Learning a Guidance Policy from Humans for Social Navigation

Luzia Knoedler[1], Bruno Brito[1], Michael Everett[2], Jonathan P. How[2] and Javier Alonso-Mora[1]

***Abstract*— Autonomous mobile robots navigating among humans must not only consider safety and efficiency but also move acceptably in the current social context. A hybrid deep reinforcement learning - model predictive control (DRL-MPC) approach can account for the complex interactions among humans while maintaining the collision avoidance guarantees and feasibility constraints inherent in the MPC formulation. However, encoding socially acceptable behavior through a reward or cost function, along with other objectives such as reaching the goal quickly, is challenging. Therefore, this work proposes a new training strategy that combines supervised and reinforcement learning to exploit human demonstration. Furthermore, it presents first results from real-world experiments.**

I. INTRODUCTION

Autonomous mobile robots coexisting and collaborating with humans must not only consider how to efficiently and safely reach their goal but also which behavior is acceptable in the current social context. Thus, finding collision-free, time efficient paths around humans is not enough. To improve the robot's acceptance the research on social navigation addresses three additional aspects (i) *Comfort*, which extends the concept of safety to the feeling of safety (ii) *Naturalness*, which refers to the similarity between the robot's and the humans' low-level behavior, and (iii) *Sociability*, which describes the adherence to high-level cultural conventions, e.g., passing on the right side [1].

However, deriving collision-free paths among humans itself remains challenging due to the unknown intents of the other agents and the complex interaction effects that arise among them. Extending the navigation approaches to account for comfort, naturalness, and sociability adds another challenge. Therefore, our work focuses on ensuring collision-free and thus safe behaviors while addressing the concepts of social navigation by learning from human demonstrations.

A common approach for collision avoidance among non-communicating, decision-making agents is predictive motion planning, e.g., model predictive control (MPC), which enables smooth collision avoidance by exploiting predictions of the other agents' trajectories and can provide collision avoidance guarantees [2]. However, to enable online planning the coupling between the agents' behavior is not considered which can result in the freezing robot problem [3]. Thus, many approaches have used deep reinforcement learning (DRL) to model the complex interactions among humans and offload the costly computations to an offline training phase [4], [5], [6]. While data-driven approaches can account for interactions among humans, they cannot guarantee to respect kinodynamic and collision avoidance constraints. Therefore, recent approaches combine DRL with local optimization techniques to benefit from both classes of methods. In the context of crowd navigation, our previous work [7] introduced Goal Oriented Model Predictive Control (GO-MPC) which enhances MPC with a learned global guidance policy. It was shown in simulation that GO-MPC improves the average time-to-goal and success rate by leveraging past experience in crowded situations. Yet, GO-MPC was never tested in the real world. Moreover, in simulation it was observed that the robot maintained a low distance to the other agents and learned to exploit the cooperation of the other agents. To achieve more social behaviors the reward or cost function can be adapted, but this is nontrivial. One approach to address this issue is to capture the comfort, naturalness and sociability aspects inherent in human navigation patterns by learning from human demonstrations.

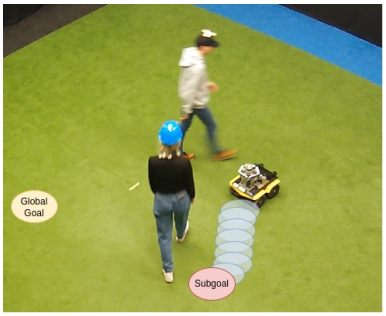


Fig. 1: Robot navigating autonomously among humans. A learned guidance policy provides subgoals to a local trajectory optimization method.

This work

(i) proposes a new training strategy combining reinforcement and supervised learning exploiting human demonstrations to learn more socially acceptable behaviors, and

(ii) demonstrates the application of GO-MPC on a real robot among humans.

II. RELATED WORK

Several state-of-the-art collision avoidance methods employ MPC with online optimization to compute motion plans that are guaranteed to respect kinodynamic and collision avoidance constraints [2]. Here, the navigation problem is typically divided into two successive steps for prediction

*This work was not supported by any organization

[1]The authors are with the Cognitive Robotics (CoR) department, Delft University of Technology, 2628 CD Delft, The Netherlands {l.knoedler, bruno.debrito j.alonsomora}@tudelft.nl

[2]The authors are with Massachusetts Institute of Technology, Aerospace Controls Laboratory, Cambridge, MA, USA. {mfe, jhow}@mit.edu

and planning. However, this can result in the freezing robot problem, during which the robot cannot find safe paths [3]. A key to resolving the freezing robot problem is to account for interactions between agents and the resulting cooperation [3]. Interaction effects can be considered by jointly planning feasible paths for all agents in the environment using, for instance, game-theoretic approaches [8], [9] or interactive Gaussian Processes [10]. However, disadvantages of these methods are their computational expensiveness and the unavailability of the other agents' intents. Thus, most works exploit learning-based methods to model interactions between agents and delegate costly computations to an offline training phase [4], [5], [6]. Recently, there has been increasing interest in combining DRL with local optimization techniques to benefit from both classes of methods [11], [12], [13].

Besides collision avoidance robot acceptance requires to address the comfort, naturalness and sociability aspects of navigation. A major drawback of RL approaches is the need for a sophisticated reward function that quantifies socially acceptable behavior. Although quantifying socially acceptable behavior might seem intuitive to humans, finding a reward or cost function is not trivial. Furthermore, RL approaches generally train the policy in simulation, since exploring in a real-world environment is expensive and unsafe. This requires realistic models of human behaviors. On the contrary, imitation learning approaches learn social navigation behavior from human demonstrations. For example, by utilizing behavior cloning (BC) [14], inverse reinforcement learning (IRL) [15], [16], [17] or adversarial imitation learning [18]. Disadvantages of these approaches are that they require expert data and thus expensive real-world data collection and that human demonstrations rarely contain failure cases and thus the expertise of the policy is very limited in critical situations.
Therefore, we propose a hybrid BC-RL approach that takes advantage of human demonstrations to warm start the RL training phase.

III. Preliminaries

A. Problem Formulation

Consider a scenario where a mobile robot, hereinafter referred to as the ego-agent, must navigate from an initial position $\mathbf{p}_0$ to a goal position $\mathbf{g}$ in an environment populated by n humans. For each agent $i \in \{0, \dots, n\}$ in the environment, where $i = 0$ refers to the robot, $\mathbf{p}^i \in \mathbb{R}^2$ denotes its position and $\mathbf{v}^i \in \mathbb{R}^2$ its velocity. The area occupied by each agent is indicated as $\mathcal{O}^i$. Hereafter, we drop the upper-script when referring to the ego-agent.

At each time step k, the ego-agent observes its state $\mathbf{s}_k$ and the set of surrounding agents' states $\mathbf{S}_k = \bigcup_{i\in\{1,\dots,n\}} \mathbf{s}_k^i$. Then, the ego-agent takes action $\mathbf{a}_k$, receives reward $R(\mathbf{s}_k, \mathbf{a}_k)$ and observes its next state $\mathbf{s}_{k+1} = h(\mathbf{s}_k, \mathbf{a}_k)$, under the transition model h. Here, we consider a partially observable setting, in which only the position and velocity of the other agents are available, but information about their goal position are assumed unknown.

The goal is to find a policy π maximizing the cumulative rewards over time to navigate from $\mathbf{p}_0$ to $\mathbf{g}$ while satisfying the ego-agent's dynamics and the static and dynamic collision avoidance constraints. The policy can be defined as the following optimization problem:

$$
\begin{aligned}
\pi^* = \underset{\pi}{\operatorname{argmax}} \quad & \mathbb{E}\left[\sum_{k=0}^{T} \gamma^k R(\mathbf{s}_k, \pi(\mathbf{s}_k, \mathbf{S}_k)\right] \\
\text{s.t.} \quad & \mathbf{x}_{k+1} = f(\mathbf{x}_k, \boldsymbol{u}_k), && \text{(1a)} \\
& \boldsymbol{s}_T = \mathbf{g}, && \text{(1b)} \\
& \mathcal{O}_k(\mathbf{x}_k) \cap (\mathcal{O}_k^i) = \emptyset && \text{(1c)} \\
& \mathbf{u}_k \in \mathcal{U},\ \mathbf{s}_k \in \mathcal{S},\ \mathbf{x}_k \in \mathcal{X}, && \text{(1d)} \\
& \forall k \in [0, T], \quad \forall i \in \{1, \dots, n\}
\end{aligned}
$$

where $\mathbf{s}$ denotes the state used in the RL formulation, $\mathbf{x}, \mathbf{u}$ denote the control state and action commands used in the optimization problem, Eq. (1a) represents the transition dynamic constraints under the dynamic model of the robot f, Eq. (1b) the terminal constraints, and Eq. (1c) formalizes the collision avoidance constraints. We denote $\mathcal{U}$, $\mathcal{S}$ and $\mathcal{X}$, Eq. (1d), as the corresponding set of admissible control inputs, states and control states, respectively.
We assume that the other agents have various behaviors which are defined in Section V-A.

B. Goal Oriented Model Predictive Control (GO-MPC)

GO-MPC addresses the optimization problem of Eq. (1) using a two-level planning architecture, consisting of a global guidance policy $\pi_\theta : (\mathbf{s}_k, \mathbf{S}_k) \rightarrow \mathbf{a}_k$ and an optimization-based motion planner. The action of the guidance policy $\mathbf{a}_k$ is defined as a position increment $\boldsymbol{\delta}_k$ providing the direction maximizing the ego-agent rewards from which a subgoal position is computed

$$
\begin{aligned}
\mathbf{p}_k^{\text{ref}} &= \mathbf{p}_k + \boldsymbol{\delta}_k && \text{(2a)} \\
\pi_\theta(\mathbf{s}_k, \mathbf{S}_k) = \mathbf{a}_k &= \boldsymbol{\delta}_k = [\boldsymbol{\delta}_{k,x}, \boldsymbol{\delta}_{k,y}]. && \text{(2b)}
\end{aligned}
$$

The recommended subgoal $\mathbf{p}_k^{\text{ref}}$ is utilized in the MPC cost function which ensures that dynamic feasibility and collision avoidance constraints are satisfied when a feasible solution to the optimization problem is found. For more details, we refer to [7].

IV. Method

This section presents a framework enhancing local trajectory optimization methods with a learned policy that provides global guidance and induces socially acceptable robot behaviors. We build upon [7] and propose an important expansion to the method. We introduce a new training strategy that combines RL and supervised learning exploiting human demonstrations to learn socially acceptable guidance policies. The applied network architecture is presented in Fig. 2

We aim to learn a socially compliant subgoal policy providing global guidance to a local optimization planner. To induce socially acceptable behavior, we propose to use a

Algorithm 1: Supervised PPO-MPC Training

1: **Inputs:** value function and policy's parameters $\{\theta^V, \theta\}$, number of supervised and RL training episodes $\{n_{\text{warm}}, n_{\text{episodes}}\}$, number of agents n, mini-batch size $n_{\text{mini-batch}}$, and reward function $R(\mathbf{s}_t, \mathbf{a}_t, \mathbf{a}_{t+1})$, loss weighting factor γ and weighting factor decay
2: Initialize scenario: $\{\mathbf{s}_0^0, \ldots, \mathbf{s}_0^n\} \sim \mathcal{S}$, $\{\mathbf{g}^0, \ldots, \mathbf{g}^n\} \sim \mathcal{S}$
3: **while** $episode < n_{\text{episodes}}$ **do**
4: Initialize $\mathcal{B} \leftarrow \emptyset$, $h_0 \leftarrow \emptyset$, $k = 0$
5: **for** $j = 0, \ldots, n_{\text{mini-batch}}$ **do**
6: **if** $episode \leq n_{\text{warm}}$ **then**
7: $\mathbf{p}_k^{\text{ref}} = \mathbf{p}_k^{\text{ref,h}}$
8: **else**
9: $\mathbf{p}_k^{\text{ref}} = \pi_\theta(\mathbf{s}_k, \mathbf{S}_k)$
10: **end if**
11: Set $\mathbf{a}_k^* = \mathbf{p}_k^{\text{ref}} - \mathbf{p}_k$
12: Solve MPC problem (see [7])
13: $\{\mathbf{s}_k, \mathbf{a}_k, r_k, \mathbf{s}_{k+1}, \text{done}\} = \text{Step}(\mathbf{s}_k^*, \mathbf{a}_k^*)$
14: Store $\mathcal{B} \leftarrow \{\mathbf{s}_k, \mathbf{a}_k, r_k, \mathbf{s}_{k+1}, \text{done}\}$
15: **if** done **then**
16: $episode \mathrel{+}= 1$, Initialize scenario, Set: $k = 0$
17: **end if**
18: **end for**
19: **if** $episode \leq n_{\text{warm}}$ **then**
20: Supervised training: Eq. (3a) and Eq. (3b)
21: **else**
22: RL combining PPO loss [19] with supervised loss L^{SV}
23: **end if**
24: $\gamma* = \text{decay}$
25: **end while**
26: **return** $\{\theta^V, \theta\}$

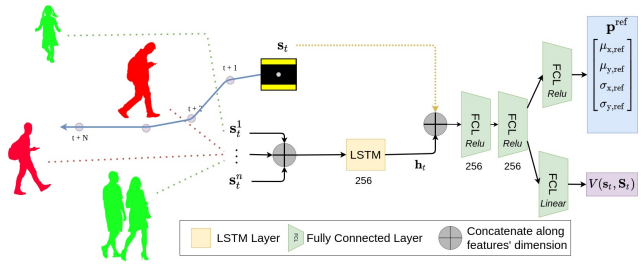


Fig. 2: Applied network architecture for the global guidance policy. It maps the ego-agent's environment observation to a probability distribution of position increments and estimates of the state-value function. The environment might contain cooperative agents (green) and non-cooperative agents (red).

human teacher to warm-start the subgoal policy. This is in contrast to [7] which uses the MPC as an expert. Furthermore, we introduce a supervised loss in the RL loss function. Algorithm 1 presents the overall training strategy that incorporates two main phases: warm-start and RL training. To induce learning a socially compliant policy, we learn the policy parameters using human expert demonstrations during the first n_{warm} steps. At each time step, the human expert provides a subgoal position, $\mathbf{p}^{\text{ref, h}}$ (Line 6-8). Then, the MPC computes the control command minimizing the robot's distance to the provided subgoal. For each time step, we store the tuple containing the state, action, reward and next state in a buffer $\mathcal{B} \leftarrow \{\mathbf{s}_k, \mathbf{a}_k^*, r_k, \mathbf{s}_{k+1}\}$, which is used at the end of the rollout phase to compute advantage estimates and perform a supervised training step. Please note that the policy's action is a position increment providing the direction maximizing the ego-agent rewards, $\mathbf{a}_k^* = \mathbf{p}_k^{\text{ref, h}} - \mathbf{p}_k$. During the warm-start phase we update the value function and policy's parameters as follows:

$$\theta_{k+1}^V = \underset{\theta^V}{\text{argmin}}\, \mathbb{E}_{(\mathbf{a}_k, \mathbf{s}_k, r_k) \sim \mathcal{D}_{\text{h}}} \left\| V_\theta(\mathbf{s}_k) - V_k^{\text{targ}} \right\| \tag{3a}$$

$$\theta_{k+1} = \underset{\theta}{\text{argmin}}\, \mathbb{E}_{(\mathbf{a}_k, \mathbf{s}_k) \sim \mathcal{D}_{\text{h}}} \left\| \mathbf{a}_k^* - \pi_\theta(s_k) \right\| \tag{3b}$$

where $\mathcal{D}_{\text{h}}$ is the set of human demonstrations, θ^V and θ are the value function and policy parameters, respectively. Afterwards, we employ an on-policy gradient method to update our policy's parameters (Line 22). Specifically, we use the Proximal Policy Optimization (PPO) method [19] which we extend by a supervised loss term

$$L^{PPO+SV} = \gamma L^{SV} + (1 - \gamma) L^{PPO}, \tag{4}$$

where $L^{SV} = -\log \pi_\theta(\boldsymbol{a}_k^* | \boldsymbol{s_k}, \boldsymbol{S_k})$ is the Negative Log-likelihood probability of the human actions. The weighting factor γ decreases over training steps.

V. Results

We apply the same computational settings as described in [7] and use the ForcesPro [20] solver to solve the non-linear and non-convex MPC problem. The implementation of the proposed training algorithm builds upon the open-source PPO implementation provided in the Stable-Baselines [21] package. For the used hyperparameters we refer to [7].

A. Training Scenarios

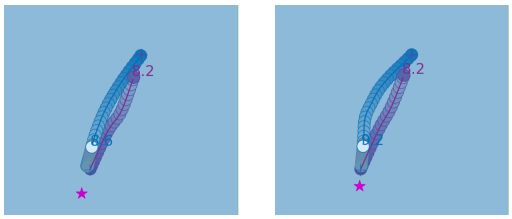
To train and evaluate our method we have selected three navigation scenarios as described in [7]. Each training episode consists of a random number of agents $n \leq n_{\max}$ and a random scenario. At the start of each episode, each other agent's policy is sampled from a binomial distribution (80% cooperative, 20% non-cooperative). The cooperative humans employ the Social Forces Model (SFM) [22] and the non-cooperative humans move with a constant velocity towards their goal. Moreover, for the cooperative agents we randomly sample a weight on the effect of inter-pedestrian interactions $c_{\text{social}} \sim \mathcal{U}(2, 10)$.
We employ a second-order unicycle model for the ego-agent [23]. During the warm-start phase a human expert provides the subgoal to the MPC using a Logitech F710 controller. A maximum number of $n_{\max,\text{h}} \leq n_{\max}$ is considered to reduce the overstrain the human.

B. Simulation Results

This section studies the effect of the proposed training procedure on the navigation behavior. We present qualitative results and compare the policy variants quantitatively over 200 random scenarios. Example trajectories are visualized in Fig. 3 and Fig. 4. The numerical results are summarized in Table I. To evaluate the socialness on top of the safety and efficiency aspects we compute the mean traveled distance of the other agents.
We observed that although we do not include any social norms in the reward function, the policy learned to pass on

TABLE I: Performance over 200 episodes of GO-MPC without human demonstrations and GO-MPC with human demonstrations. Traveled distance is displayed only for the successful episodes.

	Mean Traveled Distance Ego Agent [m]			Mean Traveled Distance Other Agents [m]			# collisions			# deadlocks		
# agents	2	4	6	2	4	6	2	4	6	2	4	6
Without human demos	7.17	6.16	6.59	4.36	10.80	21.55	0	0	0	1	2	3
With human demos	7.74	7.07	9.31	5.29	14.43	28.47	0	0	1	9	16	14



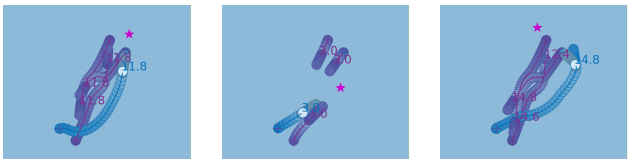
(a) Without human demos (b) With human demos

Fig. 3: Comparison of trajectories for training without [7] and with human demonstrations. The ego-agent applying GO-MPC (blue) moving from top right to bottom left and the cooperative agent (purple) swap positions.

the right side since the SFM includes a preference to pass on the right. This shows that including realistic models of human behavior in the training environment is important for learning socially acceptable navigation behaviors.
By including human demonstrations (supervised learning on 4097 transitions), we were able to influence the behavior of the resulting policy. As it can be seen in Fig. 3 the ego-agent learned to keep a larger distance to the other agent. However, including human demonstrations did not result in a significant decrease in the traveled distance of the ego agent or the traveled distances of the other agents. This can be explained by the intention of the considered human expert to increase the distance towards the other agents. Compared to the GO-MPC trained without human demonstrations, the number of deadlocks increased when using human demonstrations. Please note that the human expert was only exposed to a maximum number of $n_{\max,\mathrm{h}} = 5$ agents. We expect that adapting the decay factor and thus putting a higher weight on the PPO loss will decrease the number of deadlocks. Nevertheless, we were able to retain knowledge about the human demonstrations after the RL phase. This can be seen in Fig. 4 which compares the behavior of the ego-agent trained without and with human demonstrations against the true human demonstration in an unseen scenario. We note that further metrics need to be determined to evaluate the various aspects of socially acceptable behaviors including comfort, naturalness and sociability.

C. Hardware Experiment

We implement the GO-MPC policy on a ground robot to demonstrate its behavior among real humans and compare its performance against the MPC policy. During the experiment, we observed that the MPC policy tends to be more passive,



(a) Human Demo (b) Without demos (c) With demos

Fig. 4: Evaluation on unseen scenarios. The ego-agent applying GO-MPC (blue) and three cooperative agents (purple)

(a) MPC (b) GO-MPC

Fig. 5: Quantitative comparison real-world behavior

resulting in it waiting for the humans to pass, see black circle in Fig. 5, while the GO-MPC policy follows smooth paths towards the global goal. In future work we will apply our approach in more challenging scenarios and perform further analysis to quantify the performance improvement between different policies.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a framework to enhance local trajectory optimization methods with a learned policy providing global guidance and inducing socially acceptable robot behavior. In particular, we introduced a new training strategy combining supervised and reinforcement learning (RL) exploiting human demonstrations to train the global guidance policy. We observed that the behavior of the policy can be influenced by learning from human demonstrations. However, it remains to be evaluated to what extend the resulting behavior addresses the comfort, naturalness and sociability aspects inherent in social navigation.

ACKNOWLEDGMENT

This paper has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101017008. All content represents the opinion of the author(s), which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] 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, 2013.

[2] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[3] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, 2015.

[4] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6015–6022.

[5] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3052–3059.

[6] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 856–892, 2020.

[7] B. Brito, M. Everett, J. P. How, and J. Alonso-Mora, "Where to go next: Learning a subgoal recommendation policy for navigation in dynamic environments," 2021.

[8] S. L. Cleac'h, M. Schwager, and Z. Manchester, "Algames: A fast solver for constrained dynamic games," *arXiv preprint arXiv:1910.09713*, 2019.

[9] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin, "Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 1475–1481.

[10] 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*. IEEE, 2013, pp. 2153–2160.

[11] T. Wang and J. Ba, "Exploring model-based planning with policy networks," in *International Conference on Learning Representations*, 2019.

[12] C. Greatwood and A. G. Richards, "Reinforcement learning and model predictive control for robust embedded quadrotor guidance and control," *Autonomous Robots*, vol. 43, no. 7, pp. 1681–1693, 2019.

[13] Z.-W. Hong, J. Pajarinen, and J. Peters, "Model-based lookahead reinforcement learning," 2019.

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

[15] B. Okal and K. O. Arras, "Learning socially normative robot navigation behaviors with bayesian inverse reinforcement learning," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2889–2895.

[16] 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, 2016.

[17] D. Vasquez, B. Okal, and K. O. Arras, "Inverse reinforcement learning algorithms and features for robot navigation in crowds: an experimental comparison," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1341–1346.

[18] C.-E. Tsai and J. Oh, "A generative approach for socially compliant navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2160–2166.

[19] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[20] A. Domahidi and J. Jerez, "FORCES Professional," embotech GmbH (http://embotech.com/FORCES-Pro), Jul. 2014.

[21] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu, "Stable baselines," 2018.

[22] M. Moussaïd, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz, "Experimental study of the behavioural mechanisms underlying self-organization in human crowds," *Proceedings of the Royal Society B: Biological Sciences*, vol. 276, no. 1668, pp. 2755–2762, 2009.

[23] S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006. [Online]. Available: http://planning.cs.uiuc.edu/