IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 June 2023, accepted 8 August 2023, date of publication 11 August 2023, date of current version 16 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3304542

RESEARCH ARTICLE

Boosting Multi-Modal Unsupervised Domain Adaptation for LiDAR Semantic Segmentation by Self-Supervised Depth Completion

ADRIANO CARDACE, ANDREA CONTI, (Graduate Student Member, IEEE), PIERLUIGI ZAMA RAMIREZ, RICCARDO SPEZIALETTI, SAMUELE SALTI, AND LUIGI DI STEFANO, (Member, IEEE)
Department of Computer Science and Engineering (DISI), University of Bologna, 40126 Bologna, Italy

Corresponding author: Adriano Cardace (adriano.cardace2@unibo.it)

ABSTRACT LiDAR semantic segmentation is receiving increased attention due to its deployment in autonomous driving applications. As LiDARs come often with other sensors such as RGB cameras, multi-modal approaches for this task have been developed, which however suffer from the domain shift problem as other deep learning approaches. To address this, we propose a novel Unsupervised Domain Adaptation (UDA) technique for multi-modal LiDAR segmentation. Unlike previous works in this field, we leverage depth completion as an auxiliary task to align features extracted from 2D images across domains, and as a powerful data augmentation for LiDARs. We validate our method on three popular multi-modal UDA benchmarks and we achieve better performances than other competitors.

INDEX TERMS Depth completion, lidar segmentation, multi-modal, unsupervised domain adaptation.

I. INTRODUCTION

LiDAR semantic segmentation is the task of assigning a class label to each point of a 3D scan gathered by Light Detection and Ranging (LiDAR) sensors. These devices can record accurate depth information regardless of the lighting conditions, making them a reliable source of information for autonomous driving. However, LiDAR data is colorless, unstructured, and sparse. Consequently, scene understanding using only LiDARs is extremely challenging. Yet, nowadays, autonomous vehicles are commonly equipped also with other sensors, such as RGB cameras. For this reason, the research community has recently developed multi-modal approaches [1] exploiting both these modalities. However, akin to other tasks, multi-modal LiDAR segmentation networks suffer from the domain shift problem, i.e. models struggle to generalize to environments different from the training one. A straightforward solution would be to gather more and more annotated data in many scenarios. However this process is cumbersome and time-consuming. As an example, annotating a point cloud acquired in an urban environment of $100m^3$ needs from 1.5 to 4.5 hours by human annotators with 3D expertise [2]. Unsupervised Domain Adaptation (UDA) addresses this problem, using unlabelled data from the target domain to mitigate the domain shift. Many UDA approaches have been proposed for tasks such as classification [3], object detection [4] and 2D semantic segmentation [5]. Yet, only a few proposals deal directly with the LiDAR semantic segmentation task [6], and even fewer try to exploit multiple modalities such as RGB images and LiDAR point clouds [7], [8]. In the latter setup, referred to as multi-modal UDA for LiDAR segmentation, one can leverage both modalities as sources of information. The standard approach processes them by means of two networks, one processing the 2D images and the other the 3D point clouds. To this end, XMUDA [7] proposed a benchmark and a baseline method that uses a cross-modal loss on each domain independently forcing predictions extracted from 3D points and the corresponding 2D pixels to be similar in the same domain. Despite the effectiveness of this approach, we argue that it

The associate editor coordinating the review of this manuscript and approving it for publication was Gianluigi Ciocca.

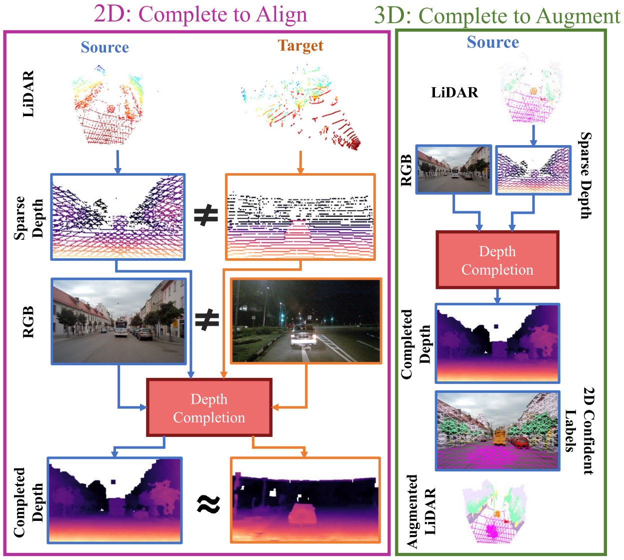

FIGURE 1. **Our multi-modal UDA framework for LiDAR segmentation exploits self-supervised depth completion as an auxiliary task. As completed depths are similar between domains, training a network for 2D segmentation and depth completion pushes source and target features to be more robust to domain shift. Moreover, we can use these depths as a data augmentation for the labeled LiDAR point clouds.**

mainly focuses on the alignment across modalities rather than the actual alignment across domains. DsCML [8] performs a step forward in domain alignment, employing adversarial training to align multi-modal features also between domains. However, adversarial learning is notoriously unstable in segmentation tasks, especially when applied to deep features, leading to variable performance.

In this paper, we explore a different direction: using depth completion as an auxiliary task to make the features of the 2D network more similar between domains and as a powerful data augmentation technique for the 3D network.

We argue that, to complete sparse depth inputs, a network needs to infer the geometrical structure of the scene, e.g., understand the shape of cars or that the road is flat. Unlike 2D appearance, which may be extremely different across domains due to environmental variables such as light and weather, or 3D scans, which may differ because of LiDAR patterns and densities, the 2D depth structure is similar, e.g., roads appear in the bottom part of the image and are flat independently of the domain, as it can be seen in the Completed Depth row of Fig. 1, left column. Following the above reasoning, a depth completion network trained jointly on the source and target data should extract features robust to the domain shift. Moreover, the geometrical structures are tightly linked to the semantics of the scene [9], thus training a network for depth completion should also push the features to be discriminative for the semantic segmentation task.

We leverage these intuitions and we project the 3D points to the image plane to obtain a sparse depth map. Then, we train a multi-task 2D network to jointly segment the RGB source images and complete the sparse depths on both domains, forcing target features to be robust to the domain shift and discriminative for semantic segmentation. However, to the best of our knowledge, no depth completion model can be trained solely on the same LiDAR input without additional data such as ground truth dense depth maps or video sequences. Thus, we propose a simple yet effective self-supervised technique to train a depth completion network without external data.

Finally, we propose to exploit the estimated dense depth maps as a powerful data augmentation technique in the source domain to boost the performance of the 3D network. To do so, we project each pixel back to the 3D space and assign the most confident 2D predictions as proxy labels to the corresponding 3D points. We add new annotated points to the source LiDAR point clouds by looking at class-specific confidence percentiles, thereby obtaining much denser 3D clouds with annotations, as shown in the bottom-right part of Fig. 1. We demonstrate that our approach can achieve state-of-the-art performance on the multi-modal UDA benchmarks introduced by [7]. In short, our contributions are:

- We design a novel self-supervised depth completion technique that uses only RGB images and LiDAR Scans.
- We show that depth completion as an auxiliary task increases the robustness to domain shift of the 2D semantic networks used in the standard RGB-LiDAR setting.
- We show that completed depths and robust 2D networks can be used to increase the 3D dataset density by synthesizing new annotated 3D points to boost performances.
- We achieve new state-of-the-art results in all standard cross-modal UDA benchmarks.

Project page: https://cvlab-unibo.github.io/cts-web/ .

II. RELATED WORKS

In this section, we review some relevant works for our paper.

A. LiDAR SEMANTIC SEGMENTATION

Scene perception from LiDAR data is getting more and more attention as these sensors are becoming a standard in assisted/autonomous driving [10]. Thus, several research datasets have been collected [2], [11], [12] with annotations for tasks such as 3D object detection or 3D semantic segmentation. In particular, segmentation has achieved a lot of popularity, and several approaches have been proposed in the last few years [13], [14], [15], [16]. Most methods for parsing the scene only use 3D information, and some leverage intensity information from LiDARs to aid segmentation. However, with the prevalence of both LiDARs and RGB cameras on autonomous vehicles, there are now multi-modal approaches that combine information from multiple sources to improve performance [1]. In our work, we also focus on multi-modal learning from LiDAR and RGB sensors, though we address it in the UDA scenario.

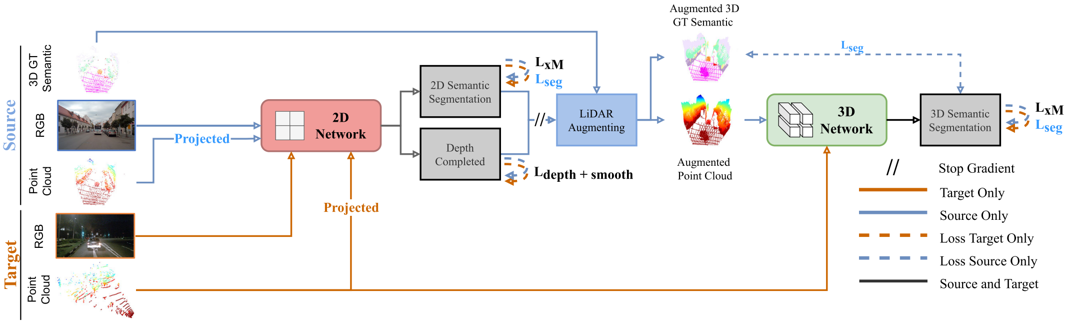

FIGURE 2. **Framework Overview. First, the 2D network outputs a densified depth map and 2D semantic labels, then these data are used to augment the ground truth on the source domain to improve the performance of the 3D network.**

B. 2D UNSUPERVISED DOMAIN ADAPTATION

In the last few years, several UDA approaches have been proposed for 2D semantic segmentation, using strategies such as style-transfer [17], [18], adversarial training [19], [20] or self-training [21], [22]. Recently, some works demonstrated that depth information can boost UDA for 2D semantic segmentation [23], [24], [25]. For instance, [25] uses self-supervised depth estimation from videos to enhance the performance of UDA methods for semantic segmentation. In contrast, we propose depth completion for aligning 2D features across domains, which we found to be more effective.

C. 3D UNSUPERVISED DOMAIN ADAPTATION

While the majority of works consider UDA for semantic segmentation of images, fewer approaches have been proposed for the 3D counterpart, with only a limited number of works addressing the problem of UDA for LiDAR Segmentation [26], [27], [28]. Very recently, some works have addressed the challenging multi-modal segmentation task from LiDARs and RGB sensors [7], [8]. XMUDA [7] is the first work that focuses on UDA in the above setting, defining a benchmark and presenting a baseline approach that employs a loss to align features across modalities. DsCML [8] is the first to explicitly address alignment across domains in this setup employing an adversarial loss, which however may lead to extremely variable performances. Our work focuses on the domain shift problem in the same multi-modal scenario but from a different perspective. We propose using depth completion as an auxiliary task to align features across domains in RGB networks and as a powerful data augmentation technique for the 3D branch.

D. DEPTH COMPLETION

Multi-modal samples retrieved through an RGB camera coupled with a LiDAR usually lead to a sparse depth map containing valid measurements only in a few pixels of the associated RGB image. The task to fill the missing coordinates with a valid depth value is referred to as Depth Completion. This task has been tackled with both non-learned [29] and deep-learning methods [30], [31], [32]. In particular, [30] initially tackled this task by feeding a deep neural network with the RGB frame and sparse depth points jointly. Reference [31] lets the network learn a propagation field which is iteratively applied to an initial depth map to gradually improve the quality of the densified depth map. While effective, these methods rely on almost dense depth ground truth for training. In contrast, [33], [34] utilize self-supervised depth completion with a video sequence. In our work, we propose a novel self-supervised approach that only requires RGB images and sparse LiDAR depth for training.

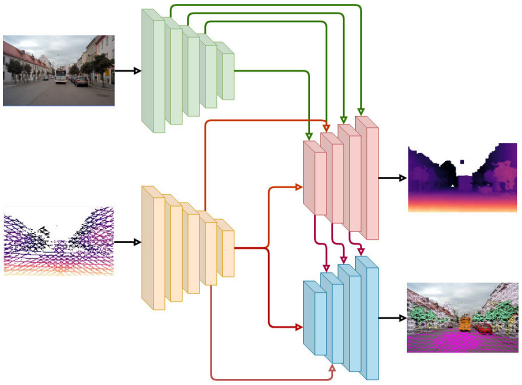

FIGURE 3. **2D Network for Depth Completion and Semantic Segmentation. It is composed of a sparse depth encoder and an RGB encoder, the depth decoder takes features at multiple scales from both to output a densified depth map. The segmentation head leverages the multi-scale features of the depth decoding step to output semantic segmentation labels.**

III. METHOD

Given a LiDAR scan and the corresponding RGB images for both domains, our goal is to solve 3D semantic segmentation on the target domain. Supervision is provided only for sparse 3D points of the source domain. Our framework dubbed Complete to Segment (CtS), is depicted in Fig. 2.

A. PRELIMINARIES

1) SETUP AND NOTATION

Given a 2D image, $\boldsymbol{x}^{2D}$, and a corresponding 3D point cloud, $\boldsymbol{x}^{3D}$, we define as $\boldsymbol{y}^{3D}$ the semantic label for each 3D point. Assuming LiDARs measurements to be expressed in the camera reference frame and the availability of the intrinsic camera matrix, we can project each 3D point into the image plane. A sparse depth map, $\boldsymbol{D}^{3D\rightarrow 2D}$, can be easily obtained by assigning the value of z to each corresponding pixel (u, v). Then, we can assign the 3D label to the corresponding 2D pixels obtaining a sparse 2D semantic map, $\boldsymbol{y}^{3D\rightarrow 2D}$. We denote as $\mathcal{S}$ the *source* domain, for which annotations are available, and as $\mathcal{T}$ the *target* domain, where no annotations are accessible. Thus, we specify by subscripts s and t whether the data belong to $\mathcal{S}$ or $\mathcal{T}$ respectively. Our full dataset is composed of: i) images, $\boldsymbol{x}_s^{2\mathrm{D}}$ and $\boldsymbol{x}_t^{2\mathrm{D}}$; ii) point clouds, $\boldsymbol{x}_s^{3\mathrm{D}}$ and $\boldsymbol{x}_t^{3\mathrm{D}}$; iii) semantic labels for points clouds, $\boldsymbol{y}_s^{3\mathrm{D}}$.

2) TWO-STREAMS ARCHITECTURE

Following the standard approach in this setup [7], [8], we deploy a two-streams architecture that processes 2D and 3D data independently. As argued in [7], having two networks is important to obtain modality-specific predictions which can be fused together effectively. Indeed, the final predictions are obtained by averaging the predictions of the 2D and 3D branches. The proposed multi-task 2D network is described in Sec. Sec. III-B. Regarding the 3D network, similarly to [7] and [8], we use SparseConvNet [35], with voxel size 5 cm to ensure that at most one point is inside each voxel.

3) SUPERVISED LEARNING

We supervise both 2D and 3D networks using the cross-entropy loss on the source domain:

$$\mathcal{L}_{\mathrm{seg}}(\boldsymbol{x}_s, \boldsymbol{y}_s) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} \boldsymbol{y}_s^{(n,c)} \log \boldsymbol{P}_{\boldsymbol{x}_s}^{(n,c)} \tag{1}$$

with $(\boldsymbol{x}_s, \boldsymbol{y}_s)$ being either $(\boldsymbol{x}_s^{2\mathrm{D}}, \boldsymbol{y_s}^{3D\rightarrow 2D})$ or $(\boldsymbol{x}_s^{3\mathrm{D}}, \boldsymbol{y_s}^{3D})$, C denoting the number of classes, N the number of labeled points in a mini-batch, and $\boldsymbol{P}_{\boldsymbol{x}_s}$ the prediction of the 2D or 3D semantic network depending on the modality of $\boldsymbol{x}_s$.

4) CROSS MODAL LEARNING

As highlighted in [7] it is important that the two branches communicate, so that each of the two modalities can take advantage of the other. Given a pair of corresponding 2D-3D points, we apply a mechanism similar to [8], though without deformable convolution. In particular, given a squared patch centered in a 2D point, we force the predictions of each pixel in the patch to be similar to that of the corresponding 3D point with a KL loss. This cross-modal loss denoted as $\mathcal{L}_{\mathrm{xM}}$, is applied by means of auxiliary heads that are trained to mimic the output of the main classifier of the other modality. In this way, the main classifier is simultaneously influenced by the features learned by the other network while keeping its strength. We rely on this simple mechanism to establish a strong baseline as a starting point on which we develop.

B. DEPTH COMPLETION

Our proposal is based on the following considerations. First, 2D depth maps are similar across domains, e.g., the bottom part of the image is smooth, cars have the same 3D shapes regardless of the time of the day, etc. Second, depth structures such as edges or blobs are tightly correlated to semantic segmentation, indeed we can easily recognize that a car is in the scene only by looking at the depth map. Finally, correlations between depth and semantics are similar across domains, e.g, a road is typically a plane or the sky is far away. Based on the above intuitions, in our work, we consider depth completion as an auxiliary task to make the features of the 2D network similar between domains, while at the same time preserving discriminability for semantic segmentation for the target domain. Specifically, given a 3D scan, we project the 3D points into the image plane to obtain a sparse depth map. Then, we train a multi-task 2D network jointly to segment the source images and complete the sparse depth map on both domains, naturally forcing target features to be robust to the domain shift and discriminative for semantic segmentation. Unluckily, current state-of-the-art techniques for depth completion [31], [36] all require either to be trained with dense depth ground truth or auxiliary information such as video sequences [33], [37]. As we can leverage only single-view sparse depths as supervision, we propose a novel technique to achieve this goal. In the next sections, we first define our multi-task architecture for semantic and depth completion, then we describe the training protocol to pursue self-supervised depth completion.

1) 2D DEPTH COMPLETION AND SEMANTIC NETWORK

We modify a standard U-Net [38] with backbone ResNet34 for 2D semantic segmentation by introducing a multi-scale depth encoder and decoder. The latter takes in input depth features at $\frac{1}{8}$, $\frac{1}{16}$ scales, and RGB features at $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$ and $\frac{1}{16}$ to output a dense depth map. Multi-scale depth feature maps from the depth decoder are then leveraged by the semantic segmentation decoder to output semantic classes. A schematic visualization of the 2D network is shown in Fig. 3.

2) DEPTH SUPERVISION

We supervise the depth branch by means of the LiDAR depth points provided as input. We employ the L1 loss alongside edge-aware smoothness [39], described by Eq. 2 and Eq. 3, respectively, where $\boldsymbol{D}$ is the dense output depth map, $M_{uv} = \boldsymbol{D}_{uv}^{3D\rightarrow 2D} > 0$ is a mask of valid reprojected depth points and N_m is the number of valid points in M.

$$L_{\mathrm{depth}}(\boldsymbol{x}^{2\mathrm{D}}, \boldsymbol{D}^{3D\rightarrow 2D}) = \frac{1}{N_m}\sum_{u,v}^{N_m} |\boldsymbol{D}_{uv} - \boldsymbol{D}_{uv}^{3D\rightarrow 2D}| \cdot M_{uv} \tag{2}$$

$$L_{\mathrm{smooth}}(\boldsymbol{x}^{2\mathrm{D}}) = \frac{1}{N_m}\sum_{u,v}^{N_m}(|\delta_u \boldsymbol{D}|e^{-|\delta_u \boldsymbol{x}^{2\mathrm{D}}|} + |\delta_v \boldsymbol{D}|e^{-|\delta_v \boldsymbol{x}^{2\mathrm{D}}|}) \tag{3}$$

We penalize abrupt depth changes in areas other than RGB edges through the L_{smooth} term. This strategy has

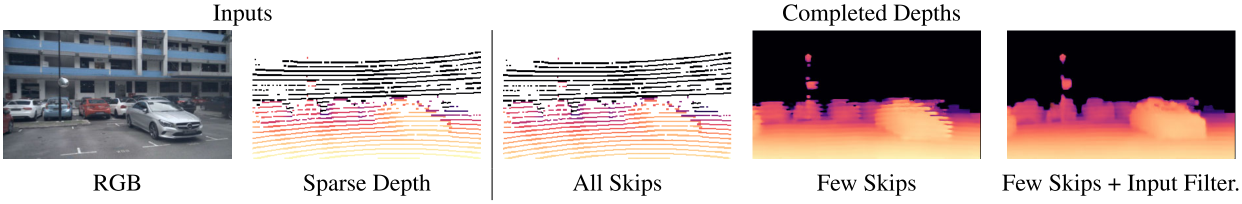

FIGURE 4. Depth Completion Ablation. From left to right: the RGB image, the input sparse depth, the depth completed using all the skip connections from the depth encoder, the depth completed using only skip connections $\frac{1}{8}$ and $\frac{1}{16}$, the depth completed applying also input filtering. Sparse depth maps are dilated for visualization purposes.

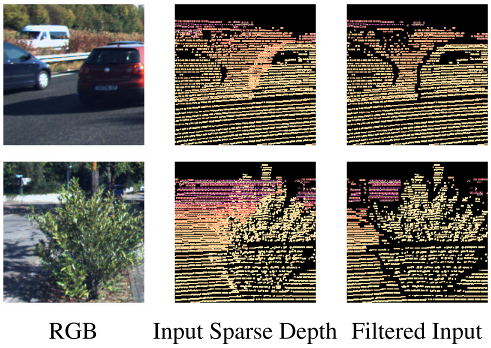

FIGURE 5. Example of input filtering to remove occluded pixels. From left to right: RGB image, sparse depth obtained from input LiDAR, filtered depth in input to our 2D network.

been proven to be effective in self-supervised depth-from-mono [39]. However, by supervising with the LiDAR depth points provided in input, the network can simply learn the identity function. As described before, we limit the usage of high-resolution skip connections from the depth encoder to prevent this behaviour, i.e., we use only skip connections at a $\frac{1}{8}$ and $\frac{1}{16}$ of the input resolution. By comparing columns 3 and 4 of Fig. 4 we note the importance of this choice.

3) INPUT FILTERING

Depth completion frameworks are usually trained without any kind of filtering of the sparse depth provided in input [31], [32], [40]. However, when the sparse input depths are obtained through re-projection from a LiDAR sensor, large areas may be affected by errors due to occlusion between the LiDAR sensor and the RGB camera. This issue yields regions where depth measurements of occluded objects mix together, typically at the borders of objects standing in front of a background far behind, as shown in Fig. 5 (middle column). Usually, depth completion networks can learn to cope with this issue, if not excessively prominent, when a cleaned and denser depth ground truth is available. However, when self-supervising the network by the LiDAR itself, these inconsistencies worsen the completion performance. To filter out the occluded depth points, we follow the coarse yet simple and fast approach proposed by [41]: for each depth point d of the projected LiDAR $\boldsymbol{D}^{3D\to 2D}$, we take into account the other valid depth points inside a patch $L_F(d)$ of size $F \times F$ and compute the minimum $m(d) = \min\{y : y \in L_F(d), y > 0\}$, then we apply a threshold over the error between the minimum and the depth value to filter out the outliers $|m(d)-d|/m(d) < \lambda_f$, with $\lambda_f = 0.1$, obtaining a filtered depth, as shown in Fig. 5 (3rd column). We set $F = 9$ in this work. Applying this filtering step to the sparse input depths improves the densified outputs provided by our depth completion network, as shown in Fig. 4 (last column).

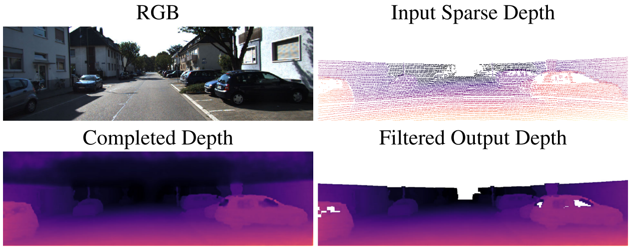

FIGURE 6. Output filtering to remove large areas without valid LiDAR measurements. From top to bottom, from left to right: RGB image, sparse input depth, completed depth map, filtered output.

4) OUTPUT FILTERING

Finally, we argue that the densified depth map really depends on the spatial distribution of the valid depth measurements. Even though LiDAR sensors usually output an almost homogeneous distribution of sparse points, large areas of the image can lack them at all, e.g. the sky, reflective or absorbent surfaces, as well as objects too far away. In these regions, the depth completion network will likely yield wrong predictions that are not good to be projected back into the 3D point cloud, which is needed for the LiDAR data augmentation strategy described in the next section. To filter these regions out, we employ the following strategy. First, we set pixels with invalid depth measurements in the input LiDAR to zero (white pixels in the top-right image of Fig. 6). Then, we apply a max pool with a large kernel size of size λ_p and stride 1 to the input sparse LiDAR obtaining a dilated depth map. We use λ_p equal to 17. In the dilated depth map, pixels with a large invalid neighborhood will have a depth equal to zero. We then select pixel coordinates with a depth equal to zero, and we filter out pixels at the same coordinates from the completed depth map produced by our 2D network (white part of the bottom-right figure in Fig. 6).

C. LiDAR DATA AUGMENTATION

Thanks to depth completion, we obtain dense depth maps that can be exploited to boost the performance of the 3D network. Assuming that LiDAR measurements are in the camera reference frame and intrinsic parameters of the RGB camera are known, we can project back each 2D pixel into the corresponding 3D point. Potentially, we can use all these 3D points to increase the number of samples in input to the 3D network. In this way, we alleviate the severe sparsity problem of LiDAR point clouds, and at the same time, we reduce the overfitting on the source input scanning pattern that can be different from the target one. However, to fully exploit the potential of the completed depth map, we ought to be able to assign a label to each new point. We do this by relying on the output of the 2D backbone as the 2D network has an inductive bias that pushes pixels in the same neighbourhood to be classified similarly, even with sparse supervision, thus producing dense semantic predictions. However, naively projecting all pixels to obtain a 3D point cloud leads to a huge input that would make the training impractical. Moreover, not all semantic predictions are correct, especially for the target domain. For these reasons, we lift proxy labels from 2D to 3D only for data from the source domain, where the network is trained supervisedly. Then, we select points based on a class-wise confidence-level strategy. Given a 2D dense semantic prediction $P^{2\mathrm{D}}_{x_s}$, for each pixel location, we apply the argmax operator to obtain the predicted semantic class, and we use the max operator on the logits after softmax to obtain a per-pixel confidence map as done in several other works [21], [42]. Then, for each class c, we sort predictions based on their confidence scores and we maintain a random 2% among the 10% most confident pixels. In this way, we take into account the class distribution and select pixels for all classes, including rarer ones. Thus, we generate new points and proxy labels only for the source domain, respectively $\boldsymbol{x}_s^{\tilde{3}\mathrm{D}}$ and $\boldsymbol{y}_s^{\tilde{3}\mathrm{D}}$. A visualization of this augmentation is illustrated in Fig. 7. The original labeled LiDAR (left column) only covers a small fraction of the whole image, while our method allows us to synthesize new correctly labeled points (right column).

D. LEARNING PROCESS

The framework is optimized in an end-to-end manner by the following objective function:

$$\begin{aligned}\mathcal{L} = {} & \mathcal{L}_{\mathrm{seg}}(\boldsymbol{x}_s^{2\mathrm{D}}, \boldsymbol{y}_{\boldsymbol{s}}{}^{3D\rightarrow 2D}) \\ & + \mathcal{L}_{\mathrm{seg}}(\boldsymbol{x}_s^{3\mathrm{D}}, \boldsymbol{y}_s^{3\mathrm{D}}) + \mathcal{L}_{\mathrm{seg}}(\boldsymbol{x}_s^{\tilde{3}\mathrm{D}}, \boldsymbol{y}_s^{\tilde{3}\mathrm{D}}) \\ & + \lambda_s \mathcal{L}_{\mathrm{xM}}(\boldsymbol{x}_s^{2D}, \boldsymbol{x}_s^{3D}) + \lambda_t \mathcal{L}_{\mathrm{xM}}(\boldsymbol{x}_t^{2D}, \boldsymbol{x}_t^{3D}) \\ & + \lambda_d L_{\mathrm{depth}}(\boldsymbol{x}_s^{2D}, \boldsymbol{D}_s^{3D\rightarrow 2D}) + \lambda_g L_{\mathrm{smooth}}(\boldsymbol{x}_s^{2D}) \\ & + \lambda_d L_{\mathrm{depth}}(\boldsymbol{x}_t^{2D}, \boldsymbol{D}_t^{3D\rightarrow 2D}) + \lambda_g L_{\mathrm{smooth}}(\boldsymbol{x}_t^{2D}) \qquad (4)\end{aligned}$$

where λ parameters are the weights applied to each loss component. We keep these hyper-parameters fixed for all settings. Note that, $\mathcal{L}_{\mathrm{seg}}(\boldsymbol{x}_s^{\tilde{3}\mathrm{D}}, \boldsymbol{y}_s^{\tilde{3}\mathrm{D}})$ is only activated after a certain amount of steps N_{aug}, as we need depth completion to be strong enough to reach a low error in its predictions and the 2D segmentation to be reasonably accurate in the source domain. Moreover, when synthesizing the new 3D points $\boldsymbol{x}_s^{\tilde{3}\mathrm{D}}$ from the completed depths, we avoid gradient propagation back to the 2D network from the 3D network.

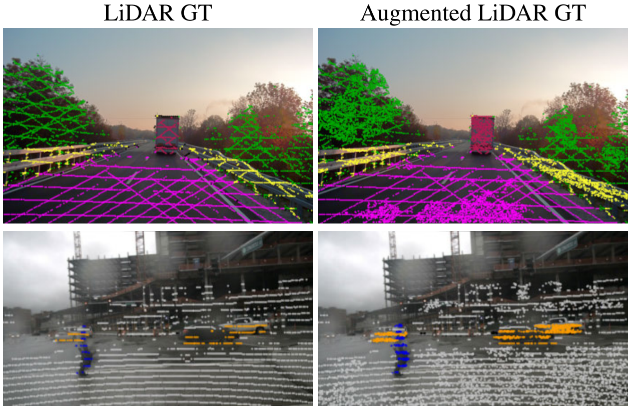

FIGURE 7. Effect of our LiDAR augmentation on A2D2 (top) and Nuscenes-USA (bottom). Left: GT LiDAR projected in 2D. Right: Augmented LiDAR with new labeled points, projected over 2D images. Colors indicate semantic class.

IV. EXPERIMENTS

A. IMPLEMENTATION DETAILS AND DATASETS

We use the same data augmentation pipeline as our competitors, i.e., random horizontal flipping and color jittering for 2D images, vertical axis flipping, random scaling, and random 3D rotations for the 3D scans. Augmentations are done independently for each branch. We train with batch size 8, alternating batches of source and target domain. The smallest dataset is repeated to match the length of the other. We use Adam optimizer, we initialize the learning rate at 0.001 and divide by 10 at the iterations 80k and 90k. We train for 100k steps. We use λ_s, λ_t, λ_d, λ_g equals to 0.8, 0.1, 0.1, 0.01 respectively. We selected these parameters based on training loss values, without performing a grid search. We use the same values for all our experiments. We evaluate our framework in the same way as our two competitors xMUDA [7] and DsCML [8] on three standard benchmarks used for multi-modal domain adaptation that provide three different scenarios: day-to-night, country-to-country, and dataset-to-dataset. The first two settings leverage the NuScenes [12] dataset by means of the Day/Night and USA/Singapore splits. In the former, the RGB images exhibit a severe gap due to the different lighting conditions, while the LiDAR shows small differences being the same sensor. For the latter, the sensor setup is the same, but objects may have different appearances as two different cities are involved. The dataset-to-dataset case is realized by adapting from A2D2 [11] to SemanticKITTI [2], which comprises both a large change in the sensors setup and in appearance.

B. UDA RESULTS

We report in Tab. 1 our results. We detail for each method the mean Intersection over Union (mIoU) for each modality

TABLE 1. Results for 3D semantic segmentation with both uni-modal and multi-modal adaptation methods. We report performance for each network stream in terms of mIoU. 'Avg' column denotes the obtained by taking the mean of the 2D and 3D predictions. * indicates the mean of three different runs with different seeds.

Modality	Method	USA → Singapore			Day → Night			A2D2 → SemanticKITTI		
		2D	3D	Avg	2D	3D	Avg	2D	3D	Avg
	Baseline (Source only)	53.2	46.8	61.2	41.8	41.4	47.6	36.4	37.3	42.2
Uni-modal	MinEnt [43]	53.4	47.0	59.7	44.9	43.5	51.3	38.8	38.0	42.7
	CyCADA [17]	54.9	48.7	61.4	45.7	45.2	49.7	38.2	43.9	43.9
	AdaptSegNet [20]	56.3	47.7	61.8	45.3	44.6	49.6	38.8	44.3	44.2
	CLAN [44]	57.8	51.2	62.5	45.6	43.7	49.2	39.2	44.7	44.5
Multi-modal*	xMUDA [7]	57.2	51.6	61.1	48.9	45.6	52.9	39.0	43.4	44.9
	DsCML [8]	58.5	52.3	62.3	47.5	45.2	53.0	38.9	40.1	43.2
	DsCML + CMAL [8]	57.5	51.0	61.9	46.9	36.2	49.2	27.4	33.3	33.6
	CtS (Ours)	**61.9**	**52.4**	**63.6**	**50.8**	**47.2**	**58.3**	**39.6**	**46.0**	**45.8**

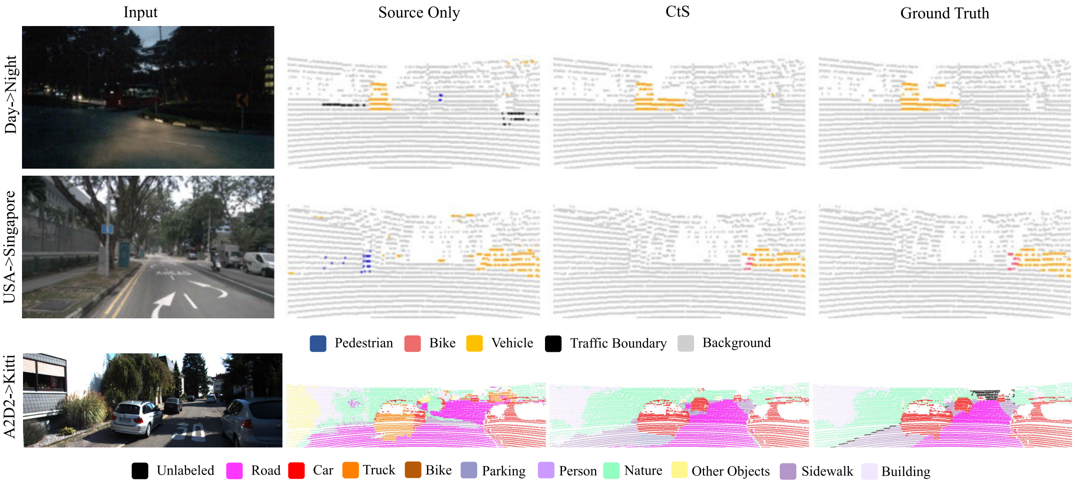

FIGURE 8. Qualitatives Adaptation Results. Left to right: Input images, source-only predictions, CtS predictions, and GT.

TABLE 2. Ablation studies for the proposed contributions. 2D-C: 2D completion, 3D-A: Augmentation for 3D network.

		USA → Singapore			Day → Night		
2D-C	3D-A	mIoU			mIoU		
		2D	3D	Avg	2D	3D	Avg
		56.2	51.3	61.8	48.2	42.8	52.8
✓		61.3	51.0	62.1	50.5	45.4	54.2
	✓	56.5	**52.6**	60.5	49.5	45.8	53.0
✓	✓	**61.9**	52.4	**63.6**	**50.8**	**47.2**	**58.3**

independently (2D and 3D), and we also show the score obtained by averaging the 2D and 3D scores after Softmax as done by our competitors (Avg). We also report results from [8] for uni-modal domain adaptation techniques applied to each modality independently as a reference. To provide more reliable results in this multi-modal setup, we report the average of three different runs, using the official code[1] provided by the authors. We highlight that we used the same number of steps, optimizers, and hyper-parameters for our competitors as well as for our method to be fair. Trainings require approximately one day for the USA → Singapore and Day → Night setups, and three days for A2D2 → SemanticKITTI on an NVIDIA 3090 RTX GPU. Model selection is done as in our competitors by selecting the best on the validation domain of the target domain, and reporting results on the test set. Our method achieves state-of-the-art performance across all scenarios and modalities. In particular CtS shines in the Day → Night adaptation scenario, where the RGB domain gap is larger. In this setting, in fact, we improve by 1.9% for the 2D branch and by 1.6% in terms of mIoU when comparing with the best previous model. When averaging the predictions from both 2D and 3D branches, which is the real and final objective, we observe an even larger improvement of 5,3% (Avg column). We attribute this to the completion auxiliary task, which is able to guide the network to classify each pixel by also reasoning on the 3D cues learned by solving the depth completion task. On USA → Singapore, we also observe good results, especially for the 2D branch where we obtain a large 3.4% improvement. This means that the proposed depth completion auxiliary task is also beneficial in presence of a smaller RGB domain gap. As regards A2D2 → SemanticKITTI, we improve by 0.6%, 2.6%, and 0.9% the previous best multi-modal framework for 2D, 3D, and Avg respectively. In this setting, where the LiDAR sensor is completely different across domains, we substantially improve the performance of the 3D network. This is due to the proposed 3D augmentation, which is indeed able to avoid overfitting of the source pattern and at the same time reduce the sparsity of the LiDAR input.

[1] https://github.com/leolyj/DsCML, https://github.com/valeoai/xmuda

TABLE 3. Comparison with different auxiliary tasks.

2D architecture	USA → Singapore			Day → Night		
	mIoU			mIoU		
	2D	3D	Avg	2D	3D	Avg
Depth from Mono	57.4	49.7	60.2	38.4	45.1	43.7
Completion	**61.3**	**51.0**	**62.1**	**50.5**	**45.4**	**54.2**

C. ADDITIONAL STUDIES

1) ABLATION OF CONTRIBUTIONS

In Tab. 2 we ablate the effect of our contributions considering both USA → Singapore and Day → Night. In the first row, we report the results of our baseline architecture, where we employ the cross-modal loss L_{xM} between the two modalities as done in [8] but without the deformable convolutions. In the second row, we activate depth completion as an auxiliary task, and we observe a large boost for the 2D network for both scenarios: +5,1% and +2,3% respectively. This confirms that forcing the network to reason about the depth structures of the input image helps to generalize better. Moreover, we can observe that our first contribution already improves the overall performance (Avg) by 0.3% and 1.4% respectively. When only activating the LiDAR augmentation, we expect the 3D branch to observe a larger improvement as we are specifically tackling the 3D modality. Indeed, we note a +1.6% for USA → Singapore and +0.4% for Day → Night in terms of mIoU when comparing the performance of this model (third row) with our baseline. Since this augmentation step needs a dense depth map, in this case, we exploit a separate depth completion network pre-trained with our self-supervised methodology. Thereby, the 2D semantic network is not multi-task. When activating all contributions, we obtain the best average results. In Fig. 8, we depict a qualitative comparison of a source-only model with our proposal.

2) AUXILIARY TASKS ALTERNATIVES

A plausible alternative to injecting 3D cues into the learning process is to use monocular depth estimation as an auxiliary task. To compare depth completion with this solution, we implement a network with a single encoder that processes RGB images and two decoders, one that predicts each pixel semantic label, and one to estimate depth as done for the depth completion task. The model is then optimized in the same way i.e., by applying Eq. (2) and Eq. (3) on both domains and Eq. (1) only for the source one. We compare this variant with the proposed auxiliary task in Tab. 3. We observe that depth completion performs better across all modalities in both Day → Night and USA → Singapore. This is due to the fact that to solve the task of monocular depth estimation, the network has to reason on RGB features that do not provide any additional improvements if the gap in the RGB space is too large. On the other hand, we argue that depth completion networks can focus also on the geometry of the scene in input and not only on RGB images to solve the task, and this is important to improve generalization on the target domain.

TABLE 4. Results on the validation split of KITTI Depth Completion. GT: ground truth, Photometric: photometric loss on videos, LiDAR: sparse depths from input LiDAR. ‡: evaluated using officially released weights.

Supervision		Method	RMSE ↓ (mm)	MAE ↓
GT Depth	‡	NLSPN [32]	788.00	199.50
	‡	PENet [40]	791.62	242.25
	‡	PackNet [46]	1027.32	356.04
		StD [33]	878.56	260.90
Photometric + LiDAR		StD [33]	1384.85	358.92
Photometric		StD [33]	1901.16	658.13
LiDAR		CTS (depth only)	1788.37	506.86

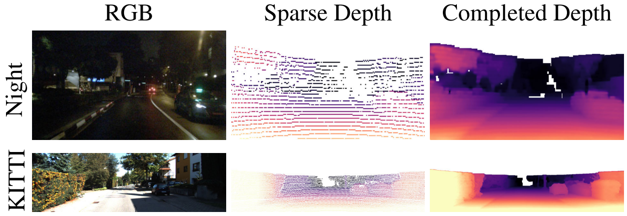

FIGURE 9. Depth completion qualitative results. From left to right, RGB, sparse depth from LiDAR, and completed depth.

3) QUANTITATIVE RESULTS ON DEPTH COMPLETION

Although we mainly employ depth completion as an auxiliary task, we investigate the quality of completed depths also from a quantitative point of view. In Table 4, we compare our self-supervised approach with state-of-the-art supervised depth completion methods that leverage the dense ground-truth of the KITTI-Depth-Completion split [45], and with methods that exploit video sequences [33]. Despite being trained with the input LiDAR only, our performances are still comparable. Indeed, our method has a Mean Absolute Error (MAE) only 300mm higher than the state-of-the-art supervised method [32] (row 1 vs 7), and performs better than [33] when using only the photometric loss on video sequences (row 6 vs 7). The quality of our completed depth maps can be assessed by looking at the results in Fig. 9.

V. CONCLUSION

We introduce CtS, a novel multi-modal UDA method for LiDAR segmentation. We show that depth completion is an effective auxiliary task to improve generalization for the 2D network. Furthermore, we propose to exploit completed depths to augment the source LiDAR to achieve better results. We believe that this task could be even more useful when applied to online adaptation, where video sequences can be available and could be used to obtain better 3D geometries, and consequently a better semantic understanding.

REFERENCES

[1] 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 *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 677–695.

[2] 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 *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9296–9306.

[3] M. Wang and W. Deng, ‘‘Deep visual domain adaptation: A survey,’’ *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018.

[4] P. Oza, V. A. Sindagi, V. V. Sharmini, and V. M. Patel, ‘‘Unsupervised domain adaptation of object detectors: A survey,’’ 2021, *arXiv:2105.13502*.

[5] G. Csurka, R. Volpi, and B. Chidlovskii, ‘‘Unsupervised domain adaptation for semantic image segmentation: A comprehensive survey,’’ 2021, *arXiv:2112.03241*.

[6] L. T. Triess, M. Dreissig, C. B. Rist, and J. Marius Zöllner, ‘‘A survey on deep domain adaptation for LiDAR perception,’’ in *Proc. IEEE Intell. Vehicles Symp. Workshops (IV Workshops)*, Jul. 2021, pp. 350–357.

[7] M. Jaritz, T.-H. Vu, R. de Charette, E. Wirbel, and P. Pérez, ‘‘XMUDA: Cross-modal unsupervised domain adaptation for 3D semantic segmentation,’’ in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12602–12611.

[8] 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 *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7088–7097.

[9] P. Z. Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. Di Stefano, ‘‘Geometry meets semantics for semi-supervised monocular depth estimation,’’ in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 298–313.

[10] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, ‘‘Deep learning for 3D point clouds: A survey,’’ *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021.

[11] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. Hoang Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth, ‘‘A2D2: Audi autonomous driving dataset,’’ 2020, *arXiv:2004.06320*.

[12] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11618–11628.

[13] K. Sirohi, R. Mohan, D. Büscher, W. Burgard, and A. Valada, ‘‘EfficientLPS: Efficient LiDAR panoptic segmentation,’’ *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1894–1914, Jun. 2022.

[14] C. Xu, B. Wu, Z. Wang, W. Zhan, P. Vajda, K. Keutzer, and M. Tomizuka, ‘‘SqueezeSegV3: Spatially-adaptive convolution for efficient point-cloud segmentation,’’ in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 1–19.

[15] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, ‘‘RangeNet++: Fast and accurate LiDAR semantic segmentation,’’ in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4213–4220.

[16] Y. Hou, X. Zhu, Y. Ma, C. C. Loy, and Y. Li, ‘‘Point-to-Voxel knowledge distillation for LiDAR semantic segmentation,’’ in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8469–8478.

[17] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, ‘‘CYCADA: Cycle-consistent adversarial domain adaptation,’’ in *Proc. ICML*, 2018, pp. 1989–1998.

[18] M. Kim and H. Byun, ‘‘Learning texture invariant representation for domain adaptation of semantic segmentation,’’ in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12972–12981.

[19] J. Hoffman, D. Wang, F. Yu, and T. Darrell, ‘‘FCNs in the wild: Pixel-level adversarial and constraint-based adaptation,’’ 2016, *arXiv:1612.02649*.

[20] 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. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7472–7481.

[21] Y. Zou, Z. Yu, B. V. Kumar, and J. Wang, ‘‘Unsupervised domain adaptation for semantic segmentation via class-balanced self-training,’’ in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 289–305.

[22] A. Cardace, P. Z. Ramirez, S. Salti, and L. Di Stefano, ‘‘Shallow features guide unsupervised domain adaptation for semantic segmentation at class boundaries,’’ in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 2010–2020.

[23] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. P. Pérez, ‘‘DADA: Depth-aware domain adaptation in semantic segmentation,’’ in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7363–7372.

[24] P. Z. Ramirez, A. Tonioni, S. Salti, and L. D. Stefano, ‘‘Learning across tasks and domains,’’ in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8109–8118.

[25] A. Cardace, L. De Luigi, P. Z. Ramirez, S. Salti, and L. Di Stefano, ‘‘Plugging self-supervised monocular depth into unsupervised domain adaptation for semantic segmentation,’’ in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 1999–2009.

[26] C. B. Rist, M. Enzweiler, and D. M. Gavrila, ‘‘Cross-sensor deep domain adaptation for LiDAR detection and segmentation,’’ in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1535–1542.

[27] P. Jiang and S. Saripalli, ‘‘LiDARNet: A boundary-aware domain adaptation model for point cloud semantic segmentation,’’ in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 2457–2464.

[28] 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 *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 2649–2655.

[29] Y. Zhao, L. Bai, Z. Zhang, and X. Huang, ‘‘A surface geometry model for LiDAR depth completion,’’ *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4457–4464, Jul. 2021.

[30] F. Ma and S. Karaman, ‘‘Sparse-to-dense: Depth prediction from sparse depth samples and a single image,’’ in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4796–4803.

[31] X. Cheng, P. Wang, and R. Yang, ‘‘Depth estimation via affinity learned with convolutional spatial propagation network,’’ in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 103–119.

[32] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. S. Kweon, ‘‘Non-local spatial propagation network for depth completion,’’ in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Jul. 2020, pp. 120–136.

[33] F. Ma, G. V. Cavalheiro, and S. Karaman, ‘‘Self-supervised sparse-to-dense: Self-supervised depth completion from LiDAR and monocular camera,’’ in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 3288–3295.

[34] A. Wong, X. Fei, S. Tsuei, and S. Soatto, ‘‘Unsupervised depth completion from visual inertial odometry,’’ *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1899–1906, Apr. 2020.

[35] B. Graham, M. Engelcke, and L. V. D. Maaten, ‘‘3D semantic segmentation with submanifold sparse convolutional networks,’’ in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9224–9232.

[36] S. Liu, S. De Mello, J. Gu, G. Zhong, M.-H. Yang, and J. Kautz, ‘‘Learning affinity via spatial propagation networks,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Long Beach, CA, USA: Curran Associates, 2017, pp. 1–11.

[37] A. Wong and S. Soatto, ‘‘Unsupervised depth completion with calibrated backprojection layers,’’ in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12727–12736.

[38] O. Ronneberger, P. Fischer, and T. Brox, ‘‘U-Net: Convolutional networks for biomedical image segmentation,’’ in *Medical Image Computing and Computer-Assisted Intervention—MICCAI.* Munich, Germany: Springer, 2015, pp. 234–241.

[39] C. Godard, O. M. Aodha, and G. J. Brostow, ‘‘Unsupervised monocular depth estimation with left-right consistency,’’ in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6602–6611.

[40] M. Hu, S. Wang, B. Li, S. Ning, L. Fan, and X. Gong, ‘‘PENet: Towards precise and efficient image guided depth completion,’’ in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13656–13662.

[41] A. Conti, M. Poggi, F. Aleotti, and S. Mattoccia, ‘‘Unsupervised confidence for LiDAR depth maps and applications,’’ in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 8352–8359.

[42] Y. Li, L. Yuan, and N. Vasconcelos, ‘‘Bidirectional learning for domain adaptation of semantic segmentation,’’ in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6929–6938.

[43] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, ‘‘ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation,’’ in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2512–2521.

[44] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, ‘‘Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation,’’ in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2502–2511.

[45] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant CNNs," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 11–20.

[46] V. Guizilini, R. Ambrus, W. Burgard, and A. Gaidon, "Sparse auxiliary networks for unified monocular depth prediction and completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11073–11083.

ADRIANO CARDACE is currently pursuing the Ph.D. degree with the Computer Vision Laboratory (CVLab), University of Bologna. He has authored several research articles covering a range of subjects, including semantic segmentation, domain adaptation, self-supervised learning, and 3D vision.

ANDREA CONTI (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree in computer science and engineering with the University of Bologna, Italy. Throughout his academic journey, he focused his research interests mainly on 3D reconstruction and active depth sensors fusion. He authored various research papers on these topics.

PIERLUIGI ZAMA RAMIREZ received the Ph.D. degree in computer science and engineering, in 2021. He was a Research Intern with Google, for six months. He is currently a Postdoctoral Researcher with the University of Bologna. He coauthored 15 publications on several computer vision research topics, such as semantic segmentation, depth estimation, optical flow, domain adaptation, virtual reality, and 3D reconstruction.

RICCARDO SPEZIALETTI received the Ph.D. degree in computer science and engineering from the University of Bologna, in 2020. He was a Postdoctoral Researcher with the Department of Computer Science and Engineering, University of Bologna, for two years. He is currently a Research Scientist with EyeCan.ai, a spinoff from the University of Bologna, founded in 2020. His research interests include machine/deep learning for 3D computer vision problems.

SAMUELE SALTI was a Co-Founder of start-up EyeCan.ai, in 2020. He is currently an Associate Professor with the Department of Computer Science and Engineering (DISI), University of Bologna, Italy. He has coauthored more than 50 publications and eight international patents. His research interests include computer vision, in particular 3D computer vision, and machine/deep learning applied to computer vision problems.

LUIGI DI STEFANO (Member, IEEE) received the Ph.D. degree in electronic engineering and computer science from the University of Bologna, in 1994. He was a Scientific Consultant for major companies, in the fields of computer vision and machine learning. He is currently a Full Professor with the Department of Computer Science and Engineering, University of Bologna, where he founded and leads the Computer Vision Laboratory (CVLab). He is the author of more than 150 papers and several patents. His research interests include image processing, computer vision, and machine/deep learning. He is a member of the IEEE Computer Society and IAPR-IC.

Open Access funding provided by 'Alma Mater Studiorum - Università di Bologna' within the CRUI CARE Agreement