Singapore (SG). We report the mIoU for each modality (2D and 3D) as well as their ensemble (2D + 3D).

Singapore. Since all three datasets are captured by different LiDAR sensors, we investigate two cross-sensor adaptations from SemanticKITTI to nuScenes-Lidarseg as well as from SemanticKITTI to SemanticPOSS.

### *B. Implementation Details*

We use the official SalsaNext [25] implementation as 2D semantic segmentation network with 32 feature channels. Similar to [20], we replace all batch normalization layers with instance normalization for better conversion and performance results of the generator module. We cut the $360°$ range image and randomly take a $512$ pixel wide cutout with the same height as the target beam size. Therefore, we are able to utilize data from all viewpoints and are not limited to a front camera view as in [4]. Further, we use random horizontal flips and randomly remove parts of the range image as a data augmentation. Finally, we normalize each channel of the range image by subtracting the mean and dividing by the standard deviation of the source dataset.

To further facilitate the adaptation process, we align the height $H$ and width $W$ of the source range images to the dimensions of the target domain. Other works have aligned the target range images to the source domain or to the one with a smaller height [20]. We argue that a transformation to the target domain benefits the learning process, as no ground truth is available in this domain.

For our 3D stream segmentation network, we use the official implementation of SparseConvNet [26] with a U-Net architecture. In the same way as [4], we implement the submanifold sparse convolution with 16 U-Net features and 6 times downsampling. To obtain a voxelized grid with only one 3D point per voxel, we set the voxel size to $5\,\mathrm{cm}$. For the 3D data augmentation, we implement random rotation, translation, and flipping of the $x$- and $y$-axis.

*a) Training:* Our models are trained from scratch on a single NVIDIA Tesla V100S (PCIe) with 32 GB RAM with a batch size of 8 for a maximum number of iterations of 100k. In each iteration, we accumulate the gradients of the source, target-like, and target batch to jointly train both modality networks. The values for the hyperparameters in our loss functions have been selected empirically and vary across the different UDA evaluation scenarios. For the sake of reproducibility, we provide a full list of all hyperparameters in the supplementary material. We optimize the 2D segmentation network with stochastic gradient descent (SGD), an initial learning rate of $2.5\times\ 10^{-3}$, and momentum $0.9$.

The 3D segmentation network is trained using ADAM with a learning rate of $10^{-3}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. For both the 2D and the 3D training procedure, we choose a Multi Step Learning rate scheduler with $\gamma = 0.1$ and milestones at iterations 80k and 90k at which the learning rate is multiplied by $\gamma$.

The discriminator networks are optimized with ADAM, a learning rate of $10^{-4}$, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. We schedule the learning rate by following a polynomial annealing procedure as used in [27]. In this policy learning rate is multiplied by $\left(1 - \frac{iter}{max\_iter}\right)^{power}$ where $iter$ is the current iteration, $max\_iter$ is the maximum number of iterations, and $power$ is the "poly" power that we set to $0.9$.

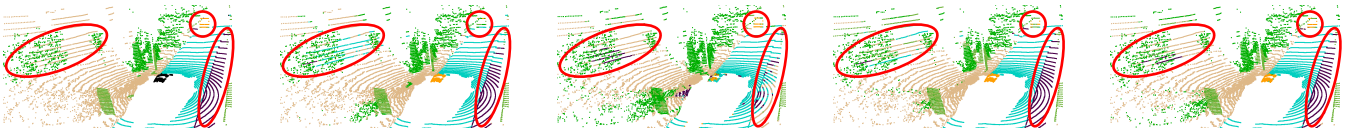
*b) Metrics:* We evaluate the performance of our approach with the well-known intersection-over-union (IoU) and corresponding mean IoU (mIoU) metric. The mIoU is reported separately for both the 2D and 3D segmentation models. The model for each modality is chosen independently based on their checkpoints achieving the best mIoU on the validation set. Finally, we report the softmax average over both probability distributions as in [4].

### *C. Quantitative Results*

The mIoU results for the USA to Singapore (SG) and the SemanticKITTI to nuScenes-Lidarseg domain adaptation scenarios are summarized in Table I. We evaluate LiOn-XA against four state-of-the-art methods xMUDA [4], AUDA [6], DsCML + CMAL [7], and CM-CL+PL [8]. The baseline is trained only on the source domain, while the oracle is trained only on the target domain. At the bottom, we further report the difference between LiOn-XA and the baseline (unsupervised advantage), the difference between oracle and baseline performance (domain gap), and how much of the domain gap has been closed by applying LiOn-XA (closed gap, i.e., $\frac{IoU(\text{LiOn-XA}) - IoU(\text{baseline})}{IoU(\text{oracle}) - IoU(\text{baseline})} \cdot 100$).

| Method | *Person* | *Rider* | *Car* | *Trunk* | *Vegetation* | *Traffic − sign* | *Pole* | *Object* | *Building* | *Fence* | *Bike* | *Ground* | *mIoU*(↑) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline (source only) | 22.77 | 1.78 | 35.91 | 16.86 | 39.84 | 7.08 | 9.73 | 0.18 | 57.03 | 1.64 | 18.17 | 41.99 | 21.08 |
| LiDARNet [20] | 31.39 | **23.98** | 70.78 | 21.43 | 60.68 | **9.59** | 17.48 | **4.97** | 79.53 | 12.57 | **0.78** | 82.41 | 34.63 |
| LiDAR-UDA [3] | **65.59** | 2.19 | 64.12 | 27.49 | **65.40** | 6.44 | **36.57** | 4.19 | 75.21 | **40.31** | 0.00 | 75.06 | **38.55** |
| LiOn-XA (ours) | 31.86 | 13.64 | **81.30** | **34.49** | 61.86 | 6.20 | 32.25 | 0.88 | **86.35** | 25.44 | 0.00 | **87.43** | 38.48 |

TABLE II: Quantitative results on the SemanticKITTI to SemanticPOSS scenario, measured by class-wise IoU and mIoU.



(a) Ground Truth. (b) Baseline. (c) LiOn-XA (2D). (d) LiOn-XA (3D). (e) LiOn-XA (2D + 3D).

Fig. 3: Qualitative results on the SemanticKITTI to nuScenes-Lidarseg adaptation scenario. ■ Vehicle, ■ Driveable surface, ■ Sidewalk, ■ Terrain, ■ Manmade, ■ Vegetation, ■ Ignore label.

From Table I, we observe that LiOn-XA outperforms the baseline by a large margin. The ensemble (2D + 3D) achieves the best results with an increase of 2.2 mIoU compared to CM-CL+PL. In addition, LiOn-XA outperforms xMUDA on the 3D and the combined stream of the SemanticKITTI to nuScenes-Lidarseg scenario. As expected, the combination of both modalities achieves the highest scores, which compensates the lower 2D performance. The results in Table I verify the primary assumption of our approach: 2D and 3D LiDAR representations are complementary in cross-modal learning setting and can learn robust domain-invariant features for UDA when combined with adversarial training.

For the SemanticKITTI to SemanticPOSS scenario, we compare LiOn-XA to the uni-modal approach LiDARNet [20] and the recently proposed self-training approach LiDAR-UDA [3]. Table II summarizes the IoU results for each of the 12 classes in addition to the mIoU. LiOn-XA reaches an overall performance of 38.48 mIoU, which is an improvement over LiDARNet and a performance comparable to LiDAR-UDA with 38.55. The results show that the adaptation capability of LiOn-XA is superior in many classes. We hypothesize that the additional geometric information in LiOn-XA from the 3D stream helps to learn a high-level, domain-invariant feature representation that transfers well between both datasets for classes, such as "Car", "Building", and "Ground". However, the decrease in performance in both our approach and LiDAR-UDA compared to the baseline for the "Traffic-sign" and the "Bike" classes likely indicates a considerable change in the data distribution of these classes. The difference is more severe for the class "Bike" as LiOn-XA seems not to be able to adapt to the large number of bikes in the campus scenes of SemanticPOSS.

### D. Ablation Study

We conduct ablation studies regarding adding the discriminator module and target-like data to LiOn-XA. The results are summarized in Table III. Without the discriminator module (Dis), the overall performance drops from 68.9 to 66.4 mIoU for the nuScenes dataset. In particular, the adversarial training technique achieves consistent improvements concerning the 3D network. The domain gap in this setting is mostly due to environmental changes and the 2D network falls slightly behind when aligning the source and target features. We argue that the combination of 2D source and 3D target or vice versa exchanges feature information from both domains, in which case the 3D network benefits.

| Method | nuScenes: USA → SG | | | SemanticKITTI → nS-Lidarseg | | |
|---|---|---|---|---|---|---|
| | 2D | 3D | 2D+3D | 2D | 3D | 2D+3D |
| LiOn-XA (w/o Dis) | **58.2** | 60.9 | 66.4 | 40.9 | 60.9 | 61.9 |
| LiOn-XA (w/ Dis) | 58.0 | **63.4** | **68.9** | 51.6 | 70.4 | 71.0 |
| LiOn-XA(w/ Dis+Tgl) | – | – | – | **51.9** | **70.7** | **71.3** |

TABLE III: Influence of discriminator (Dis) stream and target-like (Tgl) data. We report the mIoU for each modality.

For the SemanticKITTI to nuScenes-Lidarseg scenario (right-hand side Table III), we include target-like data (Tgl) based on [2] into the learning procedure. It can be observed that a discriminator module alone results in a major performance gain for all three modality options due to enforcing a domain-invariant feature space. Moreover, adding an auxiliary loss function with target-like data in addition to the discriminators further improves the performance of both network streams. Our model benefits from the domain mapping procedure into a representation that approximates the visual appearance of the target dataset.

### E. Qualitative Results

We qualitatively evaluate our approach against the baseline in Figure 3. We compare the ground truth labels in Figure 3a with the baseline approach in Figure 3b, which is trained only on the source domain without a domain adaptation strategy. In addition, we show LiOn-XA as the 2D + 3D predictions in Figure 3e. The 2D model in Figure 3c is able to detect the vehicle in the upper part of the point cloud, whereas the 3D model in Figure 3d is uncertain and primarily predicts vegetation. In this case, the 2D segmentation output is more robust to domain changes, which benefits the predictions

of their ensemble (2D + 3D). For other parts of the point cloud (red ellipsis on the right hand side), the 3D modality is stronger at predicting the sidewalk class. It becomes evident that both representations complement each other and their combination yields the best possible results from both.

## V. Conclusion

We propose LiOn-XA, a new UDA approach for 3D LiDAR point cloud semantic segmentation. LiOn-XA uses LiDAR-only cross-modal learning, where two different LiDAR representations learn from each other to mitigate the negative effects of domain shifts in sensor-to-sensor and cross-city domain adaptation. For LiDAR-only cross-modal learning, we propose combining a cross-modal mimicking task, adversarial training, and target-like data generated from the source domain. Experiments on three unsupervised domain adaptation scenarios show that LiOn-XA can successfully improve the performance over non-adaptive baselines as well as multi-modal state-of-the-art works. In the future, our approach could be applied to different modalities other than LiDAR, such as radar data.

## Acknowledgement

This work has been partially funded by the LOEWE initiative (Hesse, Germany) within the emergenCITY centre and by the Federal Ministry of Education and Research (BMBF) grant 01|S17050.

## References

[1] L. Yi, B. Gong, and T. Funkhouser, "Complete & Label: A Domain Adaptation Approach to Semantic Segmentation of LiDAR Point Clouds," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 15 358–15 368.

[2] F. Langer, A. Milioto, A. Haag, J. Behley, and C. Stachniss, "Domain Transfer for Semantic Segmentation of LiDAR Data using Deep Neural Networks," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 8263–8270.

[3] A. Shaban, J. Lee, S. Jung, X. Meng, and B. Boots, "LiDAR-UDA: Self-ensembling Through Time for Unsupervised LiDAR Domain Adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 784–19 794.

[4] M. Jaritz, T.-H. Vu, R. de Charette, E. Wirbel, and P. Perez, "xMUDA: Cross-Modal Unsupervised Domain Adaptation for 3D Semantic Segmentation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 12 602–12 611.

[5] M. Jaritz, T.-H. Vu, R. de Charette, E. Wirbel, and P. Pérez, "Cross-modal Learning for Domain Adaptation in 3D Semantic Segmentation," in *TPAMI*, 2022.

[6] W. Liu, Z. Luo, Y. Cai, Y. Yu, Y. Ke, J. M. Junior, W. N. Gonçalves, and J. Li, "Adversarial unsupervised domain adaptation for 3D semantic segmentation with multi-modal learning," vol. 176, pp. 211–221, 2021, pII: S0924271621001131.

[7] D. Peng, Y. Lei, W. Li, P. Zhang, and Y. Guo, "Sparse-to-dense Feature Matching: Intra and Inter domain Cross-modal Learning in Domain Adaptation for 3D Semantic Segmentation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2021, pp. 7088–7097.

[8] B. Xing, X. Ying, R. Wang, J. Yang, and T. Chen, "Cross-Modal Contrastive Learning for Domain Adaptation in 3D Semantic Segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 3, 2023, pp. 2974–2982.

[9] M. Mühlhäuser, C. Meurisch, M. Stein, J. Daubert, J. Von Willich, J. Riemann, and L. Wang, "Street lamps as a platform," *Communications of the ACM*, vol. 63, no. 6, pp. 75–83, 2020.

[10] T. Kreutz, M. Mühlhäuser, and A. S. Guinea, "Unsupervised 4D LiDAR Moving Object Segmentation in Stationary Settings With Multivariate Occupancy Time Series," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023, pp. 1644–1653.

[11] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu, "(AF) 2 -S3Net: Attentive Feature Fusion with Adaptive Feature Selection for Sparse Semantic Segmentation Network," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 12 542–12 551.

[12] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang, and S. Han, "Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution," in *Computer Vision – ECCV 2020*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Springer International Publishing, 2020, vol. 12373, pp. 685–702.

[13] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu, "RPVNet: A Deep and Efficient Range-Point-Voxel Fusion Network for LiDAR Point Cloud Segmentation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2021, pp. 16 004–16 013.

[14] B. Besic, N. Gosala, D. Cattaneo, and A. Valada, "Unsupervised Domain Adaptation for LiDAR Panoptic Segmentation," vol. 7, no. 2, pp. 3404–3411, 2022.

[15] M. Rochan, S. Aich, E. R. Corral-Soto, A. Nabatchian, and B. Liu, "Unsupervised Domain Adaptation in LiDAR Semantic Segmentation with Self-Supervision and Gated Adapters," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2649–2655.

[16] L. Kong, N. Quader, and V. E. Liong, "ConDA: Unsupervised Domain Adaptation for LiDAR Segmentation via Regularized Domain Concatenation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9338–9345.

[17] X. Yan, J. Gao, C. Zheng, C. Zheng, R. Zhang, S. Cui, and Z. Li, "2DPASS: 2D Priors Assisted Semantic Segmentation on LiDAR Point Clouds," in *European Conference on Computer Vision*. Springer, 2022, pp. 677–695.

[18] L. Kong, N. Quader, V. E. Liong, and H. Zhang, "ConDA: Unsupervised Domain Adaptation for LiDAR Segmentation via Regularized Domain Concatenation," 2021.

[19] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy Networks: Learning 3D Reconstruction in Function Space," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 4455–4465.

[20] P. Jiang and S. Saripalli, "LiDARNet: A Boundary-Aware Domain Adaptation Model for Point Cloud Semantic Segmentation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 2457–2464.

[21] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A Multimodal Dataset for Autonomous Driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 11 618–11 628.

[22] W. K. Fong, R. Mohan, J. V. Hurtado, L. Zhou, H. Caesar, O. Beijbom, and A. Valada, "Panoptic Nuscenes: A Large-Scale Benchmark for LiDAR Panoptic Segmentation and Tracking," vol. 7, no. 2, pp. 3795–3802, 2022.

[23] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *2019 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 9296–9306.

[24] Y. Pan, B. Gao, J. Mei, S. Geng, C. Li, and H. Zhao, "SemanticPOSS: A Point Cloud Dataset with Large Quantity of Dynamic Instances," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 687–693.

[25] T. Cortinhal, G. Tzelepis, and E. Erdal Aksoy, "SalsaNext: Fast, Uncertainty-Aware Semantic Segmentation of LiDAR Point Clouds," in *Advances in Visual Computing*, ser. Lecture Notes in Computer Science, G. Bebis, Z. Yin, E. Kim, J. Bender, K. Subr, B. C. Kwon, J. Zhao, D. Kalkofen, and G. Baciu, Eds. Springer International Publishing, 2020, vol. 12510, pp. 207–222.

[26] B. Graham, M. Engelcke, and L. van der Maaten, "3D Semantic Segmentation with Submanifold Sparse Convolutional Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2018, pp. 9224–9232.

[27] L.-C. Chen, G. Papandreou, I. Kokkinos, K. P. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 834–848, 2016.