A Survey on Imitation Learning Techniques for End-to-End Autonomous Vehicles

Article *in* IEEE Transactions on Intelligent Transportation Systems · September 2022

DOI: 10.1109/TITS.2022.3144867

4 authors, including:

Luc Le Mero
The University of Warwick
1 PUBLICATION **63** CITATIONS
SEE PROFILE

Dewei Yi
University of Aberdeen
29 PUBLICATIONS **719** CITATIONS
SEE PROFILE

Mehrdad Dianati
The University of Warwick
184 PUBLICATIONS **6,828** CITATIONS
SEE PROFILE

A Survey on Imitation Learning Techniques for End-to-End Autonomous Vehicles

Luc Le Mero, Dewei Yi, *Member, IEEE*, Mehrdad Dianati, *Senior Member, IEEE* and Alexandros Mouzakitis

***Abstract*—The state-of-the-art decision and planning approaches for autonomous vehicles have moved away from manually designed systems, instead focusing on the utilisation of large-scale datasets of expert demonstration via Imitation Learning (IL). In this paper, we present a comprehensive review of IL approaches, primarily for the paradigm of end-to-end based systems in autonomous vehicles. We classify the literature into three distinct categories: 1) Behavioural Cloning (BC), 2) Direct Policy Learning (DPL) and 3) Inverse Reinforcement Learning (IRL). For each of these categories, the current state-of-the-art literature is comprehensively reviewed and summarised, with future directions of research identified to facilitate the development of imitation learning based systems for end-to-end autonomous vehicles. Due to the data-intensive nature of deep learning techniques, currently available datasets and simulators for end-to-end autonomous driving are also reviewed.**

I. INTRODUCTION

THE autonomous vehicle industry is growing rapidly. The Boston Consulting Group estimates that the industry will be worth $77 billion by 2035 [1]. Predictions from the Brookings Institution [2] and IHS [3] assert that autonomous vehicles will make up 25% of road users by 2040 and almost all users by 2050, respectively. With these predictions driving research and competition between companies and researchers, a race is on to develop the first fully autonomous vehicle.

The current state-of-the-art autonomous vehicles utilise a modular paradigm of system design. Such a system involves the design of an autonomous vehicle such that it contains multiple unique and distinct modules, each responsible for an individual task of autonomous driving (perception, path planning, control *etc*.)

However, there is growing interest in an alternate paradigm, *end-to-end* autonomous driving systems, also referred to as the *behaviour reflex* approach. These systems involve the design of an autonomous vehicle as a single distinct module, responsible for directly mapping from a raw sensory input (camera, LiDAR *etc*.) into a control signal (steering, braking *etc*.).

Lying in between these two approaches is another growing field, first suggested by Chen *et al*. [4], referred to as *direct perception*. Such methods perform the perception and path planning tasks in an end-to-end fashion. The outputs of this module are then utilised to make control decisions. Although not strictly end-to-end, such systems still output trajectories that can be implemented by simple control schemes and are included in this work.

Modular systems offer an attractive level of verifiability, with each module having distinct outputs that can be individually evaluated. However, this comes at the cost of computational efficiency. Each individual module in the pipeline is not 'aware' of the high-level task required, and so wasted computation may occur, for example the identification of objects by the feature detecting module that are unrelated to the autonomous driving task. This problem can be mitigated through careful design of such modules, however this requires prior knowledge of the task at hand.

End-to-end systems are self-optimising; they learn to be computationally efficient, and require no prior knowledge of the task. This improvement comes at the cost of verifiability due to the black-box nature of deep learning. However, with the complexity of the task of autonomous driving, this computational efficiency is a highly desirable trait.

The construction of a single module capable of performing the complex task of autonomous driving typically utilises the processing ability of Deep Learning techniques. The training of such systems is primarily performed through *Imitation Learning*.

Imitation Learning utilises datasets of expert demonstration (typically human) to train a system to imitate the given expert for the range of scenarios presented. Alternate deep learning methods, such as deep reinforcement learning approaches, have been applied to the problem. However, these methods are limited due to the complexity and safety critical nature of the driving task. Imitation learning allows the leveraging of widely available, easily captured large-scale datasets of human driving to be used to train deep learning approaches to near human standard. Such large-scale datasets of human driving are readily available and are reviewed in Section IV. To this end, the majority of the literature focuses primarily on Imitation Learning.

Deep learning techniques have shown great promise in a wide variety of fields from image classification [5], [6] to playing Atari games [7]. However, such techniques have had limited success for complex tasks such as autonomous driving. An autonomous vehicle should be able to reliably and safely act in a wide range of environments, weather and lighting conditions. For the Imitation Learning based solutions considered in this paper, these challenges include being able to accurately replicate human driving behaviour and model external factors that can affect human driving actions

This work was supported by Jaguar Land Rover and the U.K. EPSRC grant EP/N01300X/1 as part of the jointly funded Towards Autonomy: Smart and Connected Control(TASCC) Programme. L. H. Le Mero and M. Dianati are with the Warwick Manufacturing Group, University of Warwick, Coventry CV4 7AL, U.K. Dewei Yi is with the Department of Computing Science, University of Aberdeen AB24 3UE, Aberdeen, U.K. (e-mail:l.h.le-mero, m.dianati@warwick.ac.uk, dewei.yi@abdn.ac.uk).

Alexandros Mouzakitis is with Jaguar Land Rover, Ltd, Coventry, CV4 7HS, UK.

beyond environmental observations. The generalisability of these solutions also presents a significant challenge. Models trained using Imitation Learning struggle to generalise to states that differ significantly from those present in the training dataset. A single and finite dataset cannot realistically contain all possible driving scenarios and so it is crucial to enhance the generalisability of these models.

In previous work, Hussein *et al*. [8] review Imitation Learning in depth, discussing design options for each stage in the Imitation Learning process. Tai *et al*. [9] provide a comprehensive overview of deep learning techniques for control applications. Janai *et al*. [10] review the current state-of-the-art computer vision algorithms for autonomous vehicles, with some focuses given to end-to-end paradigms. To the best of the authors' knowledge, this is the first attempt to provide a comprehensive overview of Imitation Learning based techniques for end-to-end based autonomous vehicle systems.

We have broken the literature into 3 primary sub-fields; Behavioural Cloning, Direct Policy Learning and Inverse Reinforcement Learning. This classification has been done based on the underlying algorithms of each approach. Behavioural Cloning utilises large-scale datasets to train end-to-end systems offline to mimic human driving. Alongside this, Direct Policy approaches utilise iterative online training to continuously improve end-to-end systems. Finally, Inverse Reinforcement Learning attempts to leverage large-scale datasets to learn an underlying reward function of the task of autonomous driving that is then used to train an agent. These three approaches are distinct sub-fields of IL, all utilising expert demonstration in different ways, and as such have been individually reviewed in Section II.

The main contributions of the work are as follows:

- The field of Imitation Learning is categorised into three primary sub-fields; Behavioural Cloning, Direct Policy Learning and Inverse Reinforcement Learning. The state-of-the-art works in each of these sub-fields is presented and reviewed.
- A comparative evaluation of the currently available datasets is presented for end-to-end autonomous driving systems alongside simulation tools and their potential applications.
- Open challenges in the field of autonomous driving are presented to inspire future research.

We briefly introduce the history of end-to-end driving systems in section II. In section III, the state-of-the-art works in the sub-fields of Behavioural Cloning, Direct Policy Learning and Inverse Reinforcement Learning are presented. Section IV gives an introduction and overview of various datasets and simulation tools used in the training of networks. Section V discusses the current state of the literature and section VI concludes the work.

II. Imitation Learning

Imitation Learning for autonomous vehicles can be categorised into three main approaches: Behavioural Cloning (BC), Direct Policy Learning (DPL) and Inverse Reinforcement Learning (IRL). An overview of each of these approaches alongside a review of the literature that utilises them will be presented in this section.

A. History

The first implementation of Imitation Learning for an end-to-end autonomous vehicle system was the development of ALVINN by Pomerleau *et al*. [11] in 1989.

ALVINN was trained in an online fashion, with training data captured in real time from a human driver and used to train the network sequentially to output steering angles for lane following. Alongside this, an algorithm to generate realistic road images was used to create an augmented dataset for further training.

Once fully trained, ALVINN was capable of driving the original test vehicle at speeds of up to 55mph. The early success of ALVINN paved the way for the development of the more complex and successful end-to-end driving systems that will be reviewed in the following paper.

This lane following behaviour was expanded on by [12] who trained a small vehicle, DAVE, to perform obstacle avoidance. The system utilised 2 front-facing cameras, enabling the system to extract distance information and learn to steer the vehicle in an end-to-end fashion.

Bojarski et al [13] expanded upon the DAVE system, training a 3 camera model to perform steering control of a vehicle in a range of real-world driving scenarios. It was arguably this work that brought end-to-end systems into the forefront of autonomous vehicle research. In addition, key studies and milestone work on safety-critical imitation learning [14–17] will be reviewed in the following sections.

B. Problem Formulation

Given a dataset $\mathbb{D}$ of expert state-action pairs (s, a) generated by an expert policy π^*, the general goal of IL is to train a policy $\pi_\theta(s)$ that maps any given state s to a corresponding action a as closely to the given expert as possible according to:

$$argmin_\theta E_{s\sim P(s|\theta)} L(\pi^*(s), \pi_\theta(s)) \tag{1}$$

where $P(s|\theta)$ is the state distribution of the trained policy π_θ.

Behavioural Cloning involves the reduction of the IL task to that of supervised learning. Defining $P^*(s|\pi^*)$ as the state distribution of the expert policy, the objective of BC is to treat each state-action pair in this distribution as an i.i.d example and minimise imitation loss for the trained policy according to:

$$argmin_\theta E_{(s,a^*)\sim P^*} L(a^*, \pi_\theta(s)) \tag{2}$$

where the state distribution encountered, $(s, a^*) \sim P^*$, is now provided solely by the expert policy, and is assumed to be drawn i.i.d from P^*.

In Direct Policy Learning, we utilise a sequence of distributions attained through querying a given expert (typically human) to train an optimal policy. In general, a policy π is trained and rolled out in the environment to attain a state

distribution P_m. For any given state in P_m the expert can then be queried for an optimal solution. This new distribution of state action pairs, $(\pi^*(s)|s)$, can then be used to further train the policy π. A single iteration of this method reduces to BC.

The aim of Inverse Reinforcement Learning is to infer the reward function from expert demonstration [18]. This reward function is then utilised in training a policy.

To give context to this method, we will introduce the idea of Markov Decision Processes (MDP's). An MDP is defined as a tuple:

$$(S, A, T, \gamma, R)$$

where, *S* is a finite set of states, *A* is a set of possible actions, *T* is a set of state transition probabilities, γ is a discount factor and *R* is the reward function. The discount factor is used to define whether the reward evaluation should be short or far sighted. A policy π maps from states to actions.

Given a dataset $\mathbb{D}$ of expert state-action pairs (s, a) generated by an expert policy π^*, IRL attempts to learn a reward function r^* such that:

$$\pi^* = argmax E_\pi[r^*(s, a)] \tag{3}$$

C. Behavioural Cloning

By far, the most researched implementation of Imitation Learning for autonomous vehicles is Behavioural Cloning (BC), a supervised method of approaching the task. However, Behavioural Cloning's use of expert demonstration to guide the training process separates it from classical supervised learning techniques.

In a classical, purely supervised learning approach, the assumption is made that the outputs of the learning agent do not affect the environment. Therefore, learning errors are assumed to be independent for each sample. However, this assumption does not hold true for driving-related tasks, where predictions can be temporally related. More specifically, network outputs are not independent of each other. For such tasks, errors made in the learning for an image input will be compounded which leads to the *Cascading Error Problem* [19].

Further to this, classical approaches are typically step-wise methods, with each prediction being independent of any previous predictions. This assumption does not hold true for real-world driving either. Steering control of a vehicle is a continuous action. If a vehicle is in full right-hand turn in one frame of a video, it cannot then turn left in an immediately subsequent frame and vice-versa. Such a relationship should be modelled by an end-to-end driving system.

Behavioural Cloning trains a model to map directly from input to output data pairs in a training dataset. This method therefore assumes that such a mapping exists, *i.e.* that an expert's actions can be fully explained by an observation. In reality a human driver's decisions are affected by a number of *latent variables*. Such variables include intended destination, expert driving style, the vehicle used in the dataset, *etc.* A successful driving system should effectively model these latent variables.

Fig. 1. Visualisation of Internal State of End-to-End System from [20]. Salient parts of input images were recovered using visual backpropagation. Green areas are highlighting these salient regions. As can be seen from the image, the model clearly learns to place high importance on relevant regions of images such as curbs and other vehicles.

We further categorise Behavioural Cloning into 3 paradigms; end-to-end control prediction, direct perception and uncertainty quantification. End-to-end control prediction algorithms train a model to directly output steering and/or acceleration commands. Direct perception models are trained to output information that can then be used by a simple control module to perform vehicle control. This information may include trajectories, possible future actions or environmental cost maps. Uncertainty quantification approaches attempt to train a model to output uncertainty information to improve the safety of end-to-end systems.

1) ***End-to-End Control Prediction****:* In this section we will present works that train a model to map directly from input data to autonomous vehicle control signals, such as steering angle, throttle etc., in an end-to-end fashion.

Bojarski *et al.* [13], a team from NVidia, propose the utilisation of 3 onboard cameras as inputs to an Imitation Learning end-to-end driving model. They construct an end-to-end, CNN based model (DAVE-2) utilising these inputs to generate steering control signals. The left and right cameras provide off-centre shifted viewpoints which are used to correct any drift the car encounters. The network is trained in a supervised manner to minimise the mean squared error between the networks' steering command and the command from either the expert (human) driver or corresponding shifted (off-centre) viewpoint. Evaluation is performed in both simulation and on-road testing. The number of human interventions required is used as an evaluation metric. Once the network has demonstrated good performance in simulation, it is evaluated in a roughly 12 mile on-road test. A metric of the percentage of

time the vehicle remained autonomous is used for evaluating autonomous driving performance.

One major criticism of end-to-end systems is their lack of interpretability. A further paper, from Bojarski *et al*. [20], attempts to understand the decision making process of the network from [13]. The main focus of the paper is on identifying the *salient objects*, or the regions of the input images that are most salient to the network in determining steering angles. Results show that the salient objects used by the network are extremely similar to those used by humans, as shown in Figure 1.

Cultrera *et al*. [21] also seek to increase the explainability of end-to-end models by training an IL agent equipped with an attention model. They train a model to receive an input image alongside a high-level command and output steering angles. The first part of the model consists of a feature extracting CNN which feeds into a Regions of Interest (RoI) pooling layer to extract descriptors of varying size for the image. The model is constructed to receive one of four high-level commands: follow the lane, go straight, turn left or turn right. For each of these commands a separate predictive section of network is used, consisting of an attention layer and FCN to output steering commands. The attention layers are trained to weight regions of the input images given outputs from the RoI layer.

The model is trained in the CARLA simulator [22], and evaluated using the CARLA benchmark. The authors' model is found to achieve state-of-the-art performance in the *New weather* setting, whilst also providing explainability of solution through the output of the model's attention layer as seen in Figure 2.

Hecker *et al*. [23] argue that human drivers typically have access to more data than provided to autonomous vehicles. They propose providing end-to-end systems with input information regarding their entire surroundings. A system consisting of 8 cameras is used alongside a route planner to provide high-level action information. A new dataset was created, the *Drive360* dataset, covering a wide range of driving scenarios and weather/lighting conditions. Training is only undertaken using four cameras. Each camera is fed to a network consisting of multiple CNN and LSTM sub-networks. Subsequently, an FCN then fuses all information from the cameras and the map to provide future speed and steering predictions. The inclusion of additional information leads to a significant increase in the performance, as measured using the Mean Squared Error between predicted signals and ground truths. The system is compared to [24] and [13], retrained on the *Drive360* dataset, outperforming both.

Codevilla *et al*. [25] claim that the assumption made in Imitation Learning that an expert's actions can be fully explained by a single observation does not hold for complex tasks, such as autonomous driving. They propose an extension of Imitation Learning, *Conditional Imitation Learning* (CIL). In this approach, a trained network is not only fed an input observation but also a representation of the expert agent's intention. This allows the network to be given high-level intention information at test time as a secondary input.

The work adopts the three-camera system from DAVE-2 [13] in order to recover from perturbations. Examples of expert recovery are included in the dataset to improve system robustness to the cascading error problem. The dataset is also augmented with a random set of transformations of contrast, brightness, tone and Gaussian blur. The system is evaluated both in simulation and in the real-world on a 1/5th scale truck. A dataset of input images, steering commands and driver intent is collected in the CARLA simulator [22] as well as in the real-world for training purposes.

A set of four commands is used to guide the vehicle; continue (follow the road), turn left, turn right and go straight (at a junction). These commands are used during training as inputs to the network. Following successful completion of courses in simulation, the system is evaluated in the real-world on pedestrian walkways and completed all routes. Evaluation metrics include the number of missed turnings, human interventions and time taken to complete a course.

Hawke *et al*. [28] use CIL to train an end-to-end system capable of performing both steering and speed control in complex urban environments. The model takes monocular camera images alongside a high-level route command as inputs and outputs vehicle control signals. Despite training the model in an end-to-end fashion, they visualise the network as consisting of 3 individual components: perception, sensor fusion and control. The perception part of the model consists of a network pre-trained on several large research vision datasets and is trained to receive an image and reconstruct RGB, depth and segmentation. Temporal information is encoded using an optical flow model similar to [29]. The sensor fusion component aggregates information for the 3 models studied, processing the information into a single representation. Single camera, multi-camera and optical flow models are studied. The control component then produces control signals.

Multiple variations of their models are evaluated, including using only a fraction of the training data, and comparing the performance of pre-trained and non-pre-trained perception components. The models were evaluated in 34.4km of real-world autonomous driving. A range of metrics are used for evaluation including intervention rate and manoeuvre success rate. The utilisation of temporal information into the CIL system alongside the addition of pre-trained perception components was found to improve model performance, with the addition of multiple cameras providing a further boost, but at the expense of model complexity.

Xiao *et al*. [30] construct an end-to-end autonomous vehicle control system utilising both input RGB images as well as depth information from on-board LiDAR sensors, referred to as RGB-D information. The inclusion of this depth information at varying stages of a CIL based system pipeline (early, mid and late) is investigated (Figure 4).

Early inclusion involves increasing the number of channels at the first convolutional layer P. Mid fusion requires the entire perception phase at P to be executed twice, once for RGB input and once for depth. Late fusion involves twice replicating the entire CIL architecture, once for RGB input and once for depth, with the results shared. Data is collected from 2 maps in the CARLA simulator; one for the collection and the use of a training dataset and the other for testing and validation. RGB-D based systems are found to outperform pure RGB based

Fig. 2. Examples of attention maps for each high-level command from [21]. Commands are Follow lane (Green), Turn Left (red), Turn right (cyan) and Go Straight (Yellow). The various regions of input images that are learnt by the model to be important for each command are highlighted in the images and can be output as shown to provide a level of explainability for the networks decisions.

systems in several specific testing scenarios.

Wang *et al*. [26] propose to utilise the current position and the desired vehicle path for generating a *subgoal angle* which is fed to the network. The desired vehicle path is divided into a set of uniformly distributed points, with the current subgoal angle being the angle between the vehicles current pose and the direction to the nearest of these points. The authors propose a new network architecture, shown in Figure 3, which they refer to as an *Angle Branched Network*. Inputs to the network are sequential images, the speed and the subgoal angle. The first seven layers of the network are pre-trained on the ImageNet dataset. Independent feature extraction networks extract features from input images, vehicle speed and the subgoal angle. These features are then concatenated together to predict steering angles and throttle. The network is trained in the CARLA simulator, with scene information extracted to generate the subgoal angles. The use of the subgoal angle is found to greatly increase performance while the utilisation of depth information reduced the number of collisions. The subgoal angle was found to be an effective high-level navigational command. The angle-branched network is compared to the network in [25], and experimental results indicated improvement of performance.

Further to their previous work, Codevilla *et al*. [31] propose a new benchmark, the NoCrash Benchmark. Constructed in the CARLA simulator, NoCrash is designed to evaluate the performance of autonomous vehicle systems against complex events created by traffic conditions and other road agents. The benchmark consists of 3 tasks: an empty town, regular traffic and dense traffic. Each task consists of a range of goal directed episodes, where an agent starts at a random position and follows high-level commands to reach a goal position. A range of weather conditions are also used. An episode is considered to be successfully completed if the model reaches the goal point without collisions bigger than a fixed magnitude in a defined time limit. Alongside this, traffic rule violations are recorded.

A range of models are evaluated on both the previous CARLA benchmark and on NoCrash. Models were trained using CIL to receive camera inputs and high-level commands and output steering and throttle control signals. A new model, CILRS, is also presented, utilising a deep pre-trained perception model. The proposed model was found to outperform previous models and the state-of-the-art on both benchmarks. Results for all models are found to be worse on NoCrash, indicating the benchmarks ability to more completely explore the limitations of driving models.

Haavaldsen *et al*. [32] investigate the incorporation of recurrent layers into end-to-end models. They train a traditional CNN based end-to-end model alongside another model consisting of a CNN with a recurrent layer. They use the CARLA simulator to train and evaluate the models. A 3 camera system was used to gather training data, with the model being trained to receive input images, traffic signals and a high-level command and subsequently output steering and speed control signals. A second model is constructed with

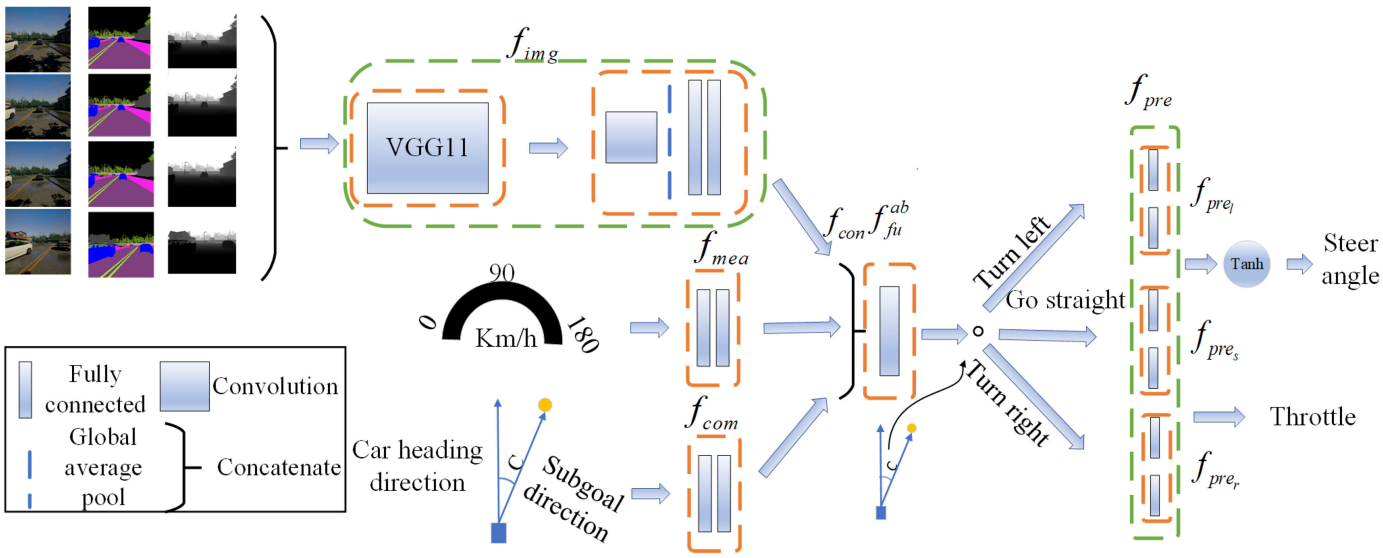

Fig. 3. The architecture of the angle branched network from [26]. The model takes as input RGB images, speed and the subgoal angle. The image processing layers are pre trained from VGG11 [27]. A fusion layer is then used to concatenate the three 512-dimensional layers from the images, speed and subgoal angles into a single layer. The subgoal angle is discretised into 3 distinct set of values representing a separate high-level command; [-180, -10] (turning left), [-10, 10] (going straight) and [10, 180] (turning right). Similar to CIL architectures, the model contains separate action layers corresponding to each of these 3 commands. Each of these layers is responsible for outputting corresponding control signals for each high-level command.

an LSTM layer to allow temporal information to be utilised. Both models achieved good performance in urban autonomous driving. However, the incorporation of temporal information was found to improve the performance.

Chi *et al*. [33] also propose a model utilising LSTM architecture. They incorporate temporal information, modelling the steering angle as a continuous variable. The LSTM network is trained to minimise the loss between predicted steering angles and those of an expert. The network architecture consists of two individual sub-networks, the first a feature-extracting network to model the visual surroundings and internal status of the vehicle. The second sub-network is the steering predicting sub-network, which is responsible for control output.

The feature extracting sub-network performs spatio-temporal convolution to fully model the sequential learning problem of autonomous steering. The steering prediction network fuses multiple kinds of temporal information in order to minimise an overall objective function, outputting steering, speed and torque predictions.

They utilise the Udacity simulator [34] for training purposes. For performance analysis, the method is compared with competing algorithms including AlexNet [5] and PilotNet [20], outperforming them at the task of steering wheel prediction.

Kebria *et al*. [35] investigate the impacts of the number of layers, filers, and filter size on the performance of end-to-end models trained to predict steering angles from camera inputs. They train, evaluate and compare 96 models as well as proposing a novel ensemble approach. The ensemble model assigns weights for each model based on their recent loss values. The Udacity simulator is used for the collection of a training dataset and model evaluation. Deeper models are found to outperform shallower ones, with the jump from 9 to 12 layers being the most significant. It was also found that 16 filters actually outperformed those that had 32. Models with a mixture of filter sizes were found to perform the best. The proposed ensemble method was compared to bagging [36] and was found to outperform it.

2) ***Direct Perception****:* In this section we will survey works that perform the direct perception paradigm of autonomous vehicle design.

Barnes *et al*. [37] propose utilising monocular cameras as opposed to purely LiDAR data. The method aims to generate potential driving paths using input data from the camera. A deep semantic segmentation network is used, with training data acquired from the trajectories taken by a test vehicle. Video odometry is used to measure vehicle motion, whilst obstacle sensing is performed by using LiDAR. The combination of this visual and LiDAR data, in conjunction with the known vehicle trajectory, is then used to segment the input images into driveable, non-driveable and unknown regions at a pixel level. The method is evaluated on both the KITTI [38] and Oxford RobotCar [39] datasets, showing good results for both the segmentation and path proposals in a variety of conditions.

Cai *et al*. [40] propose a CIL end-to-end model that receives camera images, high-level commands and the autonomous vehicles previous trajectories and learns to output collision-free trajectories 3 seconds into the future. The model consists of 3 sub-networks, one for each high-level command. These sub-networks then feed into an LSTM/FCC network to output trajectories. The Oxford RobotCar dataset is utilised for training and evaluation. The model is trained to perform a range of tasks including lane keeping, overtaking and stopping behind a parked car, outputting trajectories close to the ground truth for each task.

Bansal *et al*. [14] present ChauffeurNet, an end-to-end model trained to map from birds-eye representations of the

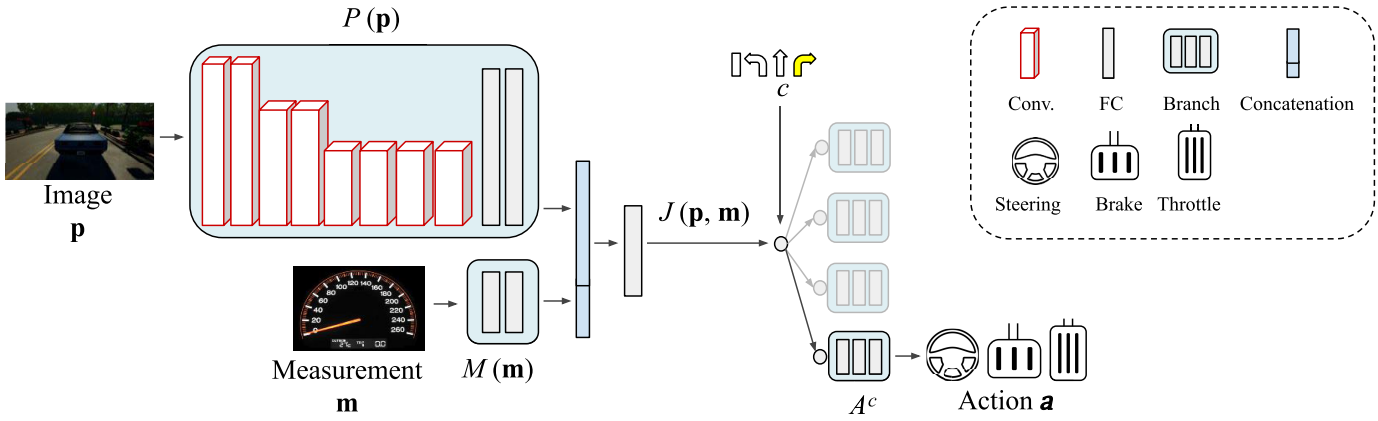

Fig. 4. Conditional Imitation Learning architecture from [30]. Inputs to the network are video frames (p), vehicle state measurements (m) and high-level command information (c). The architecture consists of a series of sub networks for each possible high-level command. The input of a particular command then selects its associated sub-network to produce a given control output. Outputs are in the form of a triplet (steering angle, throttle, brake).

environment to control outputs for an autonomous vehicle. The dataset is constructed from real-world driving with input data in the form of birds-eye representations including road masks, traffic lights, speed limits, route and past agent poses. The model is then trained to output future trajectories. The authors' further augment the training data with simulated examples of collisions and incorrect driving whilst augmenting the loss functions to discourage such undesirable behaviour. A series of defined scenarios are used to evaluate performance including nudging around a parked car, recovering from a trajectory perturbation and slowing down for a slow car. The model is found to perform well in a range of evaluation scenarios and is also demonstrated to be capable of driving an autonomous vehicle in the real-world.

Caltagirone *et al*. [41] propose to utilise a direct perception method taking in LiDAR inputs, past GPS-IMU and driving directions from google maps and outputting driving paths through the environment. An FCN architecture is used to allow for the interpreting of a 3D LiDAR point cloud, with the GPS-IMU data transformed into a spatial format. The FCN is trained to predict driveable regions of the point cloud map for a controller to navigate. The method is evaluated using the KITTI dataset [38]. Results show that the method outputs reliable driving paths for relatively short ranges, with the incorporation of driving intention improving the results further.

Xu *et al*. [24] propose to utilise a large-scale, uncalibrated, crowdsourced dataset for the training of autonomous vehicles in an attempt to address the Cascading Error Problem. To achieve this, they collect a large-scale (10,000 hours) dataset of dashcam and GPS-IMU data from internet sources. The larger scale decreases the number of potentially unseen scenarios that a vehicle will encounter. Subsequently, they develop a direct perception based approach for autonomous driving which is based on a dashcam subset of their Berkeley DeepDrive Video dataset (BDDV).

To account for the varying position of cameras within the vehicle, the model is trained to predict ego-motion. The model also utilises spatial and temporal information to make driving decisions for modelling the steering angle in a continuous fashion. A novel network architecture, consisting of the fusion of an FCN visual encoder and LSTM temporal encoder, is developed to perform this task. The goal of the method is to learn the feasibility of future actions for use in a control system.

Evaluation of the model is performed by taking the most likely action predicted by the system and comparing it to the ground truth action. Two separate output methods are considered; discrete and continuous actions. Furthermore, the use of privileged learning (having access to more information at the training stage than the evaluation stage) is investigated which uses semantic segmentation of input images. In both the discrete and continuous paradigms, the system shows a promising understanding of human driving behaviour. The incorporation of privileged learning also improves the networks performance.

Chen *et al*. [4] aim to develop a CNN based neural network to map directly from input images to output affordance indicators that can be used by a control network to perform autonomous driving. These affordance indicators include the angle of the car relative to the road, the distance to lane markings and distances to surrounding vehicles. The vehicle controller is trained to use the output indicators to minimise the gap between the car's position and the centre of the desired lane. The used dataset is collected by allowing a human driver to act in the TORCS simulator [42] for roughly 12 hours, focusing primarily on highway driving. The system is further tested on car-mounted smartphone videos and the KITTI dataset [38].

Evaluation of the system in simulation is performed by comparison with benchmark solutions. These include a behaviour reflex based CNN method, the Caltech lane detector algorithm [43] and direct perception with the hand-crafted GIST descriptor [44]. The method is found to outperform the alternate methods for the task evaluated. For the real-world evaluation, the model is compared with other methods on the KITTI dataset [38]. The comparison is performed with the state-of-the-art DPM car detector [45]. The proposed method has comparable performance levels to the DPM method.

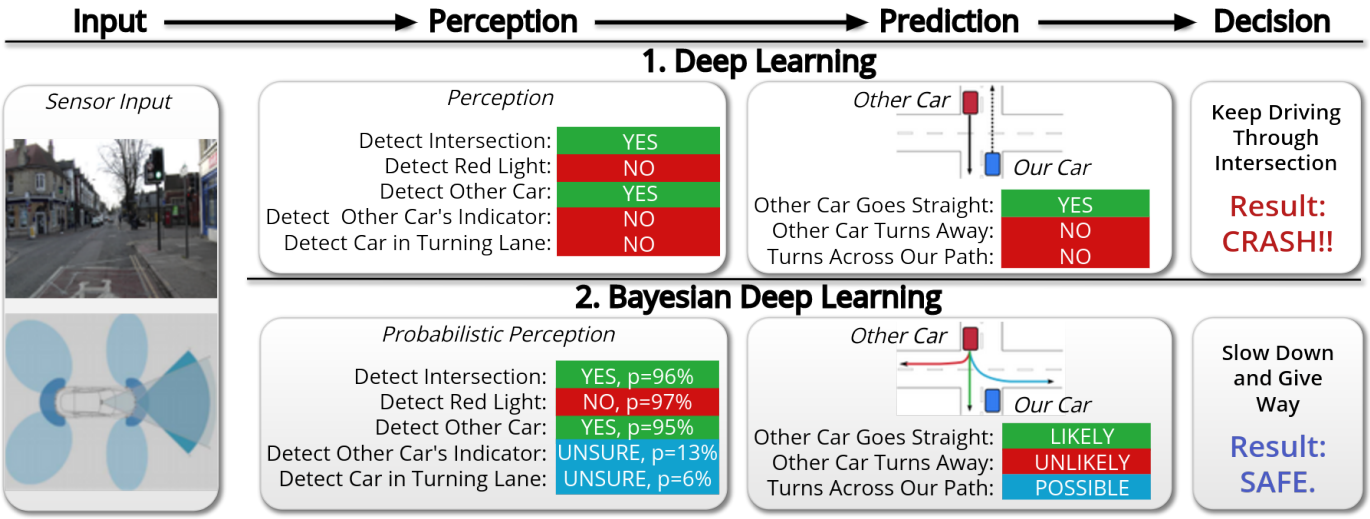

Fig. 5. Figure from [46] illustrating potential advantages of Bayesian Deep Learning architecture. The processing pipeline for traditional deep learning models is shown in the first part. The shown vehicle approaches an intersection where the red car will turn into its path. The failures to detect the red light or other cars indications and intents lead to a deterministic incorrect prediction, causing a collision. In a Bayesian model, despite the light and other vehicle signals and intentions not being detected, a level of uncertainty is associated with the incorrect detection's. This uncertainty is then propagated through the model to provide an uncertainty over the predicted actions. If the level of uncertainty is above a defined threshold then the vehicle will not take the action and the collision will be avoided.

*3) **Uncertainty Quantification**:* In this section, we will focus on works that incorporate some instance of uncertainty modelling into their end-to-end system.

Richter *et al*. [47] utilise a pair of neural networks in their autonomous system to perform collision prediction and novelty detection simultaneously. The navigation from starting point to goal is solved using geometric maps of the environment constructed using SLAM. A CNN is then trained to predict collision likelihood for a trajectory. An auto-encoder is also used in the system to perform novelty detection on input data. This is to overcome the issues with generalisation to unseen states for deep learning methods. If the auto-encoder detects that input data is from a novel, unseen scenario it allows the autonomous agent to revert back to a series of pre-programmed, 'safe' behaviours. The auto-encoder is trained using a dataset of simulated camera images from hallway type environments. The system is tested both in simulation and in the real-world using an RC car. Results show that the model is able to achieve reliable transitions between high performing trained network and the safe behaviours.

McAllister *et al*. [46] propose the utilisation of Bayesian Deep Learning (BDL) in the development of autonomous driving systems. Bayesian Neural Networks (BNN) move away from having single-valued tunable parameters (weights, biases) to having distributions of values for each parameter. This then naturally leads to the output of such a system being a distribution. Therefore, it not only can extract the networks' prediction but also a measure of uncertainty. The main focus of the work is on the implementation of such an approach for a prediction module in a modular system, namely such a systems ability to predict erroneous behaviour of other road users and dynamic objects as seen in Figure 5. However, the work also mentions the value of applying such an architecture to an end-to-end system to produce a similar ability for identifying scenarios that are either underrepresented or absent from training data sets (edge-cases).

Michelmore *et al*. [48] investigate the incorporation of model uncertainty into an end-to-end system. The challenge of steering angle prediction is treated as both a regression and classification task, with separate network architectures built for each approach. Both systems are trained and evaluated in the Udacity simulator [34]. Uncertainty information is extracted using 3 separate methods; variation ratios [49], predictive entropy [50] and mutual information [50]. Uncertainty information is able to be extracted for each approach in real time by limiting stochastic forward passes. The utilisation of uncertainty modelling is not found to improve the predictive performance of the method. However, the utilisation of uncertainty in crash prediction is also investigated. Mutual information was found to be a promising indicator of future crashes.

Amini *et al*. [51] utilise spatial dropout in a deep Bayesian Network for end-to-end autonomous vehicle control. They approximate Bayesian inference and acquire uncertainty estimates on output predictions. Dropout is performed before every weight layer using 2 approaches; element-wise Bernoulli Dropout [52] and spatial Bernoulli dropout [53]. Evaluation is performed in the real-world on a Toyota Prius in a range of weather conditions and environments. Training is performed using over 7 hours of driving data, with the network trained to map from RGB input data directly through to control signals. Spatial dropout was found to be the most suitable approach, which is utilised during training as this leads to faster convergence.

Cai *et al*. [54] focus on improving the generalisation capabilities of end-to-end models. The authors propose a multi-perception model, PMP-net, that receives as input LiDAR, radar and camera data, outputting a trajectory which is used

by a PID controller to generate a control signal. The trajectory is acquired through probabilistic motion planning, where a Gaussian Mixture Model is used to output a distribution of possible trajectories. A decision on the final trajectory is then made by evaluating the statistical properties of this distribution. This also allows a level of uncertainty on any trajectory to be attained.

A large scale dataset is collected in the CARLA simulator, focusing on the inclusion of complex environments and road agents. The authors create and utilise the DeepTest benchmark for model evaluation. The benchmark, constructed in the CARLA simulator, focuses on the inclusion of a range of weather conditions and complex multi-agent scenarios. It also has a zero tolerance for collision events, unlike other CARLA based benchmarks. The author's model was found to achieve excellent driving and generalisation performance in the DeepTest benchmark.

Chen *et al*. [15] utilise birds-eye view maps of the vehicle's environment to train an end-to-end model consisting of a trajectory planner alongside a safety and tracking controller. The HD map inputs to the network include routing, traffic light, detected objects and historical ego-vehicle states. The model is then trained to output planned trajectories through its environment. Each trajectory is used by the network to output desired speed and steering parameters, which are passed to the safety enhancement module. This model component is based on the safe set algorithm [16] and is used to ensure that outputted control signals maintain vehicle safety. The CARLA simulator is used to collect data and evaluate the model, and is found to outperform similar approaches in a range of complex environments.

Lee *et al*. [55] train a Bayesian Neural Network using dropout to perform end-to-end autonomous driving whilst allowing for real-time uncertainty estimates on model outputs. When the uncertainty of an output is higher than a defined threshold, the handling of the autonomous vehicle is handed to a predictive controller or expert. The AutoRally simulator [56] is used to evaluate the model, with the network being successfully trained to steer the vehicle and hand over control in the presence of novel scenarios.

D. Direct Policy Learning

Direct policy learning builds on Behavioural Cloning by attempting to leverage a given expert at training time to overcome BCs limitations. Initially the approach is identical to BC, with the gathering of a dataset of expert i.i.d state-action pairs which is then used to train a policy. The Direct Policy Learning algorithm then queries a given expert to evaluate the policy at run-time in order to attain further training data, typically in the form of under represented scenarios in which the initial policy fails. In this way, DPL addresses the limitations of BC, however the requirement of an expert at training time is an expensive one. In this section, we will present a review of a range of DPL algorithms.

Ross *et al*. [57] propose the DAgger algorithm, an online Imitation Learning method. A primary policy trained through Imitation Learning simultaneously collects further training examples whilst running a reference policy.

Initially, an expert dataset is used in conjunction with Imitation Learning to train an autonomous agent. Subsequent iterations then involve allowing the trained policy to acquire further trajectories which are added to the dataset. For iteration n, the next policy π_{n+1} is the policy that best imitates the expert on the entire dataset. For certain iterations, a modified policy is used which allows for queries to the expert (allows the expert to control for a fraction of the time). This allows the expert to recover mistakes, particularly for early iterations where the frequency of mistakes may be very high. This continuous addition of examples of incorrect trajectories increases the number of observed trajectories in the data and demonstrates recovery methods. Such an approach can address the cascading error problem. The algorithm is tested in simulation on Super Tux Cart, a 3D racing game. Results indicate a sharp increase in driving performance as DAgger is run.

Building on the DAgger algorithm, Zhang *et al*. [58] propose the SafeDAgger algorithm. The algorithm is constructed to minimise the number of expert queries required when running the DAgger algorithm, as this can be costly. It introduces an additional safety policy which takes both the observation of a state and the primary policy's output as inputs. This safety policy is trained to output the likelihood of the primary policy deviating from an expert trajectory. This allows the system to only query the expert driver when necessary, minimising the number of training examples collected. The method is evaluated using the TORCS simulator. Compared to the DAgger approach, the SafeDAgger algorithm is able to improve the learning rate of the vehicle and minimise expert queries.

Pan *et al*. [59] propose an Imitation Learning based network for off-road driving. The aim of the work is to develop an autonomous agent that only requires access to low-cost, on-board sensors. They perform training with an expert that is assumed to have access to much higher quality sensing equipment than equipped on the test vehicle. The work makes use of a model-predictive control (MPC) expert [60] as opposed to a human driver. An MPC expert based on incremental Sparse Spectrum Gaussian Process (SSGP) dynamics model [61] and an iSAM2 state estimator [62] was used to generate expert actions. For the training stage, Imitation Learning is used to train an initial controller from MPC generated expert data. During testing phase, the online Imitation Learning DAgger algorithm is used to further improve performance.

The testing is performed on an elliptical, real-world track, with the goal of minimising the accumulated cost function over one minute of continuous driving. Results are evaluated in a comparative method between a network trained with batch Imitation Learning and a network trained using the DAgger online Imitation Learning algorithm. In general, the policies trained with online Imitation Learning outperform those trained with batch methods.

Li *et al*. [63] propose a novel training algorithm for end-to-end systems; Observational Imitation Learning (OIL). OIL is an online learning algorithm that aims to train a model to learn to imitate only the best behaviours of several sub-optimal teachers. Multiple simple PID controllers are used as the teachers. At each training iteration, a value function of

each teacher is estimated, with the highest being taken as the expert for training as long as its value function is above 0. If the value is below 0, an additional step is required, where the optimal expert is not used to train the model, but is instead employed to correct the models' actions.

The Sim4CV [64] simulation environment is used to generate datasets and evaluate OIL. The model is trained and compared with state-of-the-art baselines as well as to the teachers that were used to train it. OIL is found to outperform all baselines in the tested scenarios, with the model also being found to outperform all of the teachers.

Kelly *et al*. [65] propose an extension to the DAgger algorithm, which they refer to as Human-Gated DAgger (HG-DAgger). Models are trained to map from camera images to speed and steering control signals. Rather than allowing the expert and model to operate the vehicle simultaneously to acquire online training data as in the DAgger algorithm, identification of model failures is left to the expert. When the expert identifies that the vehicle has entered such a state they can flip a switch and take control, simultaneously recording additional training data for the model to correct. Alongside this, HG-DAgger also learns a risk metric by monitoring when the human expert intervenes. This learnt threshold can then be used to evaluate model performance at test time. Models trained with HG-DAgger were found to outperform those trained with DAgger, as well as learning a risk threshold that was near-optimal to identify model failures for a range of metrics.

E. Inverse Reinforcement Learning

Methods such as supervised learning and reinforcement learning struggle to work for complex and real-world tasks. Reinforcement learning's reliance on a well-crafted reward function limits its potential and crafting such functions for complex tasks is extremely difficult. The trial and error nature of reinforcement learning based techniques poses significant safety challenges for complex and real-world tasks such as autonomous driving. Due to this, the majority of reinforcement learning methods for autonomous driving are applied only in simulation, limiting their effectiveness.

Feature Based Inverse Reinforcement Learning involves the assumption that an expert reward function can be represented as a linear combination of *features*. These features are task-dependent. For autonomous driving, they may include distances to cars or road markings etc. Expert demonstrations are then leveraged with the aim of extracting the reward function according to defined features. The main issue encountered with this method is *Reward Function Ambiguity* [66]. This refers to the fact that for any given dataset there may be several reward functions to explain the shown behaviour, only a small number of which can be suitable.

Abbeel *et al*. [67] employ Feature Based Inverse Reinforcement Learning with the intention of learning different driving styles in a highway simulation. They define 15 features; 5 features corresponding to lanes/shoulders and 10 corresponding to the presence of other cars. Human provided example data are then used to learn 5 different reward functions for driving styles including *nice*, *nasty*, *right lane nice*, *right lane nasty* and *middle lane*.

Sadigh *et al*. [68] propose using Feature Based Inverse Reinforcement Learning to train an autonomous vehicle controller capable of interacting with other road users. They argue that current methods fail to model that actions taken by autonomous vehicle may affect the behaviour of other road users. Inverse Reinforcement Learning was used to learn a human driving reward function to interact with a human controlled vehicle in simulation, providing promising results.

An extension to Feature Based Inverse Reinforcement Learning is Maximum Entropy Inverse Reinforcement Learning, proposed by Ziebart *et al*. [69]. This method proposes to overcome the issue of reward function ambiguity by employing the principle of maximum entropy. When selecting from possible reward functions, one should choose the function with the largest remaining uncertainty consistent and also satisfying the specified constraints of the task. For Maximum Entropy Inverse Reinforcement Learning, this is achieved alongside feature matching. They utilise the method to model route preferences of taxi drivers in urban locations.

Wulfmeier *et al*. [70] outline the development of a Maximum Entropy Deep Inverse Reinforcement Learning framework. The model accepts state features as inputs and maps them to a state reward function. They apply this method in [71] to construct an end-to-end mapping from perception to cost function for path planning urban environments. A large-scale dataset is collected, consisting of LiDAR maps of pedestrian walkways. The DeepIRL framework is then trained to map these 2D point cloud maps to cost functions for the environment which could then be used for path planning.

Generative Adversarial Imitation Learning (GAIL) [72] is a model free form of IRL, where a policy is trained directly without the use of domain knowledge. The learned policy can be thought of as the generator imitating that of the expert from the training data, whilst a discriminator is trained to identify learnt and expert state-action pairs. Although very similar to IRL, GAIL differs in that it is directly training the policy, not the learning the reward function from expert demonstration, through the minimax optimisation problem,

$$min_{\pi} max_{r \in \mathcal{R}} [\mathbb{E}_{\pi}[r(s,a)] - \mathbb{E}_{\pi^*}[r(s,a)]],$$

where $\mathbb{E}_{\pi}[r(s,a)]$ is the average reward under the learnt policy for a reward function r, and $\mathbb{E}_{\pi^*}[r(s,a)]$ is the average reward over the given expert trajectories. GAIL attempts to learn a policy which achieves comparable performance to the expert with respect to any reward function belonging to the class $\mathcal{R}$.

Kuefler *et al*. [17] utilise GAIL in the task of modelling human highway driving. A recurrent neural network, receiving LiDAR like inputs and scalar values representing the vehicles odometry, dimensions and state is trained to output vehicle trajectories. Training and evaluation are performed with the rllab reinforcement learning framework [73]. Their model is compared to standard baselines, including a behavioural cloning based approach. The proposed model outperforms the

baselines in a range of metrics and is found to perform much better over larger horizons (>3s).

Li *et al*. [74] propose an extension to Generative Adversarial Imitation Learning that attempts to model latent variables in driving datasets. Their method learns a latent variable generative model of driving trajectories to reproduce expert behaviour. The proposed extension in which the objective function is augmented with a mutual information term between latent variables and observed state-action pairs. The TORCS simulator [42] is used to evaluate the system. The method maps from input images to driving actions as well as modelling and distinguishing between high-level actions taken by the driver.

III. Datasets and Simulators

The training of neural networks for the control of autonomous vehicles requires the utilisation of large scale datasets. These datasets should cover a wide range of scenarios for both real-world and simulation based training. This section introduces the most commonly used datasets for the training of autonomous driving systems. A comparison of datasets is available in Table II. A comparison of simulators is presented in Table III.

A frequently used dataset is the KITTI dataset [38]. Consisting of 6 hours of driving scenarios, data was recorded using a Velodyne 3D laser scanner and a high precision GPS/IMU navigation system. The data consist of stereo colour images from a variety of scenarios including highway driving, urban environments and rural areas. The entire dataset is calibrated and timestamped. However, the dataset contains a limited range of weather and lighting conditions.

The Oxford Robotcar dataset [39] was collected with an autonomous vehicle traversing a route through Oxford, England twice a week over the course of a year. Containing over 1000km of recorded driving and almost 20 million images, it is an extensive dataset. Data were collected from 6 mounted cameras, LiDAR, and a GPS/IMU navigation system. Despite the lack of variety in the environment, data were collected for a wide range of lighting/weather conditions including heavy rain, night, direct sunlight and snow.

The Berkeley DeepDrive Video dataset [24] is a large-scale uncalibrated dataset, consisting of driving videos and GPS/IMU data. It is by far the largest dataset available with over 10,000 hours of driving data. It contains a wide range of environment and lighting/weather conditions, along with video sources from multiple different vehicles. However, this utilisation of multiple sources can lead to challenges regarding latent variable modelling.

The Drive360 dataset [23] consists of roughly 60 hours of driving data taken in Switzerland. By utilising 8 distinct camera inputs, the dataset provides full surround-view video. The dataset also contains information with regard to steering angle, vehicle speed and route planning.

Beyond the use of provided datasets, simulation tools also provide a useful means to synthesise a task-specific dataset. These techniques are not only easier for producing datasets, but a wide range of scenarios and conditions can be incorporated. The main issue is the difficulty in providing a powerful enough simulation for an agent trained using simulated data to be able to act in the real-world.

The CARLA simulator [22] is an open source simulation tool for autonomous driving. Various 3D models are provided to reproduce a variety of urban scenarios and the simulator allows for the addition of various lighting and weather conditions, the capability of producing maps of the environment and the ability to replicate traffic scenarios. The virtual sensor suite contains multiple camera types and LiDAR models.

An alternate simulation tool is the TORCS simulator [42]. TORCS is an open source car racing simulator. It contains multiple tracks, a sophisticated physics model and the ability to implement multiple vehicle scenarios. The simulator also includes an extensive tool suite alongside a large range of user created content.

The Udacity simulator [34] is another open source, free driving simulator. Users can build their own tracks to evaluate models in the simulator. Training data can be collected through human control and used to train an autonomous agent. The base simulator allows the training of models that utilise camera data as inputs. Models can be evaluated for a range of environments and weather/lighting conditions.

Sim4CV [64] is a photo-realistic simulation tool constructed in the Unreal Engine [75]. The simulator has full physics modelling for cars, unmanned aerial vehicles and human actors in a range of urban environments. The simulator comes with a benchmarking tool and a TensorFlow-based deep learning interface. Data including RGB images, depth, image segmentation and ground truth labelling. The simulation environment is also highly customisable, with a number of pre-built assets that the user can manually place in the scene.

GTA V is a video game, containing a wide range of driving environments and weather/lighting conditions. Beyond this, the game has the ability to simulate large scale AI controlled traffic and pedestrians.

The rFpro simulator [76] is a powerful, high-level simulation suite capable of providing an extensive range of large-scale environments with a range of weather/lighting conditions. A deep range of sensors are available, including camera, LiDAR etc.

IV. Discussion

A. Summary and Limitations

1) ***Behavioural Cloning****:* The majority of works undertaken in the field of end-to-end learning utilise Behavioural Cloning. Although simpler than alternate approaches, challenges such as the cascading error problem and generalisability of solution need to be overcome for such a system to be successfully applied in practice. In this work, we divide BC based approaches into 3 subgroups; control prediction, direct perception and uncertainty quantification.

Control prediction - These approaches train a model to output steering and/or throttle control signals directly from input data. This method is the simplest to develop, with large scale datasets of input-control pairs easily attained with very little pre-processing. However, a limitation of these approaches comes with this individual outputting of control signals with models performance decreasing over longer time horizons.

TABLE I
SUMMARY OF PRESENTED RESEARCH

Type	Ref	Dataset	Algorithm	Learning Type	Advantages	Disadvantages
Behavioural Cloning - Control	[13]	Real-world, camera, steering commands	PilotNet	Imitation Learning	-Utilisation of side facing cameras provides robustness to cascading error problems. -98% autonomy achieved highlights potential of end-to-end systems early in development.	-Method only applicable to simple driving scenarios. -Approach fails to deal with generalisability issues regarding edge cases.
	[21]	Simulation, camera, high-level commands, steering angle	CNN	Conditional Imitation Learning, Attention model	-Use of attention model improves explainability of model -Attention model can also be utilised to improve network performance.	-Attention model only explains regions of image responsible for decisions, so does not fully alleviate black box problem -Evaluated purely in simulation
	[23]	Real-world, multiple cameras, GPS/IMU	CNN, LSTM	Sensor fusion, Behavioural Cloning	-Model utilises multiple views using cameras, which could provide a cheap alternative to LiDAR based approaches. -Model is better approximation of human driving conditions than single image input models, with access to multiple views and navigation	-System requires access to large number of cameras
	[25]	Real-world, camera, steering commands, driver intent, recovery	CNN	Conditional Imitation Learning	-Use of CIL allows for vehicle to be commanded at run time -System demonstrates robustness of approach through Real-world testing	-System only evaluated in Real-world on small scale truck which is not representative of full scale vehicle -Approach fails to deal with generalisability issues regarding edge cases
	[26]	Simulation, multiple cameras, GPS/IMU	Angle Branched Network	Sensor fusion, Behavioural Cloning	-Use of subgoal as a more informative expression of intent shows good results -Subgoal angle is autonomous navigation command	-Model requires accurate definition of path to create subgoal angle
	[28]	Real-world, camera, high-level commands, steering angle	CNN	Conditional Imitation Learning	-Use of pre-trained perception network significantly boosts performance -Use of multiple cameras improves performance over a single camera -Use of temporal information can help overcome limitations of BC	-Multiple cameras increases model complexity -Policy must be evaluated online
	[31]	Simulation, camera, high-level commands, steering angles, throttle	CILRS	Conditional Imitation Learning	-Use of pre trained perception model found to improve performance -More rigorous NoCrash benchmark to evaluate end-to-end systems performance	-Evaluated purely in simulation -Benchmark does not include the ability to test models with complex, multi-agent scenarios

Type	Ref	Dataset	Architecture	Algorithms	Advantages	Disadvantages
	[32]	Simulation, camera, high-level commands, traffic signals, steering angle, speed	CNN, LSTM	Imitation Learning	-Use of temporal information found to improve networks performance	-Evaluated purely in simulation
	[33]	Simulation, camera, steering angle	CNN, LSTM	Imitation Learning	-Treats task of steering angle prediction as temporally continuous, allowing for more realistic predictions	-Evaluated purely in simulation
	[35]	Simulation, camera, steering angles	CNN, Ensemble	Imitation Learning	-Comprehensive study of the effects of network parameters on performance to guide future research	-Ensemble methods are very difficult to apply in real time
Behavioural Cloning - Trajectory	[30]	Simulation, camera, LiDAR, steering angles	CNN	CIL, Sensor Fusion, Behavioural Cloning	-Utilisation of depth information increases performance of model significantly in some cases	-Work makes fundamental assumption that LiDAR and other depth sensors will be widely available on autonomous vehicles, which may be limited due to the cost of these systems.
	[37]	Real-world, LiDAR, camera	SegNet	Semantic Segmentation, Direct Perception	-Utilisation of both visual and LiDAR data improves segmentation performance and produces reliable and accurate driving paths	-Requires access to both visual and LiDAR sensors which is an expensive requirement
	[40]	Real-world, camera, high-level commands, trajectories	CNN, LSTM	Conditional Imitation Learning	-Model capable of running in real-time -Incorporation of temporal information shown to improve performance	-Model was not evaluated under a range of weather and lighting conditions
	[14]	Real-world, birds-eye, traffic lights, speed limits, trajectories	CNN, RNN	Imitation Learning	-Use of synthesized data of interesting situations is shown to greatly improve performance -Model shown to be capable of driving a Real-world autonomous vehicle	-Model does not achieve comparable performance to motion planning approaches
Behavioural Cloning - Direct Perception	[24]	Real-world, camera, large-scale, steering angle	FCN & LSTM	Supervised learning, Privileged learning, Direct Perception	-Large scale dataset improves generalisability of system -Use of temporal encoder provides robustness to multi-modality of actions	-Issue of generalisability only partly solved by large scale dataset collection -Issue of latent variable modelling due to range of data sources
	[4]	Simulation/Real-world, visual, orientation, distances to obstacles/markings	CNN	Supervised Learning, Direct Perception	-Model is compared to current SOTA approaches, outperforming them for specified task	-Evaluation performed on ability to map to affordance indicators, separate system should be designed to utilise this information
	[41]	Real-world, LiDAR, GPS-IMU, directions	FCN	Supervised Learning, Direct Perception	-Incorporation of driving intent allows system to predict reliable, accurate driving paths for use by separate control module	-Model only predicts reliable routes over short distances

Type	Ref	Dataset	Architecture	Algorithms	Advantages	Disadvantages
Behavioural Cloning - Uncertainty Quantification	[47]	Real-world, simulation, SLAM map	CNN	SLAM/Supervised Learning, Direct Perception	-Use of secondary network to predict novelty of scenario is valuable approach to solving issues of generalisation and safety in autonomous vehicles	-Network falls back on pre-programmed 'safe behaviours', leading to a trade off between safety and performance
	[48]	Simulation, camera, steering angles	BNN	Uncertainty Modelling, Direct Perception	-Use of BNN provides robust, real-time evaluation of safety levels in vehicle -BNN can simultaneously provide control signals and uncertainty modelling	-BNN approach is more complex than standard CNN based model, potentially limiting performance
	[51]	Real-world, camera steering angles	BNN	Dropout as Inference Approximation, Behavioural Cloning	-Use of dropout is simple, quickly converging approach to approximating Bayesian Inference	-Paper does not provide detailed enough information regarding performance of system.
	[54]	Simulation, LiDAR, radar, camera, trajectory	PMP-net	Imitation Learning	-Use of probabilistic motion planning improves safety of models output -Presentation of more rigorous DeepTest benchmarking tools to allow better evaluation of current state-of-the-art approaches	-Only evaluated in simulation
	[15]	Simulation, birds-eye maps, trajectories	CNN	Imitation Learning	-Use of safety controller improves performance of network as well as improving safety of model -Use of safety controller allows network to perform successfully in complex urban environments.	-Model only evaluated in simulation -Model relies on detailed perception information that is not readily available in Real-world perception systems.
	[55]	Simulation, images, steering angle, speed	BNN	Imitation Learning	-Model successfully trained to utilise uncertainty information and hand over to expert when encountering novel scenarios	-Model only evaluated in simulation -Bayes by dropout requires computationally expensive sampling to attain uncertainty estimates
Direct Policy Learning	[57]	Simulation, camera, steering commands	CNN	DAgger	-Use of expert queries sharply increases performance and allows the system to deal with edge cases	-Requirement of system during training to have constant access to a queryable expert limits the models applicability
	[58]	Simulation, camera, steering angle	CNN	SafeDAgger	-System successfully decreases the number of expert queries, improving on large drawback of [21]	-Method still requires the presence of a queryable expert during training, limiting applicability
	[59]	Real-world, camera, steering angle	CNN	Supervised learning, DAgger	-Real-world evaluation of system more robust than simulation based one -Shows value of online learning methods in improving autonomous vehicle systems	-Use of online learning DAgger algorithm requires constant access to queryable expert

Type	Ref	Dataset	Architecture	Algorithms	Advantages	Disadvantages
	[63]	Simulation, camera, steering, speed	CNN	PID	-Use of OIL overcomes challenges with imperfect experts for IL -OIL successfully allows a policy to be trained to outperform all experts	-Requires access to multiple expert policies
	[65]	Real-world, camera, steering angles, speed	BNN	Online DAgger	-Use of HG-DAgger more expert query efficient than DAgger -Learnt risk threshold provides good measure of model performance	-Risk measurement is not employed to improve safety of model or to enable expert handover
Inverse Reinforcement Learning	[67]	Simulation, birds-eye	Feature Based IRL	Markov Decision Process	-Approach of feature based IRL provides robustness to reward function ambiguity for process as complex as autonomous driving	-Simulation is simplistic and fails to accurately model the complexity of Real-world autonomous driving
	[68]	Simulation, birds-eye	Feature Based IRL	Markov Decision Process	-Method accounts for the fact that a users actions affect the state of the environment, providing more realistic model	-Approach is again simplistic and not applicable to the development of fully autonomous vehicle
	[69]	Real-world, GPS	Maximum Entropy IRL	Markov Decision Process	-Maximum entropy approach provides strong robustness to problem of reward function ambiguity -Method is utilised on top of feature based IRL	-Approach far from being able to provide a robust control method for a fully autonomous vehicle
	[70]	Real-world, LiDAR	FCN	Maximum Entropy Deep IRL	-First approach to successfully apply deep learning to IRL framework -Provides accurate, path planning model from LiDAR pointclouds	-Task is computationally intensive and not yet applicable to controlling autonomous vehicle
	[17]	Simulation, LiDAR, state information, trajectories	RNN	GAIL	-Outperforms BC based trajectory prediction over long horizons, overcoming cascading error problem	-Outputted control signals have non human-like oscillations
GAIL	[74]	Simulation, visual, trajectories	GAIL	Supervised Learning	-Method provides resilience to latent variables and could be applied to large-scale crowd sourced datasets -Method can distinguish between high-level actions taken by the driver	-Method is basic, and needs development

Bojarski *et al*. [20] and Cultrera *et al*. [21] seek to improve the explainability of end-to-end models. Bojarski *et al*. use visual backpropagation to visualise salient regions of input images for the model's decision making. Cultrera *et al*. instead use an attention model for the task. Their investigations highlight that the salient features learnt by the network are extremely similar to those utilised by humans, a promising finding that should increase user and manufacturer trust.

The development of Conditional Imitation Learning by Codevilla *et al*. [25] presents a promising solution to the challenge of multi-modality of output, i.e. that a single input image may have multiple corresponding actions. Although the use of high-level intent information alleviates this ambiguity, the requirement of continuous expert provided high-level intention is a costly one. Wang *et al*. [26] utilise automatically generated intention information, the subgoal angle, to prevent

TABLE II
COMPARISON OF AUTONOMOUS DRIVING DATASETS [24]. U - URBAN, R - RURAL, H - HIGHWAY

Dataset	Environment			Data Type	Scale (h)	Diversity			Sensors			Advantages	Disadvantages
	U	R	H			Weather	Day	Night	Video Image	LiDAR	GPS-IMU		
KITTI	✓	✓	✓	Real-world	1.4	Single	✓	-	✓	✓	✓	-Contains all 3 environments -Contains LiDAR data	-Small dataset -Low diversity
Oxford	✓	-	-	Real-world	214	Multiple	✓	-	✓	-	✓	-Contains range of weather conditions	-Contains few environments -No LiDAR data
BDDV	✓	✓	✓	Real-world	10k	Multiple	✓	-	✓	-	✓	-Large scale -Contains wide range of environments and weather	-No night data -No LiDAR
Drive360	✓	-	✓	Real-world	60	Multiple	✓	-	✓	-	✓	-Contains multiple weather conditions	-No night data -No LiDAR
Comma.ai	-	-	✓	Real-world	7.25	Single	✓	✓	✓	-	✓	-Contains night driving	-Few environments -No LiDAR

expert intention from being required.

Chi *et al*. [33], Haavaldsen *et al*. [32] and Hawke *et al*. [28] primarily focus on overcoming the issue of multi-modality through the use of temporal information. Modelling the steering angle as a temporally continuous variable not only shows potential in alleviating this problem, but also improving performance compared to competing algorithms. However, the multi-modality of the solution is not entirely relieved. High-level intention, driving condition, and many other factors can also lead to solution ambiguity and combinations of this information as inputs to the network needs further study.

Direct Perception - Direct perception is an alternate form of end-to-end model design. Despite the task of outputting indicators for use by a control module being a slightly more complex one than control signal prediction, an advantage of this approach is the ability to train networks to provide long time horizon planning. This can significantly improve the safety of such systems whilst providing any observers with information about the systems long term intentions.

The majority of direct perception approaches utilise multi-modal inputs, straying away from the primarily vehicle-mounted camera strategies of control prediction systems. Xiao *et al*. [30] and Barnes *et al*. [37] focus on the inclusion of depth information alongside RGB camera information via LiDAR to improve their models' trajectory generation. Xiao *et al*. investigate including this additional information at varying stages of the end-to-end pipeline to generate driving trajectories, whilst Barnes *et al*. use this additional information to segment the camera images for use in trajectory generation. Both works provide good solutions to the challenge of generations of trajectory generation models, however access to LiDAR for autonomous vehicle systems can still be an issue due to availability and cost.

Works such as Bansal *et al*. [14] and Chen *et al*. [15] show the performance that can be achieved using trajectory planning approaches when they receive a comprehensive birds-eye view overview of their environment as an input. These models successfully produce successful, safe long-horizon trajectories through complex environments. However, in order to provide such comprehensive input information outside of simulation environments requires the development and implementation of state-of-the-art perception systems.

Xu *et al*. [24] attempt to overcome the generalisability issue through the acquisition of a training dataset significantly larger than any other presently available. The collection of such a dataset significantly increases the number and range of scenarios present for training. However, such an approach struggles to completely overcome the issue of generalisability, as a finite dataset cannot contain all possible driving scenarios. Beyond this, the utilisation of data from a wide range of sources, from multiple human drivers, introduces the issue of differing driving styles between each source. Known as *latent variables*, these factors should be modelled which brings a further challenge.

Caltagirone *et al*. [41] (driving path estimation), Barnes *et al*. [37] (driving path estimation) and Richter *et al*. [47] (collision/novelty prediction) all utilise the direct perception approach to provide a driving controller with navigational information. For Caltagirone, Barnes, each method is applied to only a single, specific part of the autonomous driving decision making pipeline. A direct perception based fully autonomous vehicle would require such processing to be conducted on the entire perception pipeline, which may be unfeasible. Richter *et al*. utilise such a system to perform crash prediction to greatly improve the safety of an autonomous vehicle. Performing novelty prediction provides a unique method of preventing system failures due to issues regarding generalisability.

Uncertainty Quantification - Michelmore *et al*. [48], Amini *et al*. [51] and Lee *et al*. [55] utilise uncertainty information through the training of Bayesian Neural Networks

TABLE III
COMPARISON OF AUTONOMOUS DRIVING SIMULATORS. U- URBAN, R - RACING, H - HIGHWAY

Dataset	Environment			Diversity			Sensors			Advantages	Disadvantages
	U	R	H	Weather	Lighting	Multi-Agent	Video Image	LiDAR	Depth Camera		
CARLA	✓	-	✓	✓	✓	✓	✓	✓	✓	-Can utilise custom built scenarios -Multi-agent capabilities	-No racing track environments -Complex simulation suite
TORCS	-	✓	-	✓	✓	-	✓	✓	✓	-Complex physics engine -Advanced racing simulator -Number of user made tracks/content available	-No multi-agent simulation -No urban/highway driving
Udacity	✓	✓	✓	-	-	-	✓	-	-	-Large number of user created tracks -Easy to use	-Does not allow LiDAR simulation -No multi-agent simulation
GTA V	✓	✓	✓	✓	✓	✓	✓	-	-	-Complex multi-agent traffic/pedestrian behaviour -Wide range of environments and weather conditions	-Only camera inputs available -Need software to interact with videogame
rFpro	✓	✓	✓	✓	✓	✓	✓	✓	✓	-Complex traffic scenarios -Human users can be part of testing scenarios	-Simulator is not free

to improve the safety assurances of their model. Their models perform to state-of-the-art standards whilst allowing for the extraction of model uncertainty. Michelmore *et al*. utilise the uncertainties to predict future crashes, however uncertainty information has many potential applications in such a system, such as modelling road user behaviour, identification of edge cases etc. The work presents a novel, safety driven approach to autonomous vehicle development which is much needed in the field. Whilst the system itself solely generates steering angles, the further development of such a system is of importance. However, the training of BNN's is a challenging task. The approaches approximate BNN using dropout and as such require multiple passes over the network for each input to attain the distribution of values, which can be a computationally expensive task and limit the models' abilities to act in real-time.

*2) **Direct Policy Learning**:* Direct Policy Learning approaches attempt to expand upon BC in order to overcome some of its limitations. The DAgger [57] algorithm provides a very strong framework for overcoming the issues of cascading errors and enhancing generalisability. Online Imitation Learning algorithms inherently provide a level of robustness to these challenges due to their aggregation of data. The main limitation of this approach is the requirement of constant access to an expert during the training process, which is demanding and limits the applicability of the approach. The SafeDAgger [58] and HG-DAgger [65] algorithms mitigate this issue through the efficiency of expert queries, however access to human experts at training time is still required.

Li *et al*. [74] attempt to alleviate some of the challenges of requiring access to an expert through the development of the Observational Imitation Learning algorithm. This algorithm utilises a large number of experts, provided through the use of a number of simple PID controllers. This is a much cheaper expert to provide than a human. The use of OIL finds the best expert at each iteration and over a number of iterations allows multiple experts to train the network. This has the effect of allowing the network to reach superior performance to any individual expert, a very promising result. The main drawback of the work is that it only uses simple PID controller, and finding a suitable expert that could be used on mass for complex, real-world driving may be a challenge.

*3) **Inverse Reinforcement Learning**:* Inverse Reinforcement Learning attempts to infer the reward function of driving, to allow the training of a system that can fully generalise to any unseen scenario. However, it is a challenging task, primarily due to the computational complexity of learning a reward function for highly complex tasks. Beyond this, there exist further challenges namely reward function ambiguity. A dataset can be explained by multiple reward functions, only one of which may be the desired one.

Abbeel *et al*. [67] and Ziebart *et al*. [69] apply the method successfully to the task of highway driving human reward function, learning differing driving styles. However, these methods are only applied in simple top-down simulations and further work to apply them to more complex simulations or real-world data is required if such approaches are to catch up to the level of behavioural cloning based approaches.

The most advanced and promising application of Inverse Reinforcement Learning is undertaken by Wulfmeier *et al*.

They develop the DeepIRL framework, utilising deep learning in conjunction with inverse reinforcement learning to learn cost maps of the environment. These cost maps can then be used to construct paths through the environment, similar to the behavioural cloning based approach by Caltagirone *et al*. Whilst showing potential for this task, the development of an Inverse Reinforcement Learning based autonomous vehicle will require significantly more development.

Generative Adversarial Imitation Learning [72] also shows a large amount of promise in terms of addressing all 3 of the primary issues faced by IL. Li *et al*. [74] develop a Generative Adversarial Imitation Learning based system capable of not only successfully learning a driving policy but also learning to distinguish between high-level actions and model latent variables. Kuefler *et al*. [17] use GAIL to imitate human level highway driving in simulation, achieving very promising results compared to BC baselines. These abilities highlight the potential of GAIL for the field of autonomous vehicles. However as a field GAIL is still in its infancy and as such it may be a matter of time before it is ready to be applied to real-world autonomous vehicle systems.

B. Open Challenges

1) Behavioural Cloning:

- Current datasets are not large or diverse enough for the training of a fully autonomous vehicle. For safety critical tasks such as autonomous driving, edge cases and underrepresented scenarios pose significant threats to performance. More complete and large scale datasets for BC approaches are needed to address these issues.
- Current end-to-end systems fail to account for the safety critical nature of the task. Improving safety of autonomous systems is crucial, namely via the introduction of uncertainty modelling or other safety critical approaches. Deep learning often struggles to generalise to novel or unseen scenarios that a model may encounter. This potential drop in performance can be disastrous for safety critical tasks. Models capable of outputting associated uncertainty information or performing novelty detection can address these issues and present a valuable area for future research.
- Current end-to-end approaches fail to effectively model the temporal dependence of driving decisions. This oversimplification limits their effectiveness in the real-world. Attempting to train a model to mimic human driving decision making whilst only allowing it access to a fraction of the information that humans use will almost certainly limit performance.
- The field is lacking a standardised evaluation approach. The development of a standard evaluation metric or test bed for the field is of paramount importance, to enable the comparison and assessment of current methods. Such evaluation should be rigorous, exposing models to a range of novel and dangerous scenarios to fully evaluate not only their driving performance but also their safety level. Current benchmarks utilise simulation environments however the construction of a real-world benchmarking approach would be extremely valuable.

2) Direct Policy Learning:

- The requirement of having access to human experts is a costly one, and the study of more query-efficient algorithms is an ongoing challenge. The inclusion of experts at training time can greatly mitigate the limitations of BC approaches, but currently struggle to maximise this potential due to the inefficiency of querying the expert. The use of non-human experts and the development of models that are very query efficient can minimise this issue, however further work into such approaches is needed.
- Algorithms that allow the model to reach performance greater than its experts show promise, but only currently are viable for very simple expert policies. Expansion of such approaches to more complex expert policies could greatly improve model performance.
- The majority of Direct Policy Learning approaches have only been implemented in simulation, expansion to real-world environments is an open challenge.

3) Inverse Reinforcement Learning:

- The majority of current IRL based approaches are limited to simple simulations. Expansion of IRL based approaches to more complex, realistic simulation environments/real-world datasets should be made to allow the field to catch up with Behavioural Cloning based solutions.
- Current leading deep learning based IRL approaches are focused on direct perception based approach, further expansion to attempt to utilise such methods to develop control systems should be made.
- The computational complexity of IRL approaches is a significant limitation of the method. More computationally efficient methods of acquiring reward functions are needed if the approach is to become suitable for real-world autonomous driving.
- Generative Adversarial Imitation Learning models are difficult to train and computationally intensive. They have been shown to be unstable during training, and for smaller datasets they can take a long time to converge.
- The theoretical background to GAIL is still largely unknown. For any significant progress to be made into developing the method, more work is needed to ascertain its theoretical background.
- The utilisation of online learning in GAIL based approaches could significantly relieve the issue of sample inefficiency by guiding the learning process.
- Studies such as [77] have also shown that GAIL can fail to generalise to differing environmental dynamics, and more work is needed to investigate this potentially significant limitation of the approach.

V. Conclusion

This paper provides an overview of state-of-the-art Imitation Learning based methods, their applications in the field of autonomous vehicles and discusses open challenges that still need to be addressed. The field is classified into three main approaches; Behavioural Cloning, Direct Policy Learning

and Inverse Reinforcement Learning for each of which the current state-of-the-art is presented and reviewed. Based on this review, open challenges in the field are identified such as data enhancement, robust learning mechanism, safety-critical autonomy, and the development of a standardised evaluation metric. Due to the fact that Imitation Learning, as with a large amount of deep-learning paradigms, is a data-driven approach, the review also summarises existing datasets and simulators, exploring their potential applications. It is anticipated that this survey can serve as a primary starting point to researchers who are about to enter this exciting area and to give a comprehensive overview to the existing research.

REFERENCES

[1] X. Mosquet, T. Dauner, N. Lang, N. Rubmann, A. Mei-Pochtler, R. Agrawal, and F. Schmieg, "Revolution in the driver's seat: The road to autonomous vehicles," *Boston Consulting Group*, vol. 11, 2015.

[2] D. M. West, "Securing the future of driverless cars," 2016.

[3] I. Markit, "Self-driving cars moving into the industry's driver's seat," *IHS Online Newsroom*, pp. 1–2, 2014.

[4] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2722–2730.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[8] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, p. 21, 2017.

[9] L. Tai, J. Zhang, M. Liu, J. Boedecker, and W. Burgard, "A survey of deep network solutions for learning control in robotics: From reinforcement to imitation," *arXiv preprint arXiv:1612.07139*, 2016.

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

[11] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances in neural information processing systems*, 1989, pp. 305–313.

[12] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. L. Cun, "Off-road obstacle avoidance through end-to-end learning," in *Advances in neural information processing systems*. Citeseer, 2006, pp. 739–746.

[13] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[14] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," *arXiv preprint arXiv:1812.03079*, 2018.

[15] J. Chen, B. Yuan, and M. Tomizuka, "Deep imitation learning for autonomous driving in generic urban scenarios with enhanced safety," *arXiv preprint arXiv:1903.00640*, 2019.

[16] C. Liu, J. Chen, T.-D. Nguyen, and M. Tomizuka, "The robustly-safe automated driving system for enhanced active safety," SAE Technical Paper, Tech. Rep., 2017.

[17] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 204–211.

[18] S. K. S. Ghasemipour, R. Zemel, and S. Gu, "A divergence minimization perspective on imitation learning methods," in *Conference on Robot Learning*. PMLR, 2020, pp. 1259–1277.

[19] J. A. Bagnell, "An invitation to imitation," CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, Tech. Rep., 2015.

[20] 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 preprint arXiv:1704.07911*, 2017.

[21] L. Cultrera, L. Seidenari, F. Becattini, P. Pala, and A. Del Bimbo, "Explaining autonomous driving by learning end-to-end visual attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 340–341.

[22] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," *arXiv preprint arXiv:1711.03938*, 2017.

[23] S. Hecker, D. Dai, and L. Van Gool, "End-to-end learning of driving models with surround-view cameras and route planners," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 435–453.

[24] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2174–2182.

[25] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–9.

[26] Q. Wang, L. Chen, B. Tian, W. Tian, L. Li, and D. Cao, "End-to-end autonomous driving: An angle branched network approach," *IEEE Transactions on Vehicular Technology*, 2019.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[28] J. Hawke, R. Shen, C. Gurau, S. Sharma, D. Reda, N. Nikolov, P. Mazur, S. Micklethwaite, N. Griffiths, A. Shah *et al.*, "Urban driving with conditional imitation learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 251–257.

[29] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8934–8943.

[30] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. M. López, "Multimodal end-to-end autonomous driving," *arXiv preprint arXiv:1906.03199*, 2019.

[31] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9329–9338.

[32] H. Haavaldsen, M. Aasboe, and F. Lindseth, "Autonomous vehicle control: End-to-end learning in simulated urban environments," in *Symposium of the Norwegian AI Society*. Springer, 2019, pp. 40–51.

[33] L. Chi and Y. Mu, "Deep steering: Learning end-to-end driving model from spatial and temporal visual cues," *arXiv preprint arXiv:1708.03798*, 2017.

[34] "Udacity: An open source self-driving car," 2017.

[35] P. M. Kebria, A. Khosravi, S. M. Salaken, and S. Nahavandi, "Deep imitation learning for autonomous vehicles based on convolutional neural networks," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 82–95, 2019.

[36] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[37] D. Barnes, W. Maddern, and I. Posner, "Find your own way:

Weakly-supervised segmentation of path proposals for urban autonomy," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 203–210.

[38] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[39] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[40] P. Cai, Y. Sun, Y. Chen, and M. Liu, "Vision-based trajectory planning via imitation learning for autonomous vehicles," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 2736–2742.

[41] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "Lidar-based driving path generation using fully convolutional neural networks," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[42] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner, "Torcs, the open racing car simulator," *Software available at http://torcs. sourceforge. net*, vol. 4, no. 6, 2000.

[43] M. Aly, "Real time detection of lane markers in urban streets," in *2008 IEEE Intelligent Vehicles Symposium*. IEEE, 2008, pp. 7–12.

[44] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[45] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3d traffic scene understanding from movable platforms," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 1012–1025, 2013.

[46] R. McAllister, Y. Gal, A. Kendall, M. Van Der Wilk, A. Shah, R. Cipolla, and A. V. Weller, "Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning." International Joint Conferences on Artificial Intelligence, Inc., 2017.

[47] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," 2017.

[48] R. Michelmore, M. Kwiatkowska, and Y. Gal, "Evaluating uncertainty quantification in end-to-end autonomous driving control," *arXiv preprint arXiv:1811.06817*, 2018.

[49] L. C. Freeman, *Elementary applied statistics: for students in behavioral science*. John Wiley & Sons, 1965.

[50] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[51] A. Amini, A. Soleimany, S. Karaman, and D. Rus, "Spatial uncertainty sampling for end-to-end control," *arXiv preprint arXiv:1805.04829*, 2018.

[52] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[53] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," *arXiv preprint arXiv:1506.02158*, 2015.

[54] P. Cai, S. Wang, Y. Sun, and M. Liu, "Probabilistic end-to-end vehicle navigation in complex dynamic environments with multimodal sensor fusion," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4218–4224, 2020.

[55] K. Lee, K. Saigol, and E. A. Theodorou, "Safe end-to-end imitation learning for model predictive control," *arXiv preprint arXiv:1803.10231*, 2018.

[56] "Autorally," 2019. [Online]. Available: http://autorally.github.io/

[57] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 627–635.

[58] J. Zhang and K. Cho, "Query-efficient imitation learning for end-to-end autonomous driving," *arXiv preprint arXiv:1605.06450*, 2016.

[59] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. A. Theodorou, and B. Boots, "Imitation learning for agile autonomous driving," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 286–302, 2020.

[60] Y. Pan, X. Yan, E. A. Theodorou, and B. Boots, "Prediction under uncertainty in sparse spectrum gaussian processes with applications to filtering and control," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2760–2768.

[61] J. Quinonero-Candela, C. E. Rasmussen, A. R. Figueiras-Vidal *et al.*, "Sparse spectrum gaussian process regression," *Journal of Machine Learning Research*, vol. 11, no. Jun, pp. 1865–1881, 2010.

[62] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[63] G. Li, M. Müller, V. Casser, N. Smith, D. L. Michels, and B. Ghanem, "Oil: Observational imitation learning," *arXiv preprint arXiv:1803.01129*, 2018.

[64] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, "Sim4cv: A photo-realistic simulator for computer vision applications," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 902–919, 2018.

[65] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer, "Hg-dagger: Interactive imitation learning with human experts," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8077–8083.

[66] A. Y. Ng, S. J. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, vol. 1, 2000, p. 2.

[67] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 1.

[68] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions." in *Robotics: Science and Systems*, vol. 2. Ann Arbor, MI, USA, 2016.

[69] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," 2008.

[70] M. Wulfmeier, P. Ondruska, and I. Posner, "Maximum entropy deep inverse reinforcement learning," *arXiv preprint arXiv:1507.04888*, 2015.

[71] M. Wulfmeier, D. Z. Wang, and I. Posner, "Watch this: Scalable cost-function learning for path planning in urban environments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2089–2095.

[72] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in neural information processing systems*, 2016, pp. 4565–4573.

[73] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International conference on machine learning*. PMLR, 2016, pp. 1329–1338.

[74] Y. Li, J. Song, and S. Ermon, "Inferring the latent structure of human decision-making from raw visual inputs," *arXiv preprint arXiv:1703.08840*, vol. 2, 2017.

[75] Epic Games, "Unreal engine." [Online]. Available: https://www.unrealengine.com

[76] "Driving simulation for autonomous driving, adas, vehicle dynamics and motorsport," Nov 2019. [Online]. Available: http://www.rfpro.com/

[77] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," *arXiv preprint arXiv:1710.11248*, 2017.

Luc Le Mero recieved his BSc/MPhys degree in Physics in 2015 from the University of Warwick. In 2017 he received an MSc in Scientific Computing from the University of Warwick. Now he is a PhD candidate in the Warwick Manufacturing Group, University of Warwick, UK.

Dewei Yi (M'18) received his B.Eng. degree in 2014 in Software Engineering from Zhejiang University of Technology, Zhejiang, China. In 2015, he obtained his M.Sc. degree from the Department of Computer Science, Loughborough University, Loughborough, U.K. In 2018, he received a Ph.D degree from the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, U.K. In 2019, he was a research fellow with the Warwick Manufacturing Group (WMG), University of Warwick, Coventry, U.K. He is currently a lecturer in Machine Learning, Department of Computing Science, University of Aberdeen, U.K. His current research interests include personalised driving assistance, autonomous vehicle and vehicular network, Advanced Driver Assistance Systems (ADASs),Generalised and Sustainable Artificial Intelligence (AI) Systems, and Hybrid Intelligent Systems and Robotics.

Mehrdad Dianati (Senior Member, IEEE) is a Professor of autonomous and connected vehicles, as well as, the Head of Intelligent Vehicles Research Department and the Technical Research Lead in the area of Networked Intelligent Systems at the Warwick Manufacturing Group (WMG), the University of Warwick, UK. The focus of his research is on the application of Digital Technologies (Information and Communication Technologies and Artificial Intelligent) for the development of future mobility and transport systems. He has over 29 years of combined industrial and academic experience, with 20 years in various leadership roles in multi-disciplinary collaborative R&D projects. He works closely with the Automotive and ICT industries as the primary application domains of his research. He is also the Director of Warwick's Centre for Doctoral Training on Future Mobility Technologies, training doctoral researchers in the areas of intelligent and electrified mobility systems in collaboration with the experts in the field of electrification from the Department of Engineering of the University of Warwick. In the past, he has served as an Editor for the IEEE Transactions on Vehicular Technology and several other international journals, including IET Communications. Currently, he is the Field Chief Editor of Frontiers in Future Transportation.

Alexandros Mouzakitis has over 15 years of technological and managerial experience, especially in the area of automotive embedded systems. In his previous position with JLR, he has served as the Head of the Model-based Product Engineering Department, responsible for the model-based development and automated testing standards and processes. He is currently the Head of the Electrical, Electronics and Software Engineering Research Department, Jaguar Land Rover. In his current role, he is responsible for leading a Multidisciplinary Research And Technology Department dedicated to deliver a portfolio of advanced research projects in the areas of human–machine interface, digital transformation, self-learning vehicle, smart/connected systems, and on board/off-board data platforms.