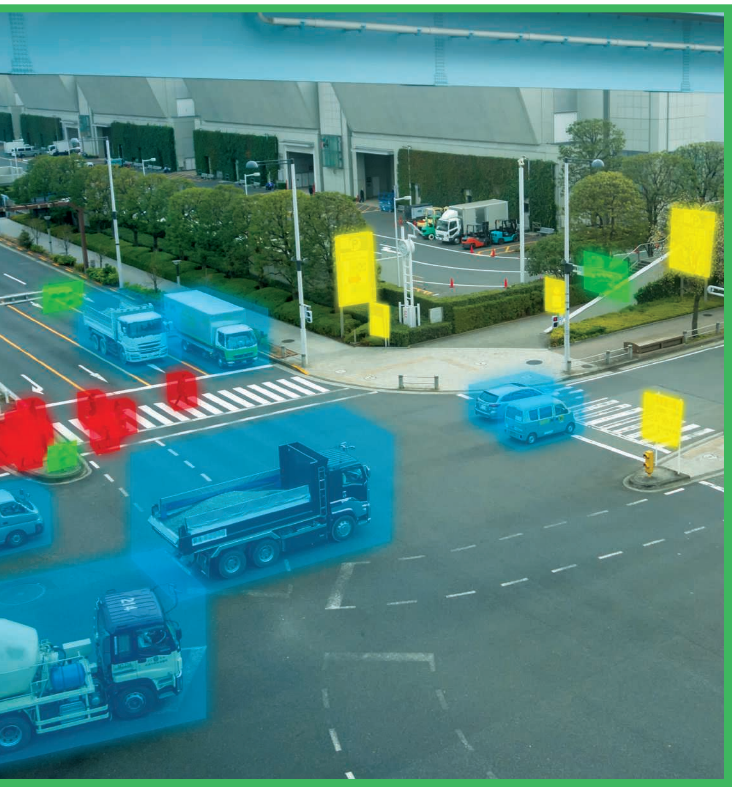
Tharindu Fernando, Simon Denman,
Sridha Sridharan, and Clinton Fookes

Deep Inverse Reinforcement Learning for Behavior Prediction in Autonomous Driving

Accurate forecasts of vehicle motion



©SHUTTERSTOCK.COM/MONOPOLY919

Accurate behavior 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 (IRL), 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 summarize the importance of accurate behavior modeling in autonomous driving and analyze the key approaches and major progress that researchers have made, focusing on the potential of deep IRL (D-IRL) to overcome the limitations of previous techniques. We provide quantitative and qualitative evaluations substantiating these observations. Although the field of D-IRL has seen recent successes, its application to model behavior in autonomous driving is largely unexplored. As such, we conclude this article by summarizing the exciting pathways for future breakthroughs.

Introduction

Consider the example shown in Figure 1. If you are driving the blue car and want to turn right at the intersection, you will try to predict the behavior of the yellow car considering aspects such as the yellow car's speed and acceleration, the distance the yellow car is from intersection, and the amount of time it would take for you to turn. You will make this decision intuitively in a split second, based on years of driving experience with similar instances as well using your intuition of human social behavior. We pose the question: How do we teach driverless cars to make these same predictions, judgments, and decisions?

Social prediction is an extraordinary feat that human drivers routinely employ to assist their decision making while traveling 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. Recognizing this, in August 2017, Sam Anthony, Harvard neuroscientist, chief technology officer, and cofounder of Perceptive Automat (an autonomous vehicle software

Digital Object Identifier 10.1109/MSP.2020.2988287
Date of current version: 24 December 2020

1053-5888/21©2021IEEE

company) said, “Self-driving cars should learn human intuition and human social behavior before they can become a part of urban life” [2]. Later, in July 2018, he mentioned 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 behavior [3].

A major hindrance to making accurate future predictions comes from the tradeoffs that humans make between arbitrary complex factors (i.e., their surroundings, the route, behavior, risk, resource, and goal-oriented factors) when making their own decisions. Through experience as humans, we have mastered this process over our lifetime, and we seamlessly adapt our behavior. To date, making such predictions autonomously has eluded the machine learning and autonomous driving community. However, recent developments in areas such as IRL have the potential to address this limitation.

Behavior modeling in autonomous driving: A review

Model-based learning and supervised learning

The main modules in a generalized autonomous driving framework can be broadly categorized as sensor fusion, localization, prediction, and motion control. The sensory inputs are captured and fused to localize and predict the future trajectory of the agents in the local neighborhood. Utilizing these predictions, the future trajectory of an autonomous vehicle is generated and subsequently passed to the motion control subsystem to generate the control commands. The prediction module in an autonomous car uses behavior-modeling techniques, and these algorithms can be broadly categorized into model- and learning-based approaches.

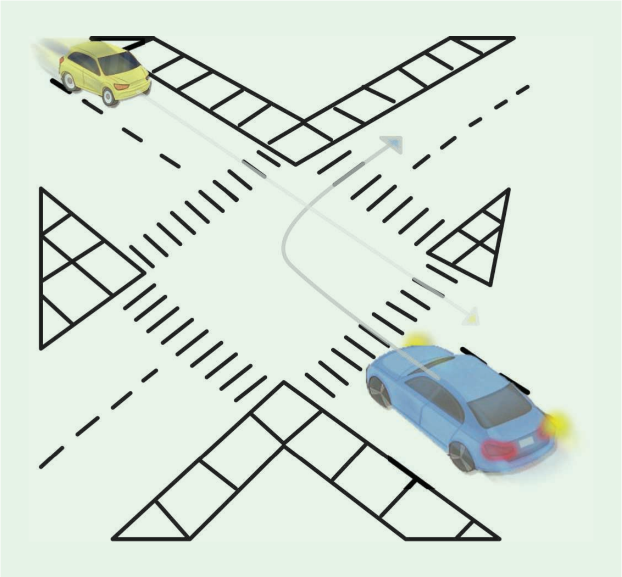


FIGURE 1. A sample driving scenario. The blue car is waiting to turn right at an intersection while there is a car coming from the opposite direction. The driver in the blue car should anticipate the future behavior of the yellow vehicle before determining its next action.

In model-based approaches, the factors that inform human behavior are hand engineered and combined to optimize a predefined 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, the underlying factors that influence human sociological factors are recovered from the data.

Among model-based approaches, Li et al. [4] uses a trajectory-planning scheme that samples trajectories from the global reference path leading to the goal state. A velocity profile generates the speed 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 course by minimizing the distance to the goal state, the distance to the centerline path, and maximizing the proximity to the obstacles. This was extended in [6] by augmenting the proximity cost to discourage picking paths that are close to dangerous drivers, cyclists, and pedestrians. Despite these attempts, it is infeasible to hand engineer a cost function that can consider all of the factors that influence the future behavior 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 algorithms is supervised learning. These algorithms apply the past observed trajectories of the autonomous agent(s) in the local neighborhood and learn to predict the future trajectory of the autonomous agent. The process is data driven, as the model minimizes the distance between the predicted and ground-truth trajectories using a predefined loss function, such as the mean square error (MSE) [7].

In [8], the authors propose the utilization of social pooling to capture interdependencies between neighboring vehicles in motion. The authors encode the past trajectories of the autonomous vehicle as well as neighboring agents using long short-term memory (LSTM) networks [9], and to capture the interdependencies of nearby agents they pool out hidden states of LSTM 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 minimize 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 maneuver classification. They illustrate that maneuver-dependent trajectory prediction is comparatively more resilient than predicting the trajectory alone.

Most recently, Zhao et al. [1] proposed a framework that encodes the past trajectories of neighboring agents using LSTMs and captures the scene context using convolutional neural networks (CNNs). 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 and adversarial loss, which is achieved through a generative adversarial network (GAN) learning process [11].

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

Generative adversarial imitation learning

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

One of the most popular approaches used for solving IRL problems is maximum entropy (MaxEnt)-IRL, where the expert behavior is modeled as a distribution to the one of the highest entropy.

Let the decision-making process of the pedestrians be modeled as a Markov decision process (MDP) [19]. The MDP $M = [S, A, \tau, R]$ is composed of state space S: a set of possible actions; A: a transition matrix; τ: a reward function; and 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, $\zeta^i = [s_0, s_1, \ldots, s_{T_{\text{obs}}}]$. 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 parameterized by θ, which directly generates the policy imitating π_E, and D_w is the discriminator network parameterized 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 the state–action pairs generated by policy π.

Numerous works [16]–[18] have utilized GAIL for predicting trajectories in simulated highway driving scenarios. In [17], the authors use eight features including vehicle speed, length, lane curvature, and distance-to-lane markers as state features and, employing the GAIL formulation, they predicted the relevant actions given this state representation. The authors in [16] propose a system to leverage variability among different expert demonstrations. They apply the information-maximization theorem to automatically discover and disentangle latent factors in the underlying expert demonstrations. In our previous work [18], we propose the use of neural memory networks (NMNs) [20] to capture relationships at a subtask level and determine how they are temporally linked in a given expert demonstration.

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

IRL

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

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

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

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

Capitalizing on the merits of IRL, several works [25]–[27] have applied it for behavior prediction. In [25], the authors first cluster the trajectories in the training set and train a multiclass classifier to label the cluster identity of a given trajectory. The authors utilize hidden Markov models 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 its future trajectory.

In [26], the authors investigate the tradeoff 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 centers, they apply the MaxEnt algorithm to learn different acceptability-dependent behaviors.

The authors in [27] address the exploding state-space problem in IRL. They propose to replace the RL inner loop in IRL with deep Q-networks to extend the IRL framework to larger state spaces.

Despite these capabilities, the original 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 modeled [23].

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 nonlinear mapping, which allows more flexibility for the learned reward structure. Hence,

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

where f is a nonlinear function. The authors of [14] try to maximize 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 from the demonstrated and inferred reward functions, respectively. Algorithm 1 illustrates the process of refining the reward network in the MaxEnt-deep-IRL (MED-IRL) framework proposed in [14], where γ is a discount factor for the value-iteration algorithm (see Algorithm 2), and α is the learning rate of the deep NN (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 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 Algorithm 2. Within Algorithm 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 presented in Figure 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 the lidar scans of urban environments to traversability maps, which are automatically learned through MED-IRL. The proposed multiscale fully convolutional network (MSFCN) architecture (see Figure 3) employs a pooling-based substream to capture spatial invariant features from lidar and a fully convolutional network (FCN) stream, which preserves the location information from the input data. The proposed system was able to learn end-to-end mapping from raw inputs to a reward map, utilizing more than 25,000 trajectories from more than 120 km 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 used in [23] is shown in Figure 4. The authors in [23] argue that, in motion planning, 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 the MED-IRL framework of [14] to incorporate this information in two stages. In the first stage, they apply a four-layer FCN to encode a color-coded point cloud. In the second stage, the authors utilize two feature maps encoding each grid cell, that is, the x and y positions of the grid cell in a vehicle centered, 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 nonlinear (MED-IRL) settings to maximize the entropy of the joint distribution over short data pieces. They show that long demonstrations

Algorithm 1. The MED-IRL.

```
Input:
D: Demonstrations; S: state space; A: set of possible actions;
  $\tau$: transition matrix; $\gamma$: discount factor for the value-iteration algorithm;
  and $\alpha$: learning rate of the deep neural network.
Output: Reward network parameters $\theta^*$
1: for iteration $i = 1$ to $M$ do
2:   $R^i(\Phi(s)) = f(\theta^i, \Phi(s))\ \forall_{s\in S}$   // forward pass in the
                                                               reward network
3:   $\pi^i = Value\_Iteration(R^i, S, A, \tau, \gamma)$   // planning step
4:   $\mathbb{E}[\mu^i] = compute\_SVF(\pi^i, S, A, \tau)$
5:   $\frac{\delta L_D^i}{\delta R^i} = \mu_D - \mathbb{E}[\mu^i]$   // gradient calculation
6:   $\theta^{i+1} = back\_propagate\left(\theta^i, \frac{\delta L_D^i}{\delta R^i}, \alpha\right)$   // reward network
                                                                  update
7: end
8: return $\theta$.
```

Algorithm 2. The value iteration.

```
Input:
R: Current approximation of the reward function; S: state space; A: set
  of possible actions; $\tau$: transition matrix; and $\gamma$: discount factor.
Output: $\phi$
$V(s) = -\infty$ 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 $\phi(a|s) = e^{Q(s,a) - V(s)}$.
```

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 that the aforementioned systems do not account for the motion of the neighbors 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 neighboring trajectories are encoded

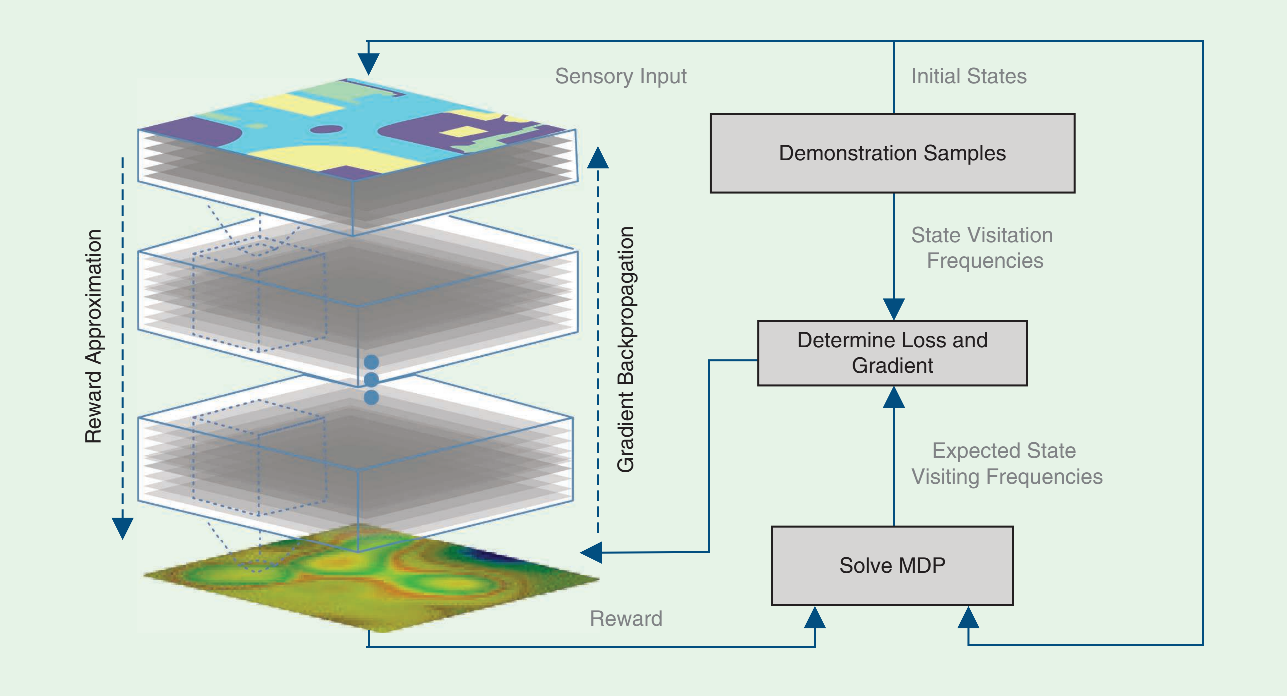


FIGURE 2. The schema proposed in [14] for training DNNs using MaxEnt-IRL. Given a set of demonstrations, a DNN is utilized to approximate the reward function. Then we calculate the difference between the state visitation frequencies from the demonstrated trajectories and from the inferred reward function.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

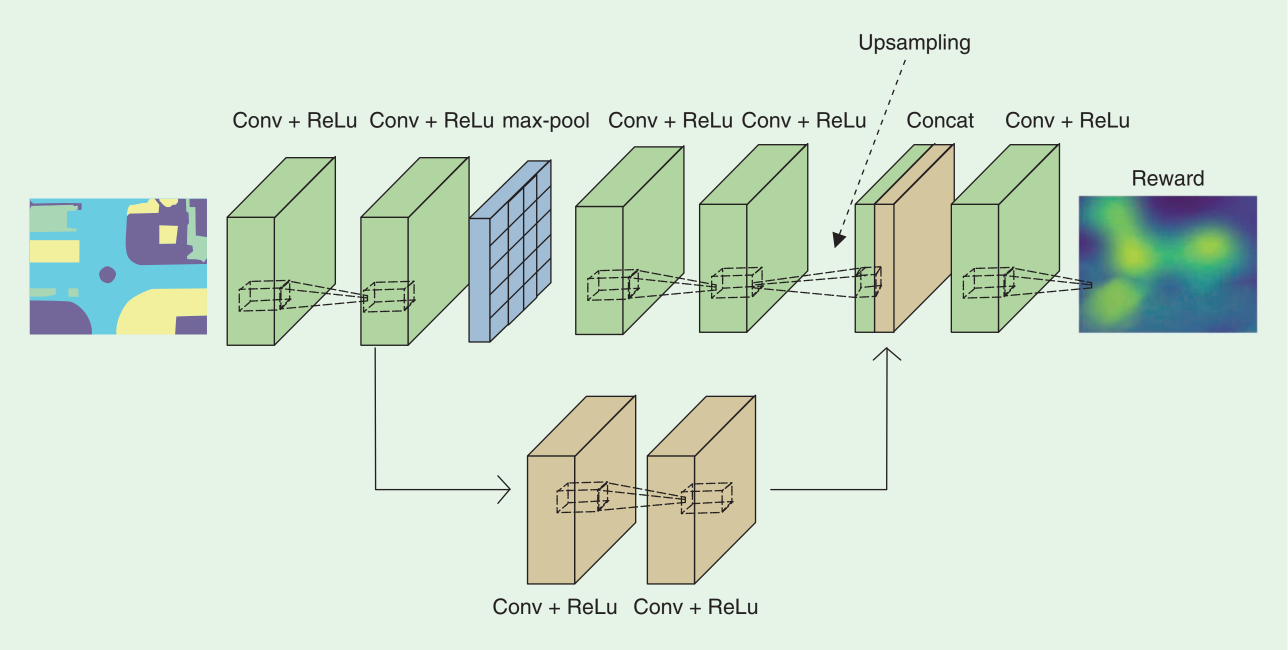


FIGURE 3. An illustration of the FCN architecture used by [14] as the reward network. By using a two-stream architecture, with an FCN-based mainstream and a pooling-based substream, we propose capturing spatially variant and invariant features, respectively. conv: convolutional; concat: concatenated; ReLu: rectified linear unit.

using LSTMs. Then we develop a combination of soft and hardwired attention [12] to aggregate the encoded trajectory information to a context vector.

As the reward network, similar to [14] and [24], we utilize an FCN. We first generate an empty map G of the environment and then assign values $\hat{h}_t^k$ from the pedestrian of interest k and $\tilde{h}_j^n$ from the neighbors to 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 Figure 5.

Experimental evaluations

In this section, we report the evaluation results of current state-of-the-art supervised learning, GAIL, linear-IRL (L-IRL), and D-IRL systems on the publicly available Next Generation Simulation (NGSIM) trajectories US-101 [31] data set and a portion of the nuScenes data set [32].

Data sets

The NGSIM US-101 [31] data set contains trajectories of real freeway traffic captured from fixed overhead cameras placed over a 640-m span of US-101, recorded at 10 Hz over a 45-min period. This data set consists of more than 6,000 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 data set [32], which is captured in multiple cities, from multiple sensors including six cameras, a lidar, five radars, a GPS sensor, and an inertial measurement unit sensor. The complete data set contains 15 h of driving data covering 242 km with dense traffic and highly challenging driving situations. The data set is divided into 1,000 scenes by the database authors, and to ensure a compatible size between the two evaluations, we use only scenes 61, 69, and 234. To generate the trajectories, we used object-bounding box annotations, and the center of the bounding box is taken to be the object position at each time step.

We report the results in terms of the root-MSE (RMSE) of the predicted trajectory with respect to the ground-truth future trajectory over different prediction horizons ranging from 1 to 5 s. Similar to [17], we simulate the behavior prediction through a trajectory-prediction task where we select each car, iteratively, to be the autonomous car and predict the future behavior of this car, exploiting the past behavior of neighboring vehicles.

Evaluated models

The following models were assessed:

- *Supervised learning*: we use the models of [8] [convolutional social (CS)-LSTM] and [1] [Multi-Agent Tensor Fusion (MATF)-GAN].
- *GAIL-gated recurrent unit (GRU)*: we consider the GAIL model from [17].
- *L-IRL*: we use the L-IRL model proposed in [22].
- *D-IRL*: to demonstrate the utility of D-IRL models, we use the models proposed in [14] (D-IRL), [23] [deep kinematics (DK)-IRL], and [30] [deep neighborhood (DN)-IRL].

For all of the considered systems, similar to [1], the neighbors appearing in the 640-m span are studied in the reasoning and prediction process. In the original works of [14] and [23], the authors utilize 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 traversable 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 (https://github.com/sisl/gail-driver).

For the D-IRL and L-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

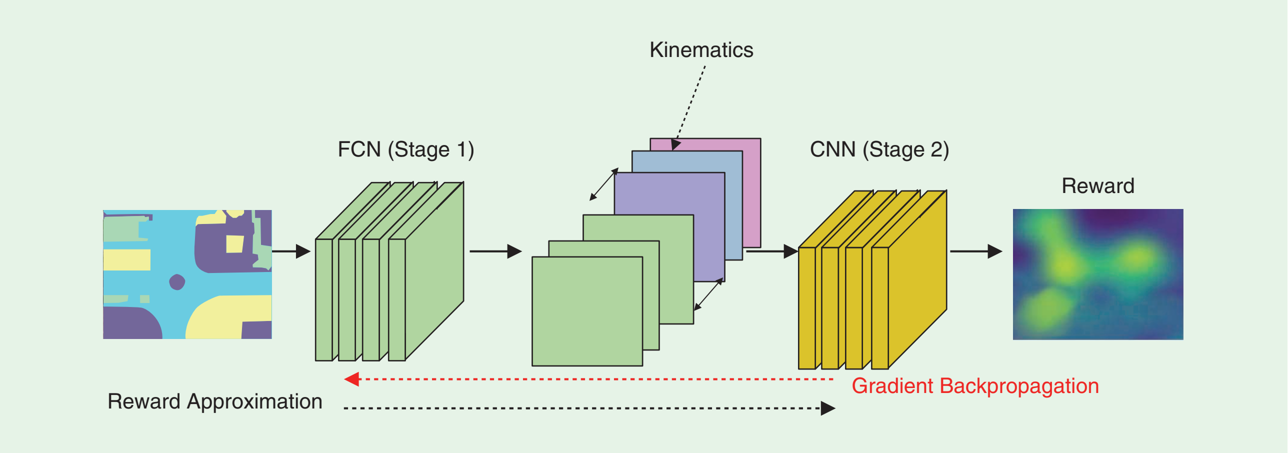


FIGURE 4. The two-state architecture proposed in [23]. We captured environmental context from input terrain maps in the first-stage network, and the resulting feature maps are concatenated with the kinematic context in the second-stage network, which outputs a reward representation. The difference between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

sample 1,000 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. 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] and 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, that it, the x and y positions of the grid cell in a pedestrian centered, 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 (https://github.com/yfzhang/vehicle-motion-forecasting), which also provided an implementation of the D-IRL framework in [14].

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

Results

Quantitative evaluations of the performance of the considered frameworks are presented in Table 1. To clearly demonstrate the utility of the D-IRL framework, we perform the trajectory predictions under different prediction horizons, estimating the trajectories from 1- to 5-s ahead. For each trajectory, we use the coordinates (positions) for the previous 3 s as the observed portion of the trajectory. We evaluated the performance for different prediction horizons, predicting trajectories from 1 s ahead to 5 s ahead. We report the RMSE as the error metric (lower is better). For clarity, supervised learning methods are shown with a blue background, the GAIL-GRU method with an orange background, the L-IRL method with a yellow background, and the D-IRL methods with a green background.

From Table 1, we observe that the performance of the supervised learning methods degrades when predicting behavior 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 intent of the driver. Furthermore, the linear IRL

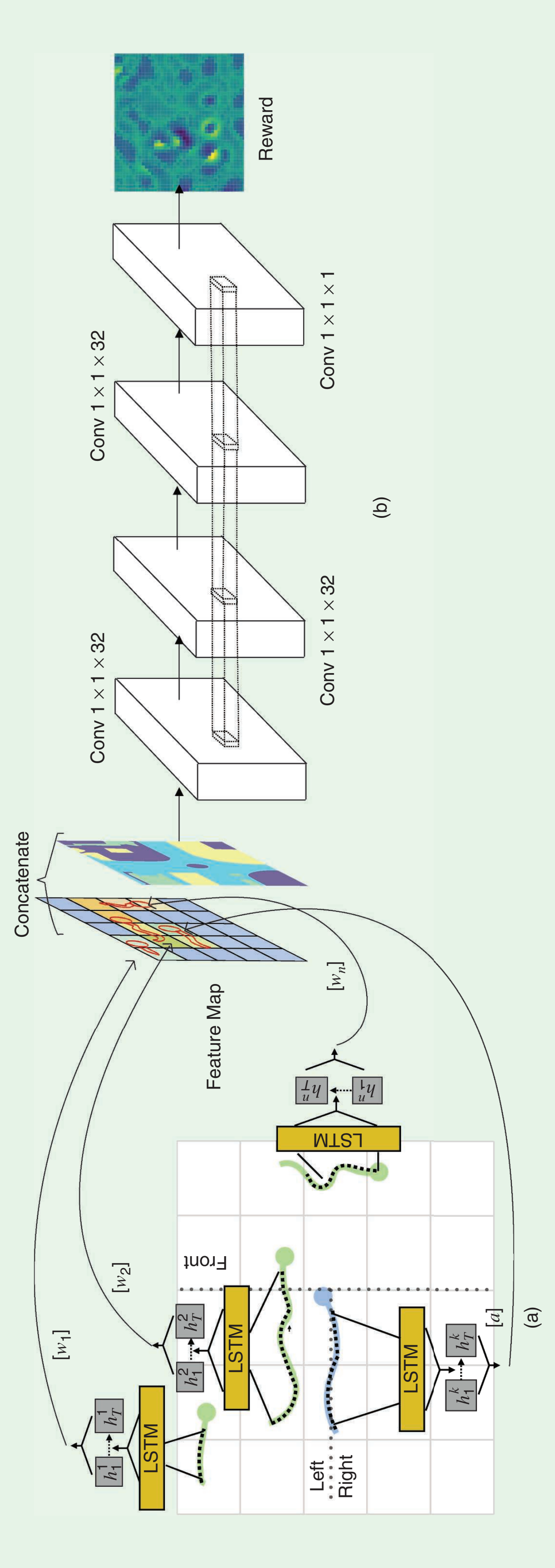


FIGURE 5. (a) The architecture used to embed neighborhood context: the trajectory of the pedestrian of interest is shown in blue, with three neighbors in green. The heading directions are indicated with circles. The trajectories are encoded using LSTMs, with soft attention used to embed information from the pedestrian of interest and hardwired attention used for the neighbors. Next, a feature map is created to spatially embed this information based on the cartesian points of each trajectory. (b) 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 an ReLu activation, and the final layer contains one 1 × 1 convolution kernel.

system fails to generate satisfactory results due to constraints with the learned feature-to-reward mapping structure.

With the introduction of the nonlinear reward mapping from the D-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 neighborhood context, which is a highly influential factor when navigating in congested environments. However, the LSTM-based neighborhood embedding scheme in the DN-IRL framework is able to capture a notion of the neighborhood, resulting in superior performance. We observe a substantial performance increase, especially when predicting behavior 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 their relative strengths and weaknesses. However, such comparisons are currently constrained by the nonpublic availability of such advanced GAIL architectures.

Qualitative results of the DN-IRL method and the recovered reward representation for four examples from the NGSIM data set [31] are given in Figure 6. By analyzing the predictions in Figure 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 its neighborhood modeling, is capable of predicting complex maneuvers such as lane changes and overtaking.

Limitations and open research challenges

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

Table 1. The evaluation results for NGSIM US-101 [31] and nuScenes [32].

Results for the NGSIM US-101 Data Set

	Prediction Horizon				
Method	1 s	2 s	3 s	4 s	5 s
CS-LSTM [8]	0.61	1.27	2.09	3.1	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	4.4
DN-IRL [30]	0.54	1.02	1.91	2.43	3.76

Results for the nuScenes Data Sets

	Prediction Horizon				
Method	1 s	2 s	3 s	4 s	5 s
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

To the best of our knowledge, the only D-IRL framework that 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, for which [30] is originally proposed. In addition, the type of neighbor, i.e., a car, truck, motorbike, or pedestrian, may also be important, yet it is not considered. Hence, further investigation is required to determine effective ways to learn the spatiotemporal contextual factors that impact an agent's behavior 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 behavior due to varied user preferences and domain knowledge, even though all experts perform the same task. 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 previous work, we investigate using NMNs 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] investigates applying an adversarial IRL (A-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 the 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 suboptimality in the given samples when the demonstrators do not follow optimal behavior.

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 utilized 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 the best ways to speed up the convergence of value iteration.

In addition, little effort has been made to leverage the multimodal data captured by autonomous vehicles. In [14] and [23],

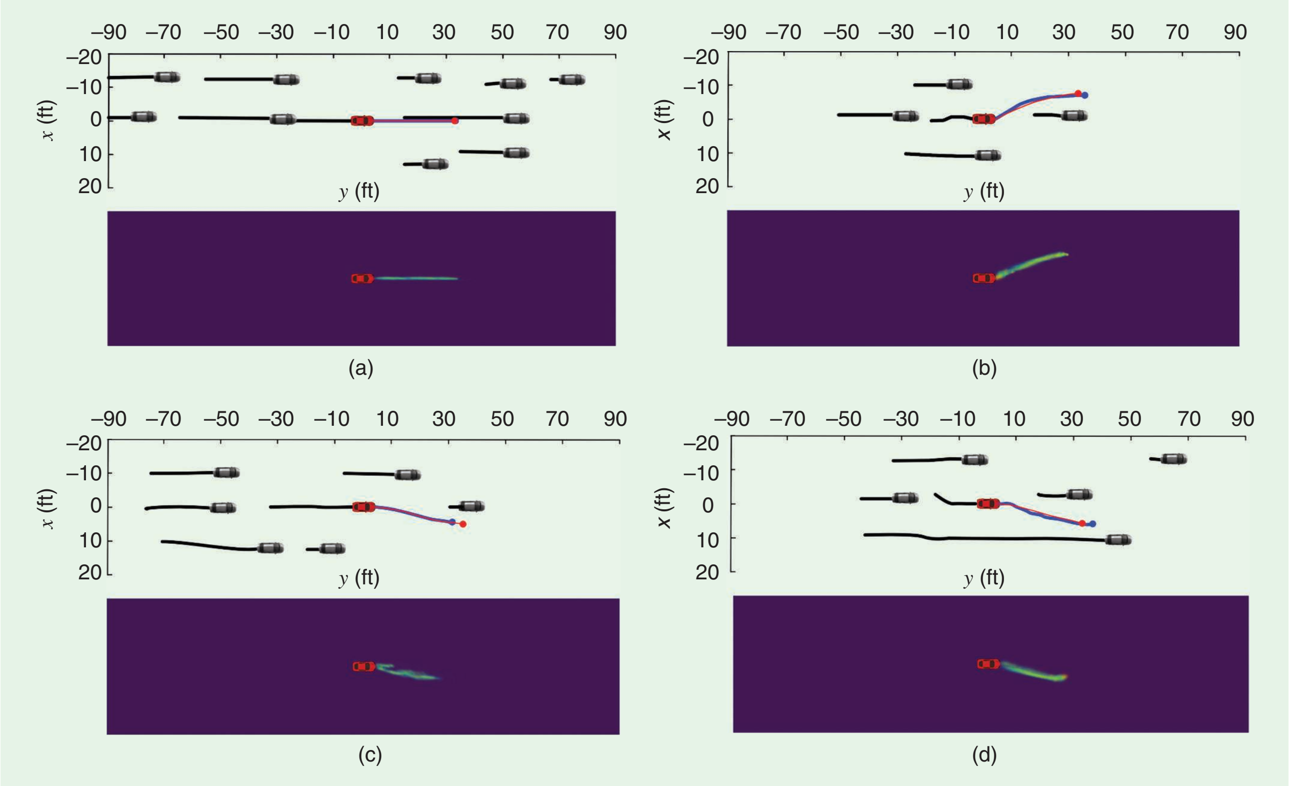


FIGURE 6. The qualitative results of the DN-IRL method: (a) a lane following behavior is shown in the ground truth and a higher probability in the prediction is given for this behavior; (b)–(d) predictions for scenarios where an overtaking behavior exists in the ground truth. The observed part of the trajectories is shown in black. The autonomous agent is indicated by the red car, and the neighboring 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 colors from blue to yellow indicate low to high probability.

the terrain maps are captured using lidar scans; however, systems can be designed to exploit the complementary information available through sources such as red-green-blue, 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 neighborhood modeling for decision making.

The lack of availability of well-annotated public benchmarks poses another hinderance. Only a limited number of data sets, such as nuScenes [32] and KITTI [36], have annotations 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 behavioral prediction systems for real-world autonomous driving systems.

Conclusions

In this article, we presented an overview of the current state-of-the-art techniques applied for behavior prediction in autonomous driving. We reviewed popular approaches, including both model-based and supervised learning and GAIL, IRL, and D-IRL methods. We quantitatively and qualitatively evaluated these frameworks on two public driving benchmark data sets and demonstrated the utility of D-IRL, especially when making predictions into the distant future. Despite the undoubted potential of D-IRL methods, there are several shortcomings at present, and a number of promising research avenues for future breakthroughs were discussed to further advance the field and realize the goal of fully autonomous vehicles.

Authors

Tharindu Fernando (t.warnakulasuriya@qut.edu.au) received his B.Sc. (special degree in computer science) and Ph.D. degrees from the University of Peradeniya, Sri Lanka, and Queensland University of Technology (QUT), Brisbane, Australia, respectively. He is currently a postdoctoral research fellow in the Speech, Audio, Image, and Video Technologies research program with the School of Electrical Engineering and Computer Science at QUT, Brisbane, Australia. His research interests focus mainly on human behavior analysis and prediction. He is a Member of IEEE.

Simon Denman (s.denman@qut.edu.au) received his B.Eng. degree in electrical engineering and his Ph.D. degree in the area of object tracking from Queensland University of Technology (QUT), Brisbane, Australia. He is currently a senior lecturer with the School of Electrical Engineering and Computer Science at QUT, Brisbane, Australia. His research interests include intelligent surveillance, video analytics, and video-based recognition. He is a Member of IEEE.

Sridha Sridharan (s.sridharan@qut.edu.au) received his B.Sc. degree in electrical engineering and his M.Sc. degree in

communications engineering, both from the University of Manchester, United Kingdom, and his Ph.D. degree from the University of New South Wales, Kensington, Australia. He is currently with Queensland University of Technology (QUT), Brisbane, Australia, where he is a professor with the School Electrical Engineering and Computer Science. He has authored more than 600 publications in the areas of image and speech technologies. He leads the Speech, Audio, Image, and Video Technologies research program at QUT, with a strong focus in the areas of computer vision, pattern recognition, and machine learning. He is a Life Member of IEEE.

Clinton Fookes (c.fookes@qut.edu.au) received his B.Eng. degree in aerospace/avionics and his M.B.A. and Ph.D. degrees from Queensland University of Technology (QUT), Brisbane, Australia. He is currently a professor and the head of discipline for vision and signal processing with the Science and Engineering Faculty at QUT, Brisbane, Australia. He serves on the editorial board of *IEEE Transactions on Information Forensics and Security*. His research focus is in the areas of computer vision, machine learning, and pattern recognition. He is a Senior Member of 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 *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019, pp. 12,126–12,134. doi: 10.1109/CVPR.2019.01240.

[2] S. Anthony, "Self-driving cars still can't mimic the most natural human behavior," Quartz, 2017. [Online]. Available: https://qz.com/1064004/self-driving-cars-still-cant-mimic-the-most-natural-human-behavior/

[3] "Introducing perceptive automata: Human intuition for self-driving cars," Medium, 2018. [Online]. Available: https://medium.com/perceptive-automata/introducing-perceptive-automata-human-intuition-for-self-driving-cars-3d2aaa05c083

[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," *Mech. Syst. Signal Process.*, vol. 87, pp. 118–137, Mar. 2017. doi: 10.1016/j.ymssp.2015.10.021.

[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 *Proc. 2017 IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 225–230. doi: 10.1109/ICRA.2017.7989028.

[6] A. B. D. Manocha, "Behavior modeling for autonomous driving," in *Proc. AAAI Fall Symp. Reasoning and Learning Real-World Systems Long-Term Autonomy (LTA)*, 2018, pp. 16–21.

[7] E. L. Lehmann and G. Casella, *Theory of Point Estimation*. New York: Springer-Verlag, 2006.

[8] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1468–1476. doi: 10.1109/CVPRW.2018.00196.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

[10] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based LSTMs," in *Proc. 2018 IEEE Intelligent Vehicles Symp. (IV)*, pp. 1179–1184. doi: 10.1109/IVS.2018.8500493.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Int. Conf. Advances Neural Information Processing Systems*, 2014, pp. 2672–2680. doi: 10.5555/2969033.2969125.

[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 Netw.*, vol. 108, pp. 466–478, Dec. 2018. doi: 10.1016/j.neunet.2018.09.002.

[13] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018, pp. 2255–2264. doi: 10.1109/CVPR.2018.00240.

[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," *Int. J. Robot. Res.*, vol. 36, no. 10, pp. 1073–1087, 2017. doi: 10.1177/0278364917722396.

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

[16] Y. Li, J. Song, and S. Ermon, "InfoGAIL: Interpretable imitation learning from visual demonstrations," in *Proc. Advances 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 *Proc. 2017 IEEE Intelligent Vehicles Symp. (IV)*, pp. 204–211. doi: 10.1109/IVS.2017.7995721.

[18] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Learning temporal strategic relationships using generative adversarial imitation learning," in *Proc. 17th Int. Conf. Autonomous Agents and MultiAgent Systems*. 2018, pp. 113–121. doi: 10.5555/3237383.3237407.

[19] R. Bellman, "A Markovian decision process," *J. Math. Mech.*, vol. 6, no. 4, pp. 679–684, 1957. doi: 10.1512/iumj.1957.6.56038.

[20] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Pedestrian trajectory prediction with structured memory hierarchies," in *Proc. Joint European Conf. Machine Learning and Knowledge Discovery in Databases*. New York: Springer-Verlag, 2018, pp. 241–256. doi: 10.1007/978-3-030-10925-7_15.

[21] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," in *Proc. Int. Conf. Learning Representation, (ICLR)*, 2018, pp. 1–15.

[22] K. Saleh, M. Hossny, and S. Nahavandi, "Long-term recurrent predictive model for intent prediction of pedestrians via inverse reinforcement learning," in *Proc. 2018 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8. doi: 10.1109/DICTA.2018.8615854.

[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," in *Proc. Conf. Robot Learning (CoRL)*, 2018, pp. 1–12.

[24] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. Conf. AAAI*, Chicago, IL, 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 *Proc. 22nd European Conf. Artificial Intelligence*. Amsterdam, The Netherlands: 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 *Proc. 2015 IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 3215–3222. doi: 10.1109/ICRA.2015.7139642.

[27] S. Sharifzadeh, I. Chiotellis, R. Triebel, and D. Cremers, "Learning to drive using inverse reinforcement learning and deep Q-networks," in *Proc. NIPS workshop Deep Learning for Action and Interaction*, 2016, pp. 1–7.

[28] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, 1967. doi: 10.1109/TIT.1967.1054010.

[29] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," *Robot. Auton. Syst.*, vol. 114, pp. 1–18, Apr. 2019. doi: 10.1016/j.robot.2019.01.003.

[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," in *Proc. IEEE Int. Conf. Computer Vision Workshops*, 2019, vol. 108, pp. 466–478. doi: 10.1109/ICCVW.2019.00149.

[31] J. Colyar and J. Halkias, "US highway 101 dataset," Federal Highway Administration (FHWA), Washington, D.C., 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 et al., nuScenes: A multimodal dataset for autonomous driving. 2019. [Online]. Available: arXiv:1903.11027

[33] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Going deeper: Autonomous steering with neural memory networks," in *Proc. IEEE Int. Conf. Computer Vision Workshops*, 2017, pp. 214–221. doi: 10.1109/ICCVW.2017.34.

[34] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *Proc. Int. Conf. Machine Learning*, 2016, pp. 49–58. doi: 10.5555/3045390.3045397.

[35] D. Wingate, "Solving large MDPS quickly with partitioned value iteration," Ph.D. dissertation, Brigham Young Univ., Provo, UT, 2004.

[36] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. 2012 IEEE Conf. Computer Vision and Pattern Recognition*, pp. 3354–3361. doi: 10.1109/CVPR.2012.6248074.