Neighbourhood Context Embeddings in Deep Inverse Reinforcement Learning for Predicting Pedestrian Motion Over Long Time Horizons

Tharindu Fernando Simon Denman Sridha Sridharan Clinton Fookes
Image and Video Research Lab, SAIVT, Queensland University of Technology (QUT), Australia
{t.warnakulasuriya, s.denman, s.sridharan, c.fookes}@qut.edu.au

Abstract

Predicting crowd behaviour in the distant future has increased in prominence among the computer vision community as it provides intelligence and flexibility for autonomous systems, enabling the early detection of abnormal events and better and more natural interactions between humans and autonomous systems such as driverless vehicles and field robots. Despite the fact that Deep Inverse Reinforcement Learning (D-IRL) based modelling paradigms offer flexibility and robustness when anticipating human behaviour across long time horizons, compared to their supervised learning counterparts, no existing state-of-the-art D-IRL methods consider path planning in situations where there are multiple moving pedestrians in the environment. To address this, we present a novel recurrent neural network based method for embedding pedestrian dynamics in a D-IRL setting, where there are multiple moving agents. We propose to capture the motion of the pedestrian of interest as well as the motion of other pedestrians in the neighbourhood through Long-Short-Term Memory networks. The neighbourhood dynamics are encoded into a feature map, preserving the spatial integrity of the observed trajectories. Utilising the maximum-entropy based non-linear inverse reinforcement learning framework, we map these features to a reward map. We perform extensive evaluations on the publicly available Stanford Drone and SAIVT Multi-Spectral Trajectory datasets where the proposed method exhibits robustness towards lengthier predictions into the distant future, demonstrating the importance of capturing the dynamic evolution of the environment using the proposed embedding scheme.

1. Introduction

In an era of automaticity, understanding and predicting crowd behaviour is critical for the accreditation of safe and effective autonomous systems. The applications vary from autonomous driving to security surveillance where the ability to understand and predict human behaviour could generate a positive impact on the safety of the system in question. This paper proposes a Deep Inverse Reinforcement Learning (D-IRL) based framework for predicting pedestrian dynamics over long time horizons.

The most common approach for predicting pedestrian behaviour is through supervised learning where function approximators, like neural networks operate directly over the input trajectory and use a pre-defined cost function, to try and mimic human behaviour [1, 6, 7, 11].

There are numerous arguments for preferring Inverse Reinforcement Learning (IRL) over such direct optimisation methods. Firstly, the direct application of supervised learning is proven to ill represent the scene context and pedestrian dynamics of any given scene, making it potentially intractable to generalise the learned knowledge to a new environment [9, 10]. Secondly, IRL based path planning frameworks have demonstrated resilient predictions over lengthier time intervals [21, 23, 24]. This utility arises from the fact that IRL methods are reward seeking algorithms which uncover the end goal or intentions of the pedestrians from the demonstrations [10]. Hence they posses the ability to segregate dynamics from scene context, allowing the platform to better model pedestrian behaviour [21].

However, the original IRL framework in [26] assumes a linear mapping from the features to a reward. The recent works of Wulfmeier et. al [22, 23] extended this to a deep learning setting, lifting this constraint and permitting a non-linear mapping which allows more flexibility in the learned reward structure. While this provides greater flexibility and robustness for a behaviour anticipation task, none of the current state-of-the-art systems have investigated it's ability in a dynamic environment where there are other moving agents.

Thus, in this paper, we are proposing a framework that utilises the motion information from the pedestrian of interest as well as the information from other moving agents, and effectively embeds this information in the learned reward representation. We demonstrate how recurrent neural networks and an attention framework can be coupled with the

D-IRL process to provide resilient long-term predictions.

The main contributions of this work can be summarised as follows:

- We extend the D-IRL framework to a dynamic environment where there are multiple pedestrians in motion.
- We incorporate recurrent neural networks to model the neighbourhood dynamics in the D-IRL framework, and augment the reward function learning process.
- We demonstrate how the hidden state representation of the recurrent network can be used as the input feature map for D-IRL, preserving the structural relationships of the neighbourhood.
- We perform extensive evaluations on multiple public benchmarks where the proposed method outperforms state-of-the-art methods.

2. Preliminaries

We model the decision making process of the pedestrians as a Markov Decision Process (MDP) [2]. 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. The goal of the learning algorithm is to find the optimal policy, π^*, that maximises the expected sum of rewards for the agent.

In the IRL setting, we assume that the reward function is unknown. Instead we are presented with a set of demonstrations, $D = [\zeta^1, \zeta^2, \ldots, \zeta^N]$, where we have examples of agents behaving in the environment. Note that each trajectory, $\zeta^i = [s_0, s_1, \ldots, s_{T_{obs}}]$. In a supervised learning setting we are directly mapping the observed states to the future states, $[s_{T_{obs}+1}, s_{T_{obs}+2}, \ldots, s_{T_{pred}}]$. With an IRL framework we first recover the reward function, R. One of the most popular approaches for solving IRL problems is Maximum Entropy (MaxEnt) IRL [26] where the expert behaviour is modelled as a distribution to the one of highest entropy [23]. 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,

$$R(\Phi(s)) = [\theta]^\top \Phi(s). \tag{1}$$

The works of [22, 23] extended this to a non-linear setting where,

$$R(\Phi(s)) = f(\theta, \Phi(s)), \tag{2}$$

where f is a non-linear function. The authors of [23] try to maximise the log likelihood of the demonstrated trajectories,

$$L(\theta) = log \prod_{\zeta^i \in D} P(\zeta^i, \theta), \tag{3}$$

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

$$\frac{\delta L_D}{\delta \theta} = \mu_D - \mathbb{E}[\mu] \frac{\delta R(\Phi(s))}{\delta \theta}, \tag{4}$$

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 [23], 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. 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 [26] to solve the forward Reinforcement Learning (RL) problem, which determines 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).

Algorithm 1: Maximum Entropy Deep IRL

Input: D, S, A, τ, γ, α
Output: Reward network parameters θ^*

1 **for** iteration i = *1* to M **do**
2 $\quad R^i(\Phi(s)) = f(\theta^i, \Phi(s)) \ \ \forall_{s \in S}$; // Forward pass in the reward network
3 $\quad \pi^i = Value\_Iteration(R^i, S, A, \tau, \gamma)$ // Planning step
4 $\quad \mathbb{E}[\mu^i] = compute\_SVF(\pi^i, S, A, \tau)$
5 $\quad \frac{\delta L_D^i}{\delta R^i} = \mu_D - \mathbb{E}[\mu^i]$ // Gradient calculation
6 $\quad \theta^{i+1} = back\_propagate(\theta^i, \frac{\delta L_D^i}{\delta R^i}, \alpha)$ // Reward network update
7 **end**
8 **return** θ

3. Related Works

There exist multiple ways to address the task of predicting an agent's future motion. Among them, one of the most popular approach is to use supervised learning. Alahi et. al [1] proposed a Social LSTM model where recurrent neural networks are utilised to encode the trajectory information from the neighbourhood of the pedestrian of interest. They pooled out the last hidden state of each LSTM [14] as

Algorithm 2: Value Iteration

Input: R, S, A, τ, γ
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 $\phi(a|s) = e^{Q(s,a)-V(s)}$

the representation of each trajectory. Prior work in [7] has extended this framework with a soft and hard-wired attention combination to use all the hidden states of the LSTMs corresponding to the pedestrians of interest as well as their neighbours. The works of [4, 11, 19, 27] have also considered the supervised learning of human trajectory patterns within a GAN learning framework, while [6, 8] incorporated neural memory architectures to capture long-term dependencies with respect to the environment in the prediction framework. However, it is a well established fact that such a direct mapping between the observed trajectory and the target lacks generalisation [22–24]. Most importantly these models doesn't possess the capacity to understand the underlying intention or the end goal that influences human behaviour, making accurate long-term predictions with supervised learning models infeasible [10, 21, 22, 24].

On the other hand IRL [26] based path planning segregates the underlying semantics of the scene such that the goal or the intentions of the agents can be recovered based on the modelled reward function. This allows more resilient predictions into the distant future as well as transferring the learnt knowledge into new environments. Numerous previous works [12, 15, 21] have capitalised on these benefits and extensively utilised IRL for trajectory prediction. However, the original IRL framework proposed in [26] assumes a linear mapping between the features and the reward representation which severely limits the degrees of freedom of the learned reward function.

The recent work of Wulfmeier et. al [22] removed this constraint with their proposed deep, non-linear IRL (D-IRL) based framework. Several attempts [23, 24] have been made to utilise this framework in real world applications, however none of the existing methods have considered crowded environments where there are multiple pedestrians in motion. Thus, this work propose a D-IRL framework which effectively embeds the spatio-temporal features of the neighbourhood of the pedestrian of interest using a combination of recurrent neural networks and an attention mechanism.

It should be noted that there exists a separate line of work, Generative Adversarial Imitation Learning (GAIL)

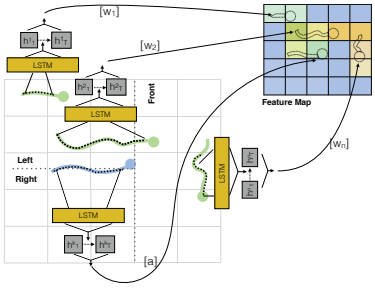


Figure 1: The architecture used to embed the neighbourhood context: The trajectory of the pedestrian of interest is shown in blue, with three neighbours shown in green. Heading directions are indicated with circles. We encode the trajectories using LSTMs where soft attention is utilised to embed the information from the pedestrian of interest and the neighbours use hard-wired attention. Next a feature map is generated to embed this information spatially, based on the cartesian points of each trajectory.

[5, 13, 16], which attempts to directly mimic the expert's policy. However, these methods suffer the same deficiencies experienced in the supervised learning setting as they also do not attempt to recover the reward function of the expert. Instead they attempt to directly mimic the expert's behaviour [10].

4. Architecture

The aim of this section is to illustrate the framework used to embed the information from the pedestrian of interest's trajectory and their neighbours (Sec. 4.1), and to describe the architecture used to map this embedded information to a reward map (Sec. 4.2).

4.1. Embedding Neighbourhood Context

Motivated by the recent success of the neighbourhood context modelling approach presented in [6, 7], we utilise a combination of soft attention and hard-wired attention to embed features from the local neighbourhood of the agent.

The approach utilised for embedding the neighbourhood context is visually illustrated in Fig. 1.

Let p^k be a vector containing the set of states of the k^{th} pedestrian trajectory, τ^k, from time instant 0 to T_{obs},

$$p^k = [s_o^k, \ldots, s_{T_{obs}}^k], \tag{5}$$

where states are composed of points in a Cartesian grid. Then each vector p^k is passed through an LSTM encoder,

$$h_t^k = \text{LSTM}(p_t^k, h_{t-1}^k). \tag{6}$$

Motivated by [7] we use soft attention to embed the features from the pedestrian of interest such that,

$$\hat{h}_t^k = \beta_{t,j} h_j^k \quad \text{for} \quad j = [0, \ldots, T_{obs}], \tag{7}$$

where the hidden states, h_j^k, are weighted based on the weights, $\beta_{t,j}$, computed by,

$$\beta_{t,j} = \frac{\exp(e_{tj})}{\sum_{j=1}^{T_{obs}} \exp(e_{tl})}, \tag{8}$$

where

$$e_{tj} = a(h_{t-1}^k, h_j^k). \tag{9}$$

a is a feed forward neural network jointly trained with the other components of the reward network.

To encode the effect of neighbouring pedestrians we use the hard-wired attention framework proposed in [7] due to its simplicity and effectiveness. The hard-wired weight, w, is computed by,

$$w_j^n = \frac{1}{\text{dist}(n,j)}, \tag{10}$$

where $\text{dist}(n,j)$ is the Euclidian distance between the pedestrian of interest, k, and the n^{th} neighbour at the j^{th} time instant. Using hard-wired weights we generate the effect of each neighbour, n, such that,

$$\tilde{h}_j^n = w_j^n h_j^n \quad \forall_{j \in T_{obs}} \text{ and } \forall_{n \in N}, \tag{11}$$

where we assume there are N neighbours in the neighbourhood and the encoded hidden state of the n^{th} neighbour at the j^{th} time instant is given by h_j^n.

4.2. Reward Prediction

The works of [23, 24] have demonstrated the utility of maintaining the reward prediction network architecture as a fully convolutional network, allowing a direct mapping between the modelled environment and the predicted reward map. This ensures that the learned reward map covers all the areas of the environment, encapsulating structural factors such as buildings and pathways that influence pedestrian behaviour.

Hence we first generate an empty 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 a Fully Convolution Network (FCN) we map G to a reward map, R. The architecture of the FCN is illustrated in Fig. 2.

5. Experiments

In this section we present the details of the datasets that are utilised in our evaluations (Sec. 5.1), implementation details of the proposed method (Sec. 5.2), information regarding the evaluation metrics (Sec. 5.3), baselines (Sec. 5.4), and quantitative and qualitative evaluations (Sec. 5.5).

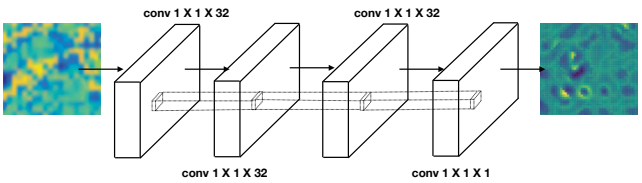


Figure 2: The architecture of the four layer fully convolution network used to map the feature map G to the reward map R. The first three layers contain 32, 1×1 convolution kernels with a ReLU activation, and the final layer contains 1, 1×1 convolution kernel.

5.1. Datasets

5.1.1 Stanford Drone (SD) Dataset [18]:

Following [11, 20, 21] we utilise the Stanford Drone (SD) dataset in our evaluation as it provides substantially lengthier trajectories compared to the frequently used Grand Central Station [25], and ETH-BIWI Walking Pedestrians [17] datasets. Furthermore, this dataset provides a challenging setting with high density crowds and different interactions among the groups [20]. This allows us to effectively evaluate the ability of the proposed D-IRL framework to anticipate human behaviour in to the distant future.

The SD dataset was collected using a downward facing camera mounted on a drone hovering above Stanford University. It was captured at 2.5fps and contains 11,216 annotated pedestrian trajectories. Similar to [21] we converted the provided bounding box annotations to x, y coordinates using the centre of the bounding box as the coordinate. Following [21], we consider four scenes from the dataset, in our evaluations: "Book Store", "Gates", "Death Circle" and "Coupa". For training and testing we used the splits provided by the dataset authors.

5.1.2 SAIVT Multi-Spectral Trajectory (MST) Dataset [6]:

To further demonstrate the proposed method we evaluate our model on a second dataset, the SAIVT Multi-Spectral Trajectory dataset [6]. This dataset is collected from synchronised CCTV and Radar feeds, captured at 5fps. From this dataset we used the trajectories from the Radar stream as it provides lengthier trajectories with average trajectory length of 1362.33 frames for the available 27,462 trajectories, compared to the CCTV stream where the average trajectory length is only 310.22 frames. After filtering out short and fragmented trajectories we are left with 20,800 trajectories. We randomly select 14,560 trajectories for training, 5,200 for testing and 1,040 for validation[1].

[1]These splits are available upon request

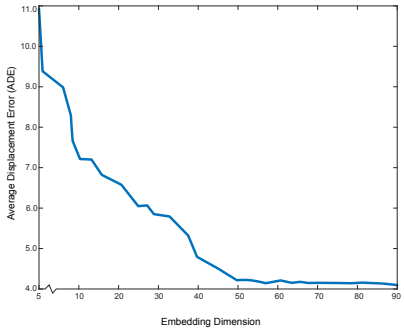


Figure 3: Hyperparameter Evaluation: Using the validation set of the SAIVT Multi-Spectral Trajectory (MST) Dataset [6]. We measure the change in Average Displacement Error (AED) with the embedding dimension of the encoder LSTMs. As the performance plateaus when the embedding dimension reaches 50, we set the embedding dimension to 50.

5.2. Implementation Details

We consider a grid size of 120 $\times$ 120 and first mapped the x, y coordinates to grid cells. Considering the neighbourhood encoding scheme, for all LSTMs we use a hidden state dimension of 50 units which we experimentally evaluated using the validation set of the MST dataset [6]. Fig. 3 illustrates the change in AED with respect to the embedding dimension of the LSTMs. We observe that the performance plateaus when the embedding dimension reaches 50.

When modelling the neighbourhood, following [7] we use the trajectories of the closest 10 neighbours in each direction, namely front, left and right. If there are more than 10 neighbours in any direction, we choose the closest 9 neighbours and the mean trajectory of the rest. In cases where there are less than 10 neighbours, we create a dummy trajectory such that we have 10 neighbours from each direction and set the hard-wired weight of that dummy trajectory to zero.

For the FCN, to enable direct comparison to other baselines, we limit the FCN architecture to four-layers. The first three layers contain 32, 1 $\times$ 1 convolution kernels followed by ReLU activation and the final layer contains 1, 1 $\times$ 1 convolution kernel.

5.3. Metrics

For quantitative evaluation of the performance, similar to [21, 23, 24] we utilise the Modified Hausdorff Distance (MHD) [3] and the Average Displacement Error (ADE) [6, 7].

MHD measures the geometric similarity between the ground truth and predicted trajectories. In order to measure the spatial similarity between the predicted and ground truth trajectories we measure the average Euclidian distance between the two.

(a) RGB Frame (b) Semantic Segmentation

Figure 4: A sample RGB frame from the Stanford Drone (SD) Dataset [18] and its semantic segmentation (grass is denoted by green, walkways by blue, trees by yellow and buildings by purple).

As the proposed framework generates a probability distribution over the grid cells, we sample 1000 trajectories from the distribution and measured the average MHD and ADE between the ground truth and the samples. We map the predictions back to the image coordinate space for clear comparisons with the baselines.

5.4. Baselines

Supervised Learning Baselines: We use the supervised models in [7] (SHA), CAR-Net [20], Social LSTM (S-LSTM) [1] and social GAN (S-GAN) [11].

Linear-IRL Baseline: We use the linear-IRL model (L-IRL) proposed in [21].

Deep-IRL Baselines: To demonstrate the utility of deep-IRL models we use the models proposed in [23] (D-IRL) and [24] (DK-IRL).

In the original works of [23] and [24] the authors utilise terrain maps captured using LIDAR. As this information is not available in our datasets, similar to [21] we use the semantic segmentation of the RGB frames which is generated manually. A sample RGB frame from the SD dataset and the segmentation map are shown in Fig. 4 (a) and (b), respectively.

For the D-IRL baseline we strictly adhere the recommendations of the authors and used the FCN architecture introduced in [23]. 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 in [24] and used the FCN model from D-IRL as the network for the first state. For the second stage, following the experimental setup of [24] 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 curvature of the input trajectory. For this implementation we use the codebase released by the au-

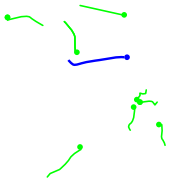


Figure 5: A Sample Neighbourhood Plot: The observed potion of the pedestrian of interest's trajectory is shown in blue, neighbouring trajectories are shown in green. Heading direction is indicated with a circle.

thors [2] which also provided an implementation of the D-IRL framework in [23]. For more details please refer to [24].

Additionally, in order to account for the motion of neighbouring pedestrians we used trajectory plots of the pedestrian of interest as well as the neighbours. A sample plot is shown in Fig. 5. Here the observed part of the trajectory is given in blue, and neighbours are indicated in green. The heading direction of each pedestrian is indicated with a circle. We extend the D-IRL and DK-IRL baseline models to use this neighbourhood information as follows. For the DN-IRL baseline model we concatenate the segmentation map input together with the neighbourhood plot and feed it to the four-layer FCN. In the DKN-IRL baseline we concatenate the segmentation map input with the neighbourhood plot and feed it to the first stage network.

5.5. Results

Quantitative evaluations of the performance of our proposed method and the baselines for the Stanford Drone (SD) dataset [18] and SAIVT Multi-Spectral Trajectory (MST) dataset [6] are presented in Tab. 1 and Tab. 2, respectively. In order to clearly demonstrate the utility of the proposed Deep IRL framework, we measure the trajectory predictions under two settings, predicting the trajectories for 5 seconds ahead and 12 seconds ahead. For each trajectory, we use the first 6 frames (i.e 2.4 seconds for SD dataset [18] and 1.2 seconds for MST dataset [6]) as the observed part of the trajectory and predicted the trajectory for the next 5 seconds or 12 seconds depending on the experiment.

From the results presented in Tab. 1 and Tab. 2 it is clearly evident that the proposed model has outperformed all the baselines, and achieved a lower ADE and MHD. Furthermore, we observe that the performance of the supervised learning methods significantly degrades when we predict into the distant future. We speculate that this is because those models are trying to directly map the inputs to the targets, without paying attention to the end goal or the intention of the pedestrian. When comparing the performance of the supervised learning methods and the linear-IRL model of [21], we observe better performance with the reward based learning framework of IRL. Though we expect to observe better performance with the introduction of the non-linear mapping with the deep-IRL framework, the performance of D-IRL and DK-IRL methods are unsatisfactory compared to the linear-IRL model of [21]. We observe that even with the augmented feature to reward mapping, the deep-IRL framework still lacks information regarding the neighbourhood context, which is a highly influential factor in crowded environments. The plotting based visual representations that we embedded in the DN-IRL and DKN-IRL baselines haven't been able to fully capture the neigh-

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

Table 1: Evaluation results for the Stanford Drone (SD) Dataset [18]. We evaluate performance under two settings, predicting 5 seconds and 12 seconds ahead. We report Modified Hausdorff Distance (MHD) [3] and the Average Displacement Error (ADE) [6, 7] as error metrics (lower is better for both metrics). For clarity supervised learning methods are shown with a blue background, the linear-IRL method with a yellow background, Deep IRL based baselines with a green background, and naive neighbourhood based deep IRL approaches with a purple background.

Method	5.0 Sec Ahead		12.0 Sec Ahead	
	ADE (pixels)	MHD (pixels)	ADE (pixels)	MHD (pixels)
S-LSTM [1]	31.19	30.13	68.26	67.49
CAR-Net [20]	25.72	-	-	-
SHA [7]	19.32	16.53	63.35	61.01
S-GAN [11]	19.27	16.51	62.11	60.93
SMN [6]	15.34	13.11	58.11	57.91
L-IRL [21]	12.93	11.95	44.35	42.91
D-IRL [23]	18.63	17.14	43.12	41.56
DK-IRL [24]	17.51	16.63	41.00	40.99
DN-IRL	16.01	15.32	39.16	38.17
DNK-IRL	15.23	14.01	37.35	36.08
Proposed	**06.15**	**04.08**	**28.15**	**27.91**

Table 2: Evaluation results for the SAIVT Multi-Spectral Trajectory (MST) Dataset [6]. We evaluate performance under two settings, predicting 5 seconds and 12 seconds ahead. We report MHD [3] and ADE [6, 7] as error metrics (lower is better for both metrics). For clarity supervised learning methods are shown with a blue background, the linear-IRL method with a yellow background, Deep IRL based baselines with a green background, and naive neighbourhood based deep IRL approaches with a purple background.

Method	5.0 Sec Ahead		12.0 Sec Ahead	
	ADE (pixels)	MHD (pixels)	ADE (pixels)	MHD (pixels)
S-LSTM [1]	20.21	19.13	43.35	40.11
SHA [7]	17.62	15.33	41.91	39.10
S-GAN [11]	17.59	15.31	41.93	40.06
SMN [6]	12.57	11.94	38.06	37.86
L-IRL [21]	10.01	09.87	32.24	31.65
D-IRL [23]	15.35	13.54	38.33	37.11
DK-IRL [24]	12.35	11.02	35.76	33.90
DN-IRL	14.51	13.55	37.71	35.41
DNK-IRL	11.43	10.68	34.55	32.61
Proposed	**04.89**	**03.59**	**26.78**	**35.67**

bourhood motion, and result in only a slight performance boost compared to [21]. We hypothesise that this is mainly due to the fact that the neighbourhood plots only offer a static representation of the neighbouring trajectories, and do not encode velocities and other kinematic factors.

With the proposed framework multiple LSTMs are utilised to effectively encode the spatial and temporal dynamics of the pedestrian of interest as well as the neighbours. This allows us to better capture motion information which leads us to attain better performance compared to state-of-the-art methods.

To further demonstrate the performance of the proposed methods we conducted an ablation experiment where we constructed ablation models as follows:

- a) M_{PI}: Uses the proposed LSTM based encoding scheme but uses the trajectory information from the pedestrian of interest only. Uses soft attention in the encoding process.

- b) $M_{PI} + \text{Neighbours}$: Uses the trajectory information from both the pedestrian of interest as well as the neighbours, however encodes everything through a single LSTM. Uses soft attention in the encoding process.

Table 3: Ablation model evaluations using the SAIVT Multi-Spectral Trajectory (MST) Dataset [6]. We evaluate performance under two settings, predicting 5 seconds and 12 seconds ahead. We report MHD [3] and ADE [6, 7] as error metrics (lower is better for both metrics).

Method	5.0 Sec Ahead		12.0 Sec Ahead	
	ADE (pixels)	MHD (pixels)	ADE (pixels)	MHD (pixels)
M_{PI}	**14.27**	**12.64**	**36.44**	**35.60**
$M_{PI} + \text{Neighbours}$	**08.13**	**07.34**	**30.57**	**29.92**
Proposed	**04.89**	**03.59**	**26.78**	**35.67**

Tab. 3 presents the ablation evaluation results. In this evaluation we used the MST dataset [6]. These results verify the importance of using the information from both the pedestrian of interest as well as from the neighbourhood. We would like to further compare the performance of the ablation model $M_{PI} + \text{Neighbours}$ and the L-IRL model in Tab. 2. This comparison clearly emphasises the utility of using the deep-IRL framework instead of the linear-IRL model. However, instead of naively passing all the trajectory information from a single LSTM, the proposed method effectively utilises the informative facts from a combination of soft and hard-wired attention, which allows us to better capture the neighbourhood dynamics.

In Fig. 6 we plot the Negative Log Likelihood (NLL) of a given test trajectory under the learned policy against the number of iterations. We normalise the NLL by the total length of the trajectories in the test set. It is clear that

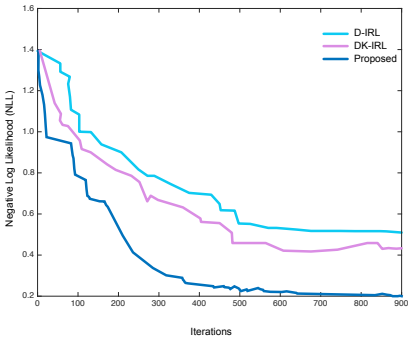


Figure 6: Change in Negative Log Likelihood (NLL) of a given trajectory in the test set under the learned policy against the number of iterations.

the proposed method understands the expected behaviour quickly from the available demonstrations compared to the baselines, denoting the value of mapping the dynamics of the environment through the proposed method.

An illustration of the predictions generated by the DK-IRL baseline method along with the proposed method for Stanford Drone [18] datasets is given in Fig. 7. The observed part of the trajectory is shown in blue. The ground truth future trajectory is in red while the neighbours are in green. We observe that the baseline method DK-IRL is accurate when predicting for short time intervals. However as we predict further into the future the model becomes more uncertain about the agent's behaviour. We speculate that this is mainly due to the fact that the motion of the neighbourhood also shapes the agent's behaviour, which the baselines fail to infer. By effectively capturing these dynamics, the proposed method generates better predictions.

Qualitative results of the proposed method and the recovered reward representation for two examples from the MST dataset [6] is given in Fig. 8, In the overlaid probability map and the reward map, colours from blue to yellow indicate low to high probabilities. Considering the structure of the environment, the model hypothesises two pathways, however assigns a higher probability to the ground truth pathway that is actually undertaken by the pedestrian.

6. Conclusion

In this paper we proposed a recurrent neural network framework for embedding neighbourhood dynamics when anticipating human trajectories using Deep Inverse Reinforcement Learning (D-IRL). We proposed the utilisation of an attention framework together with LSTMs to encode motion information for the pedestrian of interest as well as their neighbours, and we map these encoded dynamics into a feature map, preserving the structural integrity of the neighbourhood. Our experimental evaluations on multiple public benchmarks demonstrated the utility of the proposed

(a) DK-IRL (b) Proposed

Figure 7: Qualitative Results of the proposed method with DK-IRL baseline. The observed part of the trajectory is shown in blue. The ground through future trajectory is given in red while the neighbours are shown in green. In the overlaid probability map the colours from blue to yellow indicate low to high probability.

encoding mechanism, especially when anticipating pedestrian behaviour in the distant future, where the baseline systems were uncertain about the intentions or the end goals of the pedestrians, allowing us to attain state-of-the-art performance for both datasets. Encouraging results obtained under a security surveillance setting, especially when forecasting lengthier trajectories, verifies its applicability to understanding and predicting human behaviour in real world scenarios.

Acknowledgement

This research was supported by an Australian Research Council (ARC) Linkage grant LP140100282.

References

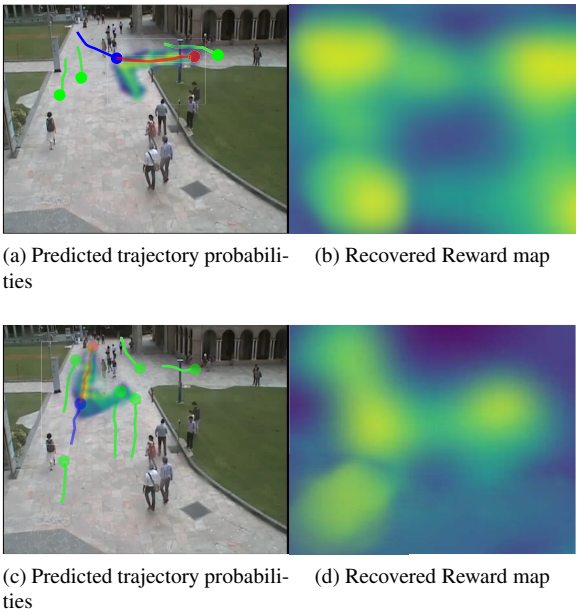
[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

[2] Richard Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, 6(5):679–684, 1957.

[3] M-P Dubuisson and Anil K Jain. A modified hausdorff distance for object matching. In *Proceedings of 12th international conference on pattern recognition*, volume 1, pages 566–568. IEEE, 1994.

[4] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Gd-gan: Generative adversarial networks for trajectory prediction and group detection in crowds. *Asian Conference in Computer VIsion (ACCV)*, 2018.

[5] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Learning temporal strategic relationships using generative adversarial imitation learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 113–121. International Foundation for Autonomous Agents and Multiagent Systems, 2018.



(a) Predicted trajectory probabilities (b) Recovered Reward map

(c) Predicted trajectory probabilities (d) Recovered Reward map

Figure 8: Qualitative Results of the proposed method and the recovered reward representation for the examples from the SAIVT Multi-Spectral Trajectory (MST) Dataset [6]. The observed part of the trajectory is shown in blue. The ground through future trajectory is given in red while the neighbours are shown in green. In the overlaid probability map and the reward map, colours from blue to yellow indicate low to high probabilities. Note that we use the radar trajectories in our evaluation, however we display these mapped to the image stream for ease of visualisation. In these examples the model hypothesises two pathways considering the structure of the environment, however assigns a higher probability to the ground truth pathway (that is actually undertaken by the pedestrian), demonstrating the utility of the proposed approach.

[6] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Pedestrian trajectory prediction with structured memory hierarchies. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 241–256. Springer, 2018.

[7] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *Neural networks*, 108:466–478, 2018.

[8] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Memory augmented deep generative models for forecasting the next shot location in tennis. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[9] Chelsea Finn, Tianhe Yu, Justin Fu, Pieter Abbeel, and Sergey Levine. Generalizing skills with semi-supervised reinforcement learning. *International Conference on Learning Representation, (ICLR)*, 2016.

[10] Justin Fu, Katie Luo, and Sergey Levine. Learning robust

rewards with adversarial inverse reinforcement learning. *International Conference on Learning Representation, (ICLR)*, 2018.

[11] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.

[12] Tsubasa Hirakawa, Takayoshi Yamashita, Ken Yoda, Toru Tamaki, and Hironobu Fujiyoshi. Travel time-dependent maximum entropy inverse reinforcement learning for seabird trajectory prediction. In *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 430–435. IEEE, 2017.

[13] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] Truc Viet Le, Siyuan Liu, and Hoong Chuin Lau. A reinforcement learning framework for trajectory prediction under uncertainty and budget constraint. In *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, pages 347–354. IOS Press, 2016.

[16] Yunzhu Li, Jiaming Song, and Stefano Ermon. Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems*, pages 3812–3822, 2017.

[17] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 261–268. IEEE, 2009.

[18] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[19] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. *arXiv preprint arXiv:1806.01482*, 2018.

[20] Amir Sadeghian, Ferdinand Legros, Maxime Voisin, Ricky Vesel, Alexandre Alahi, and Silvio Savarese. Car-net: Clairvoyant attentive recurrent network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 151–167, 2018.

[21] Khaled Saleh, Mohammed Hossny, and Saeid Nahavandi. Long-term recurrent predictive model for intent prediction of pedestrians via inverse reinforcement learning. In *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8. IEEE, 2018.

[22] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*, 2015.

[23] Markus Wulfmeier, Dushyant Rao, Dominic Zeng Wang, Peter Ondruska, and Ingmar Posner. Large-scale cost function learning for path planning using deep inverse reinforcement learning. *The International Journal of Robotics Research*, 36(10):1073–1087, 2017.

[24] Yanfu Zhang, Wenshan Wang, Rogerio Bonatti, Daniel Maturana, and Sebastian Scherer. Integrating kinematics and environment context into deep inverse reinforcement learning for predicting off-road vehicle trajectories. *Conference on Robot Learning (CoRL)*, 2018.

[25] Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2871–2878. IEEE, 2012.

[26] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[27] Haosheng Zou, Hang Su, Shihong Song, and Jun Zhu. Understanding human behaviors in crowds by imitating the decision-making process. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.