

Crowd-Robot Interaction:

Crowd-aware Robot Navigation with Attention-based Deep Reinforcement Learning

Changan Chen, Yuejiang Liu, Sven Kreiss, Alexandre Alahi

Abstract—Mobility in an effective and socially-compliant manner is an essential yet challenging task for robots operating in crowded spaces. Recent works have shown the power of deep reinforcement learning techniques to learn socially cooperative policies. However, their cooperation ability deteriorates as the crowd grows since they typically relax the problem as a one-way Human-Robot interaction problem. In this work, we want to go beyond first-order Human-Robot interaction and more explicitly model Crowd-Robot Interaction (CRI). We propose to (i) rethink pairwise interactions with a self-attention mechanism, and (ii) jointly model Human-Robot as well as Human-Human interactions in the deep reinforcement learning framework. Our model captures the Human-Human interactions occurring in dense crowds that indirectly affects the robot’s anticipation capability. Our proposed attentive pooling mechanism learns the collective importance of neighboring humans with respect to their future states. Various experiments demonstrate that our model can anticipate human dynamics and navigate in crowds with time efficiency, outperforming state-of-the-art methods.

I. INTRODUCTION

With the rapid growth of machine intelligence, robots are envisioned to expand habitats from isolated environments to social space shared with humans. Traditional approaches for robot navigation often consider moving agents as static obstacles [1]–[4] or react to them through a one-step lookahead [5]–[7], resulting in short-sighted, unsafe and unnatural behaviors. In order to navigate through a dense crowd in a socially compliant manner, robots need to understand human behavior and comply with their cooperative rules [8]–[11].

Navigation with social etiquette is a challenging task. As communications among agents (e.g., humans) are not widely available, robots need to perceive and anticipate the evolution of the crowd, which can involve complex interactions. Research works in trajectory prediction have proposed several hand-crafted or data-driven methods to model the agent-agent interactions [12]–[15]. Nevertheless, the integration of these prediction models in the decision-making process remains challenging.

Earlier works separate prediction and planning in two steps, attempting to identify a safe path after forecasting the future trajectories of the others [16], [17]. However, the probabilistic evolution of a crowd for a few steps can expand to the entire space in a dense environment, causing the freezing robot problem [18]. To address this issue, a large number of works have focused on obstacle avoidance methods that jointly plan plausible paths for all the decision-makers, in hope to make room for each other cooperatively

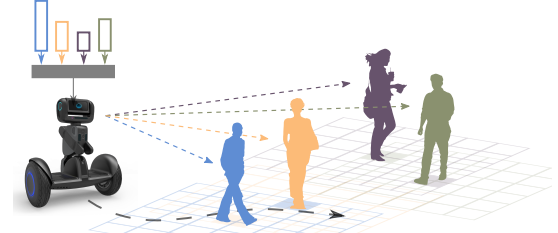


Fig. 1: In this work, we present a method that jointly model Human-Robot and Human-Human interactions for navigation in crowds.

[18]. Nevertheless, these methods suffer from the stochasticity of neighbors’ behaviors as well as high computational cost when applied to densely populated environments.

As an alternative, reinforcement learning frameworks have been used to train computationally efficient policies that implicitly encode the interactions and cooperation among agents. Although significant progress has been made in recent works [19]–[22], existing models are still limited in two aspects: i) the collective impact of the crowd is usually modeled by a simplified aggregation of the pairwise interactions, such as a maximin operator [19] or LSTM [22], which may fail to fully represent all the interactions; ii) most methods focus on one-way interactions from humans to the robot, but ignore the interactions within the crowd which could indirectly affect the robot. These limitations degrade the performance of cooperative planning in complex and crowded scenes.

In this work, we address the above issues by going beyond first-order Human-Robot interaction and dive into Crowd-Robot Interaction (CRI). We propose to: (i) rethink Human-Robot pairwise interactions with a self-attention mechanism, and (ii) jointly model Human-Robot as well as Human-Human interactions in the reinforcement learning framework. Inspired by [13]–[15], our model extracts features for pairwise interactions between the robot and each human and captures the interactions among humans via local maps. Subsequently, we aggregate the interaction features with a self-attention mechanism that infers the relative importance of neighboring humans with respect to their future states. Our proposed model can naturally take into account an arbitrary number of agents, providing a good understanding of the crowd behavior for planning. An extensive set of simulation experiments shows that our approach can anticipate crowd dynamics and navigate in time efficient paths, outperforming state-of-the-art methods. We also demonstrate the effectiveness of our model on a robotic platform in real-world environments. The code of our approach is available

II. BACKGROUND

A. Related Work

Earlier works have largely leveraged well-engineered interaction models to enhance the social awareness in robot navigation. One pioneering work is the Social Force [23], [24], which has been successfully applied to autonomous robots in simulation and real-world environments [25]–[27]. Another method named Interacting Gaussian Process (IGP) models the trajectory of each agent as an individual Gaussian Process and proposes an interaction potential term to couple the individual GP for interaction [18], [28], [29]. In multi-agent settings, where the same policy is applied to all the agents, reactive methods such as RVO [5] and ORCA [6] seek joint obstacle avoidance velocities under reciprocal assumptions. The key challenge for these models is that they heavily rely on hand-crafted functions and cannot generalize well to various scenarios for crowd-like cooperation.

Another line of work uses imitation learning approaches to program socially compliant policies based on demonstrations of desired behaviors. Navigation policies that map raw depth images and lidar measurements to actions are developed in [30], [31] respectively by directly mimicking expert demonstrations. Beyond behavioral cloning, inverse reinforcement learning has been used in [10], [11], [32] to learn the underlying cooperation features from human data using the maximum entropy method. The learning outcomes in these works highly depend on the scale and quality of demonstrations, which is not only resource consuming but also constrains the quality of the learned policy by human efforts. In our work, we adopt the imitation learning approach to warm start our model training.

Reinforcement Learning (RL) methods have been intensively studied over the last few years and applied to various fields since it started to achieve superior performance in video games [33]. In the field of robot navigation, recent works have used RL to learn sensorimotor policies in static and dynamic environments from the raw observations [21], [34] and socially cooperative policies with the agent-level state information [19], [20], [22]. To handle a variant number of neighbors, the method reported in [19] adapts from the two-agent to the multi-agent case through a maximin operation that picks up the best action against the worst-case for the crowd. A later extension uses an LSTM model to process the state of each neighbor sequentially in reverse order of the distance to the robot [22]. In contrast to these simplifications, we propose a novel neural network model to capture the collective impact of the crowd explicitly.

A variety of deep neural networks architectures have been proposed in recent years to improve the modeling of Human-Human interactions, some of which are compared in [35]. The Social LSTM method models each individual by an LSTM and shares the states of neighboring LSTMs through a social pooling module [13], which is later extended to a rich set of variants for improved accuracy and efficiency [15], [36], [37]. Some other works model the social interactions

through spatio-temporal graphs, where an attention model is introduced recently to learn the relative importance of each agent [14]. Our work built upon these models designs a social attentive pooling module to encode crowd cooperative behaviors in a deep reinforcement learning framework.

B. Problem Formulation

In this work, we consider a navigation task where a robot moves towards a goal through a crowd of n humans. This can be formulated as a sequential decision making problem in a reinforcement learning framework [19], [20], [22]. For each agent (robot or human), the position $\mathbf{p} = [p_x, p_y]$, velocity $\mathbf{v} = [v_x, v_y]$ and radius r can be observed by the others. The robot is also aware of its unobservable state including the goal position \mathbf{p}_g and preferred speed v_{pref} . We assume that the velocity of the robot \mathbf{v}_t can be achieved instantly after the action command \mathbf{a}_t , i.e. $\mathbf{v}_t = \mathbf{a}_t$. Let \mathbf{s}_t denote the state of the robot and $\mathbf{w}_t = [\mathbf{w}_t^1, \mathbf{w}_t^2, \dots, \mathbf{w}_t^n]$ denote the state of humans at time t . The joint state for robot navigation is defined as $\mathbf{s}_t^{jn} = [\mathbf{s}_t, \mathbf{w}_t]$.

The optimal policy, $\pi^* : \mathbf{s}_t^{jn} \mapsto \mathbf{a}_t$, is to maximize the expected return:

$$\begin{aligned} \pi^*(\mathbf{s}_t^{jn}) = & \underset{\mathbf{a}_t}{\operatorname{argmax}} R(\mathbf{s}_t^{jn}, \mathbf{a}_t) + \\ & \gamma^{\Delta t \cdot v_{pref}} \int_{\mathbf{s}_{t+\Delta t}^{jn}} P(\mathbf{s}_t^{jn}, \mathbf{a}_t, \mathbf{s}_{t+\Delta t}^{jn}) V^*(\mathbf{s}_{t+\Delta t}^{jn}) d\mathbf{s}_{t+\Delta t}^{jn} \quad (1) \\ V^*(\mathbf{s}_t^{jn}) = & \sum_{t'=t}^T \gamma^{t'-v_{pref}} R_{t'}(\mathbf{s}_{t'}^{jn}, \pi^*(\mathbf{s}_{t'}^{jn})), \end{aligned}$$

where $R_t(\mathbf{s}_t^{jn}, \mathbf{a}_t)$ is the reward received at time t , $\gamma \in (0, 1)$ is a discount factor, V^* is the optimal value function, $P(\mathbf{s}_t^{jn}, \mathbf{a}_t, \mathbf{s}_{t+\Delta t}^{jn})$ is the transition probability from time t to time $t + \Delta t$. The preferred velocity v_{pref} is used as a normalization term in the discount factor for numerical reasons [19].

We follow the formulation of the reward function defined in [19], [20], which awards task accomplishments while penalizing collisions or uncomfortable distances,

$$R_t(\mathbf{s}_t^{jn}, \mathbf{a}_t) = \begin{cases} -0.25 & \text{if } d_t < 0 \\ -0.1 + d_t/2 & \text{else if } d_t < 0.2 \\ 1 & \text{else if } \mathbf{p}_t = \mathbf{p}_g \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where d_t is the minimum separation distance between the robot and the humans during the time period $[t - \Delta t, t]$.

C. Value Network Training

The value network is trained by the temporal-difference method with standard experience replay and fixed target network techniques [19], [33]. As outlined in Algorithm 1, the model is first initialized with imitation learning using a set of demonstrator experience (line 1-3), and subsequently refined from experience of interactions (line 4-14). One distinction from the previous works [19], [20] is that the next state \mathbf{s}_{t+1}^{jn} in line 7 is obtained by querying the environment instead of assuming a linear motion model, mitigating the

Algorithm 1 Deep V-learning

```

1: Initialize value network  $V$  with demonstration  $D$ 
2: Initialize target value network  $\hat{V} \leftarrow V$ 
3: Initialize experience replay memory  $E \leftarrow D$ 
4: for episode = 1,  $M$  do
5:   Initialize random sequence  $s_0^{jn}$ 
6:   repeat
7:      $a_t \leftarrow \operatorname{argmax}_{a_t \in A} R(s_t^{jn}, a_t) + \gamma^{A \cdot v_{pref}} V(s_{t+\Delta t}^{jn})$ 
8:     Store tuple  $(s_t^{jn}, a_t, r_t, s_{t+\Delta t}^{jn})$  in  $E$ 
9:     Sample random minibatch tuples from  $D$ 
10:    Set target  $y_i = r_i + \gamma^{A \cdot v_{pref}} \hat{V}(s_{t+1}^{jn})$ 
11:    Update value network  $V$  by gradient descent
12:  until terminal state  $s_t$  or  $t \geq t_{max}$ 
13:  Update target network  $\hat{V} \leftarrow V$ 
14: end for
15: return  $V$ 

```

issue of system dynamics in the training process. During deployment, the transition probability can be approximated by a trajectory prediction model [12], [13], [15].

To tackle the problem (1) effectively, the value network model needs to accurately approximate the optimal value function V^* that implicitly encodes the social cooperation among agents. Previous works on this track didn't fully model the crowd interactions, which degrades the accuracy of value estimation for a densely populated scene. In the following sections, we will present a novel Crowd-Robot Interaction model that can effectively learn to navigate in crowded spaces.

III. APPROACH

When humans walk in a densely populated scene, they cooperate with others by anticipating the behaviors of their neighbors in the vicinity, particularly those who are likely to be involved in some future interactions. This motivates us to design a model that can calculate the relative importance and encode the collective impact of neighboring agents for socially compliant navigation. Inspired by the social pooling [13], [15] and attention models [14], [38]–[42], we introduce a socially attentive network that consists of three modules:

- **Interaction module:** models the Human-Robot interactions explicitly and encodes the Human-Human interactions through coarse-grained local maps.
- **Pooling module:** aggregates the interactions into a fixed-length embedding vector by a self-attention mechanism.
- **Planning module:** estimates the value of the joint state of the robot and crowd for social navigation.

In the following subsections, we present the architecture and formulations of each module. The time index t is omitted below for simplicity.

A. Parameterization

We follow the robot-centric parameterization in [19], [22], where the robot is located at the origin and the x-axis is pointing toward the robot's goal. The states of the robot and walking humans after transformation are:

$$\begin{aligned}
 s &= [d_g, v_{pref}, v_x, v_y, r], \\
 w_i &= [p_x, p_y, v_x, v_y, r_i, d_i, r_i + r],
 \end{aligned} \tag{3}$$

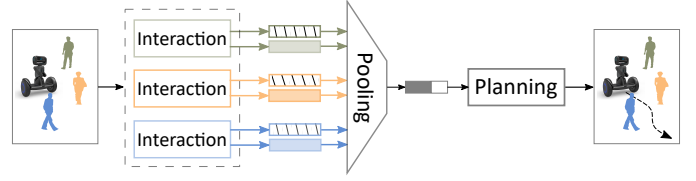


Fig. 2: Overview of our method for socially attentive navigation made of 3 modules: Interaction, Pooling, and Planning described in Section III. Interactions between the robot and each human are extracted from the interaction module and subsequently aggregated in the pooling module. The planning module estimates the value of the joint state of the robot and humans for navigation in crowds.

where $d_g = \|\mathbf{p} - \mathbf{p}_g\|_2$ is the robot's distance to the goal and $d_i = \|\mathbf{p} - \mathbf{p}_i\|_2$ is the robot's distance to the neighbor i .

B. Interaction Module

Each human has an impact on the robot and is meanwhile influenced by his/her neighboring humans. Explicitly modeling all pairs of interactions among humans leads to $O(N^2)$ complexity [14], which is not computationally-desirable for a policy to scale up in dense scenes. We tackle this problem by introducing a pairwise interaction module that explicitly models the Human-Robot interaction while using local maps as a coarse-grained representation for the Human-Human interactions.

Given a neighborhood of size L , we construct a $L \times L \times 3$ map tensor M_i centered at each human i to encode the presence and velocities of neighbors, which is referred as local map in Fig. 3:

$$M_i(a, b, :) = \sum_{j \in \mathcal{N}_i} \delta_{ab}[x_j - x_i, y_j - y_i] w'_j, \tag{4}$$

where $w'_j = (v_{xj}, v_{yj}, 1)$ is a local state vector for human j , $\delta_{mn}[x_j - x_i, y_j - y_i]$ is an indicator function which equals to 1 only if the relative position $(\Delta x, \Delta y)$ is located in the cell (a, b) , \mathcal{N}_i is the set of neighboring humans around the i^{th} person.

We embed the state of human i and the map tensor M_i , together with the state of the robot, into a fixed length vector e_i using a multi-layer perceptron (MLP):

$$e_i = \phi_e(s, w_i, M_i; W_e), \tag{5}$$

where $\phi_e(\cdot)$ is an embedding function with ReLU activations and W_e is the embedding weights.

The embedding vector e_i is fed to a subsequent MLP to obtain the pairwise interaction feature between the robot and person i :

$$h_i = \psi_h(e_i; W_h), \tag{6}$$

where $\psi_h(\cdot)$ is a fully-connected layer with ReLU non-linearity and W_h is the network weights.

C. Pooling Module

Since the number of surrounding humans can vary dramatically in different scenes, we need a model that can handle an arbitrary number of inputs into a fixed size output. Everett et al. [22] proposed to feed the states of all humans

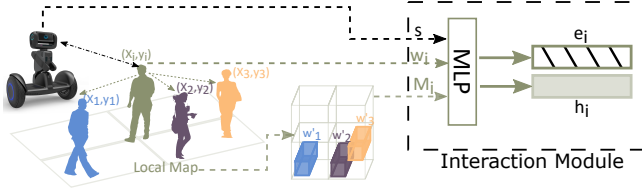


Fig. 3: Illustration of our interaction module. We use a multi-layer perceptron to extract the pairwise interaction feature between the robot and each human i . The impact of the other people on the human i is represented by a local map.

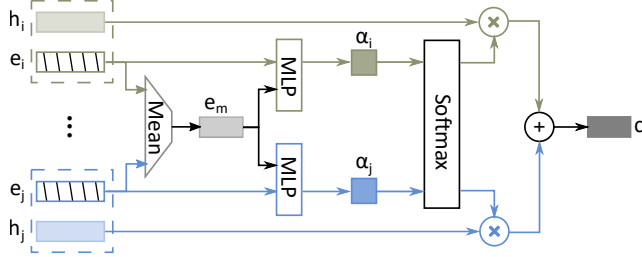


Fig. 4: Architecture of our pooling module. We use a multi-layer perceptron to compute the attention score for each person from the individual embedding vector together with the mean embedding vector. The final joint representation is a weighted sum of the pairwise interactions.

into an LSTM [43] sequentially in descending order of their distances to the robot. However, the underlying assumption that the closest neighbors have the strongest influence is not always true. Some other factors, such as speed and direction, are also essential for correctly estimating the importance of a neighbor. Leveraging the recent progress in self-attention mechanism [38], [40], [44], we propose a social attentive pooling module to learn the relative importance of each neighbor and the collective impact of the crowd in a data-driven fashion.

The interaction embedding e_i is transformed into an attention score α_i as follows:

$$e_m = \frac{1}{n} \sum_{k=1}^n e_k, \quad (7)$$

$$\alpha_i = \psi_\alpha(e_i, e_m; W_\alpha), \quad (8)$$

where e_m is a fixed-length embedding vector obtained by mean pooling all the individuals, $\psi_\alpha(\cdot)$ is a MLP with ReLU activations and W_α is the weights.

Given the pairwise interaction vector h_i and the corresponding attention score α_i for each neighbor i , the final representation of the crowd is a weighted linear combination of all the pairs:

$$c = \sum_{i=1}^n \text{softmax}(\alpha_i) h_i. \quad (9)$$

D. Planning Module

Based on the compact representation of the crowd c , we build a planning module that estimates the state value v for cooperative planning:

$$v = f_v(s, c; W_v), \quad (10)$$

where $f_v(\cdot)$ is an MLP with ReLU activations, the weights are denoted by W_v .

E. Implementation Details

The local map is a 4×4 grid centered at each human and the side length of each cell is $1m$. The hidden units of functions $\phi_e(\cdot)$, $\psi_h(\cdot)$, $\psi_\alpha(\cdot)$, $f_v(\cdot)$ are (150,100), (100,50), (100,100), (150,100,100) respectively.

We implemented the value network model in PyTorch [45] and trained it with a batch size of 100 using Adam [46]. The learning rate is 0.01 for imitation learning and 0.001 for reinforcement learning. The discount factor γ is set to be 0.9. The exploration rate of the ϵ -greedy policy decays from 0.5 to 0.1 linearly in the first 5k episodes. The RL training took approximately 10 hours to converge on an i7-8700 CPU.

This work assumes holonomic kinematics for the robot, i.e. it can move in any direction. The action space consists of 80 discrete actions: 5 speeds exponentially spaced between $(0, v_{pref}]$ and 16 headings evenly spaced between $[0, 2\pi)$.

IV. EXPERIMENTS

A. Simulation Setup

We built a simulation environment in Python to simulate robot navigation in crowds. The simulated humans are controlled by ORCA [6], the parameters of which are sampled from a Gaussian distribution to introduce behavioral diversity. We use circle crossing scenarios for both training and test, where all the agents are randomly positioned on a circle of radius $4m$ with some random perturbation added to their x,y coordinates.

Three existing state-of-the-art methods, ORCA [5], CADRL [19] and LSTM-RL [22], are implemented as baseline methods. The main distinction between our method and the RL baselines lies in the interaction and pooling module, we keep the planning module identical for a fair comparison. Note that the LSTM-RL in our implementation differs from the original one [22] in that we use the joint state instead of human's observable state as the input of the LSTM unit. We refer to our full model as LM-SARL and the model without local map as SARL for ablation experiments.

To fully evaluate the effectiveness of the proposed model, we look into two simulation settings: invisible and visible. The former one sets the robot invisible to the other humans. As a result, the simulated humans react only to humans but not to the robot. We also removed the penalty on the uncomfortable distance in the reward function to eliminate extra factors for collision avoidance. This setting serves as a clean testbed for validating the model's ability in reasoning the Human-Robot and Human-Human interaction without affecting human's behaviors. The latter visible setting resembles more realistic cases where the robot and humans have mutual impacts. Models are evaluated with 500 random test cases in both settings.

B. Quantitative Evaluation

1) *Invisible Robot*: In the invisible setting, a robot needs to forecast future trajectories of all the humans to avoid collisions. Table I reports the rates of success, collision, the average navigation time as well as the average discounted cumulative reward in test experiments.

As expected, the ORCA method fails badly in the invisible setting due to the violation of the reciprocal assumption. Among all the reinforcement learning methods, the CADRL has the lowest success rate. This is because the maximin approach used in the CADRL can only take a single pair of interaction into account while ignoring the rest. The frequent failure of the CADRL shows the necessity for a policy to simultaneously take all humans into account.

As a consequence of directly aggregating the surrounding agents' information, both LSTM-RL and SARL achieve higher success rate. However, LSTM-RL still suffers from some occasional collisions and timeouts, whereas the SARL accomplishes all the test cases. We also observe a dramatic reduction in the average navigation time in the SARL. These results demonstrate the advantages of the proposed attentive pooling mechanism in capturing the collective impact of the crowd. The full version of our model, LM-SARL, achieves the best results in the invisible experiments, outperforming the SARL in terms of both the navigation time and the cumulative reward. Though not by a large margin, this improvements indicates the benefits of encoding the interactions among humans.

2) *Visible Robot*: We further compare the navigation performance of our models with the baselines in the visible setting. The robot not only needs to understand the behavior of humans but also interact with them to obtain high rewards. We define the discomfort frequency as t_{disc}/T , where t_{disc} is the duration when the separation distance $d_i < 0.2\text{m}$. To fairly compare the ORCA baseline with the learning methods, we add an extra 0.1m as the virtual radius of the agent to maintain a comfortable distance to humans for human-aware navigation [9]. The results of all the methods are summarized in Table II.

Different from the invisible case which violates the reciprocal assumption, the ORCA policy in the visible setting achieves a very high success rate and never invade the comfort zone of the other humans. However, the ORCA baseline fails to obtain high rewards due to the short-sighted and conservative behaviors. As pointed out in [31], tuning ORCA towards an objective function can be a tedious and challenging process compared with the learning-based methods.

The Reinforcement Learning results in the visible setting are similar to the invisible ones as expected. Our SARL model outperforms the baselines significantly, and the LM-SARL shows further improvements on the final reward. Since the Human-Human interactions are not significant all the time, their effect on the quantitative results is diluted over episodes. However, we see qualitative improvements which we discuss in the next section.

Methods	Success	Collision	Time	Reward
ORCA [5]	0.43	0.57	10.86	0.054
CADRL [19]	0.78	0.22	10.80	0.222
LSTM-RL [22]	0.95	0.03	11.82	0.279
SARL (Ours)	1.00	0.00	10.55	0.338
LM-SARL (Ours)	1.00	0.00	10.46	0.342

TABLE I: Quantitative results in the invisible setting. “Success”: the rate of robot reaching its goal without a collision. “Collision”: the rate of robot colliding with other humans. “Time”: the robot’s navigation time to reach its goal in seconds. “Reward”: discounted cumulative reward in a navigation task.

Methods	Success	Collision	Time	Disc.	Reward
ORCA [5]	0.99	0.00*	12.29	0.00*	0.284
CADRL [19]	0.94	0.03	10.82	0.10	0.291
LSTM-RL [22]	0.98	0.02	11.29	0.05	0.299
SARL (Ours)	0.99	0.01	10.58	0.02	0.332
LM-SARL (Ours)	1.00	0.00	10.59	0.03	0.334

TABLE II: Quantitative results in the visible setting. “Disc.” refers to as the discomfort frequency (% of duration where robot is too close to other humans). (*) Note that ORCA has a “Collision” and “Disc.” of 0 by design.

C. Qualitative Evaluation

We further investigate the effectiveness of our model for socially compliant navigation through qualitative analysis. As shown in Fig. 5, the navigation paths of different methods are compared in an invisible test case, where the trajectories of humans are identical for a clear comparison. When encountering humans in the center of the space, the CADRL passes them aggressively. By contrast, the LSTM-RL slows down dramatically to avoid the crowd from 4.0s to 8.0s, ending up with a long navigation time. Neither the overly aggressive CADRL nor the conservative LSTM-RL are desirable for robot navigation in the crowd. In comparison to the baselines, our SARL hesitates at first but then recognizes a shorter path to the goal through the center. By taking the shortcut, the robot successfully avoids the other humans. The LM-SARL identifies the central highway from the very beginning, establishing a smart trace concerning both the safety distance and navigation time.

In addition to the overall trajectory, we take a closer look at the learned policies in a typical crowded frame where the robot is surrounded by multiple humans. Fig. 6 shows the attention scores of humans inferred by our LM-SARL model. The lowest attention score is assigned to the #4 human who has the largest distance to the robot. The human #5 located not far from the robot also receives a low score, as he is walking away from the robot. In contrast, our model pays more attention to #1, #2 and #3, all of which have a potential future influence on robot’s path planning. The #2 is the closest to the robot and obtains a high attention as a natural result. However, our model gives the highest attention score to #3, who is most likely to get closest to the robot in the next few steps. Through assigning distinctive scores to different humans, our attentive pooling module demonstrates a good ability to reason the relative importance of humans in a dense scene.

As the ultimate objective of our model is to accurately estimate the state value for planning, we finally compare the

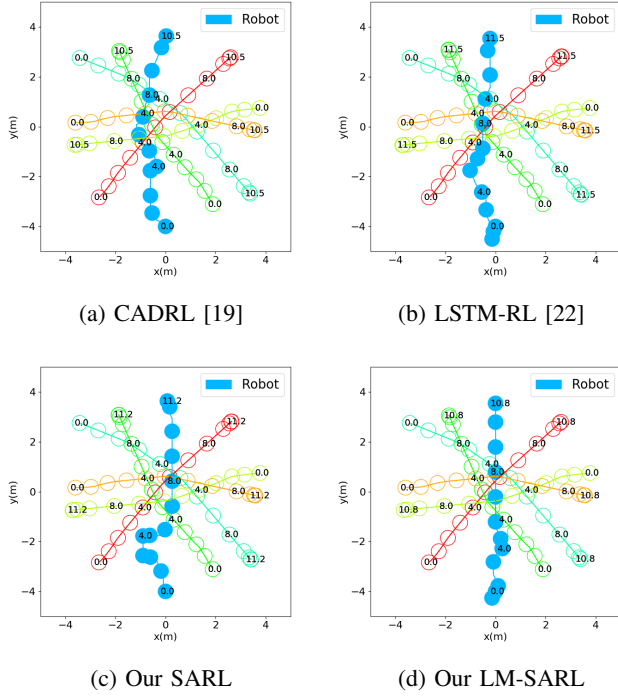


Fig. 5: Trajectory comparison in an invisible test case. Circles are the positions of agents at the labeled times. When encountering humans, CADRL and LSTM-RL demonstrate overly aggressive and conservative behaviors respectively. In contrast, our SARL and LM-SARL successfully identify a shortcut through the center, which allows the robot to keep some distance from others while navigating to the goal quickly.

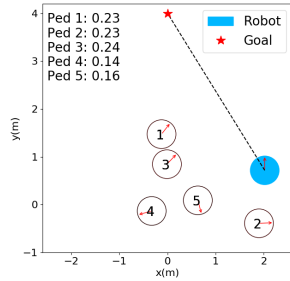


Fig. 6: Attention scores in a dense scene. Our LM-SARL assigns low importance scores to the human #4 and #5 who walk away, whereas attending with the highest weight to #3 who is most likely to get close soon.

values estimated by different methods in Fig. 7. Given that humans #1 and #3 are likely to cross the straight path from the robot to the goal, the robot is expected to either step aside or to slow down to avoid them.

Limited by the maximin selection, CADRL predicts low values only in the direction towards the closest human #2 while erroneously assigning the highest value to the direction 120°. The LSTM-RL model shifts the action preference slightly to the left but still overestimates the values of the high speeds in the directions around 120°.

In contrast, our SARL model predicts distinctly low values for the full speeds in the dangerous directions from 70° to

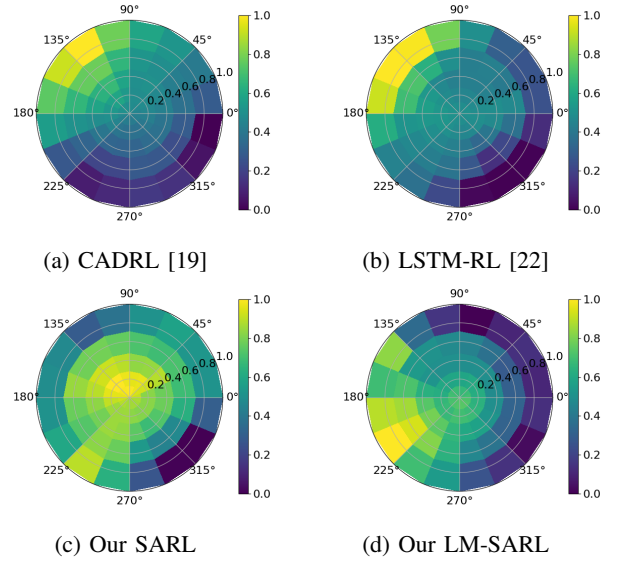


Fig. 7: Value estimations by different methods for the dense scene in Fig. 6. The baseline methods predict high values for high speeds straight towards the goal, which is dangerous because of humans #1 and #3. In contrast, our SARL slows down and waits safely, and our LM-SARL prefers to turn to 200°, preparing to pass behind them.

200°, leading to a slow down to avoid collisions instead of moving fast. More interesting patterns can be observed in the LM-SARL estimation, where even the low speeds around the 120° direction are discouraged. Considering the social repulsive forces between the person #1 and #3, person #3 might turn to the robot in the future, raising potential dangers or delays in the 120° direction. By encoding the Human-Human interactions through local maps, LM-SARL succeeds in providing a smart action in the 200° direction, which paves the way for cutting behind #1 and #3. This indicates LM-SARL’s potential for reasoning about complex interactions among agents.

D. Real-world Experiments

Aside from the simulation experiments above, we also examine the trained policy in real-world experiments on a Segway robotic platform. A video showing the effectiveness of our model for socially compliant navigation can be found in supplement materials.

V. CONCLUSION

In this work, we tackle the crowd navigation problem by decomposing the Crowd-Robot Interaction into two parts. We first jointly model the Human-Robot and Human-Human interactions and then aggregate the interactions into a compact crowd representation via a self-attention model. Our approach outperforms state-of-the-art reinforcement learning methods in terms of time-efficiency and task accomplishments. Qualitatively, we demonstrate our model’s ability to reason about Human-Human interactions and to selectively attend to humans.

REFERENCES

- [1] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, Sep. 1989.
- [2] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots in cluttered environments," in *IEEE International Conference on Robotics and Automation Proceedings*, May 1990, pp. 572–577 vol.1.
- [3] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, Jun. 1991.
- [4] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics Automation Magazine*, vol. 4, no. 1, pp. 23–33, Mar. 1997.
- [5] J. v. d. Berg, M. Lin, and D. Manocha, "Reciprocal Velocity Obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 1928–1935.
- [6] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-Body Collision Avoidance," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Springer Berlin Heidelberg, 2011, pp. 3–19.
- [7] J. Snape, J. v. d. Berg, S. J. Guy, and D. Manocha, "The Hybrid Reciprocal Velocity Obstacle," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 696–706, Aug. 2011.
- [8] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A survey of socially interactive robots," *Robotics and Autonomous Systems*, vol. 42, no. 3, pp. 143–166, Mar. 2003.
- [9] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, Dec. 2013.
- [10] N. Roy, P. Newman, and S. Srinivasa, "Feature-Based Prediction of Trajectories for Socially Compliant Navigation," in *Robotics: Science and Systems VIII*. MITP, 2013.
- [11] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1289–1307, Sep. 2016.
- [12] D. Helbing and P. Molnr, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, May 1995.
- [13] A. Alahi *et al.*, "Social LSTM: Human Trajectory Prediction in Crowded Spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 961–971.
- [14] A. Vemula, K. Muelling, and J. Oh, "Social Attention: Modeling Attention in Human Crowds," *arXiv:1710.04689 [cs]*, Oct. 2017, arXiv: 1710.04689.
- [15] A. Gupta *et al.*, "Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks," *arXiv:1803.10892 [cs]*, Mar. 2018, arXiv: 1803.10892.
- [16] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning Motion Patterns of People for Compliant Robot Motion," *The International Journal of Robotics Research*, vol. 24, no. 1, pp. 31–48, Jan. 2005.
- [17] G. S. Aoude *et al.*, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Autonomous Robots*, vol. 35, no. 1, pp. 51–76, Jul. 2013.
- [18] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 797–803.
- [19] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized Non-communicating Multiagent Collision Avoidance with Deep Reinforcement Learning," *arXiv:1609.07845 [cs]*, Sep. 2016, arXiv: 1609.07845.
- [20] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially Aware Motion Planning with Deep Reinforcement Learning," *arXiv:1703.08862 [cs]*, Mar. 2017, arXiv: 1703.08862.
- [21] P. Long *et al.*, "Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning," *arXiv:1709.10082 [cs]*, Sep. 2017, arXiv: 1709.10082.
- [22] M. Everett, Y. F. Chen, and J. P. How, "Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning," *arXiv:1805.01956 [cs]*, May 2018, arXiv: 1805.01956.
- [23] D. Helbing and P. Molnr, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, May 1995.
- [24] T. Kretz, J. Lohmiller, and P. Sukennik, "Some Indications on How to Calibrate the Social Force Model of Pedestrian Dynamics," *Transportation Research Record*, p. 0361198118786641, Jul. 2018.
- [25] A. Sud *et al.*, "Real-time Navigation of Independent Agents Using Adaptive Roadmaps," in *Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '07. New York, NY, USA: ACM, 2007, pp. 99–106.
- [26] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 1688–1694.
- [27] G. Ferrer, A. G. Zulueta, F. H. Cotarelo, and A. Sanfeliu, "Robot social-aware navigation framework to accompany people walking side-by-side," *Autonomous Robots*, vol. 41, no. 4, pp. 775–793, Apr. 2017.
- [28] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: the case for cooperation," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 2153–2160.
- [29] P. Trautman, "Sparse interacting Gaussian processes: Efficiency and optimality theorems of autonomous crowd navigation," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec. 2017, pp. 327–334.
- [30] L. Tai, J. Zhang, M. Liu, and W. Burgard, "Socially Compliant Navigation through Raw Depth Inputs with Generative Adversarial Imitation Learning," *arXiv:1710.02543 [cs]*, Oct. 2017, arXiv: 1710.02543.
- [31] P. Long, W. Liu, and J. Pan, "Deep-Learned Collision Avoidance Policy for Distributed Multiagent Navigation," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 656–663, Apr. 2017.
- [32] M. Pfeiffer *et al.*, "Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 2096–2101.
- [33] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
- [34] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 31–36.
- [35] S. Becker, R. Hug, W. Hbner, and M. Arens, "An Evaluation of Trajectory Prediction Approaches and Notes on the TrajNet Benchmark," *arXiv:1805.07663 [cs]*, May 2018, arXiv: 1805.07663.
- [36] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Soft + Hardwired Attention: An LSTM Framework for Human Trajectory Prediction and Abnormal Event Detection," *arXiv:1702.05552 [cs]*, Feb. 2017, arXiv: 1702.05552.
- [37] A. Sadeghian *et al.*, "SoPhie: An Attentive GAN for Predicting Paths Compliant to Social and Physical Constraints," *arXiv:1806.01482 [cs]*, Jun. 2018, arXiv: 1806.01482.
- [38] Y. Liu, C. Sun, L. Lin, and X. Wang, "Learning Natural Language Inference using Bidirectional LSTM model and Inner-Attention," *arXiv:1605.09090 [cs]*, May 2016, arXiv: 1605.09090.
- [39] A. Vaswani *et al.*, "Attention Is All You Need," *arXiv:1706.03762 [cs]*, Jun. 2017, arXiv: 1706.03762.
- [40] Z. Lin *et al.*, "A Structured Self-attentive Sentence Embedding," *arXiv:1703.03130 [cs]*, Mar. 2017, arXiv: 1703.03130.
- [41] Y. Hoshen, "VAIn: Attentional Multi-agent Predictive Modeling," in *Advances in Neural Information Processing Systems 30*, I. Guyon *et al.*, Eds. Curran Associates, Inc., 2017, pp. 2701–2711.
- [42] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-Attention Generative Adversarial Networks," *arXiv:1805.08318 [cs, stat]*, May 2018, arXiv: 1805.08318.
- [43] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [44] A. Conneau *et al.*, "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data," *arXiv:1705.02364 [cs]*, May 2017, arXiv: 1705.02364.
- [45] A. Paszke *et al.*, "Automatic differentiation in PyTorch," Oct. 2017.
- [46] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Dec. 2014, arXiv: 1412.6980.