Exploiting the Complementarity of 2D and 3D Networks to Address Domain-Shift in 3D Semantic Segmentation

Published Version:
Exploiting the Complementarity of 2D and 3D Networks to Address Domain-Shift in 3D Semantic Segmentation / Adriano Cardace, Pierluigi Zama Ramirez, Samuele Salti, Luigi Di Stefano. - ELETTRONICO. - (2023), pp. 98-109. (Intervento presentato al convegno IEEE/CVF Conference on Computer Vision and Pattern Recognition tenutosi a Vancouver, Canada nel 18/06/2023) [10.1109/CVPRW59228.2023.00015].

This version is available at: https://hdl.handle.net/11585/955886 since: 2024-02-06

Published:

DOI: http://doi.org/10.1109/CVPRW59228.2023.00015

Exploiting the Complementarity of 2D and 3D Networks to Address Domain-Shift in 3D Semantic Segmentation

Adriano Cardace Pierluigi Zama Ramirez Samuele Salti Luigi Di Stefano
Department of Computer Science and Engineering (DISI)
University of Bologna, Italy
{adriano.cardace2}@unibo.it

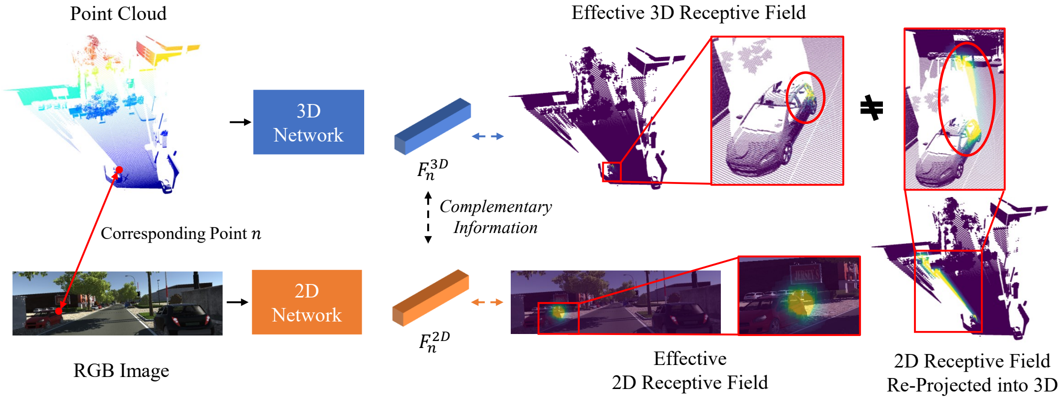
Point Cloud
3D Network
F_n^{3D}
Effective 3D Receptive Field
Corresponding Point n
Complementary Information
RGB Image
2D Network
F_n^{2D}
Effective 2D Receptive Field
2D Receptive Field Re-Projected into 3D

Figure 1. 3D (top) and 2D (bottom) networks processing point clouds and images of the same scene extract features that contain complementary information. Indeed, 2D and 3D effective receptive fields [34] centered on a point n focus on different portions of the scene, i.e., 2D or 3D neighborhoods respectively. Thus, corresponding features have different content by construction. We exploit this property to reduce the domain gap in 3D semantic segmentation.

Abstract

3D semantic segmentation is a critical task in many real-world applications, such as autonomous driving, robotics, and mixed reality. However, the task is extremely challenging due to ambiguities coming from the unstructured, sparse, and uncolored nature of the 3D point clouds. A possible solution is to combine the 3D information with others coming from sensors featuring a different modality, such as RGB cameras. Recent multi-modal 3D semantic segmentation networks exploit these modalities relying on two branches that process the 2D and 3D information independently, striving to maintain the strength of each modality. In this work, we first explain why this design choice is effective and then show how it can be improved to make the multi-modal semantic segmentation more robust to domain shift. Our surprisingly simple contribution achieves state-of-the-art performances on four popular multi-modal unsupervised domain adaptation benchmarks, as well as better results in a domain generalization scenario.

1. Introduction

3D semantic segmentation is a critical task in many real-world applications, such as autonomous driving and robotics. It involves assigning labels to 3D points in a point cloud based on their semantic meaning. However, this task can be extremely challenging due to ambiguities coming from the unstructured, sparse, and uncolored nature of the 3D point clouds. Fortunately, combining 3D information with others coming from sensors with a different modality, such as RGB cameras, can help to address these shortcomings. Indeed, by combining multi-modal data, we can leverage the strengths of each modality to produce more comprehensive and accurate segmentations. For example, in autonomous driving scenarios, RGB cameras and LiDARs are commonly used together. RGB cameras provide dense, colored, and structured information, but they may fail in dark lighting conditions. On the other hand, LiDARs are robust to light conditions, but the point clouds present the problems highlighted above. By combining these two modalities, we can obtain a richer understanding of the environment and make more robust and precise 3D segmentations.

Several recent approaches for multi-modal 3D semantic segmentation [24, 25, 39, 51, 68] leverage a peculiar two-branch 2D-3D architecture, in which images are processed by a 2D convolutional network, e.g., ResNet [18], while point clouds by a 3D convolutional backbone, e.g., SparseConvNet [15]. By processing each modality independently, each of the two branches focuses on extracting features from its specific signal (RGB colors or 3D structure information) that can be fused effectively due to their inherent complementarity in order to produce a better segmentation score. Indeed, averaging logits from the two branches provides often an improvement in performance, e.g., a mIoU gain from 2% to 4% in almost all experiments in [24]. Although we agree that each modality embodies specific information, such as color for images and 3D coordinates for point clouds, we argue that the complementarity of the features extracted by the two branches is also tightly correlated to the different information processing machinery, i.e., 2D and 3D convolutions, which makes networks focusing on different areas of the scene with different receptive fields. Indeed, in Fig. 1, given a point belonging to the red car, we visualize the effective receptive field [34] of the 2D and 3D networks (red ellipses). As we can clearly see from the receptive fields in the right part of the figure, the features extracted by the 3D network mainly leverage points in a 3D neighborhood, i.e., include points of the car surface. In contrast, the features extracted by the 2D network look at a neighborhood in the 2D projected space, and thus they depend also on pixels of the building behind the car, which are close in image space but far in 3D. We argue that this is one of the main reasons why the features from the two branches can be fused so effectively. Based on the above intuition, we propose to feed 3D and RGB signals to both networks as this should not hinder the complementarity of their predictions, with the goal of making the network more robust to the change of distributions between the training and the test scenarios. This problem is typically referred to *Domain shift* in the literature. Feeding both branches with both modalities would make: i) the 2D network more robust to domain shifts, as depth information (z coordinates of point clouds projected into image space) is more similar across different domains, as shown in several papers [6, 9, 44, 48, 61, 65]; ii) the 3D network more capable of adapting to new domains thanks to RGB information associated with each point which allows learning better semantic features for the target domain, when this is available, using Unsupervised Domain Adaptation (UDA) approaches. Thus, we propose a simple architecture for multi-modal 3D semantic segmentation consisting of a 2D-3D architecture with each branch fed with both RGB and 3D information. Despite its simplicity, our proposal achieves state-of-the-art results in multi-modal UDA benchmarks, surpassing competitors by large margins, as well as significantly better domain generalization compared to a standard 2D-3D architecture [25]. Code available at https://github.com/CVLAB-Unibo/MM2D3D. Our contributions are:

- shining a light on the intrinsic complementarity of recent multi-modal 3D semantic segmentation networks based on 2D-3D branches;
- proposing a simple yet remarkably effective baseline that injects depth cues into the 2D branch and RGB colors into the 3D branch while preserving the complementarity of predictions;
- our network achieves state-of-the-art results in popular UDA benchmarks for multi-modal 3D semantic segmentation and surpasses standard 2D-3D architectures in domain generalization.

2. Related works

Point Cloud Semantic Segmentation. 3D data can be represented in several ways such as point clouds, voxels, and meshes, each with its pros and cons. Similarly to pixels in 2D, voxels represent 3D data as a discrete grid of the 3D space. This representation allows using convolutions as done for images. However, performing a convolution over the whole 3D space is memory intense, and it does not consider that many voxels are usually empty. Some 3D CNNs [45, 54] rely on OctTree [35] to reduce the memory footprint but without addressing the problem of manifold dilation. SparseConvNet [15] and similar implementations [11] address this problem by using hash tables to convolve only on active voxels, allowing the processing of high-resolution point clouds with only one point per voxel. Aside from cubic discretization, some approaches [75, 77] employ cylindrical voxels. Other methods address the problem with sparse point-voxel convolutions [53]. Differently, point-based networks process directly each point of a point cloud. PointNet++ [42] extract features from each point, and then extract global and local features by means of max-pooling in a hierarchical way. Many improvements have been proposed in this direction, such as continuous convolutions [55], deformable kernels [55] or lightweight alternatives [23]. In this work, we select SparseConvNet [15] as our 3D network as done by other works in the field [24, 39, 51, 68] since it is suitable for 3D semantic segmentation of large scenes.

Multi-Modal Learning. Exploiting multiple modalities to learn more robust and performant networks is a well-studied field in the literature [1, 38]. Among them, several approaches address the problem of semantic segmentation exploiting RGB and 3D structure information, either with the final goal of segmenting images, e.g., RGB-D networks [17, 58] or point clouds, e.g., LiDAR + RGB ap-

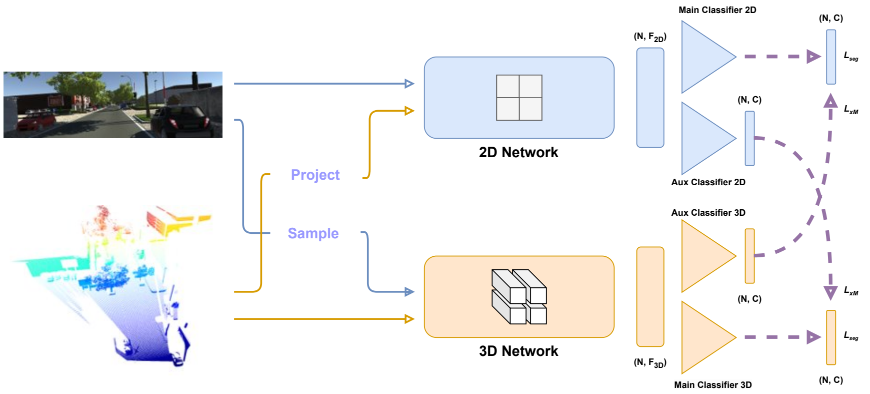

Figure 2. **Framework overview.** The RGB image and the sparse depth map obtained from the projection of the corresponding point cloud are fed to a custom 2D architecture to extract point-wise features. The same point cloud and sampled colors from the RGB image are given in input to the 3D Network. Then, two main classifiers output the main predictions to be used at test time. Moreover, two auxiliary classifiers are used at training time only to allow the exchange of information across branches.

proaches [16, 28, 68]. To speed-up research in this promising field, several datasets have been collected [3, 5, 13, 14] with 3D point clouds, images, and annotations for tasks such as 3D object detection or 3D semantic segmentation. Recently, some multi-modal methods [24, 39, 51, 68] show that a framework composed of a 2D and a 3D network can obtain very good performance in popular 3D segmentation benchmarks when averaging the scores coming from the two branches. This result is ascribed to the complementarity of the predictions due to the different modalities processed by each branch (either RGB or point clouds). In this paper, we analyze the improvement obtained by fusing the scores, and we argue that it mainly depends on the fact that the two networks extract complementary features because of the different receptive fields of the 2D and 3D networks. Based on this intuition, we propose a simple yet effective modification of the 2D-3D framework, that consists of providing both modalities as input to both branches.

Unsupervised Domain Adaptation. Unsupervised Domain Adaptation is the research field that investigates how to transfer knowledge learned from a source annotated domain to a target unlabelled domain [63]. In the last few years, several UDA approaches have been proposed for 2D semantic segmentation, using strategies such as style-transfer [8,10,19,27,31,37,41,67,71,73], adversarial training to learn domain-invariant representations [4, 20, 56, 57, 59, 62, 64, 69, 74] or self-training [7, 21, 22, 72, 78]. Recently, some works demonstrated the effectiveness of using depth information to boost UDA for 2D semantic segmentation [6, 9, 43, 44, 48, 61, 65]. In our work, we take inspiration from these findings, and we feed the projected point cloud in input to also to the 2D network, considering depth as a rich source of information robust to the domain shift. Recently some works address UDA also for semantic segmentation of point clouds [2, 26, 29, 40, 46, 49, 66, 70, 76]. Very recently, some works have addressed the challenging multi-modal 3D semantic segmentation task [24,25,39,51]. XMUDA [24] is the first work that focuses on UDA in the above setting, it defines a new benchmark and a baseline approach to adapt to a new target domain with an unsupervised cross-modal loss. [25] extend it, by proposing a more solid and comprehensive benchmark. DsCML [39] also extends XMUDA deploying adversarial training to align features across modalities and domains. In our work, we address the same multi-modal UDA scenarios introduced in [25], and we propose a simple yet effective architecture that is more robust to domain shift and can be adapted to new unlabelled target domains. Our framework, depicted in Fig. 2.

3. Method

Setup and Notation. We define input source samples $\{\boldsymbol{x}_s^{2D}, \boldsymbol{x}_s^{3D}\} \in \mathcal{S}$ and target samples $\{\boldsymbol{x}_t^{2D}, \boldsymbol{x}_t^{3D}\} \in \mathcal{T}$, with $\boldsymbol{x}^{2D}$ being the 2D RGB image and $\boldsymbol{x}^{3D}$ the corresponding point cloud, with 3D points in the camera reference frame. Note that $\boldsymbol{x}^{3D}$ contains only points visible from the RGB camera, assuming that the calibration of the two sensors is available for both domains and does not change over time. We assume the availability of annotations $\boldsymbol{y}_s^{3D}$ only for the source domain for each 3D point. When tackling the UDA scenario, we also have at our disposal the unlabeled samples from the target domain. Our goal is to obtain a point-wise prediction $N \times C$ for $\boldsymbol{x}_t^{3D}$, with N and C being the number of points of the target point cloud and the number of classes, respectively.

3.1. Base 2D/3D Architecture

We build our contributions upon the two independent branches (2D and 3D) architecture proposed in [25]. The

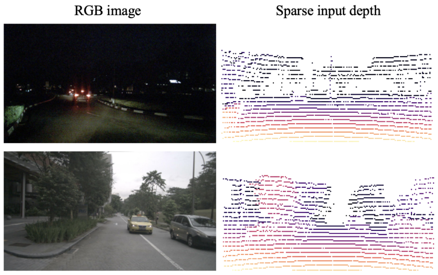

Figure 3. Depth comparison during daylight or night. Differently, from the RGB image (left column), a sparse depth map obtained by projecting a LiDAR scan into the image plane is not affected by the light conditions.

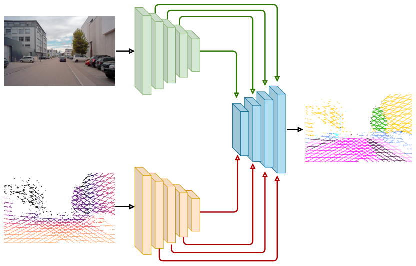

Figure 4. 2D Network of our framework. It is composed of a depth encoder and an RGB encoder to process the two inputs independently. The segmentation decoder leverages the multi-scale features of both encoders to predict semantic segmentation labels.

2D branch processes images to obtain a pixel-wise prediction given $\boldsymbol{x}^{2D}$ and it consists of a standard 2D U-Net [47]. On the other hand, the 3D branch takes in input point clouds to estimate the class of each point of $\boldsymbol{x}^{3D}$ and it is implemented as a 3D sparse convolutional network [15]. Thanks to the fact that 2D-3D correspondences are known, 3D points can be projected into the image plane to supervise the 2D branch, as supervision is provided only for the sparse 3D points. We denote the 3D semantic labels projected into 2D with the symbol $\boldsymbol{y}^{3D\to 2D}$. As argued by [25], such design choice allows one to take advantage of the strengths of each input modality, and final predictions can be obtained by averaging the outputs of the two branches to achieve an effective ensemble. In our work, we adopt the same framework, and we give an intuitive explanation of why this design choice is particularly effective. In particular, we reckon that the two predictions are complementary not only for the input signals being different but also for the fact the two branches focus on different things to determine their final predictions. Indeed, 3D convolutions produce features by looking at points that are close in the 3D space, while the 2D counterparts focus on neighboring pixels in the 2D image plane. Therefore, given corresponding 2D and 3D points, the two mechanisms implicitly produce features containing complementary information. In the right part of Fig. 1 we visualize the Effective Receptive Fields (ERF) [34] of a 2D U-Net with backbone ResNet34 [18] and of a 3D U-Net with backbone SparseConvNet [15]. It is worth highlighting that we do not focus on the theoretical yet on the effective receptive field, which is computed by analyzing the real contribution of each input point to the final prediction (the hotter the color intensity in the visualization, the larger the point contribution). Comparing the re-projected 2D ERF into 3D and the 3D ERF we can clearly appreciate that the 2D network focuses on sparse 3D regions, i.e., from the car to the building in the background, while the 3D counterpart reasons on a local 3D neighborhood (only car points). With this intuition in mind, we argue that by feeding the RGB signal to the 3D network, and the 3D information to the 2D backbone, we would still obtain complementary features that can be effectively fused together. Moreover, it is well-known that employing depth information as input to 2D segmentation networks can make it more robust to domain shift [6, 17]. At the same time, we posit that the 3D network with RGB information may be able to extract better semantic features. Differently from previous approaches that employ two independent architectures, based on the above considerations, we propose our multi-modal, two-branch framework named MM2D3D. In Sec. 3.2 we show how a point cloud can be used to obtain a stronger and more suitable input signal for the 2D network. Similarly, in Sec. 3.3 we describe our multi-modal 3D network.

3.2. Depth-based 2D Encoder

In this section, we focus on how we can use point clouds to make a 2D segmentation architecture more robust to domain shift. Inspired by [6, 17], we propose to use depth maps as an input signal that is less influenced by the domain gap. As we can observe from the two depth maps in Fig. 3, it is hard to understand which one was captured during day or night. At the same time, some objects such as the car can be distinguished by only looking at depths (bottom right of the second depth map). Thus, depth maps provide useful hints to solve the task of semantic segmentation. Given these considerations, we argue that exploiting such invariant information may alleviate the domain shifts and can be used to extract discriminative features for the segmentation task. At a first glance, injecting 3D cues into the 2D branch may seem redundant as the 3D network already has the capability to reason on the full 3D scene. However, given that the two networks have very different receptive fields, we can exploit such additional and useful information without the risk of hindering the complementarity of the two signal streams. Assuming point clouds expressed in the camera reference frame and the availability of the intrinsic camera matrix, we can project the original 3D point cloud to obtain a sparse depth map. In practice, the value of the z axis is assigned to the pixel coordinate (u, v) obtained by projecting a 3D point into the image plane. Similarly to [17], to process both inputs, we modify the 2D encoder of the 2D U-Net architecture by including an additional encoder to process the sparse depth maps obtained from the point cloud. As can be seen in Fig. 4, the two streams i.e. one for the RGB image and the other for the sparse depth map, are processed independently. Then, the concatenated depth and RGB features are processed by a decoder, composed of a series of transposed convolutions and convolutions in order to obtain semantic predictions of the same size as the input image. Moreover, features from layers of $\frac{1}{2}$ to $\frac{1}{16}$ of the input resolution are concatenated using skip connections with the corresponding layer of the decoder. This simple design choice allows semantic predictions to be conditioned also on the input depth signal, without altering the RGB encoder that provides useful classification features. Furthermore, without altering the RGB encoder, we can take advantage of a pre-trained architecture on ImageNet [12] as done by our competitors.

3.3. RGB Based 3D Network

In this work, we focus on the 3D convolutional network, SparseConvNet [15], as it can segment large scenes efficiently. In this network, the initial point cloud is first voxelized such that each 3D point is associated with only one voxel. Then, rather than processing the entire voxel grid, these models work with a sparse tensor representation ignoring empty voxels for the sake of efficiency. The network associates a feature vector to each voxel, and convolutions calculate their results based on these features. A standard choice for the voxel features is to simply assign to it a constant value, i.e., 1. Although these strategies have been shown to be effective [50, 68], the feature vector can be enriched to make it even more suitable for semantic segmentation. Based on our intuition of the different receptive fields, we can borrow information from the other modality to improve the performance of each branch, still preserving 2D-3D feature complementarity. Thus, we use RGB colors directly as features for each voxel of the SparseConvNet. Moreover, we design a simple yet effective strategy to let the 3D network decide whether to use or not this information. More specifically, the original RGB pixel values are fed to a linear layer that predicts a scalar value α to be multiplied by the color vector. For instance, learning this scaling could be useful in the UDA scenario, where we can train on unlabelled target samples, to discard RGB colors in case they do not provide any useful information, e.g., dark pixels in images acquired at night time.

3.4. Learning Scheme

Supervised Learning. Given the softmax predictions of the 2D and 3D networks, P_{2D} and P_{3D}, we supervise both branches using the cross-entropy loss on the source domain:

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

with $(\boldsymbol{x}_s, \boldsymbol{y}_s)$ being either $(\boldsymbol{x}_s^{\text{2D}}, \boldsymbol{y}_{\boldsymbol{s}}^{3D\to 2D})$ or $(\boldsymbol{x}_s^{\text{3D}}, \boldsymbol{y}_{\boldsymbol{s}}^{3D})$.

Cross-Branch Learning. To allow an exchange of information between the two branches, [25] and [39] add an auxiliary classification head to each one. The objective of these additional classifiers is to mimic the other branch output. The two auxiliary heads estimate the other modality output: 2D mimics 3D ($P_{\text{2D}\to\text{3D}}$) and 3D mimics 2D ($P_{\text{3D}\to\text{2D}}$). In practice, this is achieved with the following objective:

$$\begin{aligned} \mathcal{L}_{\text{xM}}(\boldsymbol{x}) &= \boldsymbol{D}_{\text{KL}}(\boldsymbol{P}_{\boldsymbol{x}}^{(n,c)} || \boldsymbol{Q}_{\boldsymbol{x}}^{(n,c)}) \\ &= -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} \boldsymbol{P}_{\boldsymbol{x}}^{(n,c)} \log \frac{\boldsymbol{P}_{\boldsymbol{x}}^{(n,c)}}{\boldsymbol{Q}_{\boldsymbol{x}}^{(n,c)}} \end{aligned} \tag{2}$$

with $(\boldsymbol{P}, \boldsymbol{Q}) \in \{(\boldsymbol{P}_{\text{2D}}, P_{\text{3D}\to\text{2D}}), (\boldsymbol{P}_{\text{3D}}, P_{\text{2D}\to\text{3D}})\}$ where $\boldsymbol{P}$ is the distribution from the main classification head which has to be estimated by $\boldsymbol{Q}$. Note that in Eq. (2), x can belong to either $\mathcal{T}$ or $\mathcal{S}$. This means that, in the UDA scenario, Eq. (2) can also be optimized for $\mathcal{T}$, forcing the two networks to have consistent behavior across the two modalities for the target domain as well without any labels.

Self-Training. Only in the UDA scenario, where unlabelled target samples are available, as done by [25], we perform one round of Self-Training [78] using pseudo-labels [30]. Specifically, after training the model with Eq. (1) for the source domain and Eq. (2) on both domains, we generate predictions on the unlabeled target domain dataset to be used as pseudo ground truths, $\hat{\boldsymbol{y}}_t$. Following [25], we filter out noisy pseudo-labels by considering only the most confident predictions for each class. Then, we retrain the framework from scratch the model minimizing the following objective function:

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{\text{seg}}(\boldsymbol{x}_s, \boldsymbol{y}_s) + \lambda_t \mathcal{L}_{\text{seg}}(\boldsymbol{x}_t, \hat{\boldsymbol{y}}_t) \\ + \lambda_{xs}\mathcal{L}_{\text{xM}}(\boldsymbol{x}_{\boldsymbol{s}}) + \lambda_{xt}\mathcal{L}_{\text{xM}}(\boldsymbol{x}_{\boldsymbol{t}}) \end{aligned} \tag{3}$$

4. Experiments

4.1. Datasets

To evaluate our method, we follow the benchmark introduced in [25] because it comprehends several interesting domain shift scenarios. The datasets used in the benchmark are nuScenes [5] A2D2 [14], SemanticKITTI [3], and VirtualKITTI [13] in which LiDAR point clouds and camera are synchronized and calibrated so that the projection between a 3D point and its corresponding 2D image pixel can always

Modality	Method	USA → Singapore			Day → Night			v.KITTI → Sem.KITTI			A2D2 → Sem.KITTI		
		2D	3D	Avg	2D	3D	Avg	2D	3D	Avg	2D	3D	Avg
	Baseline (Source only)	58.4	62.8	68.2	47.8	68.8	63.3	26.8	42.0	42.2	34.2	35.9	40.4
Uni-modal	MinEnt [60]	57.6	61.5	66.0	47.1	68.8	63.6	39.2	43.3	47.1	37.8	39.6	42.6
	Deep logCORAL [36]	64.4	63.2	69.4	47.7	68.7	63.7	41.4	36.8	47.0	35.1	41.0	42.2
	PL [32]	62.0	64.8	70.4	47.0	69.6	63.0	21.5	44.3	35.6	34.7	41.7	45.2
Multi-modal	xMUDA [24]	64.4	63.2	69.4	55.5	69.2	67.4	42.1	46.7	48.2	38.3	46.0	44.0
	DsCML* [39]	52.9	52.3	56.9	51.2	61.4	61.8	31.8	32.8	34.8	25.4	32.6	33.5
	MM2D3D (Ours)	**71.7**	**66.8**	**72.4**	**70.5**	**70.2**	**72.1**	**53.4**	**50.3**	**56.5**	**42.3**	**46.1**	**46.2**
	Oracle	75.4	76.0	79.6	61.5	69.8	69.2	66.3	78.4	80.1	59.3	71.9	73.6

Table 1. **Results for UDA 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 trained by us using official code.

be computed. It is important to note that only 3D points visible from the camera are used for both training and testing. NuScenes consists of 1000 driving scenes in total, each of 20 seconds, with 40k annotated point-wise frames taken at 2Hz, and it is deployed to implement two adaptation scenarios: day-to-night and country-to-country. The former exhibits severe light changes between the source and the target domain, while the latter covers changes in the scene layout. In both settings adaptation is performed on six classes: *vehicle, driveable_surface, sidewalk, terrain, manmade, vegetation*. The third challenging benchmark foresees adaptation from synthetic to real data, and it is implemented by adapting from VirtualKITTI to SemanticKITTI. Since VirtualKITTI only provides depth maps, we use the same simulated LiDAR scans from our competitor [25] for a fair comparison. Note also that to accommodate for the different classes in the two datasets, a class mapping is required and we use the same defined in [25]. The last adaptation scenario involves A2D2 and SemanticKITTI. The A2D2 dataset is composed of 20 drives, with a total of 28,637 frames. As the LiDARs sensor is very sparse (16 layers), all three front LiDARs are used. All frames of all sequences are used for training, except for the sequence 20180807_145028 which is left out for testing. The SemanticKITTI dataset features a large-angle front camera and a 64-layer LiDAR. Scenes from 0, 1, 2, 3, 4, 5, 6, 9, 10 are used for training, scene 7 as validation, and 8 as a test set. In this case, only the ten classes that are in common along the two datasets are used: *car, truck, bike, person, road, parking, sidewalk, building, nature, other-objects*.

4.2. Implementation details

We use the same data augmentation pipeline as our competitors, which is composed of random horizontal flipping and color jittering for 2D images, while vertical axis flipping, random scaling, and random 3D rotations are used for the 3D scans. It is important to note that augmentations are done independently for each branch. We implement our framework in PyTorch using two NVIDIA 3090 GPU with 24GB of RAM. We train with a batch size of 16, alternating batches of source and target domain for the UDA case and source only in DG. The smaller dataset is repeated to match the length of the other. We rely on the AdamW optimizer [33] and the One Cycle Policy as a learning rate scheduler [52]. We train for 50, 35, 15, and 30 epochs for USA → Singapore, Day → Night, v. KITTI → Sem. KITTI, and A2D2 → Sem. KITTI respectively. As regards the hyper-parameters, we follow [25] and set $\lambda_s = 0.8, \lambda_t = 0.1, \lambda_{xs} = 0.1, \lambda_{xt} = 0.01$ in all settings without performing any fine-tuning on these values.

4.3. UDA results

Following previous works in the field [25, 39], we evaluate the performance of a model on the target test set using the standard Intersection over Union (IoU) and select the best checkpoint according to a small validation set on the target domain. In Tab. 1, we report our results on the four challenging UDA benchmarks explained in Sec. 4.1. For each experiment, we report two reference methods: a model trained only on the source domain, named *Baseline (Source Only)*; a model trained only on the target data using annotations, representing the upper bound than can be obtained with real ground-truth, namely *Oracle*. We note that these two models employ the two independent stream architecture of [25]. In the columns *Avg*, we report the results obtained by the mean of the 2D and 3D outputs after softmax which is the final output of our multi-modal framework. For the sake of completeness, we also report the results of each individual branch (2D and 3D only). We compare our method with both Uni-modal and Multi-Modal approaches. In particular, we mainly focus on a comparison with xMUDA [25] and DsCML [39], as they are the current s.o.t.a. methods for UDA in our multi-modal setting. In particular, for the latter, we use the official code provided by the authors [1] to retrain the model on the new more exhaustive benchmark defined by [25]. Overall, we note how our contributions largely improve results over competitors across all settings and modalities. In USA → Singapore, we observe a large boost in both branches, and on average we report a +3%

[1] https://github.com/leolyj/DsCML

Method	USA → Singapore			Day → Night			v.KITTI → Sem.KITTI			A2D2 → Sem.KITTI		
	2D	3D	Avg	2D	3D	Avg	2D	3D	Avg	2D	3D	Avg
xMUDA* [24]	58.7	**62.3**	68.6	43.0	**68.9**	59.6	25.7	37.4	39.0	34.9	**36.7**	41.6
MM2D3D (Ours)	**69.7**	**62.3**	**70.9**	**65.3**	63.2	**68.3**	**37.7**	**40.2**	**44.2**	**39.6**	35.9	**43.6**

Table 2. **Results for 3D for semantic segmentation in the Domain Generalization setting.** 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 trained by us using official code.

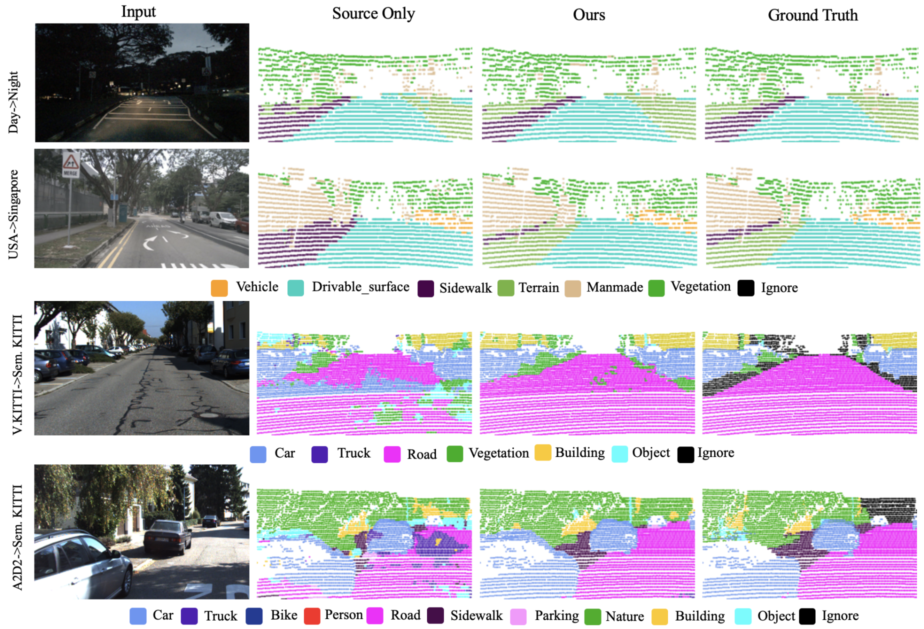

Figure 5. **Qualitative examples of the proposed framework in the UDA scenario.** From left to right: RGB images, point cloud segmentations projected into 2D for visualization purpose of the baseline source only model, our method, and the ground truth respectively. From top to bottom: the four different adaptation scenarios. Comparisons are provided for the target domain.

(third row of the Multi-modal section). The large improvement (+7.3%) for the 2D model, suggests that the depth cues injected into a common 2D decoder can be quite useful even if the light conditions are similar across domains. In Day → Night, we observe a remarkable +15% for the 2D branch, which in turn rises the average score to +4.7% when compared with the previous best model. We attribute this boost in performance to the depth encoder, which is able to provide useful hints when the RBG encoder has to deal with large changes in light conditions. Remarkably, our network surpasses even the performance of the two independent streams *Oracle*. Indeed, as discussed in Sec. 3.2, the sparse depth is able to give useful details for the task of semantic segmentation. Moreover, thanks to the fact that the cross-modal loss Sec. 3.4 is optimized for both domains, the network lean to use both encoders to make the final predictions, leading to more robust performance when the encoder receives a less informative RGB signal. In the challenging synthetic-to-real case (v. KITTI → Sem. KITTI), we also notice consistent improvements in both branches. We highlight that even though RGB colors are here likely the main source of the domain gap, they are still useful to obtain a stronger 3D model (+3.6%). In the A2D2 → Sem. KITTI setting, where the sensors setup is different, we still benefit from the depth hints provided to the 2D network, and on average, our method surpasses by 2.2% xMUDA. In general, we highlight that though we employed both modalities in the 2D and 3D branches, the Avg performances are better than those of each individual branch, supporting our core intuition. In Fig. 5, we report some qualitative results obtained with our framework.

4.4. Domain Generalization results

In this section, we test our contributions in the Domain Generalization setting, in which the target data cannot be used at training time. For this study we consider XMUDA [25] as our baseline two-branch 2D-3D method, and we show that our simple contribution can boost generalization performances. Results are reported in Tab. 2. To implement this experiment we keep the same hyper-parameters as used in the UDA scenario. We retrain [24] using the official code, but without the target data. Also in this set-

Method	Depth	RGB	USA → Singapore			Day → Night			v.KITTI → Sem.KITTI			A2D2 → Sem.KITTI		
			2D	3D	Avg	2D	3D	Avg	2D	3D	Avg	2D	3D	Avg
xMUDA [25]			64.4	63.2	69.4	55.5	69.2	67.4	42.1	46.7	48.2	38.3	46.0	44.0
MM2D3D (Ours)	✓		69.5	64.0	69.6	**71.3**	69.9	**72.8**	52.6	40.3	53.7	41.7	44.8	45.9
MM2D3D (Ours)	✓	✓	**71.7**	**66.8**	**72.4**	70.5	**70.2**	72.1	**53.4**	**50.3**	**56.5**	**42.3**	**46.1**	**46.2**

Table 3. **Modality-wise ablation of the proposed framework in the UDA scenario**. *Depth* indicates the usage of the additional sparse depth encoder, while *RGB* denotes the introduction of the RGB information in the 3D network.

Method	USA → Singapore			Day → Night			v.KITTI → Sem.KITTI			A2D2 → Sem.KITTI		
	2D	3D	Avg	2D	3D	Avg	2D	3D	Avg	2D	3D	Avg
MM2D3D (Ours)	71.7	66.8	72.4	70.5	**70.2**	72.1	53.4	50.3	56.5	42.3	46.1	46.2
xMUDA [25] + PL	67.0	65.4	71.2	57.6	69.6	64.4	45.8	51.4	52.0	41.2	**49.8**	47.5
MM2D3D (Ours) + PL	**74.3**	**68.3**	**74.9**	**71.3**	69.6	**72.2**	**55.4**	**55.0**	59.7	**46.4**	48.7	**50.7**
MM2D3D (Ours) + Fusion	x	x	74.0	x	x	71.0	x	x	**60.4**	x	x	48.8

Table 4. **Self-training Analysis.** Results with different self-training strategies in the UDA scenario.

ting, we observe overall large improvements. We believe that this can be ascribed especially to the introduction of the depth encoder, which helps to achieve a better generalization. Evidence of this is well observable in the Day → Night, where the 2D performance increases from 43% to 65.3% in terms of mIoU, but also for USA → Singapore and (v. KITTI → Sem. KITTI), where we achieve +11% and +12 respectively. In the Day → Night scenario, the 3D branch experience a drop in performance. We think that it is related to the large domain shift of RGB images. Differently from the adaptation scenario in which we can train directly on the unlabeled target data to counteract this problem, in the generalization scenario, it influences badly the 3D performance. However, we note that our final Avg prediction still outperforms xMUDA.

4.5. Ablation Studies

Modality-wise analysis. In Tab. 3, we ablate our contributions starting from the model proposed by [24] in the UDA scenario. We start by activating our depth-based network, introduced in Sec. 3.3. The performance boost given by our proposal is remarkable across all settings. In cases such as Day → Night, where the RGB gap is larger, the depth cues injected with skip connections to the semantic decoder greatly enhance performances in the target domain (+15.8% for 2D and +5.4% in "Avg). We note also a consistent improvement for the remaining settings, in particular, we highlight a +10.5% for the 2D scores on the challenging synthetic-to-real adaptation benchmark (v. KITTI → Sem. KITTI). Furthermore, when feeding RGB colors to the 3D network (last row of Tab. 3), we observe improved performances in almost all settings. The largest improvement is oberved in the synthetic-to-real setting, where we achieve a +10% in terms of mIou for the 3D, which in turn increased the average score from 53.7% to 56.5%. Better performance is also achieved for both the 3D network and the average score for A2D2 → Sem. KITTI.

Self-Training. In this section, we compare different self-training strategies and report results in Tab. 4. As explained in Sec. 3.4, for the self-training protocol we first need a model trained on the source domain to produce the pseudo-labels for the target domain in the second round. We report in the first row of Tab. 4 the performance of this starting model to better appreciate the effectiveness of self-training. First, we note how thanks to our contributions, for USA → Singapore, Day → Night, and v. KITTI → Sem. KITTI we already surpass xMUDA [24] on the *Avg* column even without the usage of pseudo-labels. When pseudo-labels from the 2D and the 3D branches are used to supervise the 2D and the 3D network respectively, we establish new state-of-the-art performances for all four settings in the average predictions (third row). Furthermore, in the fourth row of Tab. 4, we deploy the strategy proposed in [24], where point-wise features from the two networks are concatenated and used to train a unique classifier). In this case, we observe mixed results, indicating that this self-training strategy is not necessarily better across all settings when compared to the standard self-training protocol.

5. Conclusions

In this paper, we shed light on the complementarity of recent and emerging 3D-2D architectures for 3D semantic segmentation. We provide an intuitive explanation based on the notion of effective receptive field of why processing data with these two networks grants orthogonal predictions that can be effectively fused together. Based on this, we propose to feed both modalities to both branches. Despite the simplicity of our approach, we establish new state-of-the-art results in four common UDA scenarios and demonstrate superior generalization performance over the baseline 2D-3D architecture. A limitation of our work is that our method is purely multi-modal, and it requires both modalities and a valid calibration across sensors at test time. An interesting future direction is to investigate how our approach may generalize to other multi-modal 2D-3D architectures for semantic segmentation.

References

[1] Cem Akkus, Luyang Chu, Vladana Djakovic, Steffen Jauch-Walser, Philipp Koch, Giacomo Loss, Christopher Marquardt, Marco Moldovan, Nadja Sauter, Maximilian Schneider, Rickmer Schulte, Karol Urbanczyk, Jann Goschenhofer, Christian Heumann, Rasmus Hvingelby, Daniel Schalk, and Matthias Aßenmacher. Multimodal deep learning, 2023. 2

[2] Inigo Alonso, Luis Riazuelo Montesano, Ana C Murillo, et al. Domain adaptation in lidar semantic segmentation by aligning class distributions. arXiv preprint arXiv:2010.12239, 2020. 3

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019. 3, 5

[4] Matteo Biasetton, Umberto Michieli, Gianluca Agresti, and Pietro Zanuttigh. Unsupervised domain adaptation for semantic segmentation of urban scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2019. 3

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 11621–11631, 2020. 3, 5

[6] Adriano Cardace, Luca De Luigi, Pierluigi Zama Ramirez, Samuele Salti, and Luigi Di Stefano. Plugging self-supervised monocular depth into unsupervised domain adaptation for semantic segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 1129–1139, 2022. 2, 3, 4

[7] Adriano Cardace, Pierluigi Zama Ramirez, Samuele Salti, and Luigi Di Stefano. Shallow features guide unsupervised domain adaptation for semantic segmentation at class boundaries. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 1160–1170, 2022. 3

[8] Wei-Lun Chang, Hui-Po Wang, Wen-Hsiao Peng, and Wei-Chen Chiu. All about structure: Adapting structural information across domains for boosting semantic segmentation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2019. 3

[9] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1841–1850, 2019. 2, 3

[10] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Oct 2019. 3

[11] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 3075–3084, 2019. 2

[12] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 2009. 5

[13] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtualworlds as proxy for multi-object tracking analysis. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2016. 3, 5

[14] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, et al. A2d2: Audi autonomous driving dataset. arXiv preprint arXiv:2004.06320, 2020. 3, 5

[15] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun 2018. 2, 4, 5

[16] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. IEEE transactions on pattern analysis and machine intelligence, 43(12):4338–4364, 2020. 3

[17] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part I 13, pages 213–228. Springer, 2017. 2, 4, 5

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2016. 2, 4

[19] Judy Hoffman, E. Tzeng, T. Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In ICML, 2018. 3

[20] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation, 2016. 3

[21] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9924–9935, 2022. 3

[22] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. arXiv preprint arXiv:2204.13132, 2022. 3

[23] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 11108–11117, 2020. 2

[24] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Emilie Wirbel, and Patrick Pérez. xMUDA: Cross-modal unsupervised domain adaptation for 3D semantic segmentation. In CVPR, 2020. 2, 3, 6, 7, 8

[25] Maximilian Jaritz, Tuan-Hung Vu, Raoul De Charette, Émilie Wirbel, and Patrick Pérez. Cross-modal learning for domain adaptation in 3d semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(2):1533–1544, 2022. 2, 3, 4, 5, 6, 7, 8

[26] Peng Jiang and Srikanth Saripalli. Lidarnet: A boundary-aware domain adaptation model for point cloud semantic segmentation. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 2457–2464. IEEE, 2021. 3

[27] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2020. 3

[28] Georg Krispel, Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Fuseseg: Lidar point cloud segmentation fusing multi-modal data. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 1874–1883, 2020. 3

[29] Ferdinand Langer, Andres Milioto, Alexandre Haag, Jens Behley, and Cyrill Stachniss. Domain transfer for semantic segmentation of lidar data using deep neural networks. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 8263–8270. IEEE, 2020. 3

[30] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. ICML 2013 Workshop : Challenges in Representation Learning (WREPL), 07 2013. 5

[31] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2019. 3

[32] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In CVPR, 2019. 6

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In International Conference on Learning Representations, 2019. 6

[34] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. Advances in neural information processing systems, 29, 2016. 1, 2, 4

[35] Donald Meagher. Geometric modeling using octree encoding. Computer graphics and image processing, 19(2):129–147, 1982. 2

[36] Pietro Morerio, Jacopo Cavazza, and Vittorio Murino. Minimal-entropy correlation alignment for unsupervised deep domain adaptation. In International Conference on Learning Representations, 2018. 6

[37] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun 2018. 3

[38] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In Proceedings of the 28th international conference on machine learning (ICML-11), pages 689–696, 2011. 2

[39] Duo Peng, Yinjie Lei, Wen Li, Pingping Zhang, and Yulan Guo. Sparse-to-dense feature matching: Intra and inter domain cross-modal learning in domain adaptation for 3d semantic segmentation. In Proceedings of the International Conference on Computer Vision (ICCV). IEEE, 2021. 2, 3, 5, 6

[40] Florian Piewak, Peter Pinggera, and Marius Zöllner. Analyzing the cross-sensor portability of neural network architectures for lidar-based semantic labeling. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pages 3419–3426. IEEE, 2019. 3

[41] Fabio Pizzati, Raoul de Charette, Michela Zaccaria, and Pietro Cerri. Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), March 2020. 3

[42] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems, 30, 2017. 2

[43] Pierluigi Zama Ramirez, Adriano Cardace, Luca De Luigi, Alessio Tonioni, Samuele Salti, and Luigi Di Stefano. Learning good features to transfer across tasks and domains. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023. 3

[44] Pierluigi Zama Ramirez, Alessio Tonioni, Samuele Salti, and Luigi Di Stefano. Learning across tasks and domains. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Oct 2019. 2, 3

[45] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3577–3586, 2017. 2

[46] Christoph B Rist, Markus Enzweiler, and Dariu M Gavrila. Cross-sensor deep domain adaptation for lidar detection and segmentation. In 2019 IEEE Intelligent Vehicles Symposium (IV), pages 1535–1542. IEEE, 2019. 3

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, page 234–241, 2015. 4

[48] Suman Saha, Anton Obukhov, Danda Pani Paudel, Menelaos Kanakis, Yuhua Chen, Stamatios Georgoulis, and Luc Van Gool. Learning to relate depth and semantics for unsupervised domain adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8197–8207, 2021. 2, 3

[49] Khaled Saleh, Ahmed Abobakr, Mohammed Attia, Julie Iskander, Darius Nahavandi, Mohammed Hossny, and Saeid Nahvandi. Domain adaptation for vehicle detection from bird's eye view lidar point cloud data. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pages 0–0, 2019. 3

[50] Cristiano Saltori, Fabio Galasso, Giuseppe Fiameni, Nicu Sebe, Elisa Ricci, and Fabio Poiesi. Cosmix: Compositional semantic mix for domain adaptation in 3d lidar segmentation. In European Conference on Computer Vision, pages 586–602. Springer, 2022. 5

[51] Inkyu Shin, Yi-Hsuan Tsai, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Sparsh Garg, In So Kweon, and Kuk-Jin Yoon. Mm-tta: multi-modal test-time adaptation for 3d semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16928–16937, 2022. 2, 3

[52] Leslie N. Smith and Nicholay Topin. Super-convergence: very fast training of neural networks using large learning rates. Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, May 2019. 6

[53] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In European conference on computer vision, pages 685–702. Springer, 2020. 2

[54] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In Proceedings of the IEEE international conference on computer vision, pages 2088–2096, 2017. 2

[55] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF international conference on computer vision, pages 6411–6420, 2019. 2

[56] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun 2018. 3

[57] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. Domain adaptation for structured output via discriminative patch representations. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Oct 2019. 3

[58] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. International Journal of Computer Vision, 128(5):1239–1285, 2020. 2

[59] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Perez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2019. 3

[60] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In CVPR, 2019. 6

[61] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Perez Perez. Dada: Depth-aware domain adaptation in semantic segmentation. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Oct 2019. 2, 3

[62] Haoran Wang, Tong Shen, Wei Zhang, Lingyu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In The European Conference on Computer Vision (ECCV), August 2020. 3

[63] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. Neurocomputing, 312:135–153, 2018. 3

[64] Zhonghao Wang, Mo Yu, Yunchao Wei, Rogerio Feris, Jinjun Xiong, Wen-mei Hwu, Thomas S. Huang, and Honghui Shi. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2020. 3

[65] Kohei Watanabe, Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Multichannel semantic segmentation with unsupervised domain adaptation. In Proceedings of the European Conference on Computer Vision (ECCV), pages 0–0, 2018. 2, 3

[66] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In 2019 International Conference on Robotics and Automation (ICRA), pages 4376–4382. IEEE, 2019. 3

[67] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gökhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S. Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. Lecture Notes in Computer Science, page 535–552, 2018. 3

[68] Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2dpass: 2d priors assisted semantic segmentation on lidar point clouds. In European Conference on Computer Vision, pages 677–695. Springer, 2022. 2, 3, 5

[69] Jihan Yang, Ruijia Xu, Ruiyu Li, Xiaojuan Qi, Xiaoyong Shen, Guanbin Li, and Liang Lin. An adversarial perturbation oriented domain adaptation approach for semantic segmentation. Proceedings of the AAAI Conference on Artificial Intelligence, 34(07):12613–12620, Apr 2020. 3

[70] Li Yi, Boqing Gong, and Thomas Funkhouser. Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 15363–15373, 2021. 3

[71] Pierluigi Zama Ramirez, Alessio Tonioni, and Luigi Di Stefano. Exploiting semantics in adversarial training for image-level domain adaptation. In 2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS), pages 49–54, 2018. 3

[72] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. arXiv preprint arXiv:2101.10979, 2021. 3

[73] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun 2018. 3

[74] Y. Zhang and Zilei Wang. Joint adversarial learning for domain adaptation in semantic segmentation. In AAAI, 2020. 3

[75] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9601–9610, 2020. 2

[76] Sicheng Zhao, Yezhen Wang, Bo Li, Bichen Wu, Yang Gao, Pengfei Xu, Trevor Darrell, and Kurt Keutzer. epointda: An end-to-end simulation-to-real domain adaptation framework for lidar point cloud segmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 3500–3509, 2021. 3

[77] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9939–9948, 2021. 2

[78] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In Proceedings of the European Conference on Computer Vision (ECCV), pages 289–305, 2018. 3, 5