Multimodal End-to-End Autonomous Driving

Yi Xiao, Felipe Codevilla, Akhil Gurram, Onay Urfalioglu, Antonio M. López

***Abstract*—A crucial component of an autonomous vehicle (AV) is the artificial intelligence (AI) is able to drive towards a desired destination. Today, there are different paradigms addressing the development of AI drivers. On the one hand, we find modular pipelines, which divide the driving task into sub-tasks such as perception and maneuver planing and control. On the other hand, we find end-to-end driving approaches that try to learn a direct mapping from input raw sensor data to vehicle control signals. The later are relatively less studied, but are gaining popularity since they are less demanding in terms of sensor data annotation. This paper focuses on end-to-end autonomous driving. So far, most proposals relying on this paradigm assume RGB images as input sensor data. However, AVs will not be equipped only with cameras, but also with active sensors providing accurate depth information (*e.g.*, LiDARs). Accordingly, this paper analyses whether combining RGB and depth modalities, *i.e.* using RGBD data, produces better end-to-end AI drivers than relying on a single modality. We consider multimodality based on early, mid and late fusion schemes, both in multisensory and single-sensor (monocular depth estimation) settings. Using the CARLA simulator and conditional imitation learning (CIL), we show how, indeed, early fusion multimodality outperforms single-modality.**

***Index Terms*—Multimodal scene understanding, End-to-end autonomous driving, Imitation learning.**

I. Introduction

AUTONOMOUS vehicles (AVs) are core for future mobility. Thus, it is essential to develop artificial intelligence (AI) for driving AVs. Two main paradigms are under research, namely, *modular pipelines* and *end-to-end driving*.

The modular paradigm attaches to the traditional divide-and-conquer engineering principle, since AI drivers rely on modules with identifiable responsibilities; for instance, to provide environmental perception [1, 2], as well as route planning and maneuver control [3, 4]. Perception itself is already especially complex, since it involves sub-tasks such as object detection [5–10] and tracking [11–15], traffic sign recognition [16], semantic segmentation [17–22], monocular depth estimation [23–26], SLAM and place recognition [27–33], etc.

The end-to-end driving paradigm focuses on learning holistic models is able to directly map raw sensor data into control signals for maneuvering AVs [34–37], *i.e.* without forcing explicit sub-tasks related to perception or planning. Thus, advocating for learning to perceive and act simultaneously, as humans do. Moreover, such sensorimotor models are obtained through a data-driven supervised learning process as is characteristic of modern AI. End-to-end driving models can accept high-level navigation commands [38–41], or be restricted to specific navigation sub-tasks such as lane keeping [42–44] and longitudinal control [45].

Driving paradigms are highly relying on convolutional neural netwoks (CNNs). In this context, one of the main advantages of modular pipelines is the ability to explain the decisions of the AI driver in terms of its modules; which is more difficult for pure end-to-end driving models [46–48]. However, developing some of the critical modules of the modular paradigm requires hundreds of thousands of supervised data samples [49, 50], *e.g.* raw sensor data with ground truth (GT). Since the GT is most of the times provided manually (*e.g.* annotation of object bounding boxes [51], pixel-level delineation of semantic classes [52]), this is an important bottleneck for this paradigm. Conversely, end-to-end approaches are able to learn CNN-based models for driving from raw sensor data (*i.e.* without annotated GT) and associated supervision in terms of vehicle's variables (*e.g.* steering angle, speed, geo-localization and orientation [37, 53, 54]); note that such supervision does not require human intervention in terms of explicitly annotating the content of the raw sensor data. Moreover, end-to-end models are demonstrating an *unreasonable* effectiveness in practice [36, 39, 43, 45], which makes worth to go deeper in their study.

Although AVs will be multisensory platforms, equipping and maintaining on-board synchronized heterogeneous sensors is quite expensive nowadays. As a consequence, most end-to-end models for driving rely only on vision [35–37, 39, 42, 43, 45, 55–58], *i.e.* they are visuomotor models. This is not bad in itself, after all, human drivers mainly rely on vision. However, multimodality has shown better performance in key perception sub-tasks such as object detection [7–10, 59–63], tracking [15], and semantic segmentation [21, 22]. Thus, it is worth exploring multimodality for end-to-end driving.

Accordingly, in this paper we address the question *can an end-to-end driving model be improved by using multimodal sensor data over just relying on a single modality?* In particular, we assume color images (RGB) and depth (D) as single modalities, and RGBD as multimodal data. Due to its capability of accepting high level commands, this study is based on the CNN architecture known as *conditional imitation learning* (CIL) [39]. We explore RGBD from the perspective of early, mid and late fusion of the RGB and D modalities. Moreover, as in many recent works on end-to-end driving [39–41, 48, 56, 57, 64], our experiments rely on the CARLA simulator [65].

The presented results show that multimodal RGBD end-to-end driving models outperform their single-modal counterparts. Moreover, early fusion shows better performance than

Yi, Felipe, and Antonio are with the Computer Vision Center (CVC) and the Univ. Autònoma de Barcelona (UAB). Akhil and Onay are with Huawei GRC in Munich. CVC members acknowledge the financial support by the Spanish TIN2017-88709-R (MINECO/AEI/FEDER, UE). Antonio M. López acknowledges the financial support by ICREA under the ICREA Academia Program. Felipe Codevilla acknowledges Catalan AGAUR for his FI grant 2017FI-B1-00162. Yi Xiao acknowledges the Chinese Scholarship Council (CSC), grant number 201808390010. We also thank the Generalitat de Catalunya CERCA Program, as well as its ACCIO agency.

mid and late fusion. On the other hand, multisensory RGBD (*i.e.* based on camera and LiDAR) still outperforms monocular RGBD; however, we conclude that it is worth pursuing this special case of single-sensor multimodal end-to-end models.

We present the work as follows. Sect. II reviews the related literature. Sect. III presents the used CIL architecture from the point of view of early, mid, and late fusion. Sect. IV summarizes the experimental setting and the obtained results. Finally, Sect. V draws the main conclusions and future work.

II. Related work

This section focuses on two main related topics: *multimodal perception* and *end-to-end driving models learned by imitation*.

A. Multimodality

Object detection is one of the perception tasks for which multimodality has received most attention. Enzweiler *et al.* [66] developed a pedestrian detector using hand-crafted features and shallow classifiers combined as a mixture-of-experts (MoE), where multimodality relies on image luminance and stereo depth. Gonzalez *et al.* [60] detected vehicles, pedestrians and cyclists—vulnerable road users (VRUs)—, using a multimodal MoE based on space-time calibrated RGB and LiDAR depth. Chen *et al.* [61] used calibrated RGB and LiDAR depth as input for a CNN-based detector of vehicles and VRUs, which is a current trend [8, 9, 61–63]. Some of these works are inspired by Faster R-CNN [5], since they consist of a first stage for proposing regions potentially containing objects of interest, and a second stage performing the classification of those regions to provide final object detections; *i.e.* following a mid-level (deep) fusion scheme where CNN layers of features from the different modalities are fused in both stages [61–63]. Other alternatives are early fusion at raw data level [9], late fusion of independent detectors [8, 9], or just using different modalities at separated steps of the detection pipeline [7]. Other approaches focus on multispectral appearance, as in Li *et al.* [10], where different fusion schemes for RGB and Far Infrared (FIR) calibrated images are compared.

All these studies and recent surveys [67, 68] show that detection accuracy increases with multimodality. Therefore, more perception tasks have been addressed under the multimodal approach. Dimitrievski *et al.* [15] proposed a pedestrian tracker that fuses camera and LiDAR detections to solve the data association step of their tracking-by-detection approach. Schneider *et al.* [21] proposed a CNN architecture for semantic segmentation which performs a mid-level fusion of RGB and stereo depth, leading to a more accurate segmentation on small objects. Ha *et al.* [22] also proposed a mid-level RGB and FIR fusion approach in a CNN architecture for semantic segmentation. Piewak *et al.* [69] used a mid-level fusion of LiDAR and camera data to produce a Stixel representation of the driving scene, showing improved accuracy in terms of geometry and semantics of the resulting representation.

In this paper, rather than focusing on individual perception tasks such as object detection, tracking or semantic segmentation, we challenge multimodality in the context of end-to-end driving, exploring early, mid and late fusion schemes.

B. End-to-end driving

Pomerleau presented ALVINN three decades ago [34], a sensorimotor fully-connected shallow neural network that was able to perform end-to-end road following assuming no obstacles. ALVINN controlled a CMU's van, NAVLAB, along a 400m straight path, at ~ 2 Km/h and under good weather conditions. Although the addressed scenario is extremely simple compared to driving in real traffic, it was already necessary to simulate data for training the sensorimotor model and, in fact, camera images (30×32 pixels, blue channel) were already combined by early fusion with laser range finder data (8×32 depth cells). LeCun *et al.* [35] trained end-to-end a 6-layer CNN for off-road obstacle avoidance using image pairs (from a stereo rig) as input. Such CNN was able to control a 50cm-length four-wheel truck, DAVE, for avoiding obstacles at a speed of ~ 7 Km/h. During data collection for training, the truck was remotely controlled by a human operator, thus, the CNN was trained according to imitation learning in our terminology (or teleoperation-based demonstration [70]). More recently, Bojarski *et al.* [36] developed a vision-based end-to-end driving CNN which was able to control the steering wheel of a real car in different traffic conditions. Still, lane and road changing are not considered, neither stop-and-go maneuvers since throttle and brake are not controlled.

These pioneering works inspired new proposals based on imitation learning for CNNs. Eraqi *et al.* [44] applied vision-based end-to-end control of the steering angle (neither throttle nor break), focusing on including temporal reasoning by means of long short-term memory recurrent neural networks (LSTMs). Training and testing were done in the Comma.ai dataset [53]. George *et al.* [45] applied similar ideas for controlling the speed of the car. Xu *et al.* [37] presented the BDD dataset and focused on vision-based prediction of the steering angle using a fully convolutional network (FCN) and a LSTM, forcing semantic segmentation as auxiliary training task. Innocenti *et al.* [42] performed vision-based end-to-end steering angle prediction for lane keeping on private datasets, and Chen *et al.* [43] in the Comma.ai dataset.

Affordances have been proposed as intermediate tasks between enviromental perception and prediction of the vehicle control parameters [55, 56]. Affordances do not require to solve perception sub-tasks such as explicit object detection, etc; but they form a compact set of factors that influence driving according to prior human knowledge. Chen *et al.* [55] evaluated them on the TORCS simulator [71], so in car racing conditions (no pedestrians, no intersections, etc.) under clean and dry weather; while Sauer *et al.* [56] used the CARLA simulator, which supports regular traffic conditions under different lighting and weather [65]. Muller *et al.* [57] developed a vision-based CNN with an intermediate road segmentation task for learning to perform vehicle maneuvers in a semantic space; the driving policy consists of predicting waypoints within the segmented road and applying a low-level PID controller afterwards. Training and testing are done in CARLA, but neither incorporating other vehicles nor pedestrians. Using LiDAR data, Rhinehart *et al.* [64] combined imitation learning and model-based reinforcement learning to

predict expert-like vehicle trajectories, relying on CARLA but without dynamic traffic participants.

These end-to-end driving models do not accept high-level navigation instructions such as *turn left at the next intersection* (without providing explicit distance information), which can come from a global planner or just as voice commands from a passenger of the AV. Hubschneider *et al.* [38] proposed to feed a turn indicator in the vision-based CNN driving model by concatenating it with features of a mid-level fully connected layer of the CNN. Codevilla *et al.* [39] proposed a more effective method, in which a vision-based CNN consisting of an initial block agnostic to particular navigation instructions, and a second block branched according to a subset of navigation instructions (at next intersection turn-left/turn-right/go-straight, or just keep lane). In the first block, vehicle information is also incorporated as mid-level feature of the CNN; in particular, current speed is used since the CNN controls the steering angle, throttle, and break (Yang *et al.* [58] also reported the usefulness of speed feedback in end-to-end driving). Experiments are performed in CARLA for different traffic situations (including other vehicles and pedestrians), lighting and weather conditions. The overall approach is termed as conditional imitation learning (CIL). In fact, Muller *et al.* and Sauer *et al.* leveraged from CIL. Liang *et al.* [41] also used CIL as imitation learning stage before refining the resulting model by applying reinforcement learning. Wang *et al.* [40] used CIL too, but incorporating ego-vehicle heading information at the same CNN-layer level as speed.

These works focus on vision-based end-to-end driving. Here, we explore muldimodal end-to-end driving based on RGB and depth; which can be complementary to most of the cited papers. Without losing generality, we chose CIL as core CNN architecture due to its effectiveness and increasing use.

Focusing on multimodality, Sobh *et al.* [72] used CARLA to propose a CIL-based driving approach modified to process camera and LiDAR data. In this case, the information fusion is done by a mid-level approach; in particular, before fusion, RGB images are used to generate a semantic segmentation which corresponds to one of the information streams reaching the fusion layers, and there are two more independent streams based on LiDAR, one encoding a bird view and the other a polar grid mapping. Khan *et al.* [73] also used CARLA to propose an end-to-end driving CNN based on RGB and depth images, which predicts only the steering angle, assuming that neither other vehicles nor pedestrians are present. In a first step, the CNN is trained only using depth information (taken as the Z-buffer produced by UE4, the game engine behind CARLA). This CNN has an initial block of layers (CNN encoder) that outputs depth-based features, which are later used to predict the steering angle with a second block of fully connected layers. In a second step, this angle-prediction block is discarded and replaced by a new fully connected one. This new block relies on the fusion of the depth-based features and a semantic segmentation produced by a new CNN block that processes the RGB image paired with the depth image. During training, semantic segmentation is conditioned to depth-based features due to the fusion block and back-propagation. This approach can be considered a type of mid-level fusion.

In contrast to these multimodal end-to-end driving approaches, we assess early, mid and late level fusion schemes without forcing intermediate representations which are not trivial to obtain (*e.g.* semantic segmentation is an open problem in itself). Moreover, we run CARLA benchmark [65], which includes dynamic actors (vehicles and pedestrians) and generalization conditions (unseen town and weather). We show that CIL and early fusion produce state-of-the-art results.

III. Multimodal Fusion

We first detail CIL [39], and then show how we adapt it to leverage from multimodal perception data.

A. Base CIL architecture

Fig. 1 shows the CNN implementing CIL. The observations (CIL's input) are twofold, perception data, $\mathbf{p}$, and vehicle's state measurements, $\mathbf{m}$. The action (CIL's output), $\mathbf{a}$, consists of vehicle controls for maneuvering. CIL includes a CNN block to extract perception features, $P(\mathbf{p})$; and a block of fully connected layers to extract measurement features $M(\mathbf{m})$. A joint layer of features is formed by appending $P(\mathbf{p})$ and $M(\mathbf{m})$; which is further processed by a new fully connected layer to obtain the joint features $J(<P(\mathbf{p}), M(\mathbf{m})>)$, or just $J(\mathbf{p}, \mathbf{m})$ simplifying the notation. Up to this point of the neural network, the processing done with the observations is common to any driving maneuver/action. However, many times, the autonomous vehicle reaches ambiguous situations which require to incorporate informed decisions. For instance, when reaching a cross intersection, without incorporating a route navigation command (*e.g.* from a global trajectory plan), the vehicle could only take a random decision about turning or going straight. Thus, the end-to-end driving CNN must incorporate high-level commands, c, such as *'in the next intersection turn left'*, or *'turn right'*, or *'go straight'*. Moreover, $\mathbf{a}$ will take very different values depending on c. Thus, provided c takes discrete values, having specialized neural network layers for each maneuver can be more accurate a priori. All this is achieved in the CIL proposal by incorporating fully connected maneuver/action branches, A^c, selected by c (both during CNN training and vehicle self-driving).

We follow the CIL architecture proposed in [39]. Therefore, $\mathbf{p}$ is a RGB image of 200×88 pixels and 8 bits at each color channel, $\mathbf{m}$ is a real value with the current speed of the vehicle, and $\mathbf{a}$ consists of three real-valued signals which set the next maneuver in terms of steering angle, throttle, and brake. Thus, the idea is to perform vision-based self-driving, as well as taking into account the vehicle speed to apply higher/lower throttle and brake for the same perceived traffic situation. In [39], the focus is on handling intersections, then the considered c values are $\{$*turn-left, turn-right, go-straight, continue*$\}$, where the last refers to just keep driving in the current lane and the others inform about what to do when reaching next intersection (which is an event detected by the own CNN). Accordingly, there are four branches A^c. Therefore, if we term by F the end-to-end driver, we have $F(\mathbf{p}, \mathbf{m}, c) = A^c(J(\mathbf{p}, \mathbf{m}))$. As shown in [39], this manner of explicitly taking into account

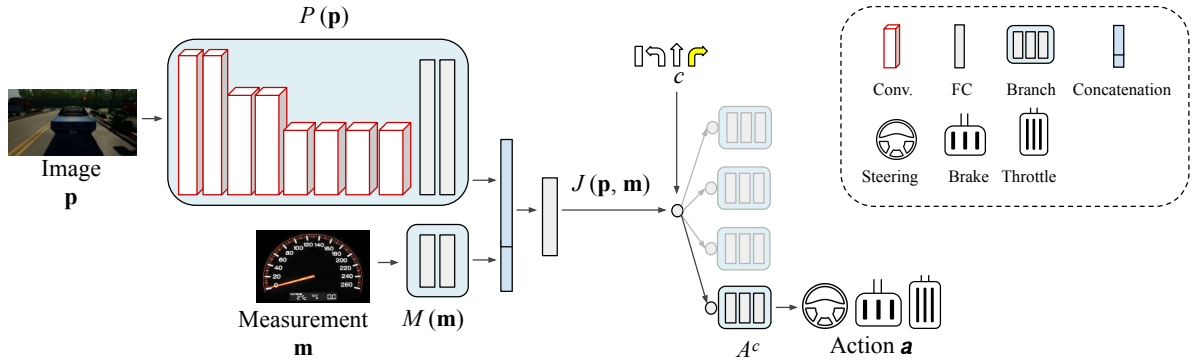

Fig. 1. CIL branched architecture: vehicle maneuvers (actions) in the form of the triplet <steering angle, throttle, brake>, depend on a high-level route navigation command (branch selector) running on {turn-left, turn-right, go-straight, continue}, as well as on the world observations in the form of perception data (*e.g.* a RBG image) and vehicle state measurements (*e.g.* speed).

high-level navigation commands is more effective than other alternatives.

B. Fusion schemes

Fig. 2 illustrates how we fuse RGB and depth information following mid, early and late fusion approaches.

Early fusion: with respect to the original CIL we only change the number of channels of $\mathbf{p}$ from three (RGB) to four (RGBD). The CIL network only changes the first convolutional layer of $P(\mathbf{p})$ to accommodate for the extra input channel, the rest of the network is equal to the original.

*Mid fusion:*we replicate twice the perception processing $P(\mathbf{p})$. One of the $P(\mathbf{p})$ blocks processes only RGB images, the other one only depth images. Then, we build the joint feature vector $<P(\text{RGB}), P(\text{D}), M(\mathbf{m})>$ which is further processed to obtain $J(\text{RGB}, \text{D}, \mathbf{m})$. From this point, the branched part of CIL is the same as in the original architecture.

Late fusion: we replicate twice the full CIL architecture. Thus, RGB and depth channels are processed separately, but the measurements are shared as input. Hence, we run $A^c(J(\text{RGB}, \mathbf{m}))$ and $A^c(J(\text{D}, \mathbf{m}))$, and their outputs are concatenated and further processed by a module of fully connected layers, the output of which conveys the final action values. Note that this is a kind of mixture-of-experts approach, where the two experts are jointly trained.

As is common practice in the literature, we assume a pixel-level correspondence of all channels and normalize all of them to be in the same magnitude range (we normalize depth values to match the range of color channels, *i.e.* $[0 \dots 255]$).

C. Loss function

Given a predicted action $\mathbf{a}$, its ground truth $\mathbf{a}^{gt}$, and a vector of weights $\mathbf{w}$, we use the L1 loss $\ell_{act}(\mathbf{a}, \mathbf{a}^{gt}, \mathbf{w}) = \sum_i^n |w_i(a_i - a_i^{gt})|$, with $n = 3$ (steering angle, throttle, brake). Note that when computing $\mathbf{a}$, only one A^c branch is active at a time. In particular, the one selected by the particular command c associated to the current input data $(\mathbf{p}, \mathbf{m})$. We make this fact explicit by changing the notation to $\ell_{act}(\mathbf{a}, \mathbf{a}^{gt}, \mathbf{w}; c)$.

In addition, as in other computer vision problems addressed by deep learning [74, 75], we empirically found that using multi-task learning helps to obtain more accurate CIL networks. In particular, we add an additional branch of three fully connected layers to predict current vehicle speed from the perception data features $P(\mathbf{p})$. This prediction relies on a L1 loss $\ell_{sp}(s, s^{gt}) = |s - s^{gt}|$, where s is the predicted speed and s^{gt} is the ground truth speed which, in this case, is already available since it corresponds to the measurement used as input. Speed prediction is only used during training.

Thus, all networks, *i.e.* both single- and multimodal, are trained according to the same total loss $\ell(\mathbf{a}, \mathbf{a}^{gt}, \mathbf{w}; c; s, s^{gt}) = \beta\ell_{act}(\mathbf{a}, \mathbf{a}^{gt}, \mathbf{w}; c) + (1 - \beta)\ell_{sp}(s, s^{gt})$, where β is used to balance the relevance of ℓ_{act} and ℓ_{sp} losses.

IV. Experiments

We start by summarizing the environment we use for our experiments, *i.e.* CARLA (Sect. IV-A). Next, we describe the driving benchmark available in CARLA (Sect. IV-B), the dataset we use for training our AI drivers (Sect. IV-C), and the training protocol that we follow (Sect. IV-D). Finally, we present and discuss the obtained results (Sect. IV-E).

A. Environment

In order to conduct our experiments, we rely on the open source driving simulator CARLA [65]. There are several reasons for this. First, many recent previous works on end-to-end driving rely on CARLA [39–41, 48, 56, 57, 64]; thus, we can compare our results with the previous literature. Second, it seems that for some scenarios the end-to-end paradigm may need exponentially more training samples than the modular one [76], so there is a trade-off between collecting driving runs (for the end-to-end paradigm) and manually annotating on-board acquired data (for the modular paradigm) which, together with the gigantic effort needed to demonstrate that an AV outperforms human drivers, really encourages to rely on simulators during the development of AI drivers [77]. Yet, a third and core reason is specific for end-to-end driving models. In particular, in [78] it is demonstrated that current offline evaluation metrics (*i.e.* based on static datasets) for assessing end-to-end driving models do not correlate well-enough with

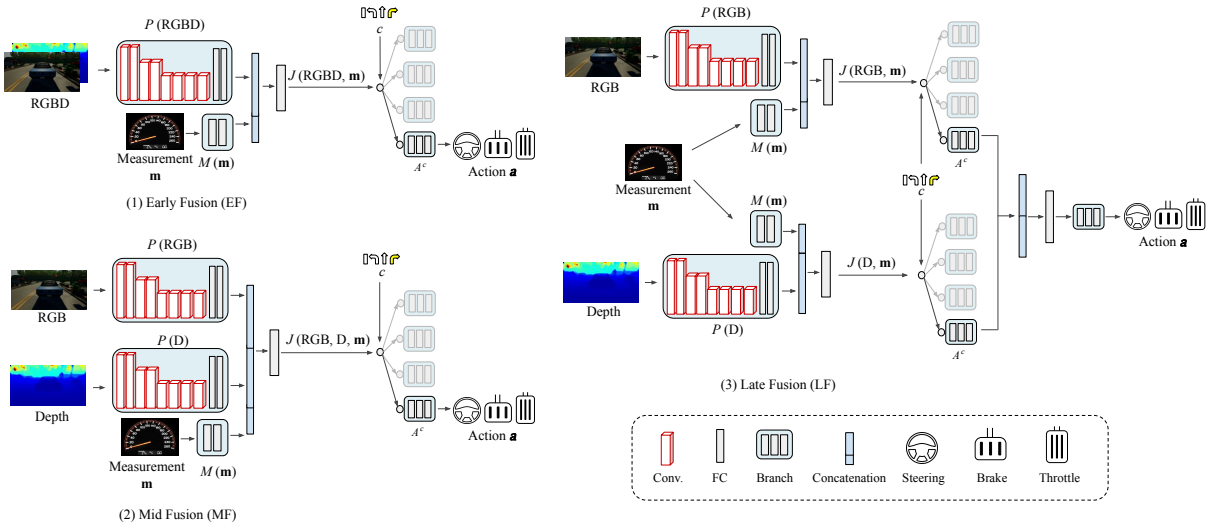

Fig. 2. Network Architectures - we explore RGBD from the perspective of early, mid and late fusion of the RGB and Depth (D) modalities. (1) Early Fusion: the raw RGB and D channels are the input of the CIL architecture; (2) Mid Fusion: intermediate CIL feature layers from RGB and D streams are fused; (3) Late Fusion: the output (maneuver controls) of the RGB and D CIL streams are fused to output the final values after further neural processing.

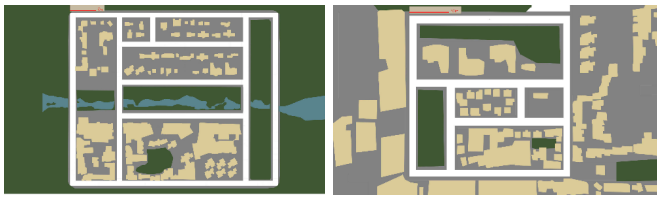

Fig. 3. Bird-view road maps of Town 1 (left) and Town 2 (right).

Fig. 4. Top, from Town 1: clear noon (left) and clear after rain (right). Middle, from Town 1: heavy rain noon (left) and clear sunset (right). Bottom, from Town 2: wet cloudy noon (left) and soft rainy sunset (right).

actual driving, an observation also seen in [79]; therefore, it is really important to evaluate these driving models in an on-board driving regime, which is possible in a realistic simulator such as CARLA.

Briefly, CARLA contains two towns (Fig. 3) based on two-directional roads with turns and intersections, buildings, vegetation, urban furniture, traffic signs, traffic lights, and dynamic objects such as vehicles and pedestrians. Town 1 deploys 2.9Km of road and 11 intersections, while Town 2 contains 1.4Km of road and 8 intersections. The different towns can be travelled under six different weather conditions (Fig. 4): 'clear noon', 'clear after rain', 'heavy rain noon', and 'clear sunset', 'wet cloudy noon' and 'soft rainy sunset'.

B. Driving benchmark

CARLA was deployed with a benchmarking infrastructure for assessing the performance of AI drivers [65]. Thus, it has been used in the related literature since then and we follow it here too. Four increasingly difficult *driving tasks* are defined:

- *straight*: the destination point is straight ahead from the starting point but no dynamic objects are present;
- *one turn*: destination is one turn away from the starting point, no dynamic objects;
- *navigation*: no restriction on the location of the destination and starting points, no dynamic objects;
- *navigation with dynamic obstacles*.

For each driving task, an AI driver is assessed over a total of E_T driving episodes. Each episode has its own starting and destination points with an associated topological route. An episode is considered as successful if the AI driver completes the route within a time budget. Collisions do not lead to the termination of an episode unless the AV runs in time-out as a consequence. If we term as E_S the total number of successfully completed episodes by the assessed AI driver, then its *success rate* is defined as $100 \times (E_S/E_T)$. E_T is determined by the selected town and weather conditions.

Table I shows how the benchmark organizes towns and weather conditions for training, validation, and testing; where,

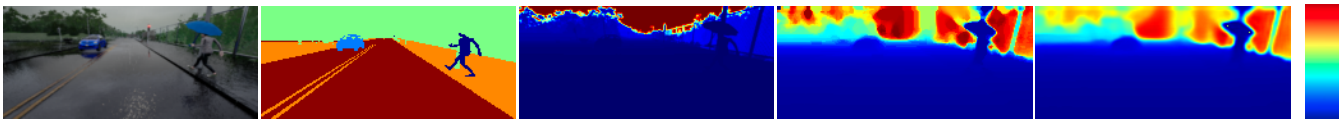

Fig. 5. From left to right: original RGB image, semantic segmentation ground truth (for the five considered classes), CARLA depth ground truth, post-processed to be closer to the capabilities of an *active* depth Sensor, and monocular depth estimation from a model trained using such a depth.

TABLE I
TRAINING, VALIDATION, AND TESTING SETTINGS. TRAINING IS BASED ON A PRE-RECORDED DATASET. VALIDATION AND TESTING ARE BASED ON ACTUAL DRIVING EPISODES. GREY MEANS 'NOT USED'.

	Training (dataset)	Validation (episodes)	Testing (episodes)
Wet cloudy noon			
Soft rainy sunset		Towns 1 & 2	
Clear noon	Town 1		Towns 1 & 2
Clear after rain			
Clear sunset			
Heavy rain noon			

TABLE II
V_P FOR FIVE TRAINING RUNS FOR RGB ONLY, DEPTH (D) ONLY, AND RGBD COMBINED BY EARLY (EF), MID (MF), OR LATE (LF) FUSION. DEPTH: FROM AN ACTIVE SENSOR OR ESTIMATED FROM RGB IMAGES.

	RGB	Active				Estimation	
		D	EF	MF	LF	D	EF
1	**48**	**74**	**91**	61	60	51	42
2	36	67	71	71	63	49	44
3	46	73	75	58	**67**	46	**51**
4	40	68	71	**74**	60	**59**	46
5	36	68	77	52	62	51	49

irrespective of the town and weather, validation and testing is always based on episodes, not in pre-recorded datasets, while training requires pre-recording a dataset. Validation is performed to select a driving model among those trained as different trials from the same training dataset, while testing is performed for actually benchmarking the selected models.

Regarding town and weather conditions, the benchmark establishes four main town-weather blocks under which the four driving tasks must be tested, assuming 25 episodes for each considered weather. Therefore, for each block, the E_T value is different as we can deduce from Table I. In particular, these are the town-weather blocks defined in the benchmark with their respective E_T value:

Training conditions: driving (*i.e.* running the episodes) in the same conditions as the training set (Town 1, four weather conditions), thus, $E_T = 100$;

New town: driving under the four weather conditions of the training set but in Town 2, $E_T = 100$;

New weather: driving in Town 1 but under the two weather conditions not seen at training time, $E_T = 50$;

New town & weather: driving in conditions not included in the training set (Town 2, two weather conditions), $E_T = 50$.

C. Training dataset

In order to train our CNNs, we use the same dataset as in [78] since it corresponds to 25h of driving in Town 1, balancing weather conditions (Table I). Briefly, this dataset was collected by a hard-coded auto-pilot with access to all the privileged information of CARLA required for driving like an expert. The auto-pilot kept a constant speed of 35 km/h when driving straight and reduced the speed when making turns. Images were recorded at 10fps from three cameras: a central forward-facing one and two lateral cameras facing 30° left and right. The central camera is the only one used for self-driving, while the images coming from the lateral cameras are used only at training time to simulate episodes of recovering from driving errors as can be done with real cars [36] (the protocol for injecting noise follows [39]). Overall, the dataset contains ~ 2.5 millions of RGB images of 800×600 pixels resolution, with associated ground truth (see Fig. 5) consisting of corresponding images of dense depth and pixel-wise semantic classes (semantic segmentation), as well as meta-information consisting of the high-level commands provided by the navigation system (continue in lane, at next intersection go straight, turn left or turn right), and car information such as speed, steering angle, throttle, and braking. In this work, we use perfect semantic segmentation to develop an upper-bound driver. Since we focus on end-to-end driving, the twelve semantic classes of CARLA are mapped to five which we consider sufficient to develop such an upper-bound. In particular, we keep the original *road-surface*, *vehicle*, and *pedestrian*, while *lane-marking* and *sidewalk* are mapped as *lane-limits* (Town 1 and Town 2 only have roads with one go and one return lane, separated by double continuous lines), and the remaining seven classes are mapped as *other*.

Focusing on depth information, as is common in the literature, we assume that RGB images have associated dense depth information; for instance, Premebida *et al.* [80] obtained it from LiDAR point clouds. In CARLA, the depth ground truth is extremely accurate since it comes directly from the Z-buffer used during simulation rendering. In particular, depth values run from 0 to 1,000 meters and are codified with 24 bits, which means that depth precision is of $\sim 1/20$ mm. This distance range coverage and depth precision is far beyond from what even *active* sensors can provide. Therefore, we post-process depth data to make it more realistic. In particular, we take as a realistic sensor reference the Velodyne information of KITTI dataset [51]. First we trim depth values to consider only those within the 1 to 100 meters interval, *i.e.* pixels of the depth image with values outside this range are considered as not having depth information. Second, we re-quantify the depth values to have an accuracy of ~ 4 cm. Third, we perform inpainting to fill-in the pixels with no information. Finally, we apply a median filter to avoid having perfect depth boundaries between objects. The new depth images are used both during training and testing. Fig. 5 shows an example of a depth image from CARLA and its corresponding post-processed version.

During a training run we use Adam optimizer with 120 training samples per iteration (minibatch), an initial learning

TABLE III

MEAN AND STANDARD DEVIATION OF SUCCESS RATES ON THE ORIGINAL CARLA BENCHMARK, BY RUNNING IT THREE TIMES. CIL BASED ON PERFECT SEMANTIC SEGMENTATION (SS) ACTS AS UPPER BOUND. EXCLUDING SS, FOR MODELS TESTED UNDER THE SAME ENVIRONMENT AND TRAFFIC CONDITIONS, WE SHOW IN BOLD THE HIGHER MEANS AND WE UNDERLINE SIMILAR SUCCESS RATES CONSIDERING STANDARD DEVIATIONS TOO.

	Active						Estimated		Active						Estimated	
Task	SS	RGB	D	EF	MF	LF	D	EF	SS	RGB	D	EF	MF	LF	D	EF
	Training Conditions								New Town							
Straight	98.00 ± 1.73	96.33 ± 1.53	98.67 ± 1.53	98.33 ± 0.58	92.33 ± 2.08	**99.00** ± 0.00	92.33 ± 1.15	97.33 ± 1.15	100.00 ± 0.00	84.00 ± 2.00	94.33 ± 0.58	**96.33** ± 0.58	87.00 ± 1.00	77.00 ± 0.00	78.33 ± 1.53	71.67 ± 2.08
One turn	100.00 ± 0.00	95.00 ± 0.00	92.00 ± 0.00	**99.00** ± 0.00	91.67 ± 2.08	90.33 ± 0.58	84.67 ± 1.15	96.33 ± 1.53	96.67 ± 0.58	68.00 ± 1.00	74.33 ± 2.52	**79.00** ± 1.73	78.00 ± 2.65	58.67 ± 2.08	46.33 ± 1.15	47.00 ± 1.00
Navigation	96.00 ± 0.00	89.00 ± 2.00	89.33 ± 2.08	92.67 ± 1.15	90.67 ± 1.15	93.67 ± 0.58	75.33 ± 1.15	**94.33** ± 0.58	96.00 ± 0.00	59.67 ± 3.06	85.33 ± 1.15	**90.00** ± 2.00	80.67 ± 0.58	52.33 ± 0.58	45.67 ± 3.06	46.67 ± 3.06
Nav.Dynamic	92.00 ± 1.00	84.00 ± 2.00	82.67 ± 0.58	89.33 ± 0.58	78.33 ± 2.89	89.00 ± 2.65	71.00 ± 1.00	**89.67** ± 1.15	99.33 ± 0.58	54.33 ± 3.79	70.33 ± 1.15	**84.33** ± 2.52	73.67 ± 2.52	55.67 ± 2.31	44.33 ± 2.52	46.67 ± 4.04
	New Weather								New Town & Weather							
Straight	100.00 ± 0.00	84.00 ± 0.00	**99.33** ± 1.15	96.00 ± 2.00	94.67 ± 3.06	96.00 ± 0.00	92.00 ± 2.00	84.67 ± 1.15	100.00 ± 0.00	84.00 ± 1.15	**97.33** ± 1.15	**97.33** ± 2.31	88.67 ± 1.15	**97.33** ± 1.15	78.00 ± 0.00	89.33 ± 1.15
One turn	100.00 ± 0.00	76.67 ± 4.16	**94.67** ± 2.31	**94.67** ± 2.31	94.00 ± 2.00	92.00 ± 2.00	93.33 ± 2.31	80.67 ± 1.15	96.00 ± 0.00	66.67 ± 4.62	72.67 ± 1.15	**82.67** ± 2.31	69.33 ± 3.06	67.33 ± 2.31	62.67 ± 1.15	64.00 ± 3.46
Navigation	95.33 ± 1.15	72.67 ± 2.31	89.33 ± 1.15	91.33 ± 2.31	90.67 ± 3.06	**96.00** ± 0.00	73.33 ± 2.31	80.67 ± 5.03	96.00 ± 0.00	57.33 ± 6.11	84.00 ± 3.46	**92.67** ± 3.06	78.67 ± 3.06	72.67 ± 1.15	55.33 ± 6.11	60.67 ± 2.31
Nav.Dynamic	92.67 ± 1.15	68.67 ± 4.62	90.00 ± 2.00	86.00 ± 4.00	80.67 ± 3.06	**92.67** ± 3.06	76.67 ± 4.16	77.33 ± 6.11	98.00 ± 2.00	46.67 ± 6.43	69.33 ± 2.31	**94.00** ± 0.00	73.33 ± 3.06	73.33 ± 2.31	54.00 ± 4.00	49.33 ± 3.06

TABLE IV

SUCC. RATE COMPARISON WITH PREVIOUS METHODS (SEE MAIN TEXT).

Task	MP	RL	CAL	CIRL	MT	Active EF	MP	RL	CAL	CIRL	MT	Active EF
	Training Conditions						New Town					
Straight	98	89	**100**	98	98	98.33 ± 0.58	92	74	93	**100**	**100**	96.33 ± 0.58
One turn	82	34	97	97	87	**99.00** ± 0.00	61	12	**82**	71	81	79.00 ± 1.73
Navigation	80	14	92	**93**	81	92.67 ± 1.15	24	3	70	53	72	**90.00** ± 2.00
Nav.dynamic	77	7	83	82	81	**89.33** ± 0.58	24	2	64	41	53	**84.33** ± 2.52
	New Weather						New Town & Weather					
Straight	**100**	86	**100**	**100**	**100**	96.00 ± 2.00	50	68	94	**98**	96	97.33 ± 2.31
One turn	95	16	**96**	94	88	94.67 ± 2.31	50	20	72	82	82	**82.67** ± 2.31
Navigation	**94**	2	90	86	88	91.33 ± 2.31	47	6	68	68	78	**92.67** ± 3.06
Nav.dynamic	**89**	2	82	80	80	86.00 ± 4.00	44	4	64	62	62	**94.00** ± 0.00

TABLE V

INFRACTIONS ON DYNAMIC NAVIGATION IN NEW TOWN & WEATHER.

Km per	Event	RGB	Active D	Active EF
Infraction	Sidewalk	0.86 ± 0.10	35.80 ± 1.30	16.76 ± 5.54
	Opposite lane	0.73 ± 0.04	1.65 ± 0.24	3.29 ± 1.96
Driven Km	(Perfect driving: 17.30 Km)	13.62 ± 0.67	35.80 ± 1.30	20.22 ± 0.54

rate of 0.0002, decreased to the half each 50K iterations. Mini-batches are balanced in terms of per A^c branch samples. We set $\mathbf{w} = (0.5, 0.45, 0.05)$ to weight the control signals (action) in the loss function. Action and speed losses are balanced by $\beta = 0.95$. For selecting the best intermediate model of a training run, we do 500K iterations monitoring a validation performance measurement, V_P, each 100K iterations (thus, five times). The intermediate model with highest V_P is selected as the resulting model of the training run. Since CIL models are trained from the scratch, variability is expected in their performance. Thus, for each type of model we perform five training runs, finally selecting the model with the highest V_P among those resulting from the five training runs.

Using Table I as reference, we define V_P to balance training-validation differences in terms of town and weather conditions. In particular, we use $V_P = 0.25V_w + 0.25V_t + 0.50V_{wt}$; where V_w is the success rate when validating in Town 1 and 'soft rainy sunset' weather (not included in training data), V_t is a success rate when validating in Town 2 (not included in training data) and 'clear noon' weather (included in training data), and V_{wt} stands for success rate when validating in Town 2 and 'soft rainy sunset' (neither town, nor weather are part of the training data). Therefore, note that V_P is a weighted success rate based on 75 episodes.

D. Training protocol

All CIL models in this paper rely on the same training protocol, partially following [39]. In all our CIL models original sensor channels (R/G/B/D) are trimmed to remove sky and very close areas (top and bottom part of the channels), and down-scaled to finally obtain channels of 200×88 pixel resolution. In our initial experiments, we found that traditional photometric and geometric recipes for data augmentation were not providing better driving models, thus, we do not use them.

E. Experimental results

We start the analysis of the experimental results by looking at Table II, which is produced during training and selection of the best CIL networks. We focus first on RGB data as well as depth based on the post-processed CARLA depth ground truth, termed here as *active* depth (Sect. IV-C) since its accuracy and covered depth range is characteristic of active sensors (*e.g.* LiDAR). We see that the best (among five training runs) validation performance V_P is 48% when using RGB data only. So we will use the corresponding CIL model as RGB-based driver in the following experiments. Analogously, for the case of using only active depth (D), the best CIL reports a performance of 74%. The best performances for early fusion (EF), mid fusion (MF), and late fusion (LF) are $91\%, 74\%$ and 67%, respectively. Again, we take the corresponding CIL models as drivers for the following experiments.

Table III reports the performance of the selected models according to the original CARLA benchmark. We have included a model trained on perfect semantic segmentation (SS) according to the five classes considered here for self-driving (see Fig. 5). Thus, we consider this model as an upper bound. Indeed, its performance is most of the times ≥ 96, reaching 100 several times. This also confirms that the CIL model is able to drive properly in CARLA conditions provided there is a proper input. We can see that, indeed, active depth is a powerful information for end-to-end driving by itself, clearly outperforming RGB in non-training conditions. However, in most of the cases RGBD outperforms the use of only RGB or only D. The most clear case is for new town and weather with dynamic objects, *i.e.* for the most challenging conditions, where RGB alone reaches a success rate of 46.67 ± 6.43, D alone 69.33 ± 2.31, but together the success rate is 94.00 ± 0.00 for early fusion. For a new town (irrespective of the weather conditions) early fusion clearly outperforms mid and late fusion. In any case, it is clear that multimodality improves CIL performance with respect to a single modality, which is the main question we wanted to answer in this paper.

In order to further analyse the goodness of multimodality, we compare it to previous single-modality methods (see Sect. II). Not all the corresponding papers provide details

about the training methodology or training datasets; thus, this comparison is solely based on the reported performances on the original CARLA benchmark and must be taken only as an additional reference about the goodness of multimodality. Early fusion, is the smaller CNN architecture in terms of weights, thus, we are going to focus on it for this comparison. Table IV shows the results. MP and RL stand for modular perception and reinforcement learning, respectively. The reported results are reproduced from [65]. CAL stands for conditional affordance learning and the results are reproduced from [56]. CIRL stands for controllable imitative reinforcement learning and the results are reproduced from [41]. Finally, MT stands for multi-task learning, and the results are reproduced from [48]. We see how, in presence of dynamic traffic participants, the RGBD early fusion (with active depth) is the model with higher success rate on the original CARLA benchmark. On the other hand, such an early fusion approach can be combined with CAL or CIRL methods, they are totally compatible. We think that this comparison with previous works reinforces the idea that multimodality can help end-to-end driving.

Once it is clear that multimodality is beneficial for end-to-end driving, we can raise the question of whether monocular depth estimation [23, 24, 81, 82] can be as effective as depth coming from active sensors in this context. In the former case, it would consists on a multisensory multimodal approach, while the later case would correspond to a single-sensor multimodal approach since both RGB and depth come from the same camera sensor (depth is estimated from RGB). In order to carry out a proof-of-concept, we use our own monocular depth estimation model [24] (it was state-of-the-art at the moment of its publication) fine-tuned on CARLA training dataset. More specifically, the dataset used for training the multimodal CIL models is also used to fine-tune our monocular depth estimation model, *i.e.* using the post-processed depth channels and corresponding RGB images. During training, we monitor the regression loss until it is stable, we do not stop training based on the performance on validation episodes. Figure 5 shows and example of monocular depth estimation.

Analogously to the experiments shown so far, we train a CIL model based on the estimated depth as well as on the corresponding multimodal (RGBD) fusion. In order to reduce the burden of experiments, we use early fusion since it is the best performing for the active depth case. The training performances for model selection can be seen in Table II. We use the CIL models of V_P 59% and 51%, respectively. In validation terms, such performances are already clearly worse than the analogous based on active depth. Table III shows the results on the original CARLA benchmark. Indeed, these are worse than using active depth, however, still when remaining in the training conditions monocular-based EF outperforms depth and RGB alone, and in fact shows similar performance as active depth. This is not the case when we change from training conditions since monocular depth estimation itself does not perform equally well in this case, and so happens to EF. However, we think that this single-sensor multimodal setting is really worth to pursuit. Moreover, although it is out of the scope of this paper, we think that performing end-to-end driving may be a good protocol for evaluating depth estimation models beyond the static metrics currently used, which are agnostic to the task in which depth estimation is going to be used. Note that even for evaluating the driving performance of end-to-end driving models in itself, it has been shown that relying only on static evaluations may be misleading [78, 79].

Finally, for the RGB, Active D and EF models, we assess additional infractions for new town and weather conditions with dynamic objects. Table V shows the driven Km per infraction of each model. Note that not all such infractions imply an accident stopping the AV. For instance, the AV can run into an opposite lane a bit without crashing with other vehicles. As a reference, we also show the amount of driven Km in which these measurements are based. All models are supposed to complete the same testing routes (*i.e.*, same total Km), termed as *perfect driving* in Table V. However, if a model fails to follow the right path at an intersection the route would be recomputed, thus, it will need more Km to reach the destination. On the contrary, if it fails to complete the routes, the driven Km will be lower. We see that RGB and Active EF models are not far, but Active D failed too much at taking the right path at intersections. RGB performs the worst in all metrics. The Active D model does not run over the sidewalk and uses the curbside as a cue that also helps on lane keeping except at intersections. Active D shows a good equilibrium between RGB and Active depth single-modality models.

V. Conclusion

In this paper, we compare single- and multimodal perception data for end-to-end driving. As multimodal perception data we focus on RGB and depth, since they are usually available in autonomous vehicles through the presence of cameras and active sensors such as LiDAR. As end-to-end driving model we use branched conditional imitation learning (CIL). Relying on a well-established simulation environment, CARLA, we assess the driving performance of single-modal (RGB, depth) CIL models, as well as multimodal CIL models according to early, mid, and late fusion paradigms. In all cases, the depth information available in CARLA is post-processed to obtain a more realistic range of distances and depth accuracy. This depth is also used to train a depth estimation model so that the experiments cover multimodality not only based on a multisensory setting (RGB and active depth) but also based on a single-sensor setting (RGB and estimated depth). Overall, the experiments clearly allow us to conclude that multimodality (RGBD) is indeed a beneficial approach for end-to-end driving. In fact, we plan to follow this line of work in the near future, focusing on the single-sensor setting since better estimation models are required in order to compete with the multisensory setting. In addition, we also plan to consider other sources of multi-modality usually available in modern vehicles, such as GNSS information which, even usually being noisy, eventually can complement direct scene sensing.

References

[1] U. Franke, "Autonomous driving," in *Computer Vision in Vehicle Technology*, 2017.

[2] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art," arXiv:1704.05519, 2017.

[3] B. Paden, M. Cáp, S. Z. Yong, D. S. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[4] W. Schwarting, J. Alonso, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Reviews of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 187–210, May 2018.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Neural Information Processing Systems (NIPS)*, 2015.

[6] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] A. Asvadi, L. Garrote, C. Premebida, P. Peixoto, and U. J. Nunes, "Multimodal vehicle detection: fusing 3D-LIDAR and color camera data," *Pattern Recognition Letters*, vol. 115, pp. 20–29, November 2018.

[9] A. Pfeuffer and K. Dietmayer, "Optimal sensor data fusion architecture for object detection in adverse weather conditions," in *Inter. Conf. on Information Fusion (FUSION)*, 2018.

[10] C. Li, D. Song, R. Tong, and M. Tang, "Illumination-aware Faster R-CNN for robust multispectral pedestrian detection," *Pattern Recognition*, vol. 85, pp. 161–171, January 2019.

[11] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," in *Inter. Conf. on Computer Vision (ICCV)*, 2015.

[12] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Inter. Conf. on Computer Vision (ICCV)*, 2015.

[13] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Inter. Conf. on Image Processing (ICIP)*, 2017.

[14] G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, and M. Felsberg, "Unveiling the power of deep tracking," in *European Conf. on Computer Vision (ECCV)*, 2018.

[15] M. Dimitrievski, P. Veelaert, and W. Philips, "Behavioral pedestrian tracking using a camera and LiDAR sensors on a moving vehicle," *Sensors*, vol. 19, no. 2, 2019.

[16] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[18] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Inter. Conf. on Computer Vision (ICCV)*, 2015.

[19] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Inter. Conf. on Learning Representation (ICLR)*, 2016.

[20] J. Uhrig, M. Cordts, U. Franke, and T. Brox, "Pixel-level encoding and depth layering for instance-level semantic labelling," in *German Conf. on Pattern Recognition (GCPR)*, 2016.

[21] L. Schneider, M. Jasch, B. Fröhlich, T. Weber, U. Franke, M. Pollefeys, and M. Rätsch, "Multimodal neural networks: RGB-D for semantic segmentation and object detection," in *Scandinavian Conference on Image Analysis (SCIA)*, 2017.

[22] Q. Ha, K. Watanabe, T. Karasawa, Y. Ushiku, and T. Harada, "MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[23] C. Godard, O. Aodha, and G. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[24] A. Gurram, O. Urfalioglu, I. Halfaoui, F. Bouzaraa, and A. M. Lopez, "Monocular depth estimation by learning from heterogeneous datasets," in *Intelligent Vehicles Symposium (IV)*, 2018.

[25] Y. Gan, X. Xu, W. Sun, and L. Lin, "Monocular depth estimation with affinity, vertical pooling, and label enhancement," in *European Conf. on Computer Vision (ECCV)*, 2018.

[26] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[27] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[28] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[29] X. Chen, H. Zhang, H. Lu, J. Xiao, Q. Qiu, and Y. Li, "Robust SLAM system based on monocular vision and LiDAR for robotic urban search and rescue," in *Safety, Security and Rescue Robotics (SSRR)*, 2017.

[30] Y.-S. Shin, Y. S. Park, and A. Kim, "Direct visual SLAM using sparse depth for camera-LiDAR system," in *Inter. Conf. on Robotics and Automation (ICRA)*, 2018.

[31] K. Qiu, Y. Ai, B. Tian, B. Wang, and D. Ca, "Siamese-ResNet: implementing loop closure detection based on siamese network," in *Intelligent Vehicles Symposium (IV)*, 2018.

[32] H. Yin, L. Tang, X. Ding, Y. Wang, and R. Xiong, "LocNet: global localization in 3D point clouds for mobile vehicles," in *Intelligent Vehicles Symposium (IV)*, 2018.

[33] J. Zhu, Y. Ai, B. Tian, D. Cao, and S. Scherer, "Visual place recognition in long-term and large-scale environment based on CNN feature," in *Intelligent Vehicles Symposium (IV)*, 2018.

[34] D. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network," in *Neural Information Processing Systems (NIPS)*, 1989.

[35] Y. LeCun, U. Muller, J. Ben, E. Cosatto, and B. Flepp, "Off-road obstacle avoidance through end-to-end learning," in *Neural Information Processing Systems (NIPS)*, 2005.

[36] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," arXiv:1712.00409, 2016.

[37] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[38] C. Hubschneider, A. Bauer, M. Weber, and J. M. Zollner, "Adding navigation to the equation: Turning decisions for end-to-end vehicle control," in *Intelligent Transportation Systems Conference (ITSC) Workshops*, 2017.

[39] F. Codevilla, M. Müller, A. M. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *Inter. Conf. on Robotics and Automation (ICRA)*, 2018.

[40] Q. Wang, L. Chen, and W. Tian, "End-to-end driving simulation via angle branched network," arXiv:1805.07545, 2018.

[41] X. Liang, T. Wang, L. Yang, and E. Xing, "CIRL: Controllable imitative reinforcement learning for vision-based self-driving," in *European Conf. on Computer Vision (ECCV)*, 2018.

[42] C. Innocenti, H. Lindén, G. Panahandeh, L. Svensson, and N. Mohammadiha, "Imitation learning for vision-based lane keeping assistance," in *Intelligent Transportation Systems Conference (ITSC)*, 2017.

[43] Z. Chen and X. Huang, "End-to-end learning for lane keeping of self-driving cars," in *Intelligent Vehicles Symposium (IV)*, 2017.

[44] H. M. Eraqi, M. N. Moustafa, and J. Honer, "End-to-end deep learning for steering autonomous vehicles considering temporal dependencies," in *Neural Information Processing Systems (NIPS)ML for ITS WS*, 2017.

[45] L. George, T. Buhet, E. Wirbel, G. Le-Gall, and X. Perrotton, "Imitation learning for end to end vehicle longitudinal control with forward camera," in *Neural Information Processing Systems (NIPS) Imitation Learning WS*, 2018.

[46] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," arXiv:1704.07911, 2017.

[47] J. Kim and J. Canny, "Interpretable learning for self-driving cars by visualizing causal attention," in *Inter. Conf. on Computer Vision (ICCV)*, 2017.

[48] Z. Li, T. Motoyoshi, K. Sasaki, T. Ogata, and S. Sugano, "Rethinking self-driving: Multi-task knowledge for better generalization and accident explanation ability," arXiv:1809.11100, 2018.

[49] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Inter. Conf. on Computer Vision (ICCV)*, 2017.

[50] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou, "Deep learning scaling is predictable, empirically," arXiv:1604.07316, 2017.

[51] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[52] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes dataset for semantic urban scene understanding," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[53] E. Santana and G. Hotz, "Learning a driving simulator," arXiv:1608.01230, 2016.

[54] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000 Km: The Oxford RobotCar dataset," *Inter. Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.

[55] C. Chen, A. Seff, A. L. Kornhauser, and J. Xiao, "DeepDriving: Learning affordance for direct perception in autonomous driving," in *Inter. Conf. on Computer Vision (ICCV)*, 2015.

[56] A. Sauer, N. Savinov, and A. Geiger, "Conditional affordance learning for driving in urban environments," in *Conf. on Robot Learning (CoRL)*, 2018.

[57] M. Müller, A. Dosovitskiy, B. Ghanem, and V. Koltun, "Driving policy transfer via modularity and abstraction," in *Conf. on Robot Learning (CoRL)*, 2018.

[58] Z. Yang, Y. Zhang, J. Yu, J. Cai, and J. Luo, "End-to-end multi-modal multi-task vehicle control for self-driving cars with visual perceptions," in *Inter. Conf. on Pattern Recognition (ICPR)*, 2018.

[59] A. González, Z. Fang, Y. Socarras, J. Serrat, D. Vázquez, J. Xu, and A. M. López, "Pedestrian detection at day/night time with visible and fir cameras: A comparison," *Sensors*, vol. 16, no. 6, 2016.

[60] A. González, D. Vázquez, A. M. López, and J. Amores, "On-board object detection: Multicue, multimodal, and multiview random forest of local experts," *IEEE Trans. on Cybernetics*, vol. 47, no. 11, pp. 3980–3990, 2017.

[61] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[62] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander, "Joint 3D proposal generation and object detection from view aggregation," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[63] Y. Wu, H. Qin, T. Liu, H. Liu, and Z. Wei, "A 3D object detection based on multi-modality sensors of USV," *Applied Sciences*, vol. 9, no. 3, 2019.

[64] N. Rhinehart, R. McAllister, and S. Levine, "Deep imitative models for flexible inference, planning, and control," arXiv:1810.06544, 2018.

[65] A. Dosovitskiy, G. Ros, F. Codevilla, A. López, and V. Koltun, "CARLA: An open urban driving simulator," in *Conf. on Robot Learning (CoRL)*, 2017.

[66] M. Enzweiler and D. Gavrila, "A multilevel mixture-of-experts framework for pedestrian classification," *IEEE Trans. on Image Processing*, vol. 20, no. 10, pp. 2967–2979, 2011.

[67] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," *IEEE Trans. on Intelligent Transportation Systems*, January 2019.

[68] D. Feng, C. Haase-Schutz, L. Rosenbaum, H. Hertlein, F. Duffhauss, C. Glaser, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," arXiv:1902.07830, 2019.

[69] F. Piewak, P. Pinggera, M. Enzweiler, D. Pfeiffer, and M. Zöllner, "Improved semantic stixels via multimodal sensor fusion," arXiv:1809.08993, 2018.

[70] B. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[71] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner, "TORCS, The Open Racing Car Simulator," http://www.torcs.org.

[72] I. Sobh, L. Amin, S. Abdelkarim, K. Elmadawy, M. Saeed, O. A. Valeo, M. Gamal, and A. El-Sallab, "End-to-end multi-modal sensors fusion system for urban automated driving," in *Neural Information Processing Systems (NIPS) MLITS WS*, 2018.

[73] Q. Khan, T. Schön, and P. Wenzel, "Towards self-supervised high level sensor fusion," arXiv:1902.04272, 2019.

[74] C. Doersch and A. Zisserman, "Multi-task selfsupervised visual learning," in *Inter. Conf. on Computer Vision (ICCV)*, 2017.

[75] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[76] S. Halev-Shwartz and A. Shashua, "On the sample complexity of end-to-end training vs. semantic abstraction training," arXiv:1604.06915, 2016.

[77] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, p. 182193, December 2016.

[78] F. Codevilla, A. M. López, V. Koltun, and A. Dosovitskiy, "On offline evaluation of vision-based driving models," in *European Conf. on Computer Vision (ECCV)*, 2018.

[79] A. Bewley, J. Rigley, Y. Liu, J. Hawke, R. Shen, V.-D. Lam, and A. Kendall, "Learning to drive from simulation without real world labels," arXiv:1812.03823v2, 2018.

[80] C. Premebida, J. Carreira, J. Batista, and U. Nunes, "Pedestrian detection combining rgb and dense LiDAR," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[81] Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, and L. Lin, "Single view stereo matching," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[82] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, "Structured attention guided convolutional neural fields for monocular depth estimation," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Yi Xiao received her Bachelor's degree in electrical engineering from Shantou University in 2017, and her Master's degree with focus on computer vision at the Universitat Autonoma de Barcelona (UAB), in September 2018. Currently, she is working toward the PhD degree in the Autonomous Driving lab of the Computer Vision Center (CVC). Her research interests include end-to-end autonomous driving, imitation learning and multi-modality.

Felipe Codevilla received his PhD's degree by the Computer Science Dpt. at the Universitat Autonoma de Barcelona in May 2019. He was with the group of Autonomous Driving at CVC (UAB). Currently, he is a Post-Doctoral researcher at the Montreal Institute for Learning Algorithms (MILA). His research interests include end-to-end autonomous driving, imitation learning, reinforcement learning, transfer learning and domain adaptation.

Akhil Gurram received his Master's degree with focus on computer vision and deep learning at the Universitat Autonoma de Barcelona (UAB) in September 2016 and currently working as a PhD-student at Huawei GRC in Munich. His area of interests are in Multi-task Learning for Depth estimation, Learnable localization and End-to-End driving modules using computer vision and deep learning.

Onay Urfalioglu is currently team leader at Huawei Technologies. He obtained a Ph.D. degree at the Leibniz University of Hannover in 2006. From 2006-2011, he was a Post-Doctoral researcher at the ISTI/CNR Pisa and at the Bilkent University, Ankara. His recent activities are using Deep Learning for Mapping & Localization applications.

Antonio M. López is the principal investigator of the Autonomous Driving lab of the Computer Vision Center (CVC) at the Univ. Autònoma de Barcelona (UAB). He has also a tenure position as associated professor at the Computer Science department of the UAB. Antonio has a long trajectory carrying research at the intersection of computer vision, computer graphics, machine learning and autonomous driving. Antonio has been deeply involved in the creation of the SYNTHIA dataset and the CARLA open-source simulator, both for democratizing autonomous driving research. He is actively working hand-on-hand with industry partners to bring state-of-the-art techniques to the field of autonomous driving. Currently, Antonio is granted by the Catalan ICREA Academia program.