Cross-Modal Learning for Domain Adaptation in 3D Semantic Segmentation

Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Émilie Wirbel, and Patrick Pérez

Abstract—Domain adaptation is an important task to enable learning when labels are scarce. While most works focus only on the image modality, there are many important multi-modal datasets. In order to leverage multi-modality for domain adaptation, we propose cross-modal learning, where we enforce consistency between the predictions of two modalities via mutual mimicking. We constrain our network to make correct predictions on labeled data and consistent predictions across modalities on unlabeled target-domain data. Experiments in unsupervised and semi-supervised domain adaptation settings prove the effectiveness of this novel domain adaptation strategy. Specifically, we evaluate on the task of 3D semantic segmentation from either the 2D image, the 3D point cloud or from both. We leverage recent driving datasets to produce a wide variety of domain adaptation scenarios including changes in scene layout, lighting, sensor setup and weather, as well as the synthetic-to-real setup. Our method significantly improves over previous uni-modal adaptation baselines on all adaption scenarios. Code will be made available upon publication.

Index Terms—Domain adaptation, unsupervised learning, semi-supervised learning, semantic segmentation, 2D/3D

1 INTRODUCTION

Scene understanding is central to many applications and, among other tasks, semantic segmentation from images has been extensively studied. However, for applications that involve interaction with the world, e.g., in robotics, autonomous driving or virtual reality, scenes should be understood in 3D. In this context, 3D semantic segmentation is gaining attention and an increasing number of datasets provide jointly annotated 3D point clouds (PCs) and 2D images. The modalities are complementary since PCs provide geometry while images capture texture and color.

Manual segmentation is tedious in images [1], but even more so in 3D PCs, because the annotator has to inspect the scene from different viewpoints [2]. This results in a high annotation cost. Unfortunately, the question whether sufficient ground truth can be obtained to train a large neural network can make or break a computer vision system. Our goal in this work is to alleviate this problem with transfer learning, in particular domain adaptation (DA), i.e., we leverage multiple modalities (2D/3D) to improve the adaptation of a model to a target domain.

We consider both unsupervised and semi-supervised DA, that is when labels are available in the source domain, but not (or only partially) in the target domain. Most of DA literature investigates the image modality [3], [4], [5], [6], [7], but only a few address the point-cloud modality [8]. Different from these, we perform DA on images and point clouds simultaneously with the aim to explicitly exploit multi-modality for the DA goal.

We use driving data from synchronized cameras and LiDARs, and want to profit from the fact that the domain gaps differ across these sensors. For example, a LiDAR is more robust to lighting changes (e.g., day/night) than a camera. On the other hand, LiDAR sensing density varies with the sensor setup while cameras always output dense images. Our work takes advantage of the cross-modal discrepancies while preserving the best performance of each sensor thanks to the dual-head architecture – thus avoiding that the limitations of one modality negatively affect the other modality’s performance.

We propose a cross-modal loss which enforces consistency between multi-modal predictions, as depicted in Fig. 1. Our specifically designed dual-head architecture enables robust training by decoupling the supervised main segmentation loss from the unsupervised cross-modal loss.

We demonstrate that our cross-modal framework proposal can be either applied in the unsupervised setting (coined xMUDA), or semi-supervised setting (coined xMoSSDA).

This paper is an extension of our work [9] which covered only UDA evaluated on three scenarios. Besides the significant expansion of the experimental evaluation (Section 4) – including the addition of two new DA scenarios (see Fig. 5), the evaluation on the newly released nuScenes-Lidar-seg [10], and the inclusion of new baselines –, we also add a completely new use case of semi-supervised DA (SSDA) in Sections 3.3 and 4.4. The original code base of [9] will be extended with new experiments and the SSDA set up.

In summary our contributions are:

- We introduce new domain adaptation scenarios (4 unsupervised and 4 semi-supervised), for the task of
We propose a new DA approach with an unsupervised cross-modal learning objective; it uncouples the cross-modal loss from the main segmentation objective; and it evaluates xMUDA and xMoSSDA, our unsupervised and semi-supervised DA scenarios respectively, and demonstrates their superior performance.

2 RELATED WORK

2.1 Unsupervised Domain Adaptation

The past few years have seen an increasing interest in unsupervised domain adaptation (UDA) for complex perception tasks like object detection and semantic segmentation. Under the hood of such methods lies the same spirit of learning domain-invariant representations, i.e., features coming from different domains should introduce insignificant discrepancies. Some works promote adversarial training to minimize the source-target distribution shift, either on pixel- [3], feature- [4] or output-space [5], [6]. Instead of adversarial learning, Fourier transform can also be used to stylize the source images as target [14]. Revisited from semi-supervised learning [11], self-training with pseudo-labels has also recently proved effective for UDA [7], [15], [16].

Recent works start addressing UDA in the 3D world, i.e., for point clouds. LiDAR DA works are surveyed in [17]. PointDAN [18] proposes to jointly align local and global features used for classification. Achituve et al. [19] improve UDA performance using self-supervised learning. Wu et al. [8] adopt activation correlation alignment [20] for UDA in 3D segmentation from LiDAR point clouds. Langer et al. [21] use resampling to stylize a 64 as 32-layer LiDAR, thereby aligning source and target in input point-cloud space. Yi et al. [22] also address gaps between LiDAR sampling patterns by chaining a LiDAR-specific completion network with a LiDAR-agnostic segmentation network. In this work, we also address domain adaptation, but from a different angle, i.e., by aligning RGB and LiDAR in output space.

To the best of our knowledge, there are no previous UDA works in 2D/3D semantic segmentation for multi-modal scenarios. Only some consider the extra modality, e.g., depth, solely available at training time on the source domain and leverage such privileged information to boost adaptation performance [23], [24]. Otherwise, we here assume all modalities are available at train and test time on both source and target domains.

2.2 Semi-Supervised Domain Adaption

While UDA has become an active research topic, semi-supervised domain adaptation (SSDA) has so far been little-studied despite being highly relevant in practical applications. In SSDA, we would like to transfer knowledge from a source domain with labeled data to a target domain with partially labeled data.

Early approaches based on SVM [25] have addressed SSDA in image classification and object detection [26], [27], [28]; few has been done for deep networks. Recently, Saito et al. [29] propose an adversarial SSDA learning scheme to optimize a few-shot deep classification model with minimax entropy. Wang et al. [30] extend UDA techniques in 2D semantic segmentation to the SSDA setting by additionally aligning feature prototypes of labeled source and target samples. Our work is the first to address SSDA in point cloud segmentation.

2.3 Cross-Modality Learning

In our context, we define cross-modality learning as knowledge transfer between modalities. This is different from multi-modal fusion where a single model is trained supervisely to combine complementary inputs, such as RGB-D [31], [32] or LiDAR and RGB [33], [34], [35].

Castrejon et al. [36] address cross-modal scene retrieval by learning a joint high-level feature representation that is agnostic to the input modality (real image, clip art, text, etc.) through enforcement of similar statistics across modalities. Gupta et al. [37] adapt the more direct feature alignment technique of distillation [38] in a cross-modal setup.

Self-supervised learning produces useful representations in the absence of labels, e.g., by forcing networks with different input modalities to predict a similar output. Sayed et al. [39] minimize the cosine distance between RGB and optical flow features. Alwassel et al. [40] use clustering to generate pseudo labels and mutually train an audio and video network. Munro et al. [41] use self-supervision with temporal consistency between RGB and flow.

Similar to us, Gong et al. [42] address UDA for segmentation with RGB and LiDAR, but focus on fusing labels from multiple partial source datasets. Instead, we use a single source dataset and explore the tasks of UDA and SSDA.

2.4 Point Cloud Segmentation

While images are dense tensors, 3D point clouds can be represented in multiple ways, which leads to competing network families evolving in parallel.

Voxels are similar to pixels, but very memory intense in their dense representation as most of them are usually empty.
Some 3D CNNs [43], [44] rely on OctTree [45] to reduce memory usage but without addressing the problem of manifold dilation. Graham et al. [13] and similar implementation [46] address the latter by using hash tables to convolve only on active voxels. This allows for very high resolution with typically only one point per voxel. Aside from the cubic ones, cylindrical voxels are also employed [47], [48]. Finally, sparse point-voxel convolutions [49] can benefit from lightweight support of the high resolution point-based branch.

Point-based networks perform computation in continuous 3D space and can thus directly accept point clouds as input. PointNet++ [50] uses point-wise convolution, max-pooling to compute global features and local neighborhood aggregation for hierarchical learning akin to CNNs. Many improvements have been proposed in this direction, such as continuous convolutions [51], deformable kernels [52] or lightweight alternatives [53].

In this work, we select SparseConvNet [13], which is top performing on ScanNet [54], as our 3D network.

We present the network architecture in Section 3.1, our framework for cross-modal unsupervised domain adaptation, coined ‘xMUDA’, in Section 3.2 and its semi-supervised version, analogously called ‘xMoSSDA’, in Section 3.3.

### 3.1 Architecture

Our architecture predicts point-wise segmentation labels. It consists of two independent streams which respectively take a 2D image and a 3D point cloud as inputs, and output features of size \((N, F_{2D})\) and \((N, F_{3D})\) respectively, where \(N\) is the number of 3D points within the camera field of view.

In a naive approach, each modality has a single segmentation head (Fig. 3a) and the cross-modal optimization objective aligns the outputs of both modalities. Unfortunately, this setup is not robust as the mimicking objective competes
directly with the main segmentation objective. The risk is that a negative transfer from the weak modality might degrade the performance of the strong one. This is why, in practice, one needs to down-weight the mimicry loss w.r.t. the segmentation loss to boost performance. However, this is a serious limitation, because down-weighting the mimicry loss also decreases its adaptation effect.

In order to address this problem, we propose to disentangle the mimicry from the main segmentation objective. Therefore, we propose a dual-head architecture as depicted in Figs. 2 and 3a. In this setup, the 2D and 3D streams both have two segmentation heads: one main head for the best possible prediction, and one mimicry head to estimate the other modality’s output. The outputs of the four segmentation heads (see Fig. 2) are of size \((N, C)\), with \(C\) the number of classes, such that we obtain a vector of class probabilities for each 3D point. The two main heads produce the best possible segmentation predictions, \(P_{2D}\) and \(P_{3D}\) respectively for each branch. The two mimicry heads estimate the other modality’s output: 2D estimates 3D (\(P_{2D-3D}\)) and 3D estimates 2D (\(P_{3D-2D}\)).

In the following, we introduce how we use the described architecture for cross-modal learning in unsupervised (Section 3.2) and semi-supervised (Section 3.3) domain adaptation, respectively.

### 3.2 Unsupervised Domain Adaptation (xMUDA)

We propose xMUDA, cross-modal unsupervised domain adaptation, which considers a source-domain dataset \(\mathcal{S}\), where each sample consists of a 2D image \(\mathbf{x}^{2D}\) \(\in \mathbb{R}^{H \times W \times 3}\), a 3D point cloud \(\mathbf{x}^{3D} \in \mathbb{R}^{N \times 3}\) and 3D segmentation labels \(\mathbf{y}^{3D} \in [1, C]^N\) with \(C\) classes, as well as a target-domain dataset \(\mathcal{T}\), lacking annotations, where each sample only consists of an image \(\mathbf{x}^{2D}\) and a point cloud \(\mathbf{x}^{3D}\).

In the following, we define the usual supervised learning setup, our cross-modal loss \(\mathcal{L}_{XM}\), and an additional variant ‘xMUDA\(_{PL}\)’ that further uses pseudo-labels to boost performance. An overview of the learning setup is in Fig. 4a. The difference between our cross-modal learning and existing uni-modal UDA techniques, such as as Pseudo-labels [11], MinEnt [5] or Deep logCORAL [20] is visualized in Fig. 4b.

#### 3.2.1 Supervised Learning

The main goal of 3D segmentation is learned through cross-entropy in a classical supervised fashion on the source-domain data. Denoting \(P_x \in [0, 1]^{N \times C}\) the soft-classification map associated by the segmentation model to the \(N\) 3D points of interest, for a given input \(x\), the segmentation loss \(\mathcal{L}_{seg}\) of each network stream (2D and 3D) for a given training sample in \(S\) reads:

\[
\mathcal{L}_{seg}(x, y^{ID}) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} y^{(n,c)} \log P_x^{(n,c)},
\]

where \(x\) is either \(x^{2D}\) or \(x^{3D}\) and \(y^{ID}\) equals \(y^{3D}\). We denote tensor entries’ indices as superscript.

#### 3.2.2 Cross-Modal Learning

The goal of unsupervised learning across modalities is twofold. Firstly, we want to transfer knowledge from one modality to the other on the target-domain dataset; for example, if one modality is more sensitive than the other to the domain shift, then the robust modality should teach the sensitive one the correct class in the target domain where no labels are available. Secondly, we want to design an auxiliary objective on source and target domains, where the task is to estimate the other modality’s prediction. By mimicking not only the class with maximum probability, but the whole distribution like in teacher-student distillation [38], more information is exchanged, leading to softer labels.

We choose the KL divergence for the cross-modal loss \(\mathcal{L}_{XM}\) and define it as follows:

\[
\mathcal{L}_{XM}(x) = D_{KL}(P_x^{(n,c)} || Q_x^{(n,c)})
\]

\[
= -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} P_x^{(n,c)} \log \frac{P_x^{(n,c)}}{Q_x^{(n,c)}},
\]
with \((P, Q) \in \{(P_{2D}, P_{3D-2D}), (P_{3D}, P_{2D-3D})\}\) where \(P\) is the target distribution from the main prediction which is to be estimated by the mimicking prediction \(Q\). This loss is applied on the source and the target domain as it does not require ground-truth labels and is the key to our proposed domain adaptation framework. In the source domain, \(\mathcal{L}_{2D}\) can be seen as an auxiliary mimicry loss in addition to the main segmentation loss \(\mathcal{L}_{seg}\).

The final objective for each network stream (2D and 3D) is the combination of the segmentation loss \(\mathcal{L}_{seg}\) on source-domain data and the cross-modal loss \(\mathcal{L}_{3D}\) on both domains:

\[
\min_{\theta} \left[ \frac{1}{|S|} \sum_{x_s \in S} \left( \mathcal{L}_{seg}(x_s, y^{3D}_s) + \lambda_s \mathcal{L}_{3D}(x_s) \right) + \frac{1}{|T|} \sum_{x_t \in T} \lambda_t \mathcal{L}_{3D}(x_t) \right], \tag{4}
\]

where \(\lambda_s, \lambda_t\) are hyperparameters to weight \(\mathcal{L}_{2D}\) on source and target domain respectively and \(\theta\) are the network weights of either the 2D or the 3D stream.

There are parallels between our approach and Deep Mutual Learning \[55\] in training two networks in collaboration and using the KL divergence as mimicry loss. However, unlike this work, our cross-modal learning establishes consistency across modalities (2D/3D) without supervision.

### 3.2.3 Self-Training With Pseudo-Labels

Cross-modal learning is complementary to pseudo-labeling [11] used originally in semi-supervised learning and recently in UDA \[7, 15\]. To benefit from both, once having optimized a model with Eq. (4), we extract pseudo-labels offline, selecting highly-confident labels based on the predicted class probability. Then, we train again from scratch using the produced pseudo-labels for an additional segmentation loss on the target-domain training set. The optimization problem writes:

\[
\min_{\theta} \left[ \frac{1}{|S|} \sum_{x_s \in S} \left( \mathcal{L}_{seg}(x_s, y^{3D}_s) + \lambda_s \mathcal{L}_{3D}(x_s) \right) + \frac{1}{|T|} \sum_{x_t \in T} \left( \lambda_{PL} \mathcal{L}_{seg}(x_t, \hat{y}^{3D}_t) + \lambda_t \mathcal{L}_{3D}(x_t) \right) \right], \tag{5}
\]

where \(\lambda_{PL}\) weights the pseudo-label segmentation loss and \(\hat{y}^{3D}\) are the pseudo-labels. For clarity, we will refer to the xMUDA variant that uses additional self-training with pseudo-labels as xMUDA\_PL.

### 3.3 Semi-Supervised Domain Adaptation (xMoSSDA)

Cross-modal learning can also be used in semi-supervised domain adaptation, thus benefitting from a small portion of labeled data in the target domain.

Formally, we consider in xMoSSDA a labeled source-domain dataset \(S\) where each sample contains an image \(x^{2D}_s\), a point cloud \(x^{3D}_s\) and labels \(y^{3D}_s\). Different from unsupervised learning, the target-domain dataset consists of a usually small labeled part \(T_t\) where each sample holds an image \(x^{2D}_{3T}\), a point cloud \(x^{3D}_{3T}\) and labels \(y^{3D}_{3T}\), as well as an, often larger, unlabeled part \(T_u\) where each sample consists only of an image \(x^{2D}_{3u}\) and a point cloud \(x^{3D}_{3u}\).

#### 3.3.1 Supervised Learning

Unlike xMUDA, we do not only apply the segmentation loss \(\mathcal{L}_{seg}(x, y^{3D})\) of Eq. (1) on the source-domain dataset \(S\), but also on the labeled target-domain dataset \(T_t\). The segmentation loss in Eq. (1) thus applies both to samples \((x, y^{3D}) \in \{(x^{2D}_s, y^{3D}_{3s}), (x^{3D}_s, y^{3D}_{3s})\}\) in \(S\) and to samples \((x, y^{3D}) \in \{(x^{2D}_{3t}, y^{3D}_{3T}), (x^{3D}_{3T}, y^{3D}_{3T})\}\) in \(T_t\). Note that, in practice, we train on source and target domains at the same time by concatenating examples from both in a batch.

#### 3.3.2 Cross-Modal Learning

We apply the unsupervised cross-modal loss \(\mathcal{L}_{3D}\) of Eq. (2) on all datasets, i.e., labeled source-domain \(S\), labeled target-domain \(T_t\), and unlabeled target-domain dataset \(T_u\). The latter, \(T_u\), is a typically large portion of unlabeled data compared to a usually much smaller labeled portion \(T_t\). Subsequently, it is beneficial to also exploit \(T_u\) with an unsupervised loss, such as cross-modal learning. The complete objective is a combination of supervised segmentation loss \(\mathcal{L}_{seg}\) where labels are available (i.e., in sets \(S\) and \(T_t\)) and unsupervised cross-modal loss \(\mathcal{L}_{3D}\) everywhere (i.e., on \(S, T_t\) and \(T_u\)) which enforces consistency between the 2D and 3D predictions. It writes:

\[
\min_{\theta} \left[ \frac{1}{|S|} \sum_{x_s \in S} \left( \mathcal{L}_{seg}(x_s, y^{3D}_s) + \lambda_s \mathcal{L}_{3D}(x_s) \right) + \frac{1}{|T_t|} \sum_{x_t \in T_t} \left( \mathcal{L}_{seg}(x_t, y^{3D}_t) + \lambda_t \mathcal{L}_{3D}(x_t) \right) \right], \tag{6}
\]

where \(\lambda_s, \lambda_t\) and \(\lambda_{PL}\) are the weighting hyperparameters for \(\mathcal{L}_{3D}\). In practice we choose \(\lambda_s = \lambda_{PL}\) for simplicity.

#### 3.3.3 Self-Training With Pseudo-Labels

As in the unsupervised setting, we extend semi-supervised cross-modal learning to also benefit from pseudo-labels. After having trained a model with Eq. (6), we use the model to generate predictions on the unlabeled target-domain dataset \(T_u\) and extract highly-confident pseudo-labels which are used to train again from scratch with the following objective:

\[
\min_{\theta} \left[ \frac{1}{|S|} \sum_{x_s \in S} \left( \mathcal{L}_{seg}(x_s, y^{3D}_s) + \lambda_s \mathcal{L}_{3D}(x_s) \right) + \frac{1}{|T_t|} \sum_{x_t \in T_t} \left( \mathcal{L}_{seg}(x_t, y^{3D}_t) + \lambda_t \mathcal{L}_{3D}(x_t) \right) + \frac{1}{|T_u|} \sum_{x_{3u} \in T_u} \left( \lambda_{PL} \mathcal{L}_{seg}(x_{3u}, \hat{y}^{3D}_{3u}) + \lambda_{PL} \mathcal{L}_{3D}(x_{3u}) \right) \right], \tag{7}
\]

where \(\lambda_{PL}\) is weighting the pseudo-label segmentation loss and \(\hat{y}^{3D}\) are pseudo-labels. We call this variant xMoSSDA\_PL.
4 EXPERIMENTS

For evaluation, we identified five domain adaptation (DA) scenarios relevant to autonomous driving, shown in Fig. 5, and evaluated our proposals against recent baselines.

Hereafter, we first describe the datasets (Section 4.1), the implementation backbone and training details (Section 4.2), and then evaluate xMUDA (Section 4.3) and xMoSSDA (Section 4.4). Finally, we extend our cross-modal framework to fusion (Section 4.5), demonstrating its global benefit.

4.1 Datasets

To compose our domain adaptation scenarios displayed in Fig. 5, we leveraged public datasets nuScenes-Lidarseg [10], VirtualKITTI [56], SemanticKITTI [2], A2D2 [57] and Waymo Open Dataset (Waymo OD) [58]. The split details are in Table 1. Our scenarios cover typical DA challenges like change in scene layout, between right and left-hand-side driving in the nuScenes-Lidarseg: USA/Singapore scenario, lighting changes, between day and night in nuScenes-Lidarseg: Day/Night, synthetic-to-real data, between simulated depth and RGB to real LiDAR and camera in VirtualKITTI/SemanticKITTI, different sensor setups and characteristics like resolution/FoV in A2D2/SemanticKITTI and weather changes between sunny San Francisco, Phoenix, Mountain View and rainy Kirkland in Waymo OD: SF,PHX,MTV/KRK.

In all datasets, the LiDAR and the camera are synchronized and calibrated, allowing 2D/3D projections. For consistency across datasets, we only use the front camera’s images, even when multiple cameras are available.

Waymo OD does not provide point-wise 3D segmentation labels so we leverage 3D object bounding-box labels. Points lying inside a box are labeled as that class and points outside of all boxes are labeled as background.

To compensate for source/target classes mismatch (e.g., VirtualKITTI/SemanticKITTI) or accommodate for classes scarcity, we apply a custom class mapping detailed in supplementary. Note that VirtualKITTI provides depth maps so we simulate LiDAR scanning via uniform point sampling.

All training data and splits can be reproduced with our code, and more details are in the supplementary.

4.2 Implementation Details

In the following, we briefly introduce our implementation. Please refer to our code for further details.

2D Network. We use a modified version of U-Net [12] with ResNet34 [59] encoder and a decoder with transposed convolutions and skip connections. To lift the 2D features to 3D, we subsample the output feature map of size \( \left( H, W, F^{2D} \right) \) at the pixel locations where the \( N \) 3D points project. Hence, the 2D network takes an image \( x^{2D} \) as input and outputs features of size \( \left( N, F^{2D} \right) \).

3D Network. We use the official SparseConvNet [13] implementation and a U-Net architecture with 6 times downsampling. The voxel size is set to 5cm which is small enough to only have one 3D point per voxel. Thus, the 3D network takes a point cloud \( x^{3D} \) as input and outputs features of size \( \left( N, F^{3D} \right) \).

Training. We employ standard 2D/3D data augmentation and log-smoothed class weights to address class-imbalance. In PyTorch, to compute the KL divergence for the cross-modal loss, we detach the target variable to only backpropagate in either the 2D or the 3D network. We train with a batch size of 8, Adam optimizer with \( \beta_1 = 0.9, \beta_2 = 0.999 \), and train 30k iterations for the scenario with the small VirtualKITTI dataset and 100k iterations for all other scenarios.

### TABLE 1

| Scenario                        | Source S | Target T |
|--------------------------------|----------|----------|
| nuScenes-Lidarseg: USA/Singapore| 15,695   | 9,665    |
| nuScenes-Lidarseg: Day/Night    | 24,745   | 2,770/2,929 |
| Virt.KITTI/Scm.KITTI            | 2,126    | 16,029   |
| A2D2/Sem.KITTI                  | 27,695   | 16,029   |
| Waymo OD: SF,PHX,MTV/KRK        | 156,081  |

While there is a single target-domain training set in UDA (T), there are two in SSDA: a labeled target-domain training set \( T_1 \) and a (much larger) unlabeled set \( T_u \).
At each iteration we compute and accumulate gradients on the source and target batch, jointly training the 2D and 3D stream. To fit the training into a single GPU with 11GB of memory, we resize the images and additionally crop them in VirtualKITTI and SemanticKITTI.

For the pseudo-label variants, xMUDA_PL and xMoSSDA_PL, we generate the pseudo-labels offline as in [7] with trained models xMUDA and xMoSSDA, respectively. Then, we retrain from scratch, additionally using the pseudo-labels, optimizing Eqs. (5) and (7), respectively. Importantly, we only use the last checkpoint to generate the pseudo-labels – as opposed to using the best weights which would provide a supervised signal.

### 4.3 xMUDA

We evaluate xMUDA on four unsupervised domain adaptation scenarios and compare against uni-modal UDA methods: Deep logCORAL [20], entropy minimization (MinEnt) [5], pseudo-labeling (PL) [7] and Fourier domain adaptation (FDA) [14]. For [7] the image-2-image translation part was excluded due to its instability, high training complexity and incompatibility with LiDAR data. Regarding the three other uni-modal techniques, we adapt the published implementations to our settings. For all, we searched for the best hyperparameters. For the 2D-only baseline FDA [14], we implement the full MTB method, i.e., with entropy, ensembling of three models to generate pseudo-labels and re-training. We found no 3D-only UDA baseline directly applicable to our scenarios. Instead, we compare against LiDAR transfer [21] – largely outperformed on their own scenario – in the supplementary.

We report mean Intersection over Union (mIoU) of the target test set for 3D segmentation in Table 2. We evaluate on the test set using the checkpoint that achieved the best score on the validation set. In addition to the scores of the 2D and 3D model, we show the ensembling result (2D+3D) which is obtained by taking the mean of the predicted 2D and 3D probabilities after softmax. The uni-modal UDA baselines [5], [7], [20] are applied separately on each modality, and FDA [14] is a 2D-only UDA baseline.

Furthermore, we provide the results of a lower bound, `Baseline (src only)`, which is not domain adaptation as it is only trained on the source-domain dataset and an upper bound, `Oracle`, trained only on target with labels\(^1\). We also indicate the `Domain gap (O-B)` computed as the difference between Oracle and Baseline. It shows that the intra-dataset domain gaps (nuScenes-Lidarseg: USA/Singapore, Day/Night), in [1.1, 17.0], are much smaller than the inter-dataset domain gaps (A2D2/SemanticKITTI, VirtualKITTI/SemanticKITTI), in [25.1, 39.5]. It suggests that a change in sensor setup (A2D2/SemanticKITTI) is actually a very hard domain adaptation problem, similar to the synthetic-to-real case (VirtualKITTI/SemanticKITTI). Importantly, note that the scores are not comparable between A2D2/SemanticKITTI and VirtualKITTI/SemanticKITTI, as they use a different number of classes, 10 and 6 respectively.

xMUDA –using the cross-modal loss but not PL– brings a significant adaptation effect on all four UDA scenarios compared to `Baseline` and most often outperforms all uni-modal UDA baselines. xMUDA_PL achieves the best score everywhere with the only exception of Day/Night 2D+3D where xMUDA is better. Further, cross-modal learning and self-training with pseudo-labels (PL) are complementary as their combination in xMUDA_PL typically yields a higher score, up to +4.7, than each separate technique. The 2D+3D oracle scores indicate that overall LiDAR (3D) is always the strongest modality, which resonates with the choice of 3D segmentation task. However, xMUDA consistently improves both modalities (2D and 3D) i.e., even the strong modality can learn from the weaker one. A notable example for 3D is at Night where xMUDA (69.2) outperforms `Baseline` (68.8), despite a narrow Domain gap (1.1) related to the active LiDAR sensing capability. The dual-head architecture might be key here: each modality can improve its main

\(^1\) Except for the Day/Night oracle, where we use 50%/50% source/target batches to prevent overfitting due to the small target size.

| Method                  | nuSc-Lidarseg: USA/Singap. | nuSc-Lidarseg: Day/Night | Virt.KITTI/Sem.KITTI | A2D2/Sem.KITTI |
|-------------------------|-----------------------------|--------------------------|----------------------|----------------|
|                         | 2D  | 3D  | 2D+3D | 2D  | 3D  | 2D+3D | 2D  | 3D  | 2D+3D |
| Baseline (src only)     | 58.4| 62.8| 68.2  | 47.8| 68.8| 63.3  | 26.8| 42.0| 42.2  |
| Deep logCORAL [20]      | 64.4| 63.2| 69.4  | 47.7| 68.7| 63.7  | 41.4| 36.8| 47.0  |
| MinEnt [5]              | 57.6| 61.5| 66.0  | 47.1| 68.8| 63.6  | 39.2| 43.3| 47.1  |
| PL [7]                  | 62.0| 64.8| 70.4  | 47.0| 69.6| 63.0  | 21.5| 44.3| 35.6  |
| FDA [14]                | 60.8| -   | -     | 48.4| -   | -     | 32.8| -   | -     |
| xMUDA                   | 64.4| 63.2| 69.4  | 55.5| 69.2| 67.4  | 42.1| 46.7| 48.2  |
| xMUDA_PL                | 67.0| 65.4| 71.2  | 57.6| 69.6| 64.1  | 45.8| 51.4| 52.0  |
| Oracle                  | 75.4| 76.0| 79.6  | 61.5| 69.8| 69.2  | 66.3| 78.4| 80.1  |
| Domain gap (O-B)        | 17.0| 13.3| 11.5  | 13.6| 11.1| 5.9   | 39.5| 36.4| 37.9  |

* The 2D network is trained with batch size 6 instead of 8 to fit into GPU memory. We report the mIoU result (with best and 2nd best) on the target set for each network stream (2D and 3D) as well as the ensemble result taking the mean of the 2D and 3D probabilities (2D+3D). We provide the lower bound ‘Baseline (src only)’ which is trained on the source set S, but not on the target set T, as well as the upper bound ‘Oracle’ which is trained supervely on the target set T using labels. We further indicate the ‘Domain gap’ which is the difference between the Oracle and Baseline score. ‘Deep logCORAL’ [20], ‘MinEnt’ [5] and ‘PL’ [7] are 2D/3D uni-modal UDA baselines, whereas ‘FDA’ is 2D-only. The two variants ‘xMUDA’ and ‘xMUDA_PL’ are our methods. We evaluate on four UDA scenarios (see Fig. 5). For the nuScenes-Lidarseg dataset [10] (‘nuSc-Lidarseg’), we generate the splits with different locations (USA/Singapore) and different time (Day/Night). VirtualKITTI [56] (‘Virt.KITTI’) to SemanticKITTI [2] (‘Sem.KITTI’) lies mainly in sensor resolution.
segmentation head independently from the other modality, because the consistency is achieved indirectly through the mimicking heads.

We also observe a regularization effect thanks to xMUDA. For example on VirtualKITTI/SemanticKITTI, the methods ‘Baseline’ and ‘PL’ perform very poorly on the 2D modality (26.8 and 21.5) due to overfitting on the very small VirtualKITTI dataset, while 3D is more stable (42.0 and 44.3). In contrast, xMUDA performs better as 3D can regularize 2D. This regularization even enables the benefit of pseudo-labels, because xMUDA PL achieves an even better score.

Qualitative results are presented in Fig. 6, showing the versatility of xMUDA across all proposed UDA scenarios. Here, the benefit of xMUDA PL over uni-modal PL baseline is evident in inter-dataset scenarios (last two rows), and more subtly on the nuScenes scenarios (first two rows) looking at the vehicle class. We provide additional qualitative results in the supplementary and a video at http://tiny.cc/cross-modal-learning.

We also successfully experiment our method in some opposite adaptation directions, with details in supplementary.

4.4 xMoSSDA

In this section, we evaluate xMoSSDA on domain adaptation scenarios nuSc-LidarSeg: USA/Singap., A2D2/SemanticKITTI and Waymo OD. To create practically relevant SSDA conditions, we make sure that the unlabeled target dataset \( T_u \) is much larger than the labeled target dataset \( T_t \). Hence, splits differ from UDA (see Table 1).

We compare xMoSSDA against eight baselines. Three baselines are purely supervised, either trained on source only \((S)\), labeled target only \((T_t)\) or on source and labeled target \((S + T_t)\). Additionally we report two UDA baselines, xMUDA and xMUDA PL, which use source and unlabeled target \((S + T_u)\). Last, we report three SSDA baselines (trained on \(S + T_t + T_u\)) adapted from uni-modal UDA baselines [5], [7], [20] as follows: we train similarly to the supervised baseline on \(S + T_t\) with 50%/50% batches, but add the respective domain adaptation loss on \(T_u\). Our semi-supervised proposals, xMoSSDA and MoSSDA PL, are also trained in this manner.

To achieve \(|T_u| \gg |T_t|\), we include unannotated data for Waymo OD and SemanticKITTI (we use hidden test set) into \(T_u\). Hence, it is impossible to train an Oracle like in Table 2. Instead, we answer the question: “How much can we improve over the supervised baseline by additionally training on \(T_u\)?” We coin it the ‘unsupervised advantage’ computed as the difference between xMoSSDA PL \((S + T_u + T_t)\) and supervised baseline \((S + T_t)\). Note that we exclude nuSc-LidarSeg: Day/Night and Virt.KITTI/Sem.KITTI due to small target/source datasets respectively, but still evaluate Virt.KITTI/Sem.KITTI in the supplementary.

We report the mIoU for 3D segmentation in Table 3. Note that the latter results cannot be compared to Table 2 since splits differ. We observe in Table 3, similar to Table 2, that the domain gap is much larger in the inter-dataset adaptation.

2. The latter is trained with 50%/50% examples from \(S\) and \(T_t\), i.e., a training batch of size 8 has 4 random samples from \(S\) and 4 from \(T_t\).
A2D2/SemanticKITTI (max. 29.5) than in intra-dataset adaptation on nuSc-Lidarseg: USA/Singap. (max. 13.5) and Waymo OD, (max. 4.5). As expected, xMUDA and xMUDA\textsubscript{PL} improve over the baseline (S), but are (with the exception of Waymo OD) worse than the baseline (S + T\textsubscript{u}). We also observe that xMoSSDA improves 2D and 3D more (max. 2.0) than ensemble result 2D+3D (max. 0.4), w.r.t. to Baseline (S + T\textsubscript{u}). We ascribe this behavior to the ensembling (2D+3D) performing best when 2D and 3D predictions differ, though our cross-modal loss aligns 2D/3D predictions. In xMoSSDA\textsubscript{PL}, the separate 2D/3D pseudo-labels act as counterweight to this alignment, resulting in comparable 2D, 3D and 2D+3D improvement.

Finally, xMoSSDA\textsubscript{PL} outperforms all baselines by up to +2.2 in 8 out of 9 cases. Results in Fig. 7 show better segmentation of thin structures (bush, bike, pedestrian).

### 4.5 Extension to Fusion
So far, we used an architecture with independent 2D/3D streams. However, can xMUDA also be applied in a fusion setup where both modalities make a joint prediction?

A common fusion architecture is late fusion where the features from different sources are concatenated (see Fig. 8a). However, when merging the main 2D/3D branches into a unique fused head, we can no longer apply cross-modal learning (as in Fig. 3a). To address this problem, we propose...
that are used to mimic the fusion output and to the fusion $P_{\text{fuse}}$. (b) In xMUDA Fusion, we add two uni-modal outputs $P_{2D\rightarrow\text{fuse}}$ and $P_{3D\rightarrow\text{fuse}}$ that are used to mimic the fusion output $P_{\text{fuse}}$.

**Table 4** Comparison of the Fusion Methods

| Method                  | Arch. | nuSc-Lidarseg: USA/Singap. | A2D2/ Sem.KITTI |
|-------------------------|-------|---------------------------|-----------------|
| Baseline (src only)     | Vanilla | 66.5 | 34.2 |
| Deep logCORAL [20]     | Vanilla | 64.0 | 36.2 |
| MinEnt [5]             | Vanilla | 65.4 | 39.8 |
| PL [7]                 | Vanilla | 70.1 | 36.6 |
| xMUDA Fusion           | xMUDA | 69.3 | 42.6 |
| xMUDA$_{A_2}$ Fusion  | xMUDA | 70.7 | 42.2 |
| Oracle                 | xMUDA | 80.6 | 65.7 |

Performance in mIoU for the two UDA scenarios: nuScenes-Lidarseg [10]: USA/Singapore and A2D2[57]/SemanticKITTI[2]. We adapt the supervised baseline ‘Baseline (src only)’ and the UDA baselines (‘Deep logCORAL’, ‘MinEnt’, ‘PL’) to the vanilla fusion architecture depicted in Fig. 8a. We propose ‘xMUDA Fusion’ which uses the architecture of Fig. 8b.

‘xMUDA Fusion’ where we add an additional segmentation head to both 2D and 3D network streams prior to the fusion layer with the purpose of mimicking the central fusion head (see Fig. 8b). Note that this idea could also be applied on top of other fusion architectures.

In Table 4 we show results for different fusion approaches where we specify which architecture was used (Vanilla late fusion from Fig. 8a or xMUDA Fusion from Fig. 8b). We observe that the xMUDA fusion architecture leads to better results than the UDA baselines with the Vanilla architecture. This demonstrates how cross-modal learning can be applied effectively in fusion setups.

5 ABLATION STUDIES

5.1 Single versus Dual Segmentation Head

Here we justify our dual head over the simpler single-head architecture, shown in Fig. 3. In the single-head architecture (Fig. 3a), the cross-modal loss $L_{\text{CM}}$ is directly applied between the 2D and 3D main heads. This enforces consistency by aligning the two outputs in addition to the supervised segmentation loss $L_{\text{seg}}$. Thus, the heads must satisfy the two objectives –segmentation and consistency– at the same time. To showcase the disadvantage of this architecture, we train xMUDA (as in Eq. (4)) and vary the weight $\lambda_s$ for the cross-modal loss $L_{\text{CM}}$ on target, which is the main driver for UDA. The results in Fig. 9 for the single-head architecture (Fig. 3a) show that increasing $\lambda_s$ from 0.001 to 0.01 slightly improves the mIoU, but that increasing $\lambda_s$ further to 0.1 and 1.0, has a hugely negative effect. In the extreme case $\lambda_s = 1.0$, 2D and 3D always predict the same class, thus only satisfying the consistency, but not the segmentation objective.

The dual-head architecture (Fig. 3b) addresses this problem by introducing a secondary mimicking head which purpose is to mimic the main head of the other modality during the training and can be discarded afterwards. This effectively disentangles the mimicking objective which is applied to the mimicking head from the segmentation objective which is applied to the main head. Fig. 9 shows that increasing $\lambda_s$ to 0.1 for dual-head produces the best results overall –better than any value for $\lambda_s$ for single head– and that the results are robust even at $\lambda_s = 1.0$.

5.2 Cross-Modal Learning on Source

In Eq. (4), the cross-modal loss $L_{\text{CM}}$ is applied on source and target domains, although we already have the supervised segmentation loss $L_{\text{seg}}$ on source domain. We observe a gain of 1.6 mIoU on 2D and 1.5 on 3D when adding $L_{\text{CM}}$ on source domain as opposed to applying it on target domain only. This shows that it is important to train the mimicking head on source-domain data, stabilizing predictions, which can be exploited during adaptation on target-domain inputs.

5.3 Cross-Modal Supervised Learning

To evaluate the possible benefits of cross-modal learning for purely supervised settings, we conducted experiments with and without adding the cross-modal loss $L_{\text{CM}}$ on two different target-domain datasets: nuScenes-Lidarseg [10] and Waymo OD [58]. The results are shown in Table 5 and show...
a performance gain when adding $\mathcal{L}_{XM}$. We hypothesize that the extra cross-modal objective can be beneficial, similar to multi-task learning. On the Waymo OD dataset, we observe a strong improvement on 2D. We observe in the training curve (validation) that cross-modal learning reduces overfitting in 2D. We hypothesize that 3D, which suffers less from overfitting, can have a regularizing effect on 2D.

6 Conclusion

In this work, we proposed cross-modal learning for domain adaptation in unsupervised (xMUDA) and semi-supervised (xMoSSDA) settings. To this end, we designed a two-stream, dual-head architecture and applied a cross-modal loss to the image and point-cloud modalities in the task of 3D semantic segmentation. The cross-modal loss consists of KL divergence applied between the predictions of the two modalities and thereby enforces consistency.

Experiments on four unsupervised and four semi-supervised domain adaptation scenarios show that cross-modal learning outperforms uni-modal adaptation baselines and is complementary to learning with pseudo-labels.

We think that cross-modal learning could generalize to many tasks that involve multi-modal input data and is not constrained to DA or to image and point-cloud modalities.

References

[1] M. Cordts et al., “The cityscapes dataset for semantic urban scene understanding,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2016, pp. 3213–3223.
[2] J. Behley et al., “SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 9297–9307.
[3] J. Hoffman et al., “Cycada: Cycle-consistent adversarial domain adaptation,” in Proc. Int. Conf. Mach. Learn., 2018.
[4] J. Hoffman, D. Wang, F. Yu, and T. Darrell, “FCNs in the wild: Pixel-level adversarial and constraint-based adaptation,” 2016, arXiv:1612.02649.
[5] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, “Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2019, pp. 2517–2526.
[6] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, “Learning to adapt structured output space for semantic segmentation,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2018, pp. 7472–7481.
[7] Y. Li, L. Yuan, and N. Vasconcelos, “Bidirectional learning for domain adaptation of semantic segmentation,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2019, pp. 6929–6938.
[8] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, “SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud,” in Proc. Int. Conf. Robot. Auton., 2019, pp. 4376–4382.
[9] M. Jaritz, T.-H. Vu, R. D. Charette, E. Wirbel, and P. Pérez, “xMUDA: Cross-modal unsupervised domain adaptation for 3 semantic segmentation,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2020, pp. 12602–12611.
[10] H. Caesar et al., “nuScenes: A multimodal dataset for autonomous driving,” Proc. Conf. Comput. Vis. Pattern Recognit., 2020, pp. 11618–11628.
[11] D.-H. Lee, “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in Proc. Int. Conf. Mach. Learn. Workshops, 2013, p. 896.
[12] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Medical Image Comput. Computer-Assisted Intervention, 2015, pp. 234–241.
[13] B. Graham, M. Engelcke, and L. van der Maaten, “3D semantic segmentation with submanifold sparse convolutional networks,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2018, pp. 9224–9232.
[14] Y. Yang and S. Soatto, “FDA: Fourier domain adaptation for semantic segmentation,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2020, pp. 4082–4091.
[15] Y. Zou, Z. Yu, X. Liu, B. V. Kumar, and J. Wang, “Confidence regularized self-training,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 5981–5990.
[16] A. Saporta, T.-H. Vu, M. Cord, and P. Pérez, “ESL: Entropy-guided self-supervised learning for domain adaptation in semantic segmentation,” in Proc. Conf. Comput. Vis. Pattern Recognit. Workshop, 2020.
[17] T. T. Triess, M. Dressigg, C. B. Rist, and J. M. Züllner, “A survey on deep domain adaptation for lidar perception,” in Proc. IV Workshop Autonomy Scale, 2021, pp. 350–357.
[18] C. Qin, H. You, L. Wang, C.-C. J. Kuo, and Y. Fu, “PointDAN: A multi-scale 3D domain adaption network for point cloud representation,” in Proc. 33rd Int. Conf. Neural Inf. Process. Syst., 2019, pp. 7792–7803.
[19] J. Donahue, H. Maron, and G. Chechik, “Self-supervised learning for domain adaptation on point clouds,” in Proc. IEEE/CVF Winter Appl. Conf. Comput. Vis. 2021, pp. 123–133.
[20] P. Moreiro, J. Cavazza, and V. Murino, “Minimal-entropy correlation alignment for unsupervised deep domain adaptation,” in Proc. Int. Conf. Learn. Representations, 2018.
[21] F. Langer, A. Milioto, A. Haag, J. Behley, and C. Stachniss, “Domain transfer for semantic segmentation of lidar data using deep neural networks,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2020, pp. 8263–8270.
[22] L. Yi, B. Gong, and T. Funkhouser, “Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2021, pp. 15338–15368.
[23] K.-H. Lee, G. Ross, J. Li, and A. Gaidon, “SPIGAN: Privileged adversarial learning from simulation,” in Proc. Int. Conf. Learn. Representations, 2019.
[24] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, “DADA: Depth-aware domain adaptation in semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 7363–7372.
[25] C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, pp. 273–297, 1995.
[26] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, “Semi-supervised domain adaptation with instance constraints,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2013, pp. 668–675.
[27] T. Yao, Y. Pan, C.-W. Ngo, H. Li, and T. Mei, “Semi-supervised domain adaptation with subspace learning for visual recognition,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2015, pp. 2142–2150.
[28] S. Ao, X. Li, and C. X. Ling, “Fast generalized distillation for semi-supervised domain adaptation,” in Proc. 31st AAAI Conf. Artif. Intell., 2017, pp. 1719–1725.
[29] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, “Semi-supervised domain adaptation via minimax entropy,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2019, pp. 8049–8057.
[30] Z. Wang et al., “Alleviating semantic-level shift: A semi-supervised domain adaptation method for semantic segmentation,” in Proc. Conf. Comput. Vis. Pattern Recognit. Workshops, 2020, pp. 936–957.
[31] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, “Fusenet: Incorporating depth into semantic segmentation via fusion-based CNN architecture,” in Proc. Asian Conf. Comput. Vis., 2016, pp. 213–228.
[32] A. Valada, R. Mohan, and W. Burgard, “Self-supervised model adaptation for multimodal semantic segmentation,” Int. J. Comput. Vis., vol. 128, pp. 1239–1285, 2019.
[33] M. Liang, B. Yang, Y. Chen, F. Yu, and R. Urtasun, “Multi-task multi-sensor fusion for 3D object detection,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2019, pp. 7337–7345.
[34] M. Liang, B. Yang, S. Wang, and R. Urtasun, “Deep continuous fusion for multi-sensor 3D object detection,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 663–678.
[35] G. P. Meyer, J. Charland, D. Hegde, A. Laddha, and C. Valltes-i-Cardona, “Sensing fusion for joint 3D object detection and semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshop, 2019, pp. 1230–1237.
[36] L. Castrejon, Y. Aytar, C. Vondrick, H. Pirsiavash, and A. Torralba, “Learning aligned cross-modal representations from weakly aligned data,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2940–2949.
Maximilian Jaritz received the master’s degrees in a dual degree program from TU Berlin and Ecole Centrale Paris, in 2016, and the PhD degree from Inria, and Valeo/Valeo.ai under the supervision of Raoul de Charette, Emilie Wirbel and Patrick P. Perez. He is an applied scientist with Amazon Robotics AI since 2020.

Tuan-Hung Vu received the engineering master’s degree in machine learning from Télécom ParisTech, and the PhD degree in computer vision from Ecole Normale Supérieure Paris. He is a research scientist with Valeo.ai since 2018. His research interests include deep learning, scene understanding, domain adaptation and data augmentation.

Raoul de Charette received the MSc degree in arts and computer graphics from Paris 8 Uni. and the PhD degree in computer vision from Mines ParisTech, in 2012. He is a researcher with Inria Paris since 2015. He was consecutively with Mines ParisTech (2008-2011, 2012-2013), Carnegie Mellon University (2011), and University of Makedonia (2014). He now leads the Computer Vision Group in RITS Team Inria.

Emilie Wirbel received the engineering degree from Mines ParisTech, in 2011, and the PhD degree in robotics, in 2014. She is a senior robotics engineer with NVIDIA. Before, she was with valeo driving assistance research and Valeo.ai, working on prediction and control for autonomous driving, in particular relying on end-to-end imitation learning.

Patrick Perez is currently the scientific director of Valeo.ai, a Valeo research lab on artificial intelligence for automotive applications. Before joining Valeo, Patrick Perez has been distinguished scientist with Technicolor (2009–2018), researcher with Inria (1993–2000, 2004–2009) and at Microsoft Research Cambridge (2000–2004). His research interests include revolves around machine learning for scene understanding, data mining and visual editing.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.