Towards Safe and Socially Compliant Map-less Navigation by Leveraging Prior Demonstrations

Shiqing Wei, Xuelei Chen, Xiaoyuan Zhang, and Chenkun Qi*

Shanghai Jiao Tong University,
800 Dongchuan Rd., Shanghai, China,
{weishiqing, chenkqi}@sjtu.edu.cn

Abstract. This paper presents a learning-based approach for safe and socially compliant map-less navigation in dynamic environments. Our approach maps directly 2D-laser range findings and other measurements to motion commands, and a combination of imitation learning and reinforcement learning is deployed. We show that, by leveraging prior demonstrations, the training time for RL can be reduced by 60% and its performance is greatly improved. We use Constrained Policy Optimization (CPO) and specially designed rewards so that a safe and socially compliant behavior is achieved. Experiment results prove that the obtained navigation policy is capable of generalizing to unseen dynamic scenarios.

Keywords: Mobile robot, navigation, deep reinforcement learning, end-to-end motion planning.

1 Introduction

One of the main challenges in the field of ground robot navigation is to enable robots to move around autonomously and intelligently. In environments where global knowledge of the map is known, navigation algorithms are now well studied [1], and optimization goals such as shortest travel path or minimal travel time can also be applied [2]. However, exploration of unknown environments remains a common problem for tasks such as search, rescue, mining, etc. Particularly, in rapidly changing environments, e.g., social scenarios with the presence of people, it can be very difficult to get a reliable information of the global environment and robots have to navigate merely based on their local perception of the environment. Thus, map-less navigation strategies are required.

Learning-based methods prove to be very suitable for this kind of real-time end-to-end navigation problems, and can be divided into two broad categories. The imitation learning (IL) based ones are trained on expert demonstrations and try to mimic the behavior of the expert, while the reinforcement learning (RL)

* Corresponding author.

** This work was funded by the National Key Research and Development Plan of China (2017YFE0112200).

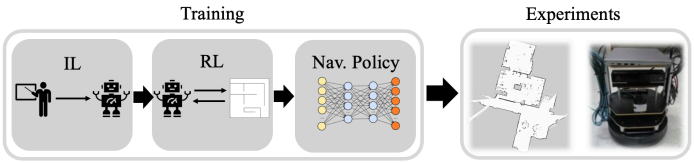


Fig. 1. A safe and socially compliant navigation policy is obtained by combining imitation learning and reinforcement learning, and then tested in simulations.

based ones employ the trial-and-error strategy to learn their navigation policy with the help of reward signals. IL is sample efficient, but usually has weak generalization abilities and a tendency of overfitting, and the learned navigation policy is limited by the quality of expert demonstrations. On the contrary, RL is theoretically more robust to unseen scenarios, since agents learn to act by trial and error, gradually improving their performance at the task as learning progresses. Still, weaknesses of RL include sample inefficiency and high time consumption, as training data are forward simulated. However, RL offers the possibility to encode desired behaviors through reward or constraint design, which makes it a conceptually better alternative to end-to-end motion planning.

In this work, we focus on the problem of navigating an unknown environment to a goal position in a *safe* and *socially compliant* way. Safety is of crucial significance for mobile robot navigation, as collision could do damage to both pedestrians and robots. In the meanwhile, in pedestrian-rich environments, robots are additionally required to understand and comply with mutually accepted rules, i.e., behave in socially compliant manners to better interact with people [3]. To this end, we adopt an approach that combines the advantages of IL and RL: the navigation policy is pre-trained by a supervised IL and then subsequently trained by RL. We use Constrained Policy Optimization (CPO)[4] for RL because of its ability to incorporate constraints during training.

In summary, the main contributions of this paper are:

- an effective deep RL-based approach for safe and socially compliant navigation through raw laser range findings and other measurements
- a case study for combining IL and RL for map-less navigation
- deployment and tests on a simulated robotic platform in unknown dynamic environments

2 Related Work

Imitation Learning Based Methods: IL, also known as behavior cloning (in the narrow sense), takes expert demonstrations as training data and directly learn the navigation policy. Tai *et al.* [5] achieve a model-less obstacle avoidance behavior by using a convolutional neural network (CNN) and taking raw depth images as input, but their approach can only generate discrete steering

commands, limiting their application to a continuous state problem. Pfeiffer *et al.* [6] and Sergeant *et al.* [7] adopt an end-to-end approach mapping 2D laser range findings to control commands. Other methods using inverse reinforcement learning (IRL) ([3], [8]) and generative adversarial imitation learning (GAIL) ([9], [10]) have also occurred in recent years.

Reinforcement Learning Based Methods: Zhu *et al.* [11] apply a deep siamese actor-critic model based on deep RL to target-driven visual navigation. Li *et al.* [12] propose an approach incorporating a neural network to learn an exploration strategy to extend a continuously updated map. Zhele *et al.* [13] augment the normal external reward for deep RL algorithms with an additional term in function of curiosity. However, these approaches mentioned above remain basically in the range of static environments and do not deal with dynamic obstacles. Long *et al.* [14] use Proximal Policy Optimization (PPO) and obtain a decentralized collision avoidance policy in a multi-robot situation. Lütjens *et al.* [15] embed MC-Dropout and Bootstrapping in a RL framework to achieve uncertainty estimates and uncertainty-aware navigation around pedestrians. These methods are primarily collision avoidance strategies. Although they address motion planning in dynamic environments, their navigation capabilities are limited.

Socially Compliant Navigation: Simplistic approaches that treat pedestrians as dynamic obstacles with simple kinematics often generate unnatural robot behaviors [3], while predictive approaches that reason about nearby pedestrians' hidden intents can lead to the freezing robot problem once the environment surpasses a certain level of dynamic complexity [16]. One possible solution to this problem is to anticipate the impact of the robot's motion on nearby pedestrians. Existing work on socially compliant navigation can be split into model-based ([17], [18]) and learning-based ([19], [20]) approaches.

In this work, we introduce a deep RL based method that uses IL as pretraining. The expert demonstrations are generated by Timed-Elastic-Bands (TEB) [2] and a navigation policy mapping the raw measurements to motion commands is learned. This temporary navigation policy is rudimentary, and will be improved by the subsequent deep RL. What's more, we design specific rewards to encourage human-like navigation through cooperative collision avoidance, thus addressing the problem of socially compliant navigation

3 Approach

3.1 Problem Formulation

Socially compliant navigation can be formulated as a Markov Decision Process (MDP). Let $\mathbf{s}_t$, $\mathbf{a}_t$ denote an agent's states and action at time t. Each agent has a current position $\mathbf{p} = [p_x, p_y]$, a current velocity $\mathbf{v} = [v_x, v_y]$ and a collision radius r, and they form the agent's external information $\mathbf{y}^{ext}$. Furthermore, each

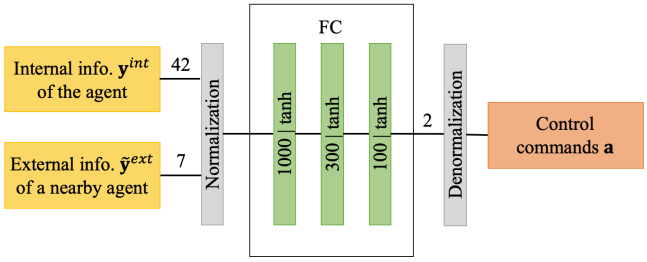


Fig. 2. Network structure of the policy $\pi_{\boldsymbol{\theta}}$.

agent has its goal position $\mathbf{p}_g = [p_{gx}, p_{gy}]$, orientation ψ (impossible to observe since agents are seen as discs) and laser range findings $\mathbf{r}_{laser}$, and they form the agent's internal information $\mathbf{y}^{int}$. To reduce redundancy, we reparameterize the internal information of a given agent $\mathbf{y}^{int}$ and external information of another agent $\tilde{\mathbf{y}}^{ext}$ in the local frame of the given agent, with the x-axis pointing to the front and y-axis to the left:

$$\mathbf{y}^{int} = [d_g, \psi_g, v_x, v_y, \psi, r, \mathbf{r}_{laser}], \tag{1}$$

$$\tilde{\mathbf{y}}^{ext} = [\tilde{d}_a, \tilde{p}_x, \tilde{p}_y, \tilde{v}_x, \tilde{v}_y, \tilde{r}, \tilde{\phi}], \tag{2}$$

where d_g is the agent's Euclidean distance to goal, ψ_g is the agent's relative orientation to goal, $\tilde{d}_a$ is the Euclidean distance between the two agents and $\tilde{\phi} = \arctan(\tilde{v}_y/\tilde{v}_x)$ is the nearby agent's heading direction. Thus, the states of the MDP are $\mathbf{s} = [\mathbf{y}^{int}, \tilde{\mathbf{y}}^{ext}]$[1]. We want to find a navigation policy $\pi_{\boldsymbol{\theta}}$ parameterized by $\boldsymbol{\theta}$, which maps $\mathbf{s}$ to a motion control command $\mathbf{a}$:

$$\mathbf{a} = \pi_{\boldsymbol{\theta}}(\mathbf{s}). \tag{3}$$

The control command $\mathbf{a}$ is composed of the translational and rotational velocities. Since the mapping from the states and the desired control commands can be really complicated, a neural network representation of the navigation policy shows great potential.

3.2 Neural Network Model and Pre-training

In this work, we combine IL and RL to obtain a robust navigation. The neural network representation of $\pi_{\boldsymbol{\theta}}$ is shown in Fig. 2. Compared with [5] and [6], where a convolutional neural network is deployed to extract environmental features, our model is simplified but still adequate. Unlike image-based methods which often have to deal with virtual-to-real problems, our method uses 2D laser range

[1] Here we only consider a two-agent collision avoidance problem, but the resulting two-agent navigation policy is parallelizable in a multiagent scenario, because it consists of a large number of independent queries of the trained neural network.

findings and can be easily implemented on a real-world robotic platform. We use minimum pooling to downsample the laser range data and reduce input dimensions. In our case, $\mathbf{r}_{laser} \in \mathbb{R}^{36}$, i.e., we keep the minima out of 36 equally divided intervals of laser range data. The inputs $\mathbf{s}$ are normalized to $[-1, 1]$, and the outputs of the neural network are denormalized to obtain actual control commands.

In the hope of improving RL performance and reducing training time, we use supervised IL, also called behavior cloning, to pre-train the navigation policy. This is similarly done as in [5] and [6], and the resulting policy of IL will continue to be ameliorated in the subsequent RL.

3.3 Reinforcement Learning

Background: A Markov Decision Process (MDP) can be defined as a tuple (S, A, R, P, μ), where S is the set of states, A is the set of actions, $R : S \times A \times S \to \mathbb{R}$ is the reward function, $P : S \times A \times S \to [0, 1]$ is the transition probability function and $\mu : S \to [0, 1]$ is the starting state distribution. A stationary policy $\pi : S \to \mathcal{P}(A)$ is a map from states to probability distribution over actions. In RL, we aim to find a policy $\pi_{\boldsymbol{\theta}}$ parameterized by $\boldsymbol{\theta}$ which maximizes the expected discounted return of reward $J(\boldsymbol{\theta})$, i.e.,

$$J(\boldsymbol{\theta}) = \underset{\tau \sim \pi_{\boldsymbol{\theta}}}{E}[\sum_{t=0}^{T} \gamma^t R(s_t, a_t, s_{t+1})], \tag{4}$$

where T is the time horizon of an episode, $\gamma \in [0, 1)$ is the discount factor, τ denotes a trajectory ($\tau = (s_0, a_0, s_1, ...)$), and $\tau \sim \pi_{\boldsymbol{\theta}}$ indicates that the distribution over trajectories depends on $\pi_{\boldsymbol{\theta}}$.

Trust Region Policy Optimization (TRPO) has recently gained much attention among policy optimization algorithms because of its ability to avoid the problem of high gradient variance commonly seen in policy gradient methods and guarantee monotonic improvement. However, when a system involves physical interaction with or around humans, it is indispensable to define and satisfy safety constraints. Therefore, we introduce Constrained Policy Optimization (CPO), a constrained version of TRPO, and applies it to obtain a safe and socially compliant navigation policy.

We augment the MDP with a an auxiliary cost function $C : S \times A \times S \to \mathbb{R}$ and a limit d. This augmented version of MDP is called Constrained Markov Decision Process (CMDP). Let $J_C(\boldsymbol{\theta})$ denote the expected discounted return of policy $\pi_{\boldsymbol{\theta}}$ with respect to cost function C:

$$J_C(\boldsymbol{\theta}) = \underset{\tau \sim \pi_{\boldsymbol{\theta}}}{E}[\sum_{t=0}^{T} \gamma^t C(s_t, a_t, s_{t+1})]. \tag{5}$$

CPO finds the optimal policy $\pi_{\boldsymbol{\theta}^\star}$, i.e.,

$$\boldsymbol{\theta}^\star = \arg\max \quad J(\boldsymbol{\theta}) \tag{6}$$

$$\text{s.t.} \quad J_C(\boldsymbol{\theta}) \leq d \tag{7}$$

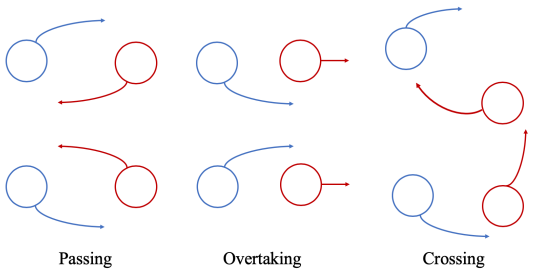


Fig. 3. Left-handed (top row) and right-handed (bottom row) rules for the blue agent to pass, overtake and cross the red agent.

Reward and Cost Design: The agent aims to reach the target position while learning to confirm to social norms and avoiding collisions with the environment or other agents. The design of the reward is crucial, since the reward function gives feedback to the agent so that desired behaviors can be learned. In our case, the overall reward function consists of two parts: the navigation-inducing and the norm-inducing rewards, i.e.,

$$R = R_{nav} + R_{norm} \tag{8}$$

To avoid the problem of sparse reward, we define the navigation-inducing reward as

$$R_{nav}(\mathbf{s}_t) = \begin{cases} 10, & \text{if goal reached} \\ -\alpha(d_{g,t} - d_{g,t-1}), & \text{otherwise,} \end{cases} \tag{9}$$

where $d_{g,t}$ and $d_{g,t-1}$ are lengths of the agent's shortest path to goal position at timestamps t and $t-1$, and α is a positive scaling parameter ($\alpha = 1$). In this way, we guarantee that the agent receives a feedback at all states, and that there is a significant increase in total reward when the target is reached.

As shown in Fig. 3, there are symmetries in a collision avoidance scenario. To induce particular social norms, an additional reward can be introduced in the learning process so that one set of social norms is preferred over the other. In this work, we choose to encourage the right-handed rules, and the reward function R_{norm} is specified as follows,

$$R_{norm}(\mathbf{s}) = \beta I(\mathbf{s} \in \mathcal{S}_{norm}) \tag{10}$$

$$\text{s.t.} \quad \mathcal{S}_{norm} = \mathcal{S}_{pass} \cup \mathcal{S}_{overtake} \cup \mathcal{S}_{cross} \tag{11}$$

$$\mathcal{S}_{pass} = \{\mathbf{s} \mid d_g > 3,\ 1 < \tilde{p}_x < 4, 0 < \tilde{p}_y < 2,\ |\tilde{\phi} - \psi| > 3\pi/4\} \tag{12}$$

$$\mathcal{S}_{overtake} = \{\mathbf{s} \mid d_g > 3,\ 0 < \tilde{p}_x < 3,\ |\mathbf{v}| > |\tilde{\mathbf{v}}|, -1 < \tilde{p}_y < 0,\ |\tilde{\phi} - \psi| < \pi/4\} \tag{13}$$

$$\mathcal{S}_{cross} = \{\mathbf{s} \mid d_g > 3,\ \tilde{d}_a < 2,\ 0 < \tilde{p}_y < 2,\ \pi/2 < |\tilde{\phi} - \psi| < \pi\}, \tag{14}$$

where β is a positive parameter ($\beta = 0.5$), $I(\cdot)$ is the indicator function, and $\tilde{\phi} - \psi$ is wrapped in $[-\pi, \pi]$.

Algorithm 1 CPO with Two Agents

Initialize the policy network $\pi_{\boldsymbol{\theta}}$ and value network V randomly from a normal distribution or use pre-trained weights from IL.
for epoch $= 1, 2, ...$ **do**
  // Collect data in parallel
  for agent $i = 1, 2$ **do**
    Run policy $\pi_{\boldsymbol{\theta}}$ for T_i timesteps, and collect $\{\mathbf{s}_{i,t}, R_{i,t}, \mathbf{a}_{i,t}, C(\mathbf{s}_{i,t})\}$, where $t \in [0, T_i]$
    Compute advantages and safety cost using GAE [21]
  end for
  Update policy $\pi_{\boldsymbol{\theta}}$ using CPO
  Update value network V using Conjugate Gradient Decent
end for
return $\pi_{\boldsymbol{\theta}}$, V

Enforcing negative rewards to discourage collisions has been a common practice for RL-based navigation methods as in [14]. However, this penalty-oriented approach could lead to undesirable trade-offs between policy exploration and policy performance in the training process. Luckily, in CMDPs, we can impose a constraint on the average number of collisions per episode. Let $S_{collision} \in S$ denote the set of all collision states, a cost function C can be defined as

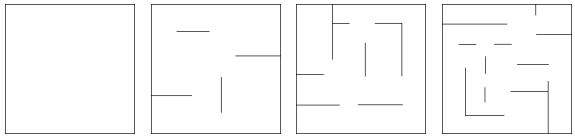
$$C(\mathbf{s}) = I(\mathbf{s} \in S_{collision}). \tag{15}$$

We set the constraint value d to 0.1, so that we can expect a very safe navigation policy after the learning process.

Training: A value network V is trained at the same time as the navigation policy $\pi_{\boldsymbol{\theta}}$. Its network structure is similar to that in Fig. 2 except that its output is a value estimation for the input states. We adopt the *centralized learning, decentralized execution* paradigm in the training process. As summarized in Algorithm 1, each agent execute the policy in parallel and generate trajectories from the shared policy $\pi_{\boldsymbol{\theta}}$, and the policy $\pi_{\boldsymbol{\theta}}$ and value network V are updated at the end of each epoch. This parallel algorithm can be easily scaled to multi-agent systems ($n > 2$) and reduce the time of sample collection.

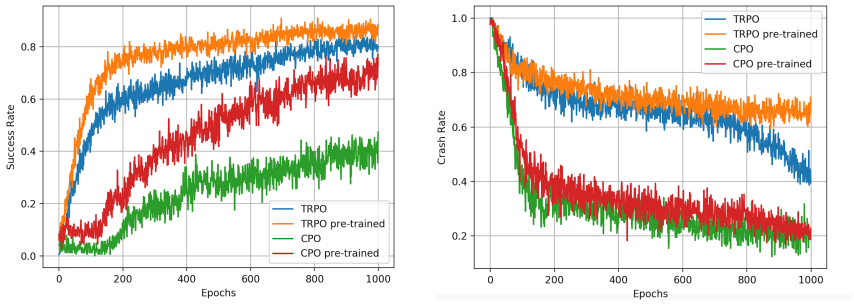
4 Experiments

This section contains training process of four different models and experiments conducted in simulations. We aim to investigate the effect of pre-training by IL and the use of safety constraint. This work is not intended to outperform a model-based planner with global knowledge of the map. We would like to show the effectiveness of RL-based method in achieving safe and socially compliant behaviors without the use of a map and its generalization ability to unseen dynamic environments.



(a) Pre-train (b) Map 1 (c) Map 2 (d) Map 3

Fig. 4. Training maps for IL and RL.



(a) Success Rate (b) Crash Rate

Fig. 5. Training results of four different models.

4.1 Model Training

We apply two different procedures to train the model[2]: (i) pure RL and (ii) a combination of IL and RL. As shown in Fig. 4, the map 4(a) is used for IL, and the maps 4(b), 4(c) and 4(d) for RL. The three training maps for RL vary in complexity in an increasing order from left to right.

In pure RL, at the beginning of each episode, a map is randomly selected among the three maps (4(b), 4(c) and 4(d)), and then a start position and a target position are also randomly generated for each agent. The motion commands are published at a frequency of 5 Hz, and a maximum of 300 motion commands is allowed per agent in each episode. The training lasts 1,000 epochs and one epoch consists of 60,000 episodes.

In the other procedure, we pre-train the navigation policy in 4(a). We put two agents in the map with one wandering randomly and the other receiving control commands from the TEB algorithm [2]. Ten trajectories including approximately 2,000 state-action pairs are sampled and fed into a supervised IL. The pre-trained navigation policy is then used to initialize RL as described above.

Fig. 5(a) and 5(b) show respectively the evolution of the percentage of successful navigation (goal reached) and that of navigation with collision(s) among

[2] The models are trained on a computer equipped with an Intel i7-8700 processor and an NVIDIA GeForce GTX 1660 GPU, running Ubuntu 16.04 and ROS Kinetic. The training is conducted in the accelerated Stage simulator with a differential drive Kobuki TurtleBot2 platform.

one epoch of four different models, and two points can be made:

Influence of the Pre-training: As shown in Fig. 5(a) and 5(b), for both TRPO and CPO based methods, the trainings that start from a pre-trained result show a better performance and a greater learning rate with a similar crash rate to their not pre-trained version. Even though the pre-training is conducted on a rather simplistic map (Fig. 4(a)), this rudimentary pre-trained navigation policy helps reduce the RL training time by 60% (if we compare the number of epochs used to reach the same success rates, e.g., 0.8 for TRPO-based methods and 0.4 for CPO-based methods), and the overall performance is improved.

Influence of Imposing a Safety Constraint: For TRPO-based methods, a fixed penalty of -0.2 is used for each collision during the training process. We can see that although the two TRPO-based methods reach a high success rate at the end of the training process, they also have a very high crash rate compared with CPO-based methods. This blind pursue of augmenting expected discounted rewards could compromise the safety of both RL training and its real-world application. On the contrary, methods using a safety constraint, the CPO-based methods, have lower crash rates. In particular, the curve *CPO Pre-trained* reaches a similarly high success rate but with a much lower crash rate.

4.2 Experiments on Navigation Ability

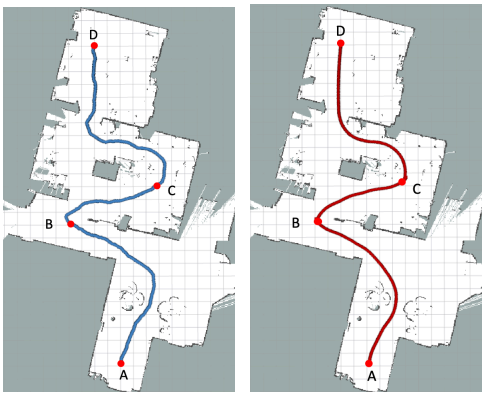
In this part, we conduct several experiments in a pre-recorded map to evaluate the navigation ability of the four models trained in section 4.1. The experiment map (see Fig. 6) consists of a corridor and two connected labs, and the edge length of the grid is 1 m. During the experiment, each model is given 20 runs, and in each run, the agent starts from point A, and is allowed 300 seconds to navigate the points B, C and D in order.

As summarized in Table 1, we note the number of successful runs (no collisions or time-outs), the number of runs with collisions, the average time of successful runs $\bar{t}_s$ and the average time till the first collision $\bar{t}_c$. The experiment results correspond well with the training curves in Fig. 5. The pre-trained models show higher success rate, and the CPO-based models, which are trained using a safety constraint, have lower crash rate. In particular, the model *TRPO Pre-trained* has both shortest $\bar{t}_s$ and $\bar{t}_c$, which is due to its risky decision making. On the contrary, the model *CPO Pre-trained* shows better safety with an acceptable trade-off with its navigation time. The model *CPO* has zero crash rate as the result of many time-outs.

By observing the sampled trajectories from the model *CPO Pre-trained* and TEB in Fig. 6, we can see that our method generates nearly the same trajectories but with more oscillations. As said at the beginning of section 4, this work is not intended to outperform a model-based planner with global knowledge of the map. This experiment proves that our method is effective in addressing the problem of map-less navigation and generalizes well to unseen environments although trained in only four maps (see Fig. 4).

Table 1. Experiment Results in a Pre-recorded Map

Model name	Success (%)	$\bar{t}_s$ (s)	Crash (%)	$\bar{t}_c$ (s)
TRPO	40	152.8	55	65.6
TRPO pre-trained	55	**93.3**	45	25.5
CPO	20	180.1	-	-
CPO pre-trained	**75**	150.3	**20**	**101.3**



(a) CPO Pre-trained (b) TEB

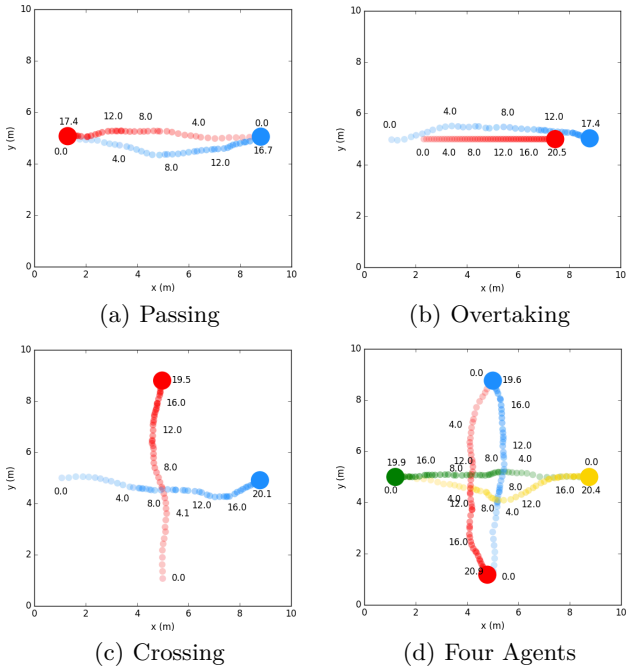
Fig. 6. Two trajectories sampled from our method CPO Pre-trained and TEB [2].

4.3 Experiments on Social Compliance

As shown in Fig. 7, four different experiments are conducted to test if the trained navigation policy (*CPO Pre-trained*) has a good collision avoidance ability while respecting the introduced right-handed rules (see Fig. 3). Timestamps (in seconds) are marked around the trajectories. In experiments 7(a), 7(b) and 7(c), the blue agent is set to pass, overtake and cross the red agent. In experiment 7(d), each agent is set to switch its position with the agent on the opposite side. We can see in all four experiments, the agents successfully reach their target position in a safe and socially compliant way. What's more, although our method uses a two-agent navigation policy for multiagent scenarios, more complex interaction patterns have occurred.

5 Conclusion

This work developed a learning-based approach for safe and socially compliant map-less navigation in dynamic environments. Our method uses a neural network representation and combines imitation learning and reinforcement learning. Experiments show that by leveraging prior demonstrations, the training time for RL can be reduced by around 60% and the overall performance is improved.



(a) Passing (b) Overtaking

(c) Crossing (d) Four Agents

Fig. 7. Tests on social compliance and collision avoidance.

Different tests in simulations prove that our method generalizes well to unseen environments, and a safe and socially compliant behavior is achieved. Future work will consider alternatives to better handle situations with a high number of nearby agents.

References

1. LaValle, S.M.: Planning Algorithms. Cambridge University Press (2006)
2. Rosmann, C., Hoffmann, F., Bertram, T.: Timed-Elastic-Bands for time-optimal point-to-point nonlinear model predictive control. In: 2015 European Control Conference (ECC). pp. 3352–3357. IEEE, Linz, Austria (Jul 2015)
3. Kretzschmar, H., Spies, M., Sprunk, C., Burgard, W.: Socially compliant mobile robot navigation via inverse reinforcement learning. The International Journal of Robotics Research 35(11), 1289–1307 (Sep 2016)
4. Achiam, J., Held, D., Tamar, A., Abbeel, P.: Constrained Policy Optimization. International Conference on Machine Learning (May 2017), arXiv: 1705.10528
5. Tai, L., Li, S., Liu, M.: A deep-network solution towards model-less obstacle avoidance. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2759–2764. IEEE, Daejeon, South Korea (Oct 2016)
6. Pfeiffer, M., Schaeuble, M., Nieto, J., Siegwart, R., Cadena, C.: From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 1527–1533. IEEE, Singapore, Singapore (May 2017)

7. Sergeant, J., Suenderhauf, N., Milford, M., Upcroft, B.: Multimodal deep autoencoders for control of a mobile robot. In: Li, H., Kim, J. (eds.) Proceedings of the Australasian Conference on Robotics and Automation 2015, pp. 1–10. Australian Robotics and Automation Association, Australia (2015)
8. Wulfmeier, M., Wang, D.Z., Posner, I.: Watch this: Scalable cost-function learning for path planning in urban environments. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2089–2095. IEEE, Daejeon, South Korea (Oct 2016)
9. Li, Y., Song, J., Ermon, S.: InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations. NIPS 2017 (Nov 2017), arXiv: 1703.08840
10. Kuefler, A., Morton, J., Wheeler, T., Kochenderfer, M.: Imitating driver behavior with generative adversarial networks. In: 2017 IEEE Intelligent Vehicles Symposium (IV). pp. 204–211. IEEE, Los Angeles, CA, USA (Jun 2017)
11. Zhu, Y., Mottaghi, R., Kolve, E., Lim, J.J., Gupta, A., Fei-Fei, L., Farhadi, A.: Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 3357–3364. IEEE, Singapore, Singapore (May 2017)
12. Li, H., Zhang, Q., Zhao, D.: Deep Reinforcement Learning-Based Automatic Exploration for Navigation in Unknown Environment. IEEE Transactions on Neural Networks and Learning Systems pp. 1–13 (2019)
13. Zhelo, O., Zhang, J., Tai, L., Liu, M., Burgard, W.: Curiosity-driven Exploration for Mapless Navigation with Deep Reinforcement Learning. ICRA 2018 Workshop in Machine Learning in the Planning and Control of Robot Motion, At Brisbane, May 2018 (May 2018), arXiv: 1804.00456
14. Long, P., Fanl, T., Liao, X., Liu, W., Zhang, H., Pan, J.: Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 6252–6259. IEEE, Brisbane, QLD (May 2018)
15. Lutjens, B., Everett, M., How, J.P.: Safe Reinforcement Learning With Model Uncertainty Estimates. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 8662–8668. IEEE, Montreal, QC, Canada (May 2019)
16. Trautman, P., Ma, J., Murray, R.M., Krause, A.: Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation. The International Journal of Robotics Research 34(3), 335–356 (Mar 2015)
17. Ferrer, G., Garrell, A., Sanfeliu, A.: 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. pp. 1688–1694. IEEE, Tokyo (Nov 2013)
18. Mehta, D., Ferrer, G., Olson, E.: Autonomous navigation in dynamic social environments using Multi-Policy Decision Making. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1190–1197. IEEE, Daejeon, South Korea (Oct 2016)
19. Kim, B., Pineau, J.: Socially Adaptive Path Planning in Human Environments Using Inverse Reinforcement Learning. International Journal of Social Robotics 8(1), 51–66 (Jan 2016)
20. Kuderer, M., Kretzschmar, H., Sprunk, C., Burgard, W.: Feature-Based Prediction of Trajectories for Socially Compliant Navigation. Robotics:Science and Systems, 2012 p. 8 (2012)
21. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-Dimensional Continuous Control Using Generalized Advantage Estimation. ICLR 2016 (Oct 2018), arXiv: 1506.02438