SELMA: SEmantic Large-Scale Multimodal Acquisitions in Variable Weather, Daytime and Viewpoints

Paolo Testolina, *Student Member, IEEE*, Francesco Barbato, *Student Member, IEEE*, Umberto Michieli, *Graduate Student Member, IEEE*, Marco Giordani, *Member, IEEE*, Pietro Zanuttigh, *Member, IEEE*, and Michele Zorzi, *Fellow, IEEE*

***Abstract*— Accurate scene understanding from multiple sensors mounted on cars is a key requirement for autonomous driving systems. Nowadays, this task is mainly performed through data-hungry deep learning techniques that need very large amounts of data to be trained. Due to the high cost of performing segmentation labeling, many synthetic datasets have been proposed. However, most of them miss the multi-sensor nature of the data, and do not capture the significant changes introduced by the variation of daytime and weather conditions. To fill these gaps, we introduce SELMA, a novel synthetic dataset for semantic segmentation that contains more than 30K unique waypoints acquired from 24 different sensors including RGB, depth, semantic cameras and LiDARs, in 27 different weather and daytime conditions, for a total of more than 20M samples. SELMA is based on CARLA, an open-source simulator for generating synthetic data in autonomous driving scenarios, that we modified to increase the variability and the diversity in the scenes and class sets, and to align it with other benchmark datasets. As shown by the experimental evaluation, SELMA allows the efficient training of standard and multi-modal deep learning architectures, and achieves remarkable results on real-world data. SELMA is free and publicly available, thus supporting open science and research.**

***Index Terms*— Synthetic dataset, CARLA, autonomous driving, domain adaptation, semantic segmentation, sensor fusion.**

I. Introduction

RECENT advances in the automotive sector have paved the way toward Connected Intelligent Transportation Systems (C-ITSs) to achieve safer and more efficient driving. Not only can C-ITSs reduce the number of traffic accidents (up to 90%, according to some estimates [1]) or improve traffic management via smart platooning, cruise control and/or traffic light coordination, but it also holds the promise to improve fuel economy and contribute to a 60% fall in carbon emissions [2]. Overall, C-ITSs represent a huge market of more than 7 trillion USD [3], hence stimulating significant research efforts.

To these goals, future connected vehicles will be equipped with heterogeneous sensors, including Light Detection and Ranging (LiDAR) and RGB camera sensors, able to provide an accurate perception of the environment. In particular, LiDARs generate a 3D omnidirectional representation of the environment in the form of a point cloud, and stand out as the most accurate sensors for geometry acquisition under several weather and lighting conditions [4]. On the other side, RGB cameras offer advantages like cheaper price, higher resolution and higher frame rate than LiDARs, even though they suffer from severe sensitivity to illumination and visibility conditions [5]. In this sense, sensor fusion appears as a promising solution to provide more robust scene understanding, at the expense of the additional processing overhead for collecting and combining observations from multiple sensors [6].

Autonomous driving tasks, in particular semantic segmentation (SS) and Vehicle-to-Everything (V2X) communication, raise several challenges [7], [8], also in view of the complex and dynamic environment in which autonomous vehicles move and operate. In these regards, machine learning (ML) and deep learning (DL) represent valuable tools to address these issues and optimize driving decisions [9]. However, these techniques require the availability of massive amounts of labeled data for proper training, whose acquisition and labeling is extremely expensive and time consuming. Hence, existing open-source datasets, like Waymo [10], Cityscapes [11], and KITTI [12], are scarce and generally lack diversity. Moreover, many datasets are too small to capture the many challenges of the urban scenario, do not encompass multiple (and diverse) sensors, and come with unlabeled scenes, undermining the training of ML models [13].

To fill these gaps, the scientific community has been investigating the usage of synthetic (computer-generated) datasets, where the full control of the data generation pipeline is delegated to simulations, hence ensuring lower costs, greater flexibility, better repeatability, and larger amount of samples than real-world data [14], [15], [16], [17], [18]. For example,

Manuscript received 20 April 2022; revised 13 December 2022 and 7 February 2023; accepted 3 March 2023. Date of publication 28 March 2023; date of current version 7 July 2023. This work was supported in part by the European Union through the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (Program "RESTART") under Grant PE0000001. The work of Paolo Testolina was supported by the Fondazione Cariparo under Grant "Dottorati di Ricerca" 2019. The work of Francesco Barbato, Umberto Michieli, and Pietro Zanuttigh was supported in part by the University of Padova under the SID project "Semantic Segmentation in the Wild." The Associate Editor for this article was L. M. Bergasa. *(Paolo Testolina and Francesco Barbato are co-first authors.) (Corresponding author: Paolo Testolina.)*

The authors are with the Department of Information Engineering, University of Padova, 35131 Padua, Italy (e-mail: testolina@dei.unipd.it).

This article has supplementary downloadable material available at https://doi.org/10.1109/TITS.2023.3257086, provided by the authors.

Digital Object Identifier 10.1109/TITS.2023.3257086

annotation and quality control on a single real-world image may require more than 1.5 hours of work according to [11], thus large-scale real-world datasets would involve huge investments for labeling. Notably, simulations facilitate data acquisition in different conditions and scenarios, and consideration of diverse sets of sensors. An open-source simulator to generate synthetic data is CAR Learning to Act (CARLA) [19], which includes urban layouts, a wide range of environmental conditions, vehicles, buildings and pedestrians models, and supports a flexible setup of sensors. At the time of writing, several synthetic datasets exist for SS in autonomous driving [14], [20], [21], [22], [23], [24], [25]. These datasets, however, present limitations. In particular, samples are generally captured in a limited number of settings, in similar viewpoints, weather, lighting, and daytime conditions, and often from a single sensor. Moreover, they do not provide end-users with fine-grained control over the weather setup or the same semantic class set as common benchmarks, like Cityscapes [11].

To overcome these limitations, in this paper we present SEmantic Large-scale Multimodal Acquisitions in Variable Weather, Daytime and Viewpoints (SELMA), a new multimodal synthetic dataset for autonomous driving, built using a modified version of the CARLA simulator. Our dataset stands out as one of the largest, most complete and diverse datasets to ensure adequate design, prototyping, and validation of autonomous driving models, in particular to solve complex tasks like SS. Specifically, the SELMA dataset consists of:

- Data acquired in 30 909 independent locations from 7 RGB cameras, 7 depth cameras, 7 semantic cameras, and 3 LiDARs coupled with semantic information. The multimodal setup of SELMA promotes complementary and diversity of data, and permits higher accuracy and performance of learning tasks [26].
- Acquisitions generated in variable weather, daytime and viewpoint conditions, and across 8 maps, for a total of 216 unique settings. To this aim, the CARLA simulator has been modified to increase the photo-realism of the weather conditions, and to maximize the visual variability, e.g., by adding parked bikes or traffic lights and signs.
- Semantic labeling for both camera and LiDAR data into 36 distinct classes, with complete overlap with the training set of common benchmarks like Cityscapes [11], obtained by modifying the source code of the simulator.

We validate the accuracy and realism of our dataset starting from a set of baseline experiments, and show that DL models for semantic understanding trained on our dataset outperform the same models trained on competing synthetic datasets when tested on a real (i.e., non-simulated) domain.

The dataset is freely available for download,[1] thus supporting open science and stimulating further research in the field of autonomous driving.

The remainder of this paper is organized as follows. In Sec. II we describe the existing real and synthetic datasets related to our work. Sec. III presents the CARLA simulator and the additions we introduced to acquire the data. The SELMA dataset is then described in detail in Sec. IV, while Secs. V and VI show some numerical results validating the accuracy of models trained on our dataset. Finally, in Sec. VII we provide the conclusions and some future research directions.

[1]The SELMA dataset is available at https://scanlab.dei.unipd.it/app/dataset

TABLE I

COMPARISON AMONG THE MOST POPULAR SS DATASETS. WHITE ROWS REFER TO REAL DATASETS, DARK GREY REFERS TO SYNTHETIC DATASETS, AND LIGHT GREY REFERS TO A COMBINATION OF THE TWO. BEST IN **BOLD**, RUNNER-UP UNDERLINED. T: TYPE, R: REAL, M: MIXED, S: SYNTHETIC, BB: BOUNDING BOXES, W: WEATHERS, ToD: DAYTIME, AA: ANTI-ALIASING, †: ESTIMATED DEPTH, *: RANDOM, ‡: FISHEYE CAMERA

Name	T	Cams	Depth	LiDAR	BB	Labels RGB	Labels LiDAR	SS/BB CS all	SS/BB CS train	SS/BB CS classes	W	ToD	AA	Positions
A2D2 [30]	R	6	-	**5**	✓	1	1	38	38	12	-	-	-	41280
ACDC [31]	R	1	-	✗	✗	1	-	19	19	**19**	**3**	2	-	4006
Apolloscape [32]	R	6	2†	2	✓	1	1	36	22	12	*	*	-	143906
Argo [33]	R	**9**	2†	2	✓	-	-	15	15	5	-	-	-	N/A
BDD 100k [34]	R	1	✗	✗	✓	1	-	40	19	**19**	*	*	-	10000
CamVID [35]	R	1	✗	✗	✓	1	-	32	11	11	-	-	-	701
Cityscapes [11]	R	2	2†	✗	✓	1	-	35	19	**19**	-	-	-	25000
DADA-seg [36]	R	2	✗	✗	✗	✗	-	19	19	**19**	-	-	-	N/A
DarkZurich [37]	R	1	✗	✗	✗	✗	-	19	19	**19**	-	**3**	-	8779
DRIV100 [38]	R	1	✗	✗	✗	✗	-	19	19	**19**	-	-	-	N/A
IDD [39]	R	1	✗	✗	✗	1	-	34	25	**19**	-	-	-	10003
KITTI [40]	R	2	4†	1	✓	✗	✗	8	8	8	-	-	-	N/A
KITTI-360 [41]	R	2‡+1	✗	1	✓	✗	✗	37	19	**19**	-	-	-	N/A
Mapillary [42]	R	1	✗	✗	✓	1	-	**66**	**66**	**19**	*	*	-	25000
NTHU [43]	R	1	✗	✗	✗	✗	-	13	13	13	-	-	-	12800
Nuscenes [44]	R	6	✗	1	✓	✗	1	23	23	-	-	-	-	40000
RainCouver [45]	R	1	✗	✗	✗	✗	-	3	3	-	1	**3**	-	N/A
SemanticKITTI [46]	R	-	-	1	✗	-	1	28	20	15	-	-	-	43552
Nightcity [47]	R	1	✗	✗	✗	1	-	19	19	**19**	-	1	-	4297
WoodScape [48]	R	4‡	4†,‡	1	✓	4	-	40	10	12	-	-	-	N/A
CS Fog [49]	M	2	2†	✗	✓	1	-	35	19	**19**	1	-	-	5000
CS Rain [50]	M	2	2†	✗	✓	1	-	34	18	18	1	-	-	5000
EventScape [51]	S	1	1	✗	✗	1	-	24	16	16	-	-	-	N/A
GTA5 [20]	S	1	✗	✗	✗	1	-	35	19	**19**	-	*	✓	24966
IDDA [23]	S	1	1	✗	✗	1	-	24	24	16	**3**	-	✗	16000
SHIFT [24]	S	6	6	1	✓	6	-	23	23	16	-	5	✗	N/A
SynPASS [25]	S	6	✗	✗	✗	6	-	22	19	13	4	2	✗	9080
SYNTHIA [21]	S	1	1	✗	✗	1	-	23	16	16	*	*	✗	9400
SynWoodScape [52]	S	4‡	4‡	1	✓	4	1	25	23	12	**9**	2	✓	155
SELMA (Ours)	S	7	7	3	✓	7	**3**	36	19	**19**	**9**	**3**	✓	30909

II. RELATED WORK

The development of DL architectures seen in recent years goes along with the design of extensive datasets, needed for their optimization. One computer vision (CV) task where such advancements have been particularly significant is scene understanding, which evolved in several sub-tasks, each requiring appropriate data for training. Among them, three tasks are worth mentioning, given the strong push they provided to datasets design: image classification, object detection and semantic segmentation. The first sparked the generation of widely used datasets, e.g., ImageNet [27]. The second and third tasks have been widely applied to many problems, and especially to autonomous driving systems. Here, vehicles require accurate recognition of the surrounding environment to appropriately plan driving actions. This translated into a wide range of real and synthetic datasets to support the training of autonomous driving applications [28], [29]. In this work, we focus on the semantic segmentation task, generally recognized as the most challenging of the three. The most popular SS datasets existing in the literature and their characteristics are reported in Table I.

A. Real Datasets

Given the high complexity and cost of labeling, most wide-scale real datasets tend not to provide the ground truth,

e.g., SS labels or bounding boxes, thus limiting their use in tasks like semantic segmentation and object detection. One of the first works to introduce labeled SS samples in the context of autonomous driving was CamVid [35], which consists of over 700 images labeled in 32 classes. Based on this, a huge effort was made by the creators of KITTI [12], [40], [53], [54] to provide the first multimodal (stereo RGB and LiDAR) dataset for road scenes. This dataset, unfortunately, consists of only a small subset of 200 SS training images. The next fundamental step was the acquisition of the Cityscapes [11] dataset, which was the first collection of labeled samples large enough to support training of deep architectures to a satisfactory level. It includes 5 000 finely-labeled samples and 20 000 coarsely-labeled samples captured in several German cities, and has become an important benchmark for the segmentation task. Recent works focus more on the volume [34], [39], [42] and variability [32], [44], [55], [56] of data. A noteworthy example is also the DADA-seg dataset [36], consisting of 2000 driving sequences obtained from mainstream video sites and focusing specifically on semantic segmentation during traffic accident events. Moreover, the DADA-seg dataset also allows to obtain event-camera data. Even more recently, researchers are supporting the advent of LiDAR sensors, and some datasets have been generated accordingly [46], [57], [58], [59]. Furthermore, fisheye and wide-field of view (FoV) cameras are being considered by the research community thanks to their already widespread adoption on commercial vehicles, thus prompting the generation of new datasets of the same kind. Namely, WoodScape [48] is among the first and most complete datasets providing both LiDAR and fisheye captures, where the omni-directional view of the surroundings is obtained combining the 4 fisheye cameras. Similarly, KITTI-360 [41], the successor of the KITTI dataset, contains data from two lateral $180° FoV$ fisheye cameras and a 90° degree perspective stereo camera in the front, and from a Velodyne HDL-64E and a SICK LMS 200 laser scanner. Finally, DensePASS [60] was generated using Google Street View as a target dataset for the domain transfer task from pinhole to panoramic data.

B. Synthetic Datasets

To circumvent the cost involved in the labeling of large-scale datasets, particularly those for SS, many synthetically-generated datasets have been proposed over the years. The first two important benchmarks are GTA5 [20] and SYNTHIA [21], both introduced in 2016. The former was generated exploiting the homonymous game, and provides 25 000 samples of realistic high-quality images. The semantic labels are provided in the same class set as Cityscapes [11], although they were inferred by the authors from secondary shader data, and the classes assigned to objects are not always consistent. The latter was the first dataset to provide depth ground truth for each of its 9 000 samples. The class set is different than that used by Cityscapes, and the overlap is limited to 16 classes (see the Suppl. Mat.[2] for details on the class splits). A third important synthetic dataset is Virtual KITTI [14], [22] which, like its real counterpart, focused heavily on the multimodal aspect. It was the first to provide ground truth optical-flow and instance segmentation data, in addition to color, depth and semantic. More recently, the IDDA dataset [23] was introduced to address the lack of weather conditions variability in the available datasets. It was developed using CARLA [19] and includes semantic labels (with an overlap of 16 classes with Cityscapes), depth and RGB data. Similarly, OmniScape [61] and SynWoodScape [52] modified the CARLA simulator to produce catadioptric (the former) and fisheye (both) datasets, whereas PanoFlow [62] used 6 $90° \times 90°$ FoV cameras to produce panoramic images. Finally, the event camera introduced in CARLA was used to generate EventScape [51], a new dataset that combines semantic segmentation with the new possibilities offered by event cameras.

From Table I we can see that, among the synthetic datasets, SELMA is the only one to provide labeled data for multiple LiDAR and camera sensors. Moreover, it is the only one that provides multiple weather conditions while supporting the full Cityscapes [11] class set, as opposed to IDDA. Even more, it is the only one that provides 3D bounding boxes. Finally, compared to GTA5 [20], i.e., the only competitor to provide anti-aliased color images, SELMA provides more samples, and considers a much higher variability of setups and sensors.

III. Simulator Setup

A. The CARLA Simulator

The CARLA simulator is used to generate synthetic data relative to autonomous driving systems. It is designed as an open-source layer over Unreal Engine 4 (UE4) to provide high-quality rendering, realistic physics based on the NVIDIA PhysX engine, and basic Non-Player Character (NPC) logic [19]. Reproducible and reliable physics simulations, as well as realistic and synchronized sensor data, can be obtained through the CARLA Application Programming Interface (API). Hereby, we briefly report the main characteristics of release 0.9.12, which was the starting point for the customization we made to meet the desired characteristics of the dataset, as detailed in Sec. III-B.

1) Unreal Engine Models: CARLA offers a wide variety of carefully designed UE4 models for static (e.g., buildings, vegetation, traffic signs) and dynamic objects (e.g., vehicles and pedestrians), sharing a common scale, and with realistic sizes. In release 0.9.12, the blueprint library includes the model of 24 cars, 6 trucks, 4 motorbikes, 3 bikes, each with customizable colors, and 41 pedestrian models of different ethnicity, build, and dressed with a wide variety of clothes. Furthermore, 8 towns (Town01-07 and Town10HD) were carefully designed by the CARLA team using more than 40 building models. Each town has its unique features and landmarks, thus offering 8 simulation environments with diverse visual characteristics.

2) Sensors: Data from the simulated world can be retrieved through a number of different sensors (see the Suppl. Mat. for a detailed list of supported sensors), that can be placed at an exact location and a given rotation, and attached to a

[2]The Supplementary Material is available on arXiv (https://arxiv.org/abs/2204.09788) and on our website (https://scanlab.dei.unipd.it/selma-dataset/)

Fig. 1. Desk view at three different times of the day.

Fig. 2. Samples in 9 variable weather conditions at Noon.

parent actor, thus following its movements with a rigid or spring-arm-like behavior. Sensors data can be collected at each simulation step. When working with multiple, high-resolution sensors, synchronous mode is required to guarantee that the GPU completes the rendering and delivers the data to the client before the following simulation step. Thus, the sensor acquisition rate is the same for all.

3) Weather Conditions and Daytime: Leveraging the underlying UE4 graphics, CARLA offers a variety of *daytime* and *weather* conditions. The combination of daytime and weather will be referred to as *environmental conditions* in the rest of the paper. Such conditions differ in the position and color of the sun, and in the intensity and color of diffuse sky radiation (daytime), as well as ambient occlusion, fog, cloudiness, and precipitation (weather). In release 0.9.12, there are 14 predefined environmental conditions, obtained by the combination of two daytimes (Noon and Sunset), and seven weather conditions (Clear, Cloudy, Wet, WetCloudy,[3] SoftRain, MidRainy, HardRain).

B. Customization

To enhance the quality of the collected data, we customized the source code of CARLA, as detailed in the following.

First, we adjusted the parameters of the predefined environmental conditions, modifying the weather scattering and fog properties, and the position of the sun, to maximize the diversity between the environmental conditions and their photo-realism. Then, we introduced the Night daytime (Fig. 1) and the Mid Fog and Hard Fog weather conditions (Fig. 2). Thus, the number of daytimes and weather conditions has been increased from 2 and 7 to 3 and 9, respectively, bringing the total number of environmental conditions to 27.

Second, we modified the CARLA semantic classes, to make them compatible with existing benchmark datasets, and added new vehicle models to increase the class diversity. Specifically, the remapping of the classes was done in the source code to affect both the semantic LiDAR and the semantic camera. We introduced the *Train* class, adding a train and a tram model, and we added two bus and two truck models to the existing classes. Then, our modifications to the source code allowed us to introduce the *Rider* class, adding the corresponding tag and separating the rider from its bike/motorbike tag. A visual example is reported in Fig. 3. Finally, as of release 0.9.12, the parked vehicles were not labeled correctly. Exploiting the CARLA API, we removed the corresponding map layer, saving the location information to place vehicles with the correct tag in the exact same position.

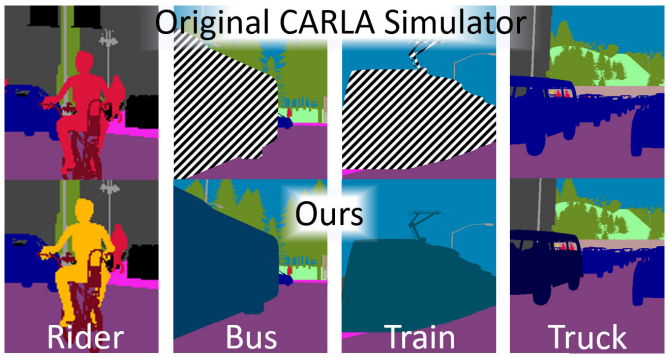

Fig. 3. Comparison between the original version of the CARLA simulator and that with our modifications. Dashed regions indicate classes that were originally missing in CARLA. Notice that our implementation now distinguishes riders from people and trucks from cars.

Then, the UE4 content was modified to meet the strict requirements that we set for the SELMA dataset. Namely, bikes could only exist along with their rider on board, which prevented parked bikes to be deployed. Nonetheless, bikes are amongst the main road actors, and CV algorithms greatly benefit from visual variability. Indeed, we deemed fundamental for our dataset to include the bike class in all the contexts. Therefore, we added hundreds of parked bikes into the existing CARLA maps. Similarly, the number of traffic lights and signs in the default towns did not reflect their distribution in a real setting, thus possibly compromising the learning ability of the algorithms. Thus, we distributed tens of additional traffic lights and signs in every town. The final class distribution matches that of real-world reference datasets such as Cityscapes, as shown in Fig. 4. The customized simulator is freely available.[4]

IV. SELMA Dataset Design

In this section we present our SELMA dataset, with a focus on the acquisition setup (Sec. IV-A) and splits (Sec. IV-B).

A. Acquisition Setup

We designed the acquisition pipeline to exploit the full potential of CARLA, while maximizing the diversity of the acquired data. Acquisitions were made equipping a vehicle,

[3] *Wet* and *WetCloudy* indicate that the road is more reflective and contains puddles. Notably, the former (latter) specifies that observations are acquired in clear (cloudy) sky.

[4] Our customized version of CARLA is available at https://github.com/LTTM/SELMA.

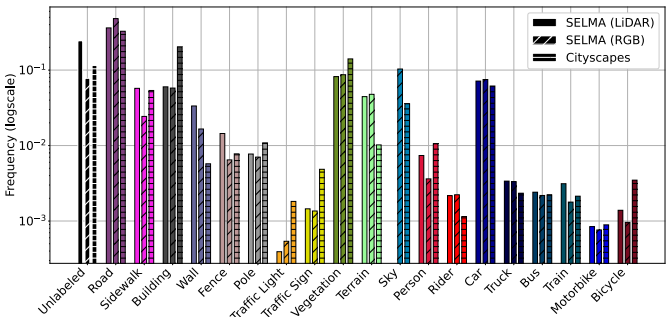

Fig. 4. Class distributions in the SELMA dataset.

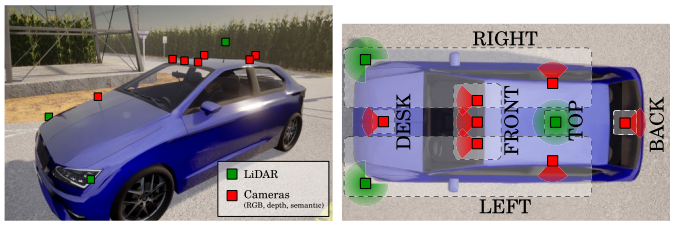

Fig. 5. Side and top views of the sensor setup in SELMA. RGB, depth and semantic cameras are co-located in 7 spots. LiDARs are placed in 3 locations.

named the ego vehicle, with a full sensor suite depicted in Fig. 5, consisting of:

- 7 RGB cameras, with the post-processing effects enabled, a 90-degree horizontal FoV, and a native resolution of 5120 × 2560, which is downsampled to 1280 × 640 to achieve a ×4 anti-aliasing enhancement. The post-processing effects include vignette, grain jitter, bloom, auto exposure, lens flare and depth of field. RGB images are saved in JPEG format.
- 7 depth cameras with a 90-degree horizontal FoV and 1280 × 640 resolution. For the depth images, anti-aliasing is not required and would compromise depth information. Depth images are saved in PNG format.
- 7 semantic cameras, that have the exact same attributes of the depth cameras.
- 3 semantic LiDARs, each with 64 vertical channels, generating 100 000 points per second, with a range of 100 meters. Point clouds are saved in PLY format.

The different camera types are co-registered at 7 different locations, and are set up to have the same FoV and resolution, so that their data can be easily matched. The variability of viewpoints and maps is shown in Fig. 6.

Furthermore, we compute the 3D surface normals at each pixel of the image acquired by the desk camera for all samples of the default random split (see Sec. IV-B). To do so, we employ a state-of-the-art differential technique as done in [63] and [64]. While the result is an approximation of the true normals, the overall precision is very high, as can be seen in the last column of Fig. 7, thanks to the detailed (i.e., synthetic ground truth) and dense depth maps used for the estimation procedure.

Data were acquired in 30 909 independent locations, across 8 virtual towns as reported in Table II and in the Suppl. Mat.

TABLE II
NUMBER OF WAYPOINTS (WPS) PER TOWN. DATA ARE ACQUIRED AT EACH WAYPOINT, INDEPENDENTLY FOR ALL ENVIRONMENTAL CONDITIONS

Town ID	01	02	03	04	05	06	07	10HD	**Total**
# of WPs	1634	756	3636	8565	6250	6579	1922	1568	**30909**

The locations were selected extracting a list of waypoints at a given distance. For our dataset, we selected points on the roads on every lane and junction, at the distance of 4 meters, which was chosen after several empirical tests as it offered the best trade-off between area coverage and acquisition diversity. The full list of waypoints with their ID is provided with the dataset for each town. At each position, the ego vehicle is created, traffic is generated around it, and pedestrians are randomly placed on the sidewalks. After one second of simulation for the transient to end, the sensors are fired simultaneously, and their data retrieved and saved. The server is then reset and the simulation goes on with the following waypoint.

The same process is repeated in 27 different environmental conditions. These include 3 daytimes (i.e., Noon, Sunset and Night) and 9 weather conditions (i.e., Clear, Cloudy, Wet, Wet and Cloudy, 2 Fog intensities and 3 Rain intensities). Traffic and pedestrians are generated randomly at every iteration, thus the same waypoint simulated under different environmental conditions presents different traffic conditions. We refer to the combinations of environmental conditions and towns as scenes. Our dataset consists of 216 scenes, obtained considering all the sensors and the complete combinations of the available weather and daytime conditions, viewpoints, and towns.

B. Splits

Exploiting the fine-grained control on environmental conditions offered by SELMA, we designed some default splits. Particularly, we considered 6 different weather distributions: Random (SELMA default), Mostly Clear (MC), Noon, Night, Rain and Fog. For more details on the splits, we refer the interested readers to the Suppl. Mat.

The Random split contains samples from all weather conditions and daytimes, sampled according to the probability distributions reported in Table III. Most of the samples come from high-visibility weather conditions: Clear, Wet (road), Cloudy and WetCloudy make up for 75% of the split.

In order to preserve the separation among training, validation and test samples, the splits are provided in CSV format, which allows to easily assign a given weather condition to a sample (and to override it, if needed). The samples separation was done according to an 80:10:10 split rule for training, validation and test, respectively.

V. EXPERIMENTAL VALIDATION

In this section we carefully analyze and validate the SELMA dataset. We start with a series of baseline experiments which serve as a reference benchmark for future studies (Sec. V-A). Then, we analyze the thematic subsets of our

Fig. 6. Randomly sampled images from the SELMA dataset in clear noon setup, demonstrating its diversity. Rows show different cameras, while columns show different (synthetic) towns (thus settings).

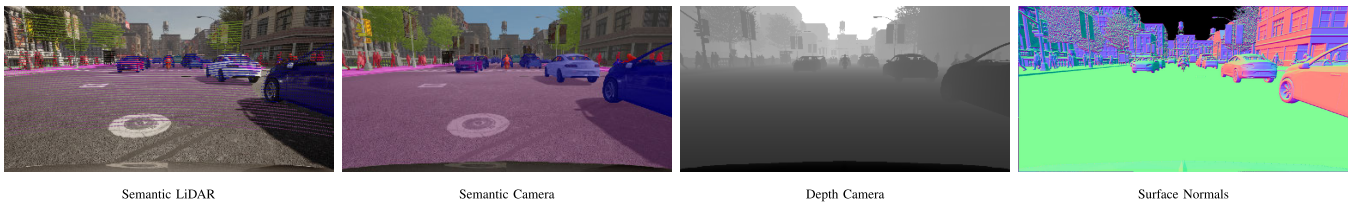

Fig. 7. Sample acquisitions with semantic LiDAR, semantic camera overlaid to the RGB samples, depth camera, and surface normals.

TABLE III
PROBABILITY DISTRIBUTIONS OF ENVIRONMENTAL CONDITIONS IN THE DIFFERENT SPLITS

Split	Noon	Sunset	Night	Clear	Cloudy	Wet Road	Wet Road and Cloudy	Mid Fog	Hard Fog	Soft Rain	Mid Rain	Hard Rain
SELMA/Noon/Night	50/100/0%	25/0/0%	25/0/100%	35%	20%	10%	10%	3.5%	3.5%	6%	6%	6%
Mostly Clear/Rain/Fog	50%	25%	25%	25/0/0%	25/0/0%	25/0/0%	25/0/0%	0/0/50%	0/0/50%	0/34/0%	0/33/0%	0/33/0%

TABLE IV
MIOU OF BASELINE SS ARCHITECTURES ON CITYSCAPES (CS) AND SUBSETS OF SELMA FOR BOTH RGB IMAGES (FIRST 7 COLUMNS) AND DEPTH (LAST COLUMN)

	CS	SELMA	Noon	Night	MC	Fog	Rain	Depth
DeeplabV2 [65]	67.4	68.9	72.3	68.2	69.9	68.0	68.7	73.4
DeeplabV3 [66]	68.2	70.7	71.7	68.4	70.8	69.0	67.8	72.7
FCN [67]	64.8	68.2	71.1	66.1	69.1	64.5	66.8	73.7
PSPNet [68]	65.3	68.4	71.2	67.2	69.8	66.8	69.0	73.6
UNet [69]	36.8	36.2	41.8	35.7	36.3	28.6	37.8	28.8
SegFormer [70]	**71.9**	**77.2**	**80.2**	**75.1**	**78.2**	**76.5**	**76.6**	**80.3**

TABLE V
DETAILED VIEW OF THE LABEL SETS CONSIDERED

	road	swalk	building	wall	fence	pole	tlight	tsign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	Count
City19	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	19
Idd17	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓	17
Synthia16	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓		✓	✓	16
Idda16	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓	16
Idda-Synthia-15	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓				✓	✓	15
Synthia-13	✓	✓	✓				✓	✓	✓		✓	✓	✓	✓				✓	✓	13
Idda-Synthia-12	✓	✓	✓				✓	✓	✓		✓	✓	✓					✓	✓	12

dataset (Sec. V-B). To conclude, we show how different sensors can be employed jointly to improve the final segmentation accuracy (Sec. V-C), and report some experiments exploiting multiple viewpoints (Sec. V-D).

A. Baseline Experiments

The first set of experiments is designed to provide a series of benchmark results for the SELMA dataset. The results in Table IV show the performance achieved employing different baseline SS architectures, i.e., UNet [69], FCN [67], PSPNet [68], DeepLab-V2 [65], [66], DeepLab-V3 [66] and SegFormer [70]. All the networks are trained with SGD with momentum of rate 0.9. The learning rate was decreased according to a polynomial decay of coefficient 0.9, starting from 2.5×10^{-4}. The batch size was set to 3 and the weight decay to 10^{-4}. Additional results for the individual classes are reported in the Suppl. Mat.

1) RGB: Initially, we perform a series of experiments using the RGB images from Cityscapes and from the SELMA desk camera in different environmental conditions. First, we observe that the UNet architecture achieves poor results, since it is unable to deal with the large visual variability of the SELMA dataset. The other architectures share the same encoder module, i.e., a ResNet-101, and they all achieve similar mean Intersection over Union (mIoU) performance. As expected, the more recent transformer-based architecture SegFormer [70]

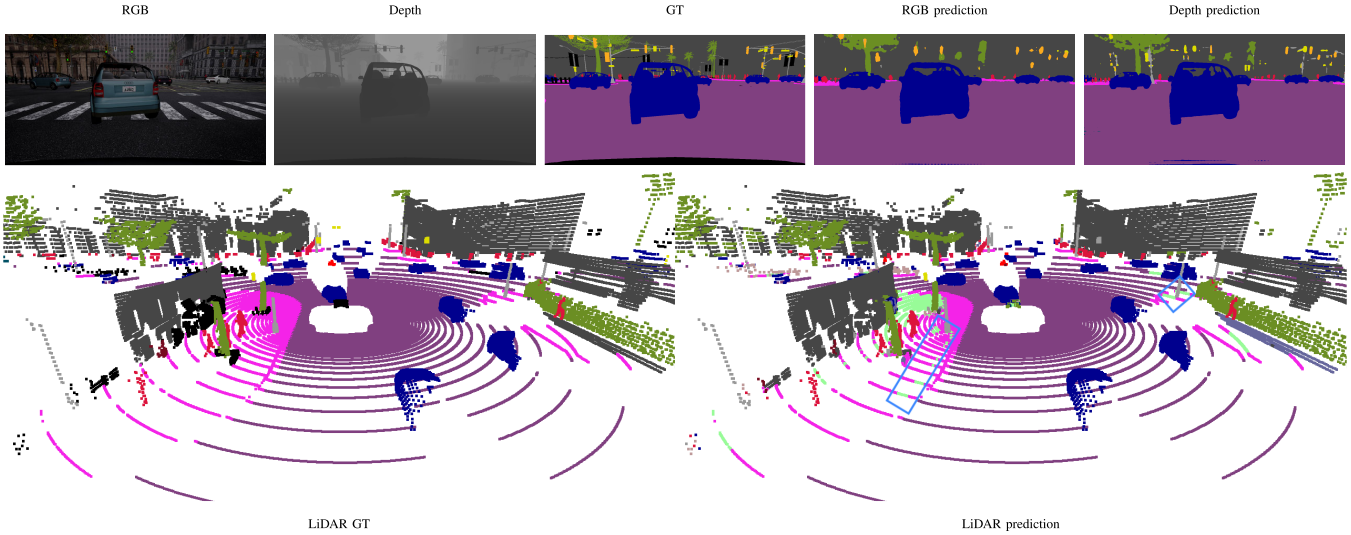

Fig. 8. Qualitative results for the SS task from RGB images (DeepLab-V2 [65]), depth maps (DeepLab-V2 [65]) or point clouds (Cylinder3D [71]).

TABLE VI

PER-CLASS AND MEAN IOU RESULTS FOR THE SUPERVISED TRAINING AND TESTING ON THE SAME DOMAIN WITH THE DEEPLAB-V2 ARCHITECTURE. THE MEAN IS COMPUTED OVER DIFFERENT LABEL SETS

	per-class IoUs																			mIoUs on different label sets					
	road	swalk	building	wall	fence	pole	tlight	tsign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	city19	idda16	synthia16	synthia13	idda-synthia-15	idda-synthia-12
SELMA	99.1	87.5	83.9	83.5	57.3	47.1	36.8	61.1	78.4	77.2	89.4	63.9	62.8	88.7	71.6	48.9	72.3	62.0	37.7	68.9	69.8	68.0	69.2	69.3	70.9
Noon	99.3	88.9	86.7	86.7	59.9	49.1	40.1	61.8	82.8	83.3	92.8	65.5	63.3	89.5	76.8	48.8	75.0	65.7	37.7	71.2	72.1	69.9	71.0	71.3	72.8
Night	99.0	86.3	80.0	82.2	53.6	45.9	36.0	60.8	71.1	67.4	84.1	62.0	62.1	90.1	78.0	57.7	71.3	60.9	34.9	67.5	67.3	66.7	68.1	67.3	68.9
Mostly Clear	99.2	88.1	85.3	85.1	57.7	48.0	37.4	61.1	79.9	79.3	91.0	64.7	63.5	89.0	65.5	54.6	73.5	66.8	38.5	69.9	70.9	69.4	70.7	70.4	72.0
Rain	99.0	86.3	82.9	84.0	57.5	46.6	38.6	62.0	77.5	74.8	88.0	64.5	63.5	88.7	65.1	55.1	71.8	64.0	36.2	68.7	69.6	68.4	69.7	69.3	70.9
Fog	99.2	88.6	77.3	80.6	55.0	46.4	40.0	61.6	69.7	66.9	81.8	63.6	63.4	88.3	57.4	47.5	69.6	65.3	37.8	66.3	67.8	66.6	68.0	67.9	69.7
GTA5 [20]	95.8	83.1	86.9	59.4	45.4	52.1	51.7	50.3	80.8	68.6	95.1	67.9	44.1	88.6	85.2	84.6	72.8	56.3	31.6	68.4	66.1	67.1	70.5	65.9	69.4
IDDA [23]	98.7	92.1	93.2	85.3	68.8	59.2	65.0	60.2	84.8	84.3	97.4	57.9	54.2	97.8	-	-	-	65.3	30.6	62.9	74.7	69.4	69.0	74.0	74.8
SYNTHIA [21]	92.2	90.9	92.3	79.6	59.0	54.4	25.0	42.1	79.6	-	94.6	74.6	49.8	87.2	-	87.5	-	63.4	33.0	58.2	63.6	69.1	70.2	67.8	68.7
ACDC [31]	92.7	71.3	82.6	46.5	37.2	50.5	61.1	49.3	83.9	44.5	94.6	49.0	17.9	80.8	41.0	71.1	81.7	29.5	41.7	59.3	58.3	60.0	63.5	59.2	62.9
CityScapes [11]	97.1	77.4	89.2	50.1	46.2	45.2	48.9	61.4	89.5	55.2	92.2	69.9	46.5	91.4	66.9	75.2	60.3	52.2	65.8	67.4	67.4	68.6	73.6	68.2	73.5
Cityscapes Fog [49]	97.1	77.2	87.9	47.7	46.5	44.9	50.4	59.7	88.5	54.3	86.6	69.2	47.4	90.6	59.2	73.0	59.7	50.3	64.6	66.0	66.4	67.6	72.5	67.2	72.5
Cityscapes Rain [50]	97.6	77.3	85.1	25.9	5.1	29.4	25.4	35.9	87.4	61.1	93.9	49.7	33.4	85.1	0.2	25.2	-	8.3	60.2	46.6	53.8	51.6	58.8	53.3	61.6
IDD [39]	96.8	72.2	73.4	70.0	35.2	45.2	9.9	70.4	87.8	-	94.9	63.8	69.1	88.5	83.5	87.5	-	65.2	26.8	60.0	60.6	66.0	69.7	64.6	68.2
Mapillary [42]	92.8	59.7	85.5	47.7	52.5	49.4	53.7	65.3	88.3	64.2	97.7	66.9	44.3	88.6	58.8	63.4	21.5	50.2	56.7	63.5	66.5	66.4	70.2	66.6	70.8
NightCity [47]	89.7	42.2	81.3	44.2	49.0	25.7	20.4	47.5	58.5	25.2	86.3	47.5	27.1	78.6	52.4	62.1	27.6	25.2	30.1	48.5	48.7	51.0	53.6	50.2	52.9

outperforms the others, achieving an mIoU of 80.3. Overall, also DeepLab-V2 and V3 offer good performance. Then, to facilitate the comparison with the other methods considered in this paper, in the following experiments we decided to refer to the standard DeepLab-V2 architecture as a baseline, which represents a common benchmark for the evaluation. The highest accuracy is obtained with the SELMA Noon split, as RGB images are easier to segment. On the contrary, Night, Fog and Rain are more challenging.

In Table VI we show the per-class IoU scores training DeepLab-V2 on our dataset (first group), on common synthetic benchmarks (second group), and on real-world datasets (last group). For each dataset, we evaluate the trained model on the label splits most commonly considered in the literature. Table V reports the names of the classes of each label set. The most widely used label set is City19, which is also the most complete. For the sake of fairness with respect to all the datasets, in Table VI we test our models on all the possible label sets. SELMA demonstrates similar, or often higher, performance compared to other synthetic and real benchmarks due to its extreme variability in daytime and weather conditions, as well as its large-scale property with more than 30 000 unique (labeled) samples.

2) Depth: We run the same experimental evaluation using a single input channel representing the depth of the scene. More precisely, since the range of true values is extremely unbalanced and their distribution is highly skewed, we normalize and rescale the depth values: starting from the original depth produced by the simulator, normalized to 1, we compute its fourth root to compress the high-distance information and expand the low-distance information. This is necessary as the sky is marked with the maximum distance possible, and overshadows the other pixels. Then, we rescale and shift the values to the $[-1, 1]$ range. Also in this case, the best performing

TABLE VII
BASELINE LIDAR SS METHODS ON SELMA POINT CLOUDS. RESULTS ON THE CS LABEL SPLIT, REMOVING SKY. LAST ROW: RGB AND LIDAR FUSION

Architecture	mIoU
RangeNet++ [76] (SqueezeSeg-V2)	61.9
RangeNet++ [76] (DarkNet-21)	67.4
Cylinder3D [71]	**80.3**
DLV2 [65] on spherical projections	57.3
RGB+LiDAR (DLV2-SP)	63.5

TABLE VIII
MIOU PERFORMANCE OF RGB AND DEPTH FUSION

Method	mIoU
RGB	68.9
Grayscale	68.0
Depth	73.4
RGBD	72.4
RGBD @layer1	74.3
CMX	91.7

architectures achieve comparable performance, as shown in the rightmost column of Table IV. The models trained on depth images can more easily segment objects of different classes, outperforming the results achieved on RGB samples. The reason for this behavior is that such experiments are based on ground-truth depth maps, synthetically originated from the 3D geometry of the scene and unaltered by atmospheric conditions (that would instead heavily degrade the RGB performance) and by noise, whereas depth maps derived from real data are generally noisy.

Indeed, the lack of noise is one of the major factors contributing to the gap between synthetic and real data. Modeling the sensor data noise is a complex task [72], and depends on several factors, e.g., the model of the sensor [73] or the processing technique [74]. SELMA provides noiseless, sensor-agnostic ground-truth maps, and leaves the users the freedom to choose the preferred noise model, according to their requirements and the target sensor of interest. For example, an advanced model based on a Generative Adversarial Network (GAN) can be found in [75].

3) LiDAR: Table VII reports the LiDAR SS results obtained with RangeNet++ [76] with two backbones (SqueezeSeg-V2 [77] and DarkNet-21 [78]) and with Cylinder3D [71]. Furthermore, we report the results obtained by flattening the point cloud via spherical projection and creating an image with 4 channels containing the depth from the LiDAR and the RGB color from the cameras that is fed to a DeepLab-V2 network for segmentation. All the backbones are trained with batch size of 4 for 40 epochs with early stopping enabled. The other learning parameters are left to the default values provided in the respective codebases. We can observe that Cylinder3D outperforms the other architectures, achieving an outstanding mIoU of 80.3.

Fig. 8 reports the qualitative results for the best segmentation architectures, i.e., the DeepLab-V2, for the RGB and the depth samples, and Cylinder3D for the point clouds. Comparing the RGB and depth-based prediction, we can appreciate that the latter offers great improvements in the recognition of far, small and challenging items in the background, such as poles, traffic lights and traffic signs. However, the use of geometric information leads to uncertainty in the prediction of traffic lights and signs, which are mixed up in the depth prediction, but not in the RGB one. On the other hand, looking at the point cloud segmentation, we can appreciate the great overall precision, as expected given the high quantitative score. Nevertheless, some artifacts are still present since the prediction is based solely on geometric information, e.g., the sidewalk region on the left is partially confused for ground in proximity of the vegetation class. Another interesting artifact lies in the geometrical arrangement of the errors. Due to the intrinsic working principle of Cylinder3D [71], we observe that most of the errors are propagated along the same radial coordinates. For instance, we see that the network predicts ground in spite of sidewalk for a few consecutive scans in a couple of regions denoted by light blue rectangles. Finally, the prediction performance of semantic labels is poor for small classes such as traffic signs or lights, which are often confused for poles.

B. Thematic Subsets

Then, to highlight the capability of SELMA to incorporate different visual domains, we define 6 subsets, the so-called *thematic* splits, sampling images with specific daytime or weather conditions, as mentioned in Sec. IV-B.

The mIoU results for the splits are shown in Fig. 9, where we report the supervised accuracy in the diagonal elements and the source-only accuracy (i.e., trained on the source domain and tested on the target domain) on off-diagonal elements. Here, we can appreciate that training and testing on the same visual domain gives almost always the highest mIoU (diagonal elements), except for some cases where the target domain is much easier than the source domain, as is the case for the SELMA Noon as target dataset.

In absolute terms, the hardest subsets (i.e., lowest supervised accuracy) are SELMA Night, Fog, and Rain, respectively. Adapting the source knowledge acquired from a subset containing a single daytime (e.g., Noon or Night) proves to be less robust to domain variability at test time, rather than adapting knowledge from a subset containing multiple daytime domains (e.g., Rain or Fog): in the first case we can observe lower off-diagonal accuracy scores compared to the second case.

C. Fusion Experiments

To prove the importance of SELMA as a multimodal dataset, we show how we can improve the segmentation quality by coupling acquisitions from different sensors. Indeed, different sensors have variable performance depending on the visibility conditions of the scenes, and can be used jointly to leverage understanding scores. To highlight this aspect, we report some experiments in Table VIII, starting from the single sensors, proposing two simple approaches to give some insights on the potential of multimodal segmentation, and finally testing a state-of-the-art algorithm on SELMA.

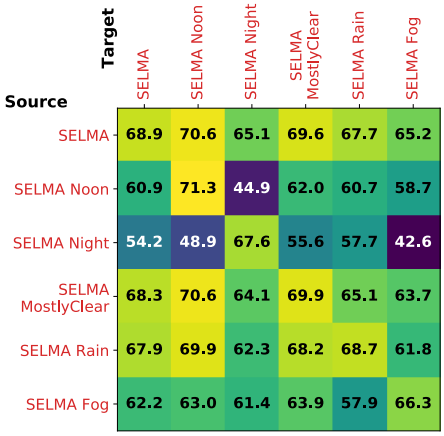

Fig. 9. mIoU results on thematic splits sub-sampled from the complete SELMA dataset.

Notably, using RGB images allows to achieve an mIoU of 68.9, while using the depth alone could improve the mIoU to 73.4. As a comparison, using the grayscale version of the image (single channel) we achieve an mIoU of 68.0, which is lower than the result on RGB images, as expected.

Building a combined RGB and depth representation at the input level (denoted as RGBD, i.e., an input of 4 channels), we achieve an mIoU of 72.4, which is higher than using RGB alone, but lower than using depth alone. Hence, we argue that simply combining the input representations as they are provided is not enough to increase the SS accuracy. Therefore, we include an additional convolutional layer (followed by batch normalization and ReLU activation function) to extract features from RGB and depth samples separately. Then, we concatenate the outputs and feed the result as the input of the first layer of the ResNet101. We denote this fusion method as "RGBD @layer1" in Table VIII. With this simple provision, we could achieve an mIoU of 74.3, an improvement of 5.4 points with respect to using RGB alone, and by 0.9 points with respect to using depth alone.

Finally, the highly performing multi-modal CMX architecture [79] trained on SELMA achieves an mIoU of 91.7, improving by 22.8 and by 17.4 points the RGB-only and the simple fusion approach, respectively. Encouraged by these promising results, we believe that future research could leverage the multimodal design of SELMA to extensively investigate a wide range of fusion strategies for different sensors, such as RGB, depth and LiDAR.

D. Multi-View Experiments

As baseline experiments for the multi-view aspect of our dataset, we consider an architecture trained on the desk camera and tested on the available points of view. This permits to verify the intra-domain shift caused by the variable camera viewpoints, which we expect to be more significant in the cameras facing different directions (like Left, Right and Back). We measure the domain shift by computing the Kullback-Leibler (KL) divergence between the label distribution of the DESK camera and that of the other cameras, reported in Table IX together with the average mIoU score and the corresponding degradation with respect to the reference DESK camera.

TABLE IX

MIOU MULTI-VIEW RESULTS. DEEPLABV2 [65] IS TRAINED ON SAMPLES FROM THE DESK CAMERA AND TESTED ON OTHER CAMERAS. THE KL-DIVERGENCE BETWEEN THE LABEL DISTRIBUTION SEEN BY THE DESK CAMERA AND BY THE OTHER CAMERAS IS ALSO REPORTED, TO QUANTIFY THE DOMAIN SHIFT. THE ARROWS INDICATE WHETHER A METRIC IS HIGHER-IS-BETTER (↑) OR LOWER-IS-BETTER (↓)

Camera	KL-Divergence ↓	mIoU ↑	Δ mIoU ↓
DESK	0.0000	68.9	0.0
FRONT LEFT	0.2806	66.6	2.3
FRONT	0.2812	66.6	2.3
FRONT RIGHT	0.2820	66.6	2.3
LEFT	0.2175	66.2	2.7
BACK	0.1642	65.9	3.0
RIGHT	0.3581	62.2	6.7

The mIoU scores confirm our expectations, i.e., all front-facing cameras have minimal performance degradation (2.3 mIoU) and are all similar to each other due to their reciprocal proximity (see Fig. 5), as confirmed by the KL score, equal for all the three cameras. The LEFT camera also shows limited degradation (only -0.4 compared to the front-facing cameras), with an mIoU of 66.2. On the other hand, the RIGHT camera shows a significant loss of 6.7 mIoU compared to the DESK camera, as also demonstrated by the large KL divergence. This is because CARLA is a right-lane driving environment, meaning that the point of view of the right-facing camera changes significantly with respect to the front-facing camera used for training, which leads to performance degradation.

Moreover, we can observe an inverse correlation between the mIoU score and the KL divergence from the DESK camera, as expected, for all the cameras except for the BACK camera. This discrepancy is probably due to the fact that, while the frequencies of objects from the BACK camera is consistent with that of the FRONT cameras, they might still look different on some aspects (e.g., vehicles will be turned towards the camera, rather than away from it). Still, the BACK camera performs similarly to the LEFT camera, with an mIoU score of 65.9, that is only 0.3 lower than the latter.

VI. EXPERIMENTS ON REAL-WORLD DATASETS

In the last set of experiments we validate the SELMA dataset on different SS models. We run an extensive evaluation by training the DeepLab-V2 segmentation architecture on the SELMA dataset and testing it on different real-world datasets.

A. Training With a Mixture of Synthetic and Real Data

To start, we show in Fig. 10 the mIoU accuracy on source and target sets when training the segmentation network on samples drawn either from the target real-world domain, i.e., the Cityscapes dataset (with probability r), or from the source domain, i.e., our SELMA dataset (with probability $1 - r$). Adding as few as 5% to 10% of data from a different domain improves domain generalization, i.e., the network can perform well on both domains. Even more, we highlight that

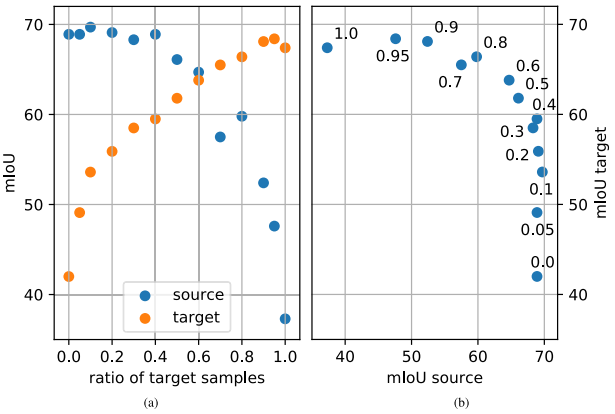

Fig. 10. Accuracy (mIoU) with images sampled either from the Cityscapes dataset (target) or from our SELMA dataset (source). Using more samples from the target dataset in the training (i.e., moving along the positive direction of the x-axis in (a)) degrades the mIoU on the source data (blue), while improving that of the target (orange). Similarly, (b) compares the mIoU of the source with that of the target, showing how the former increases when using only source data ($r = 0.0$) and reaches its minimum when using only target data ($r = 1.0$) for the training.

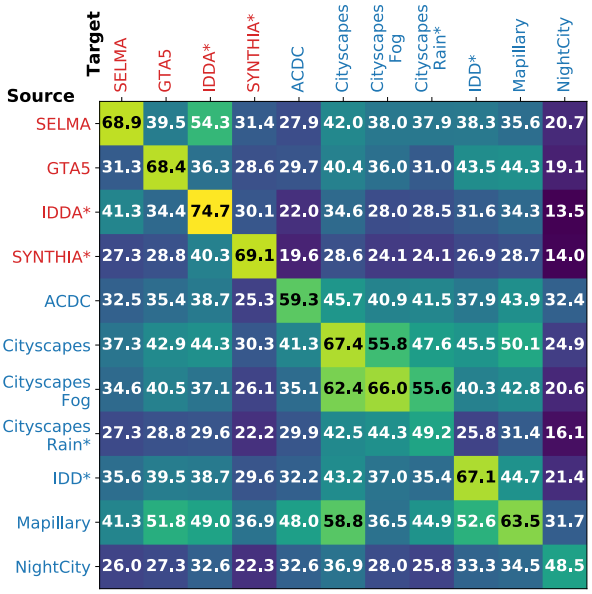

Fig. 11. mIoU performance for different synthetic and real-world datasets. The network is trained on a source domain (rows) and tested on a target domain (columns). Off-diagonal elements correspond to the presence of domain shift. The asterisk (*) indicates that a different label set is employed for testing (see Suppl. Mat. for further details).

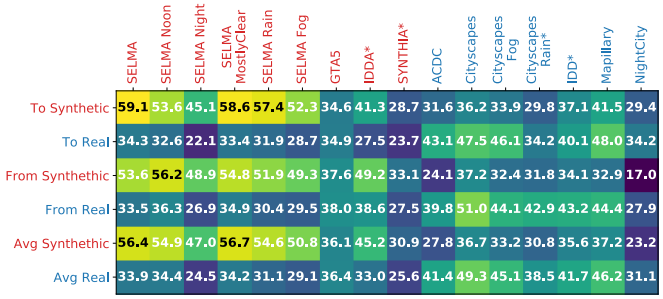

Fig. 12. mIoU performance results taken from the heatmap in Fig. 11, and aggregated by synthetic or real domain. The asterisk (*) indicates a different label set (see Suppl. Mat.).

a 5% of samples from SELMA can improve the performance on the target domain from 67.4 to 68.4. Similarly, 10% of target samples improve the performance on the source domain from 68.9 to 69.7. The per-class accuracy is reported in the Suppl. Mat.

To analyze the performance gain when considering imprecise labels, we trained the same architecture using a mixture of SELMA and coarsely-annotated samples from Cityscapes. We achieved an mIoU score of 59.5, which is 7.9 lower than the fully supervised training on Cityscapes, and 2.3 lower than the mixed training score when $r = 0.5$. This demonstrates that we can bridge the domain gap with few coarsely-supervised samples in the target domain, thus reducing the cost for accurate labeling.

B. Training With Synthetic Data Only

We then considered an unsupervised setup where no data from the target dataset were used for the training. We performed an extensive validation in the presence of domain shift, whose results are reported in Fig. 11: we considered four synthetic datasets (in red) and seven real-world datasets (in blue) with extremely variable time and weather conditions. We trained a DeepLab-V2 architecture (with ResNet-101 as backbone) on each dataset, and performed the testing on all the domains. Values on the diagonal correspond to training and testing performed on the same dataset, i.e., standard supervised training, while values off the diagonal correspond to training on a source domain (on the rows) and testing on a different target domain (on the columns). As expected, the values on the diagonal are larger than the others, since there is no domain shift.

In general, the mIoU performance depends on the source domain where training is performed. For example, a source training on SELMA performs well on IDDA, and vice-versa, since the rendering engine is common for the two datasets. Also, a source training on Cityscapes performs well on Cityscapes Fog or Cityscapes Rain, and vice-versa, since the city-level domain is common for the two datasets.

On the other side, training on datasets which only account for daytime images, such as Cityscapes, SYNTHIA or IDDA, struggles to generalize to nighttime images, e.g., sampled from the NightCity dataset. Training on statistically variable datasets, such as SELMA or Mapillary, can greatly improve the generalization capabilities in challenging domains.

Furthermore, we can observe that architectures trained on the SELMA dataset outperform those trained on other synthetic datasets on the widely-used real-world Cityscapes dataset. Indeed, source knowledge acquired on SELMA transfers well to Cityscapes, achieving 42.0 of mIoU, higher than transferring knowledge from GTA5 (40.4), IDDA (22.0) or SYNTHIA (28.6). Remarkably, SELMA outperforms both GTA5 (which is the most popular dataset for this task) by 1.6 of mIoU, and IDDA (i.e., the most similar dataset based on the same graphic engine) by an outstanding 20 of mIoU.

Then, in Fig. 12 we show the results averaged according to synthetic or real domains. We observe that evaluating a model trained on synthetic data on another set of synthetic data - not necessarily from the same domain - achieves better performance than applying the same model to real data, and vice versa. This is due to different textures, colors, and brightness rendered by the synthetic graphic engines versus the true target properties of real-world datasets. In general, we observe that acquiring source knowledge on the SELMA dataset (or its subsets) leads to much higher accuracy scores (e.g., 34.3 from SELMA) on both source and target domains, rather than IDDA [23] (27.5) or SYNTHIA [21] (23.7) datasets. Overall, SELMA achieves similar scores as obtained by acquiring source knowledge from the GTA5 dataset.

Finally, to provide an Unsupervised Domain Adaptation (UDA) baseline, we applied the AdaptSegNet [80] approach to SELMA. In particular, to align with the other experiments, the AdaptSegNet architecture was trained in the single-level configuration, and obtained a final performance score of 42.5 mIoU, corresponding to an improvement of 1.1 mIoU compared to the score reported in [80] when adapting from GTA5 to Cityscapes.

VII. Conclusion and Future Work

In this paper we presented SELMA, a synthetic dataset with driving scenes that contain a large amount of labeled samples acquired considering several different sensors, weather, daytime and viewpoint conditions. The experimental evaluation shows that SELMA allows to efficiently train deep learning models for scene understanding in the autonomous driving context, achieving a good generalization to real-world data.

In general, we noticed that the accuracy of synthetic data compared to real data is limited by the quality and realism of the rendering engine and by the implementation of the synthetic sensors. Another limiting factor, common to many synthetic datasets, is the lack of proper models for the data noise and the artifacts generated by the sensor imperfections. Given that, in future releases, the CARLA Simulator may fill these gaps, SELMA will be updated accordingly. Finally, since by design SELMA contains independent samples, it cannot be used for temporal analyses, thus limiting its application to the training and evaluation of static segmentation networks. As part of our future work we may consider adding temporal sequences of multimodal data to the dataset.

The SELMA dataset is publicly available, in the hope that it will be useful to the research community.

The availability of large-scale multimodal acquisitions in variable weather, daytime and viewpoints in SELMA promotes research towards key challenges, for example scene understanding for autonomous driving applications like multimodal sensor exploitation, domain generalization from synthetic datasets to real scenes, and autonomous driving in adverse weather conditions.

References

[1] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations," *Transp. Res. A, Policy Pract.*, vol. 77, pp. 167–181, Jul. 2015.

[2] G. Silberg, R. Wallace, G. Matuszak, J. Plessers, C. Brower, and D. Subramanian, "Self-driving cars: The next revolution," Center Automot. Res., Ann Arbor, MI, USA, KPMG LLP & Center Automot. Res., White Paper 36, 2012.

[3] L. M. Clements and K. M. Kockelman, "Economic effects of automated vehicles," *Transp. Res. Rec.*, vol. 2606, no. 1, pp. 106–114, 2017.

[4] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 50–61, Jul. 2020.

[5] F. Secci and A. Ceccarelli, "On failures of RGB cameras and their effects in autonomous driving applications," in *Proc. IEEE 31st Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2020, pp. 13–24.

[6] V. Rossi, P. Testolina, M. Giordani, and M. Zorzi, "On the role of sensor fusion for object detection in future vehicular networks," in *Proc. Joint Eur. Conf. Netw. Commun. 6G Summit (EuCNC/6G Summit)*, Jun. 2021, pp. 247–252.

[7] M. Giordani, A. Zanella, T. Higuchi, O. Altintas, and M. Zorzi, "Performance study of LTE and mmWave in vehicle-to-network communications," in *Proc. 17th Annu. Medit. Ad Hoc Netw. Workshop (Med-Hoc-Net)*, Jun. 2018, pp. 1–7.

[8] M. Giordani, A. Zanella, T. Higuchi, O. Altintas, and M. Zorzi, "On the feasibility of integrating mmWave and IEEE 802.11p for V2V communications," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–7.

[9] S. Zhang, J. Chen, F. Lyu, N. Cheng, W. Shi, and X. Shen, "Vehicular communication networks in the automated driving era," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 26–32, Sep. 2018.

[10] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. CVPR*, Jun. 2020, pp. 2446–2454.

[11] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. CVPR*, Jun. 2016, pp. 3213–3223.

[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[13] J. Mao et al., "One million scenes for autonomous driving: Once dataset," 2021, *arXiv:2106.11037*.

[14] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proc. CVPR*, Jun. 2016, pp. 4340–4349.

[15] M. Toldo, A. Maracani, U. Michieli, and P. Zanuttigh, "Unsupervised domain adaptation in semantic segmentation: A review," *Technologies*, vol. 8, no. 2, p. 35, Jun. 2020.

[16] F. Barbato, M. Toldo, U. Michieli, and P. Zanuttigh, "Latent space regularization for unsupervised domain adaptation in semantic segmentation," in *Proc. CVPRW*, Jun. 2021, pp. 2835–2845.

[17] F. Barbato, U. Michieli, M. Toldo, and P. Zanuttigh, "Road scenes segmentation across different domains by disentangling latent representations," 2021, *arXiv:2108.03021*.

[18] U. Michieli, M. Biasetton, G. Agresti, and P. Zanuttigh, "Adversarial learning and self-teaching techniques for domain adaptation in semantic segmentation," *IEEE Trans. Intell. Vehicles*, vol. 5, no. 3, pp. 508–518, Sep. 2020.

[19] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. ACRL*, 2017, pp. 1–16.

[20] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Proc. ECCV*, 2016, pp. 102–118.

[21] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. CVPR*, Jun. 2016, pp. 3234–3243.

[22] Y. Cabon, N. Murray, and M. Humenberger, "Virtual KITTI 2," 2020, *arXiv:2001.10773*.

[23] E. Alberti, A. Tavera, C. Masone, and B. Caputo, "IDDA: A large-scale multi-domain dataset for autonomous driving," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 5526–5533, Oct. 2020.

[24] T. Sun et al., "SHIFT: A synthetic driving dataset for continuous multi-task domain adaptation," in *Proc. CVPR*, Jun. 2022, pp. 21371–21382.

[25] J. Zhang et al., "Behind every domain there is a shift: Adapting distortion-aware vision transformers for panoramic semantic segmentation," 2022, *arXiv:2207.11860*.

[26] D. Lahat, T. Adali, and C. Jutten, "Multimodal data fusion: An overview of methods, challenges, and prospects," *Proc. IEEE*, vol. 103, no. 9, pp. 1449–1477, Sep. 2015.

[27] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, Jun. 2009, pp. 248–255.

[28] D. Feng et al., "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021.

[29] G. Rizzoli, F. Barbato, and P. Zanuttigh, "Multimodal semantic segmentation in autonomous driving: A review of current approaches and future perspectives," *Technologies*, vol. 10, no. 4, p. 90, 2022.

[30] J. Geyer et al. (2020). *A2D2: Audi Autonomous Driving Dataset*. [Online]. Available: https://www.a2d2.audi

[31] C. Sakaridis, D. Dai, and L. Van Gool, "ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding," in *Proc. ICCV*, Oct. 2021, pp. 10765–10775.

[32] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2702–2719, Oct. 2020.

[33] M.-F. Chang et al., "Argoverse: 3D tracking and forecasting with rich maps," in *Proc. CVPR*, Jun. 2019, pp. 8748–8757.

[34] F. Yu et al., "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. CVPR*, Jun. 2020, pp. 2633–2642.

[35] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 88–97, 2009.

[36] J. Zhang, K. Yang, and R. Stiefelhagen, "Exploring event-driven dynamic context for accident scene segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2606–2622, Mar. 2022.

[37] C. Sakaridis, D. Dai, and L. V. Gool, "Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," in *Proc. ICCV*, Oct. 2019, pp. 7374–7383.

[38] H. Sakashita, C. Flothow, N. Takemura, and Y. Sugano, "DRIV100: In-the-wild multi-domain dataset and evaluation for real-world domain adaptation of semantic segmentation," 2021, *arXiv:2102.00150*.

[39] G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, and C. Jawahar, "IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments," in *Proc. WACV*, Jan. 2019, pp. 1743–1751.

[40] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The kitti vision benchmark suite," in *Proc. CVPR*, Jun. 2012, pp. 3354–3361.

[41] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3292–3310, Mar. 2023.

[42] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proc. ICCV*, Oct. 2017, pp. 4990–4999.

[43] Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C. Frank Wang, and M. Sun, "No more discrimination: Cross city adaptation of road scene segmenters," in *Proc. ICCV*, Oct. 2017, pp. 1992–2001.

[44] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. CVPR*, Jun. 2020, pp. 11621–11631.

[45] F. Tung, J. Chen, L. Meng, and J. J. Little, "The raincouver scene parsing benchmark for self-driving in adverse weather and at night," *IEEE Robot. Automat. Lett.*, vol. 2, no. 4, pp. 2188–2193, Oct. 2017.

[46] J. Behley et al., "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. ICCV*, Oct. 2019, pp. 9297–9307.

[47] X. Tan, K. Xu, Y. Cao, Y. Zhang, L. Ma, and R. W. H. Lau, "Night-time scene parsing with a large real dataset," *IEEE Trans. Image Process.*, vol. 30, pp. 9085–9098, 2021.

[48] S. Yogamani et al., "WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *Proc. ICCV*, Oct. 2019, pp. 9308–9318.

[49] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," *Int. J. Comput. Vis.*, vol. 126, no. 9, pp. 973–992, 2018.

[50] X. Hu, C.-W. Fu, L. Zhu, and P.-A. Heng, "Depth-attentional features for single-image rain removal," in *Proc. CVPR*, Jun. 2019, pp. 8022–8031.

[51] D. Gehrig, M. Rüegg, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, "Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2822–2829, Apr. 2021.

[52] A. R. Sekkat et al., "SynWoodScape: Synthetic surround-view fisheye camera dataset for autonomous driving," 2022, *arXiv:2203.05056*.

[53] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 1693–1700.

[54] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. CVPR*, 2015.

[55] R. Kesten et al., "Level 5 perception dataset 2020," Woven Planet Holdings, Tokyo, Japan, 2019. [Online]. Available: https://level-5.global/level5/data/

[56] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "TrafficPredict: Trajectory prediction for heterogeneous traffic-agents," in *Proc. AAAI*, vol. 33, Jul. 2019, pp. 6120–6127.

[57] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "Semantic3D.Net: A new large-scale point cloud classification benchmark," in *Proc. ISPRS Ann.*, 2017, pp. 91–98.

[58] Y. Pan, B. Gao, J. Mei, S. Geng, C. Li, and H. Zhao, "SemanticPOSS: A point cloud dataset with large quantity of dynamic instances," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct./Nov. 2020, pp. 687–693.

[59] P. Jiang and S. Saripalli, "LiDARNet: A boundary-aware domain adaptation model for point cloud semantic segmentation," 2020, *arXiv:2003.01174*.

[60] J. Zhang, C. Ma, K. Yang, A. Roitberg, K. Peng, and R. Stiefelhagen, "Transfer beyond the field of view: Dense panoramic semantic segmentation via unsupervised domain adaptation," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9478–9491, Jul. 2022.

[61] A. R. Sekkat, Y. Dupuis, P. Vasseur, and P. Honeine, "The omniscape dataset," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2020, pp. 1603–1608.

[62] H. Shi et al., "PanoFlow: Learning 360° optical flow for surrounding temporal understanding," 2022, *arXiv:2202.13388*.

[63] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia, "Unsupervised learning of geometry from videos with edge-aware depth-normal consistency," in *Proc. AAAI*, Feb. 2018, vol. 32, no. 1, pp. 1–8.

[64] V. Guizilini, J. Li, R. Ambruş, and A. Gaidon, "Geometric unsupervised domain adaptation for semantic segmentation," in *Proc. ICCV*, Oct. 2021, pp. 8537–8547.

[65] L. C. Chen, G. Papandreou, and I. Kokkinos, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Jun. 2017.

[66] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 833–851.

[67] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, Jun. 2015, pp. 3431–3440.

[68] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. CVPR*, Jul. 2017, pp. 2881–2890.

[69] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015, pp. 234–241.

[70] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. NIPS*, vol. 34, 2021, pp. 12077–12090.

[71] X. Zhu et al., "Cylindrical and asymmetrical 3D convolution networks for lidar-based perception," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6807–6822, Oct. 2022.

[72] S. Lee, "Depth camera image processing and applications," in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep./Oct. 2012, pp. 545–548.

[73] T. H. Thai, R. Cogranne, and F. Retraint, "Camera model identification based on the heteroscedastic noise model," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 250–263, Jan. 2014.

[74] A. Haider and H. Hel-Or, "What can we learn from depth camera sensor noise?" *Sensors*, vol. 22, no. 14, p. 5448, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/14/5448

[75] K.-C. Chang et al., "Learning camera-aware noise models," in *Computer Vision–(ECCV)*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, 2020, pp. 343–358.

[76] A. Milioto, L. Mandtler, and C. Stachniss, "Fast instance and semantic segmentation exploiting local connectivity, metric learning, and one-shot detection for robotics," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, May 2019, pp. 5481–5487.

[77] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, "SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, May 2019, pp. 4376–4382.

[78] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[79] H. Liu, J. Zhang, K. Yang, X. Hu, and R. Stiefelhagen, "CMX: Cross-modal fusion for RGB-X semantic segmentation with transformers," 2022, *arXiv:2203.04838*.
[80] 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. CVPR*, Jun. 2018, pp. 7472–7481.

Paolo Testolina (Student Member, IEEE) received the Ph.D. degree in information engineering from the University of Padova in 2022. He spent research periods with Northeastern University, Boston, MA, USA. He is currently a Post-Doctoral Researcher with the University of Padova. His research interests include mmWave networks, from channel modeling to link layer simulation, traffic modeling, radio frequency interference analysis, and vehicular networks.

Francesco Barbato (Student Member, IEEE) received the M.Sc. degree (Hons.) in telecommunication engineering from the University of Padova in 2020, where he is currently pursuing the Ph.D. degree. His research interests include unsupervised domain adaptation, multi-modal learning, and continual learning applied to computer vision tasks, particularly to semantic segmentation for autonomous vehicles.

Umberto Michieli (Graduate Student Member, IEEE) received the Ph.D. degree in information engineering from the University of Padova in 2021. He spent research periods with Technische Universität Dresden and Samsung Research, U.K. He is currently a Post-Doctoral Researcher and an Adjunct Professor with the University of Padova. His research interests include intersection of foundation AI problems applied to semantic understanding. In particular, he focuses on domain adaptation, continual learning, coarse-to-fine learning, and federated learning.

Marco Giordani (Member, IEEE) is currently an Assistant Professor (tenure-track) with the University of Padova, Italy. During his Ph.D., he visited NYU and TOYOTA Info Technology Center Inc., USA. His research interests include protocol design for 5G/6G wireless networks. He was a recipient of several awards, including the 2018 IEEE Daniel E. Noble Fellowship Award from the IEEE Vehicular Technology Society and the 2021 IEEE ComSoc Outstanding Young Researcher Award for EMEA. He is currently serves as an Editor for the IEEE TRANSACTIONS OF WIRELESS COMMUNICATIONS.

Pietro Zanuttigh (Member, IEEE) received the Ph.D. degree from the University of Padova in 2007. He is currently an Associate Professor with the Department of Information Engineering. He works in the computer vision and machine learning fields, with a special focus on domain adaptation and incremental learning in semantic segmentation, 3D acquisition with ToF sensors, depth data processing, sensor fusion, and hand gesture recognition.

Michele Zorzi (Fellow, IEEE) is currently with the Department of Information Engineering, University of Padova, focusing on wireless communications. He served ComSoc as a Member-at-Large for the Board of Governors from 2009 to 2011 and from 2021 to 2023, the Director for Education and Training from 2014 to 2015, and the Director for Journals from 2020 to 2021. He was the Editor-in-Chief of IEEE WIRELESS COMMUNICATIONS from 2003 to 2005, IEEE TRANSACTIONS ON COMMUNICATIONS from 2008 to 2011, and IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING from 2014 to 2018.

Open Access funding provided by 'Università degli Studi di Padova' within the CRUI CARE Agreement