Deep Inverse Reinforcement Learning for Behaviour Prediction in Autonomous Driving

Tharindu Fernando, *Member, IEEE,* Simon Denman, *Member, IEEE,* Sridha Sridharan, *Life Senior Member, IEEE,* and Clinton Fookes, *Senior Member, IEEE*

Abstract

Accurate behaviour anticipation is essential for autonomous vehicles when navigating in close proximity to other vehicles, pedestrians and cyclists. Thanks to the recent advances in deep learning and inverse reinforcement learning we observe a tremendous opportunity to address this need, which was once believed impossible given the complex nature of human decision making. In this article, we will summarise the importance of accurate behaviour modelling in autonomous driving and analyse the key approaches and the major progress that researchers have made, focusing on the potential of Deep Inverse Reinforcement Learning (D-IRL) to overcome the limitations of previous techniques. We provide quantitative and qualitative evaluations substantiating these observations. While the field of D-IRL has seen recent successes, it's application to model behaviour in autonomous driving is largely unexplored. As such we conclude this article by summarising the exciting pathways for future breakthroughs.

I. Introduction

Consider the example shown in Fig. 1 If you are driving the blue car and want to turn right at the intersection, you will try to predict the behaviour of the yellow car, considering aspects

T. Fernando, S.Denman, S. Sridharan, C. Fookes are with Image and Video Research Laboratory, SAIVT, Queensland University of Technology, Australia.

E-mail: t.warnakulasuriya@qut.edu.au

Manuscript received

including the yellow car's speed and acceleration, the distance the yellow car is from intersection, and the amount of time it would take you to turn. You will make this decision intuitively in a split second, based on years of driving experience and likely experience of several similar instances, as well using your intuition of human social behaviour. We pose the question: How do we teach the driverless cars to make these same predictions, judgments and decisions?

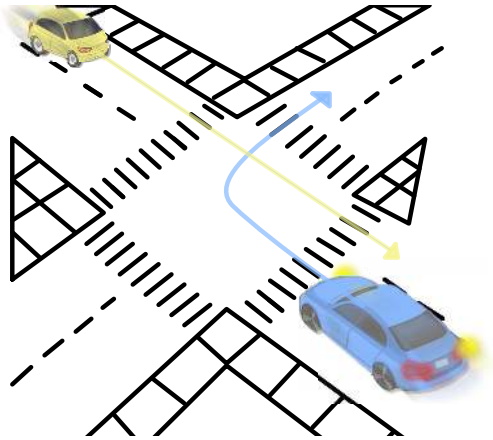

Fig. 1: A sample driving scenario. The blue car is waiting to turn right at an intersection and there is a car coming from the opposite direction. The driver in the blue car should anticipate the future behaviour of the yellow vehicle before determining their next action.

Social prediction is an extraordinary feat that human drivers routinely employ to assist their decision making while travelling in close proximity to other vehicles, with conflicting objectives and incomplete information regarding the objectives of other people in the scene [1]. As such, prediction is a pivotal component in self driving cars. Recognising this, in February 2018 Sam Anthony, Harvard neuroscientist, CTO and cofounder of Perceptive Automat (an autonomous vehicle software company) said "self driving cars should learn human intuition and human social behaviour before they can become a part of urban life" [2]. This followed his earlier remarks in August 2017 that one of the key challenges for safety in self driving cars is the inability of machine learning algorithms to look at a person on the road, and irrespective of whether they are walking, driving a car, or riding a bike predict their future behaviour [3].

A major hindrance to making accurate future predictions comes from the trade-offs that humans make between arbitrary complex factors (i.e. their surroundings, the route, behaviour, risk, resource and goal oriented factors) when making their own decisions. However, as humans,

through experience we have mastered this process over our lifetime, and we seamlessly adapt our behaviour. To date, making such predictions autonomously has eluded the machine learning and autonomous driving community. However recent developments in areas such as Inverse Reinforcement Learning (IRL) have the potential to address this limitation.

The rest of this paper is organised as follows. Section II reviews current state-of-the-art methods for behaviour modelling in autonomous driving, which are evaluated on two public driving benchmark datasets in Section III. Section IV details the limitations and challenges in deep IRL based behaviour modelling, and Section V concludes the paper.

II. Behaviour Modelling in Autonomous Driving: A Review

A. Model-based Learning and Supervised Learning

The main modules in a generalised autonomous driving framework can be broadly categorised as sensor fusion, localisation, prediction and motion control. The sensory inputs are captured and fused in order to localise and predict the future trajectory of the agents in the local neighbourhood. Utilising these predictions, the future trajectory of the autonomous vehicle is generated and subsequently passed to the motion control subsystem to generate the control commands. The prediction module in the autonomous car is able to use behaviour modelling techniques and these algorithms can be broadly categorised into model based and learning based approaches.

In model based approaches factors that inform human behaviour are hand engineered and combined to optimise a pre-defined objective such as proximity to other vehicles, the number of lane changes or the risk of taking a particular trajectory. In contrast, in learning based systems underlying factors that influence human sociological factors are recovered from the data.

Among model based approaches, Li et. al [4] uses a trajectory planing scheme which samples trajectories from the global reference path leading to the goal state. A velocity profile generates the velocity for each state along the generated path. Finally the best path is chosen based on a cost function considering safety and comfort. The trajectory generation algorithm of [5] computes a trajectory by minimising the distance to the goal state, the distance to the centreline path and maximising the proximity to the obstacles. This was extended in [6] by augmenting the proximity cost to discourage picking paths which are close to dangerous drivers, cyclists and pedestrians.

Despite these attempts it is infeasible to hand engineer a cost function that can consider all factors that influence the future behaviour of people in the vicinity of an autonomous vehicle.

As opposed to model based systems, learning based systems try to automatically recover these factors from the data. A popular family of learning based algorithm is supervised learning. These algorithms utilise past observed trajectories of the autonomous agent and agents in the local neighbourhood, and learn to predict the future trajectory of the autonomous agent. The process is data driven as the model minimises the distance between the predicted and the ground truth trajectories using a pre-defined loss function, such as the Mean Square Error (MSE) [7].

In [8] the authors propose the utilisation of social pooling to capture inter dependencies between neighbouring vehicles in motion. The authors encode past trajectories of autonomous vehicle as well as the neighbouring agents using Long Short-Term Memory networks (LSTMs) [9], and to capture the inter-dependencies of nearby agents they pool out LSTM states based on their spatial configuration in the scene. These states are subsequently passed through a series of convolutional and pooling layers and the future trajectory is generated by a decoder LSTM. The framework is trained to minimise the negative log likelihood loss between the predicted and ground truth trajectories. In [10] the authors extend this encoder-decoder LSTM framework for joint trajectory prediction and manoeuvre classification. They illustrate that manoeuvre dependent trajectory prediction is comparatively more resilient than predicting the trajectory alone.

Most recently Zhao et al. [1] proposed a framework which encodes past trajectories of neighbouring agents using LSTMs and captures the scene context using convolutional neural networks. Then this information is fused and passed to a decoder LSTM to generate the future trajectory of the autonomous agent. This framework is learned through a combination of MSE loss and adversarial loss which is achieved through a GAN learning process [11].

In addition to [1] which has achieved favourable results on both highway driving and pedestrian trajectory datasets, it is worth noting that supervised learning systems such as Soft + Hard-wired Attention [12] and Social GAN [13] have been proposed to automatically recover human social navigation behaviour in crowded environments. However, these systems were developed and evaluated using pedestrian trajectory data.

B. Generative Adversarial Imitation Learning

Despite their reasonable success, supervised learning approaches cannot recover the underlying factors that influence human social behaviour [14] as they operate using a pre-defined cost function which does not fully capture human reasoning. There exists another class of algorithms,

Generative Adversarial Imitation Learning (GAIL) [15], which seeks to directly mimic the expert's policy, and has been extensively applied for autonomous driving tasks [16]–[18].

Let the decision making process of the pedestrians be modelled as a Markov Decision Process (MDP) [19]. The MDP, $M = [S, A, \tau, R]$, is composed of state space, S; set of possible actions, A; a transition matrix, τ; and a reward function, R. A policy, π, defines the selection of an action given a particular state. We are presented with a set of demonstrations, $D = [\zeta^1, \zeta^2, \ldots, \zeta^N]$, where each demonstration, ζ^i, is composed of state (s_t) and action (a_t) pairs, $\zeta^i = [s_0, s_1, \ldots, s_{T_{obs}}]$. Then the GAIL objective is denoted by,

$$\min_\theta \max_w V(\theta, w) = \mathbb{E}_{\pi_\theta}[\log D_w(s, a)] + \mathbb{E}_{\pi_E}[\log D_w(s, a)], \tag{1}$$

where policy π_θ is a neural network parameterised by θ which directly generates the policy imitating π_E, and D_w is the discriminator network parameterised by w which tries to distinguish state-action pairs from π_θ and π_E. $\mathbb{E}_\pi[f(s, a)]$ denotes the expectation of f over state action pairs generated by policy π.

Numerous works [16]–[18] have utilised GAIL for predicting trajectories in simulated highway driving scenarios. In [17] the authors utilise eight features including vehicle speed, lengths, lane curvature and distance to lane markers as state features and utilising the GAIL formulation they predicted the relevant actions given this state representation. In [16] the authors propose a system to leverage the variability among different expert demonstrations. They utilise the information maximisation theorem to automatically discover and disentangle latent factors in the underlying expert demonstrations. In our prior work [18] we proposed the use of Neural Memory Networks (NMNs) [20] to capture relationships at a sub-task level and determine how they are temporally linked in a given expert demonstration.

It should be noted that similar to supervised methods, GAIL does not attempt to recover the reward function. Instead it attempts to directly mimic the expert's policy. Hence, it's applicability to environments with data constraints and its generalisability to new environments remain questionable [21].

C. Inverse Reinforcement Learning (IRL)

Inverse reinforcement learning (IRL), however, has shown promise in being able to address the deficiencies of supervised and imitation learning. Unlike GAIL which directly tells the learner how to act, IRL recovers the underlying reward function, which provides better understanding

regarding modelled behaviour [21]. In an IRL framework, given a set of demonstrations, $D = [\zeta^1, \zeta^2, \ldots, \zeta^N]$, we recover the reward function, R, followed by the demonstrators in the samples. Then, using the recovered reward function a machine can imitate natural human behaviour.

IRL based behaviour prediction techniques segregate the underlying semantics of the scene such that the goal or intention of the agents can be recovered from the modelled reward function. This makes the system more tractable and able to generalise to new environments [21] while demonstrating more accurate predictions into the distant future [22], [23].

One of the most popular approaches for solving IRL problems is Maximum Entropy (MaxEnt) IRL [24], where the expert behaviour is modelled as a distribution to the one of highest entropy [14]. The MaxEnt formulation assumes that the reward function can be calculated as a weighted linear combination of the features, $\Phi(s)$, where Φ is a function that outputs the features of the state, s, and the set of weights θ,

$$R(\Phi(s)) = [\theta]^\top \Phi(s). \tag{2}$$

Capitalising on the merits of IRL, several works [25]–[27] have applied it for behaviour prediction. In [25] the authors first cluster the trajectories in the training set and train a multi-class classifier to classify the cluster identity of a given trajectory. The authors utilise Hidden Markov Models (HMMs) to transform the observed trajectories in each cluster into a set of finite states. Then they recover the reward matrices, R_i, for each cluster, i, using an IRL framework. In the test phase, given an observed partial trajectory, they first predict the cluster identity and using the recovered reward matrix of that particular cluster and the Viterbi algorithm [28], they find the most probable sequence of states for the future trajectory.

In [26] the authors investigate the trade off between social accessibility and task related constraints for navigation. For each demonstrated trajectory they define an acceptability-dependent criteria based on its social acceptability. Then combining this feature together with other task related features such as acceleration, steering, velocity and deviation from lane centres; they apply the MaxEnt algorithm to learn different acceptability-dependent behaviours.

In [27] the authors address the exploding state-space problem in IRL. They propose to replace the reinforcement learning inner loop in IRL with Deep-Q Networks to extend the IRL framework to larger state spaces.

Despite these capabilities, the original Maximum Entropy (MaxEnt) IRL framework [24] and subsequent works [25]–[27] assume that the reward function can be calculated as a weighted

linear combination of the features [21]. This linear mapping from features to the reward severely restricts the reward structure that can be modelled [23].

D. Deep Inverse Reinforcement Learning (D-IRL)

The recent works of Wulfmeier et. al [14] extend IRL to a deep learning setting, lifting the MaxEnt IRL constraints and permitting a non-linear mapping which allows more flexibility for the learned reward structure. Hence,

$$R(\Phi(s)) = f(\theta, \Phi(s)), \tag{3}$$

where f is a non-linear function. The authors of [14] try to maximise the log likelihood of the demonstrated trajectories,

$$L(\theta) = log \prod_{\zeta^i \in D} P(\zeta^i, \theta), \tag{4}$$

where $P(\zeta^i, \theta)$ is the probability of the trajectory ζ^i in demonstration D and

$$\frac{\delta L_D}{\delta \theta} = \mu_D - \mathbb{E}[\mu] \frac{\delta R(\Phi(s))}{\delta \theta}, \tag{5}$$

where μ_D and $\mathbb{E}[\mu]$ are the State Visitation Frequencies (SVF) from the demonstrated and inferred reward functions, respectively. Alg. 1 illustrates the process of refining the reward network in the Maximum Entropy Deep IRL (MED-IRL) framework proposed in [14], where γ is a discount factor for the value iteration algorithm (See. Alg. 2), and α is the learning rate of the deep neural network (DNN). In each iteration, i, of the algorithm, they first evaluate the reward based on the state features, $\Phi(s)$, and the current reward network parameters, θ^i. Then, using the current reward function they apply value iteration [24] to solve the forward Reinforcement Learning (RL) problem, determining the current policy, π^i, based on the current approximation of the reward, $R^i(\Phi(s))$, and the transition matrix, τ. The value iteration algorithm is illustrated in Alg. 2. Within Alg. 1, line 5 computes the gradient with respect to the reward which determines how to update the reward network parameters (line 6). The process is illustrated in Fig. 2.

Recently, MED-IRL has been applied for autonomous driving tasks [14], [23], [29]. Wulfmeier et. al [14] demonstrated the utility of Fully Convolutional Neural (FCN) networks for mapping LIDAR scans of urban environments to traversability maps, which are automatically learned through MED-IRL. The proposed Multi-Scale Fully Convolutional Network (MSFCN) architecture (see Fig. 3) utilises a pooling-based sub-stream to capture spatial invariant features from

Algorithm 1: Maximum Entropy Deep IRL

Input:

D: Demonstrations; S: State space; A: Set of possible actions; τ: Transition matrix; γ: Discount factor for the value iteration algorithm; α: Learning rate of the DNN.

Output: Reward network parameters θ^*

for iteration i = *1* to M **do**

  $R^i(\Phi(s)) = f(\theta^i, \Phi(s))\ \forall_{s \in S}$ `// Forward pass in the reward network`

  $\pi^i = Value\_Iteration(R^i, S, A, \tau, \gamma)$ `// Planning step`

  $\mathbb{E}[\mu^i] = compute\_SVF(\pi^i, S, A, \tau)$

  $\frac{\delta L_D^i}{\delta R^i} = \mu_D - \mathbb{E}[\mu^i]$ `// Gradient calculation`

  $\theta^{i+1} = back\_propagate(\theta^i, \frac{\delta L_D^i}{\delta R^i}, \alpha)$ `// Reward network update`

end

return θ

Algorithm 2: Value Iteration

Input:

R: Current approximation of the reward function; S: State space; A: Set of possible actions; τ: Transition matrix; γ: Discount factor.

Output: ϕ

V(s) = -∞ **repeat**

  $V_t(s) = V(s)$

  $Q(s,a) = r(s,a) + E_{\tau(s,a,s')}[V(s')]$

  $V(s) = max_a(Q_i(s,a))$

until $max_s(V(s) - V_t(s)) < \epsilon$;

return $\phi(a|s) = e^{Q(s,a)-V(s)}$

LIDAR, and a Fully Convolutional Network (FCN) stream which preserves location information from the input data. The proposed system was able to learn an end-to-end mapping from raw inputs to a reward map, utilising more than 25,000 trajectories from over 120km of driving.

In [23], Zhang et al. couple low-level LIDAR scan features together with kinematic features to augment the performance of the MED-IRL framework. The network architecture utilised in [23] is

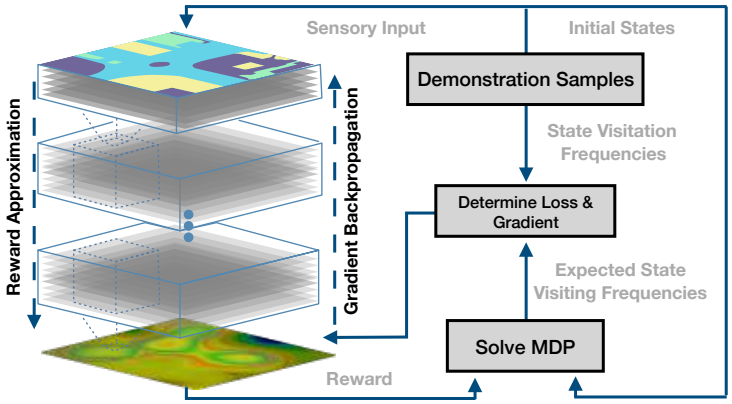

Fig. 2: The schema proposed in [14] for training DNNs using maximum entropy IRL. Given a set of demonstrations, a DNN is utilised to approximate the reward function. Then we calculate the difference between the state visitation frequencies from the demonstrated trajectories and from the inferred reward function. This difference acts as the network's loss and we backpropagate it's gradients, updating the network. Reproduced from [14].

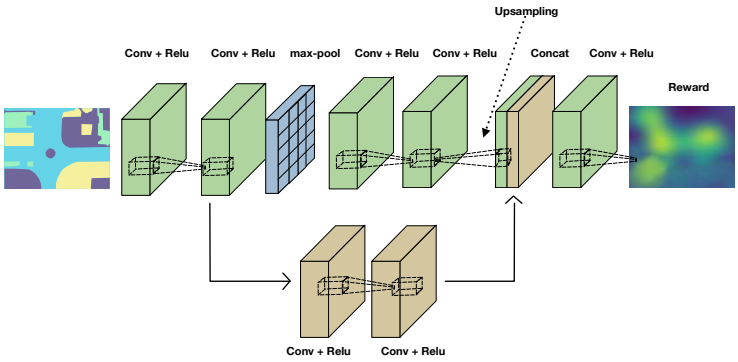

Fig. 3: Illustration of the FCN architecture used by [14] as the reward network. By using a two stream architecture, with an FCN-based main stream and a pooling-based sub-stream, the authors propose to capture spatially variant and invariant features, respectively. Reproduced from [14].

illustrated in Fig. 4. [23] argue that in motion planing human drivers consider kinematic aspects such as the vehicle's current velocity and past trajectory in addition to evaluating the spatial attributes of the environment such as the distance to obstacles. Hence they propose to augment

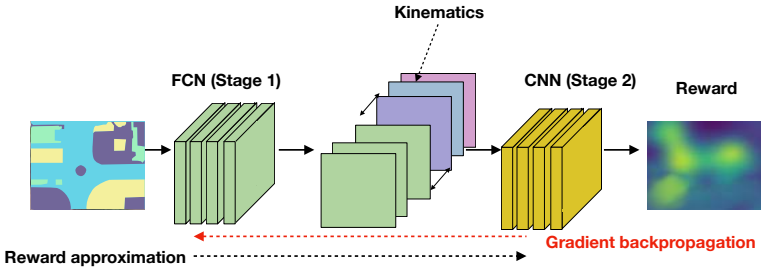

Fig. 4: The two-state architecture proposed in [23]. The authors capture environmental context from input terrain maps in the first-stage network, and the resulting feature maps are concatenated with the kinematic context in the second-stage network, which outputs a reward representation. The difference between the state visitation frequencies from the demonstrated trajectories and the learned policy is used to compute gradients for backpropagation. Reproduced from [23].

the MED-IRL framework of [14] to incorporate this information in two stages. In the first stage they utilise a four layer FCN to encode a colour coded point cloud. In the second stage, the authors utilise two feature maps encoding each grid cell, the x and y positions of the grid cell in a vehicle centred, world-aligned frame. Another three feature maps are generated encoding kinematic information: Δx, Δy and the curvature of the input trajectory. Their evaluations demonstrated greater robustness in predictions compared to both supervised learning and MaxEnt IRL systems.

In [29] the authors refine the MaxEnt [24] formulation, considering both linear and non-linear (MED-IRL) settings to maximise the entropy of the joint distribution over short data pieces. They show that long demonstrations are hard to use in a model free IRL setting, as the prediction error is accumulated over long time horizons. However, this system is validated in simulations of highway driving where the environment is simplified compared to complex urban driving.

Considering the fact that the above mentioned systems do not account for the motion of the neighbours when predicting future motion, most recently we proposed a novel MED-IRL framework for pedestrian trajectory prediction. The trajectories of the agent of interest and the neighbouring trajectories are encoded using LSTMs. Then we utilise a combination of soft and hard-wired attention [12] to aggregate the encoded trajectory information to a context vector.

As the reward network, similar to [14], [24] we utilise a FCN. We first generate an empty

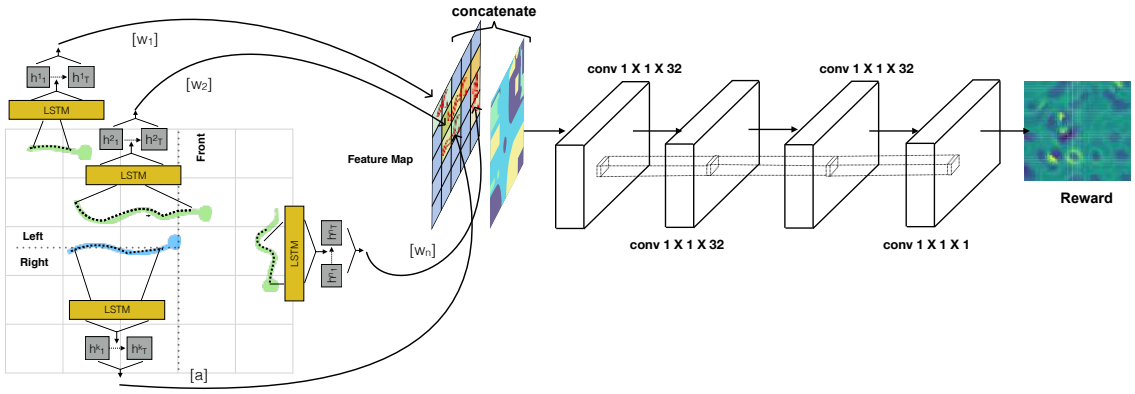

Fig. 5: Left: The architecture used to embed neighbourhood context: The trajectory of the pedestrian of interest is shown in blue, with three neighbours in green. Heading directions are indicated with circles. Trajectories are encoded using LSTMs with soft attention used to embed information from the pedestrian of interest, and hard-wired attention used for the neighbours. Next, a feature map is created to spatially embed this information, based on the cartesian points of each trajectory. Right: The architecture of the four layer FCN used to map the feature map G to the reward map R. The first three layers contain 32, 1×1 convolution kernels with a ReLU activation, and the final layer contains 1, 1×1 convolution kernel. Reproduced from [30].

map, G, of the environment and then we assign values, $\hat{h}_t^k$, from the pedestrian of interest, k, and $\tilde{h}_j^n$ from the neighbours, to the grid, G, based on the Cartesian coordinates that the specific hidden state comes from (i.e based on the position of the trajectory). Then using the FCN we map G to a reward map, R. This architecture is illustrated in Fig. 5.

III. Experimental Evaluations

In this section we report the evaluation results of current state-of-the-art supervised learning, GAIL, Linear-IRL and Deep-IRL systems on the publicly available NGSIM US-101 [31] dataset and a portion of the nuScenes dataset [32].

1) Datasets: The NGSIM US-101 [31] dataset contains trajectories of real freeway traffic captured from fixed overhead cameras placed over a 640-meter span of US101, recorded at 10Hz over a 45 minute period. This dataset consists of more than 6k vehicle annotations and provides varying traffic conditions where the traffic flow varies from mild to moderate to congested.

In addition, we use trajectories from the nuScenes dataset [32], which is captured in multiple cities, from multiple sensors including 6 Cameras, a Lidar, 5 Radars, a GPS sensor and an IMU sensor. The complete dataset contains 15 hours of driving data covering 242km with dense traffic and highly challenging driving situations. The dataset is divided into 1000 scenes by the database authors, and in order to ensure a compatible size between the two evaluations we use only scenes 61, 69, and 234. To generate trajectories we used object bounding box annotations and the centre of the bounding box is taken to be the object position at each time step.

We report the results in terms of Root Mean Squared Error (RMSE) of the predicted trajectory with respect to the ground truth future trajectory over different prediction horizons ranging from 1 second up to 5 seconds. Similar to [17] we simulate the behaviour prediction through a trajectory prediction task where we select each car, iteratively, to be the autonomous car and predict the future behaviour of this car, utilising the past behaviour of the neighbouring vehicles.

2) Evaluated Models:

- Supervised Learning: We use the models of [8] (CS-LSTM), and [1] (MATF-GAN).
- GAIL-GRU: We consider the Generative Adversarial Imitation learning model from [17].
- Linear-IRL: We use the linear-IRL model (L-IRL) proposed in [22].
- Deep-IRL: To demonstrate the utility of deep-IRL models we use the models proposed in [14] (D-IRL), [23] (DK-IRL) and [30] (DN-IRL).

For all the considered systems, similar to [1] the neighbours appearing in the 640-meter span are considered in the reasoning and prediction process. In the original works of [14] and [23] the authors utilise terrain maps captured using LIDAR. As this information is not available in the NGSIM US-101 data, we use the semantic segmentation of the scene which indicates the traversible lanes and for the nuScenes we use the traversability maps generated through LIDAR.

For the GAIL-GRU baseline we follow the policy network architecture of [17], which uses five feedforward layers that decrease in size from 256 to 32 neurons, and an additional GRU layer consisting of 32 neurons. We use the implementation released by the authors[1].

For the Deep-IRL and Linear-IRL systems we consider a grid size of 120×120, and map the x, y coordinates to grid cells. As they generate a probability distribution over the cells, we sample 1000 trajectories from the distribution and measure the average RMSE between the ground truth and samples. We map predictions back to the image coordinate space for clear comparison.

[1]Available in: https://github.com/sisl/gail-driver

For the D-IRL baseline we strictly adhere to the recommendations of the authors and used the FCN architecture introduced in [14]. This takes the semantic segmentation map as the input and generates the reward map purely based on the environment.

For the DK-IRL baseline we follow the two stage architecture of [23] an used the FCN model from D-IRL as the network for the first state. For the second stage, following [23] we generate two feature maps encoding each grid cell, the x and y positions of the grid cell in a pedestrian centred, world-aligned frame. Another three feature maps are generated encoding the kinematic information: Δx, Δy and the input trajectory curvature. We use the codebase released by the authors [2] which also provided an implementation of the D-IRL framework of [14].

For the DN-IRL baseline, as per [30] we consider trajectories of the closest 10 neighbours in front, left and right directions. If there are more than 10 neighbours in any direction, we choose the closest 9 and the mean trajectory of the rest. If there are less than 10 neighbours, we create a dummy trajectories such that we have 10 neighbours for each direction and set dummy trajectory hard-wired weights to zero. For all LSTMs we use a hidden state dimension of 50 units.

3) Results: Quantitative evaluations of the performance of the considered frameworks are presented in Tab. I. To clearly demonstrate the utility of the Deep IRL framework, we perform the trajectory predictions under different prediction horizons, predicting the trajectories from 1 second ahead to 5 seconds ahead. For each trajectory, we use the trajectory for the previous 3 seconds as the observed portion of the trajectory.

From Tab. I we observe that performance of the supervised learning methods degrade when predicting behaviour into the distant future. This is caused by deficiencies in the supervised learning structure, as these models try to directly map inputs to targets, without paying attention to the end goal or intention of the driver. Furthermore, the linear IRL system fails to generate satisfactory results due to constraints with the learnt feature to reward mapping structure.

With the introduction of the non-linear reward mapping from the deep-IRL framework, we observe a slight performance increase with the DK-IRL methods compared to the L-IRL method of [22], however these methods fail to outperform the MATF-GAN system. This is a result of the lack of input information that DK-IRL frameworks receive regarding the neighbourhood context, which is a highly influential factor when navigating in congested environments. However, the LSTM based neighbourhood embedding scheme in the DN-IRL framework is able to capture a

[2]https://github.com/yfzhang/vehicle-motion-forecasting

TABLE I: Evaluation results for NGSIM US-101 [31] and nuScenes [32]. We evaluate performance for different prediction horizons, predicting trajectories from 1 second ahead to 5 seconds ahead. We report RMSE as the error metric (lower is better). For clarity, supervised learning methods are shown with a blue background, the GAIL-GRU method with a orange background, the linear-IRL method with a yellow background and Deep IRL methods with a green background

Results for the NGSIM US-101 dataset [31]

Method	Prediction horizon				
	1s	2s	3s	4s	5s
CS-LSTM [8]	0.61	1.27	2.09	3.10	4.37
MATF GAN [1]	0.66	1.34	2.08	2.97	4.13
GAIL-GRU [17]	0.69	1.51	2.55	3.65	4.71
L-IRL [22]	1.12	2.29	2.31	3.38	4.45
D-IRL [14]	1.35	2.57	2.83	3.69	4.88
DK-IRL [23]	1.09	2.05	2.27	2.91	4.40
DN-IRL [30]	**0.54**	**1.02**	**1.91**	**2.43**	**3. 76**

Results for the nuScenes [32] datasets

Method	Prediction horizon				
	1s	2s	3s	4s	5s
Social GAN [13]	0.93	1.49	2.67	3.32	5.89
GAIL-GRU [17]	1.39	2.02	2.98	4.05	5.87
L-IRL [22]	1.44	2.68	3.57	3.59	5.51
D-IRL [14]	1.61	2.93	3.12	4.21	5.19
DK-IRL [23]	1.23	2.53	3.03	3.52	4.94
DN-IRL [30]	**0.75**	**1.25**	**2.35**	**2.59**	**4. 55**

notion of the neighbourhood, and this results in superior performance. We observe a substantial performance increase, especially when predicting behaviour into the distant future.

Due to the architectural differences between the GAIL-GRU and DN-IRL methods, their performance is not directly comparable. Further evaluation is necessary with identical network architectures to compare the relative strengths and weaknesses. However, such comparisons are currently constrained by the non-public availability of such advanced GAIL architectures.

Qualitative results of the DN-IRL method and the recovered reward representation for four examples from the NGSIM dataset [31] are given in Fig. 6. Analysing the predictions in Fig. 6 we observe that the DN-IRL method achieves good performance when predicting lengthier trajectories. Furthermore, we observe that the DN-IRL method, by virtue of it's neighbourhood modelling, is capable of predicting complex maneuvers such as lane changes and overtaking.

IV. Limitations and Open Research Challenges

While MED-IRL provides flexibility and robustness for behaviour anticipation, it is yet to be widely adopted for autonomous driving systems. Despite great potential for generating realistic hypotheses of human behaviour, there are several open research questions requiring further study.

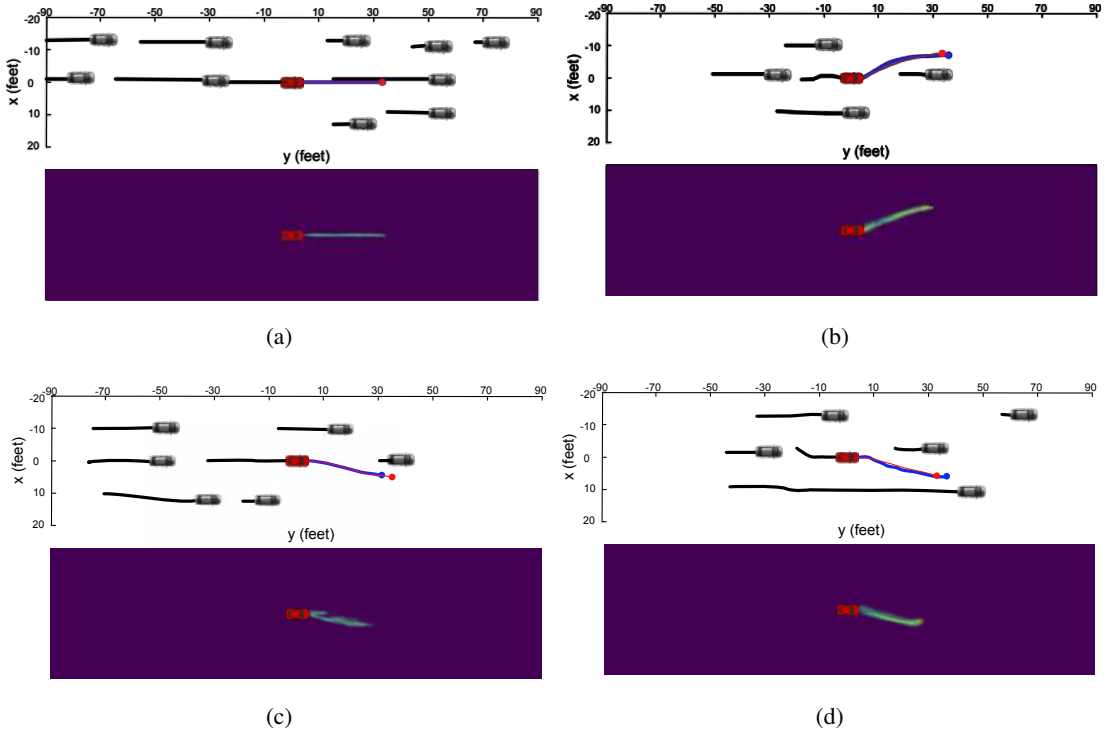

Fig. 6: Qualitative results of the DN-IRL method. The observed part of the trajectories is shown in black. The autonomous agent is indicated by the red car and the neighbouring vehicles are denoted by black vehicles. The ground truth future trajectory is given in blue while the predictions are in red. In the probability map the colours from blue to yellow indicate low to high probability.

To the best of our knowledge, the only D-IRL framework which considers complex dynamic environments with multiple agents in motion is presented in [30]. Yet, in [30] the temporal nature of the agent's motion is ill represented through the current formulation of the reward network. Furthermore, the hardwired attention formulation of [30] is perhaps less impactful in a driving context than it is for pedestrian motion, which [30] is originally proposed for. In addition, the type of neighbour, i.e. a car, truck, motorbike or pedestrian, may also be important, yet is not considered. Hence, further investigation is required to determine effective ways to learn spatio-temporal contextual factors that impact an agent's behaviour when there are large numbers of different types of mobile agents.

Another interesting pathway for investigation is a methodology to capture subtle differences among different expert demonstrations via the reward network formulation. There are often clear

differences in expert behaviour due to varied user preferences and domain knowledge, even though all experts perform the same task. The work of Li et. al [16] learns these differences by conditioning the learned low-level actions on a latent variable and discriminates expert demonstrations based on their structure in the GAIL setting. In our prior work we investigated using neural memory networks to capture these factors in GAIL [18] and supervised learning settings [33]. The viability of these methods in the MED-IRL setting is an open question. In addition, MED-IRL assumes a fixed transition model, τ. However this formulation may limit the robustness of learned policies when there are changes in dynamics such as significant environmental variations (i.e. changes in weather or traffic conditions).

The work of Fu et. al [21] investigated applying an Adversarial IRL framework to disentangle the policy and reward function. A-IRL has been formulated by combining GAIL and Guided Cost Learning (GCL) [34]. Compared to GAIL it learns both the reward function and the policy, and compared to GCL it learns in an adversarial learning setting. Although evaluations in [21] demonstrate increased robustness in high-dimensional environments with significant domain shifts between demonstrations, further investigation is required to enable the method to mitigate the sub-optimality in the given samples when the demonstrators do not follow optimal behaviour.

In the current formulation of the MED-IRL algorithm, value iteration (see Algorithm 2) is used to solve the forward RL problem in the loop. Numerous works have demonstrated that value iteration has a very slow convergence rate [23], [35]. In the work of Zhang et. al [23] the authors utilised a technique called annealed softmax where they artificially increase the probability of the most likely action being chosen. However, more investigation is required to determine best ways to speedup the convergence of value iteration.

In addition, little effort has been made to leverage the multi-modal data captured by autonomous vehicles. In [14] and [23] the terrain maps are captured using LIDAR scans; however systems can be designed to utilise the complementary information that is available through sources such as RGB, infrared and thermal cameras, and radar sensors, which are readily available in a typical autonomous driving setting. These sensors could provide information at different granularities and different ranges, enabling better neighbourhood modelling for decision making.

The lack of availability of well annotated public benchmarks poses another hinderance. Only a limited number of datasets such as Nuscenes [32] and KITTI [36] have annotation relating to other agents in the scene including pedestrians and cyclists. Hence introducing public benchmarks

with richer annotations could promote the swift implementation and evaluation of behavioural prediction systems for real world autonomous driving systems.

V. Conclusion

We have presented an overview of the current state-of-the-art techniques applied for behaviour prediction in autonomous driving. We reviewed popular approaches, including model-based learning, supervised learning, generative adversarial imitation learning, IRL and Deep IRL methods. We quantitatively and qualitatively evaluated these frameworks on two public driving benchmark datasets and demonstrated the utility of D-IRL, especially when making predictions into the distant future. Despite the undoubted potential of Deep IRL methods there are several shortcomings at present and a number of promising research avenues for future breakthroughs are discussed to further advance the field, and realise the goal of fully autonomous vehicles.

VI. Authors

Tharindu Fernando received his BSc (special degree in computer science) and Ph.D. degrees from the University of Peradeniya, Sri Lanka and the Queensland University of Technology (QUT), Australia, respectively. He is currently a Postdoctoral Research Fellow in the SAIVT Research Program in the School Electrical Engineering and Computer Science at QUT. His research interests focus mainly on human behaviour analysis and prediction.

Simon Denman received a BEng (Electrical), BIT, and PhD in the area of object tracking from the Queensland University of Technology (QUT) in Brisbane, Australia. He is currently a Senior Lecturer within the School of Electrical Engineering and Computer Science at QUT. His active areas of research include intelligent surveillance, video analytics, and video-based recognition.

Sridha Sridharan has a BSc (Electrical Engineering) and MSc (Communication Engineering) from the University of Manchester, UK and a PhD from University of New South Wales, Australia. He is currently with the Queensland University of Technology (QUT) where he is a Professor in the School Electrical Engineering and Computer Science. Professor Sridharan is the Leader of the Research Program in Speech, Audio, Image and Video Technologies (SAIVT) at QUT, with strong focus in the areas of computer vision, pattern recognition and machine learning. He has 600 publications in the areas of Image and Speech technologies.

Clinton Fookes received his B.Eng. (Aerospace/Avionics), MBA, and Ph.D. degrees from the Queensland University of Technology (QUT), Australia. He is currently a Professor and Head of

Discipline for Vision and Signal Processing within the Science and Engineering Faculty at QUT. He actively researchers across computer vision, machine learning, and pattern recognition areas. He serves on the editorial board for the IEEE Transactions on Information Forensics & Security. He is a Senior Member of the IEEE, an Australian Institute of Policy and Science Young Tall Poppy, an Australian Museum Eureka Prize winner, and a Senior Fulbright Scholar.

REFERENCES

[1] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 126–12 134.

[2] (2019). [Online]. Available: https://www.daimler.com/innovation/next/what-does-the-machine-need-human-intuition.html

[3] (2019). [Online]. Available: https://qz.com/1064004/self-driving-cars-still-cant-mimic-the-most-natural-human-behavior/

[4] X. Li, Z. Sun, D. Cao, D. Liu, and H. He, "Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles," *Mechanical Systems and Signal Processing*, vol. 87, pp. 118–137, 2017.

[5] V. Cardoso, J. Oliveira, T. Teixeira, C. Badue, F. Mutz, T. Oliveira-Santos, L. Veronese, and A. F. De Souza, "A model-predictive motion planner for the iara autonomous car," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 225–230.

[6] A. B. D. Manocha, "Behavior modeling for autonomous driving," *AAAI Fall Symposium on Reasoning and Learning in Real-World Systems for Long-Term Autonomy (LTA)*, 2018.

[7] E. L. Lehmann and G. Casella, *Theory of point estimation*. Springer Science & Business Media, 2006.

[8] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1468–1476.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1179–1184.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[12] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection," *Neural networks*, vol. 108, pp. 466–478, 2018.

[13] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2255–2264.

[14] M. Wulfmeier, D. Rao, D. Z. Wang, P. Ondruska, and I. Posner, "Large-scale cost function learning for path planning using deep inverse reinforcement learning," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1073–1087, 2017.

[15] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 4565–4573.

[16] Y. Li, J. Song, and S. Ermon, "Infogail: Interpretable imitation learning from visual demonstrations," in *Advances in Neural Information Processing Systems*, 2017, pp. 3812–3822.

[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] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Learning temporal strategic relationships using generative adversarial imitation learning," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2018, pp. 113–121.

[19] R. Bellman, "A markovian decision process," *Journal of Mathematics and Mechanics*, vol. 6, no. 5, pp. 679–684, 1957.

[20] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Pedestrian trajectory prediction with structured memory hierarchies," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 241–256.

[21] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," *International Conference on Learning Representation, (ICLR)*, 2018.

[22] K. Saleh, M. Hossny, and S. Nahavandi, "Long-term recurrent predictive model for intent prediction of pedestrians via inverse reinforcement learning," in *2018 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2018, pp. 1–8.

[23] Y. Zhang, W. Wang, R. Bonatti, D. Maturana, and S. Scherer, "Integrating kinematics and environment context into deep inverse reinforcement learning for predicting off-road vehicle trajectories," *Conference on Robot Learning (CoRL)*, 2018.

[24] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *Aaai*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[25] T. V. Le, S. Liu, and H. C. Lau, "A reinforcement learning framework for trajectory prediction under uncertainty and budget constraint," in *Proceedings of the Twenty-second European Conference on Artificial Intelligence*. IOS Press, 2016, pp. 347–354.

[26] M. Herman, V. Fischer, T. Gindele, and W. Burgard, "Inverse reinforcement learning of behavioral models for online-adapting navigation strategies," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3215–3222.

[27] S. Sharifzadeh, I. Chiotellis, R. Triebel, and D. Cremers, "Learning to drive using inverse reinforcement learning and deep q-networks," *NIPS workshop on Deep Learning for Action and Interaction*, 2016.

[28] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[29] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," *Robotics and Autonomous Systems*, vol. 114, pp. 1–18, 2019.

[30] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Neighbourhood context embeddings in deep inverse reinforcement learning for predicting pedestrian motion over long time horizons," *Proceedings of the IEEE*

International Conference on Computer Vision Workshops, vol. 108, pp. 466–478, 2019.

[31] J. Colyar and J. Halkias, "Us highway 101 dataset," *Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT-07-030*, 2007.

[32] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[33] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Going deeper: Autonomous steering with neural memory networks," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 214–221.

[34] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *International conference on machine learning*, 2016, pp. 49–58.

[35] D. Wingate, "Solving large mdps quickly with partitioned value iteration," 2004.

[36] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.