2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Learning World Transition Model for Socially Aware Robot Navigation

Yuxiang Cui, Haodong Zhang, Yue Wang, Rong Xiong

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9561973

***Abstract*— Moving in dynamic pedestrian environments is one of the important requirements for autonomous mobile robots. We present a model-based reinforcement learning approach for robots to navigate through crowded environments. The navigation policy is trained with both real interaction data from multi-agent simulation and virtual data from a deep transition model that predicts the evolution of surrounding dynamics of mobile robots. A reward function considering social conventions is designed to guide the training of the policy. Specifically, the policy model takes laser scan sequence and robot's own state as input and outputs steering command. The laser sequence is further transformed into stacked local obstacle maps disentangled from robot's ego motion to separate the static and dynamic obstacles, simplifying the model training. We observe that the policy using our method can be trained with significantly less real interaction data in simulator but achieve similar level of success rate in social navigation tasks compared with other methods. Experiments are conducted in multiple social scenarios both in simulation and on real robots, the learned policy can guide the robots to the final targets successfully in a socially compliant manner. Code is available at https://github.com/YuxiangCui/model-based-social-navigation.**

I. Introduction

Autonomous navigation through previously unseen and crowded environments like airports or shopping malls is in demand for robots but remains a challenging research field. Expanding habitats from static and isolated environments to human-robot sharing environments needs the robots to be capable of moving in a socially compliant manner. Therefore, the robots should make decisions considering the evolution of surrounding dynamics in case of endangering self-safety or interfering human comfort.

Traditional solution to this problem is a modular pipeline that consists of human detection, trajectory prediction and local path planning. It works efficiently in some common scenarios but may fail in crowded environments because of the gap between these modules. For example, over-conservative estimation from detection and prediction modules usually leaves no place for path planning causing robot frozen problems [1]. Not only that, the high dependence on accurate and robust scene understanding with multiple sensors also constrains its performance in application.

Deep learning based methods try to tackle this problem in an end-to-end style. Existing learning based methods can be divided into two categories, imitation learning methods and reinforcement learning methods. Imitation learning methods optimize the policy over abundant expert demonstrations, while reinforcement learning methods obtain policy by interacting with the environment. Those methods use deep networks to directly map sensor data to steering commands [2] [3] [4]. Taking the advantage of sensors like 2D laser, light-weight environment information with respect to robots' own coordinates can be obtained in real time. Nevertheless, these methods still require extensive training data or long time searching.

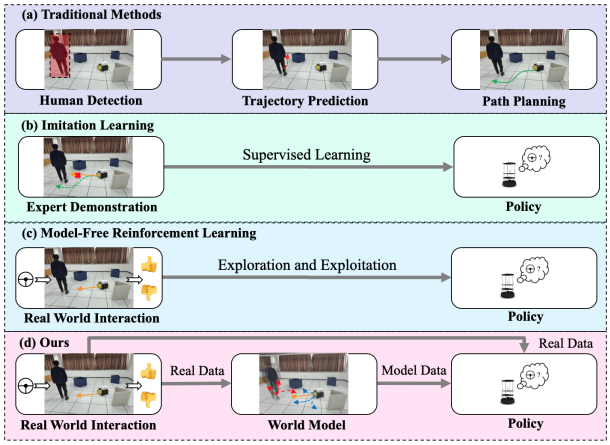


Fig. 1: Illustration comparing traditional method(a), imitation learning method(b), model-free reinforcement learning method(c) and our model-based reinforcement learning method(d). Note that our method utilizes a world model to predict the evolution of surrounding dynamics and produce abundant virtual data for social navigation policy training, which significantly improves the sample efficiency.

In this paper, we develop a framework of socially aware navigation policy training with model-based reinforcement learning using only 2D laser scans. Specifically, we propose a deep world transition model to predict the future observations of robots and the corresponding rewards. By leveraging the learned world transition model, we can train the policy with data composed of limited real data from actual interaction and abundant simulated data from the transition model, substantially improving the sample efficiency. To disentangle the ego motion from the observations, we transform the laser scans into the same frame, resulting in stacked local obstacle maps. Compared with the angle range representation of laser scans [2], the obstacle map representation provides a clearer distinction between static and dynamic obstacles, which is expected to accelerate the learning. To the best of our knowledge, this is the first work utilizing model-based reinforcement learning for social navigation. Besides, our

Yuxiang Cui, Haodong Zhang, Yue Wang and Rong Xiong are with the State Key Laboratory of Industrial Control and Technology, Zhejiang University, Hangzhou, P.R. China. Yue Wang is the corresponding author wangyue@iipc.zju.edu.cn.

network is trained in a policy-sharing multi-agent simulation environment which better resembles actual crowds, rather than previous methods using only one decision-making agent and leaving others to move with predefined rules. Based on these designs, the learned policy can be directly transferred to real robots. Particularly, this paper presents the following contributions:

- We propose a deep model of environment dynamics trained in a self-supervised way that predicts future observations and produces simulated data for policy training. The disentangled local obstacle map representation provides a clear distinction between dynamic and static obstacles, which improves the performance of world model and policy network.
- We introduce a model-based reinforcement learning approach that efficiently tackles socially aware navigation problems under limited interaction data with the environment by simulating abundant virtual data.
- We train the policy in a decentralized policy-sharing multi-agent simulation environment that better resembles social interactions and evaluate the policy in multiple social scenarios and on real robots.

II. Related Works

A. Social Navigation with Human Detection

Aided by human detection with multiple sensors developed these years, some of the prior works on social navigation problems use the results of human detection to estimate the states of surrounding pedestrians and identify a safe path through. Existing detection-based approaches can be divided into two categories, model-based methods and learning-based methods.

Model-based methods like Social Force [5], RVO [6] and ORCA [7], tend to design a specific rule describing multi-agent interactions, so that a hand-crafted potential field of the surrounding environment can be constructed, transforming the social navigation problem into an optimization problem. Different social characteristics, like motion behaviours [8], are considered in extensions of those methods, but it is still unclear whether pedestrians do follow those rules. These methods are computationally efficient and interpretable but limited by applicability as the parameters of these models vary greatly between different environments or even different pedestrians.

In contrast, learning-based methods obtain policy by optimizing the network over a large amount of training data. Earlier works adopt the supervised learning paradigm to let the robots mimic expert behaviors or make decisions using reward functions from inverse reinforcement learning [9] [10]. As an alternative, reinforcement learning based methods learn from interactions. [11] proposes a method using a value network to describe cooperative interactions in a multi-agent environment and navigate robots by querying the best action from it. Their later works extend it to an arbitrary number of surrounding pedestrians without any assumptions of human behaviors using LSTM model [12] and introduce common social norms into the framework [13]. Interactions within the crowds like human-human interactions that indirectly affect robots have also been considered with an attentive pooling mechanism in [14].

B. Social Navigation with Sensor-level Data

Recent works have used sensor-level data to train socially compliant navigation behaviors. Imitation learning based ones directly extract expert driving behavior from demonstrations [15], which need a large amount of expert labeled data. In another line of work, Tai et al. [16] use reinforcement learning to train a mapless navigation policy in simulation taking only 2D laser data as input, but only works well in nearly static environments due to the little information with sparse range findings. Chen et al. [11] propose a decentralized collision avoidance method with PPO using dense laser input and integrate it into a hybrid control framework in [3], which still makes oscillatory moves in crowds as it dose not distinguish moving pedestrians from the raw sensor input. Tai et al. [4] develop a GAIL-based strategy using raw depth images describing the surrounding environment including the relative positions of pedestrians, but it usually makes short-sighted decisions with only single frame observations. In contrast, sequential laser scans disentangled from robots' ego motion are used in [2] to tackle the partial observability of the environment, which also inspires our sensor processing method in this work.

C. Model-based Reinforcement Learning

Using models of environments to predict the state evolution of robots has a great appeal for reinforcement learning based approaches, as the model-free reinforcement learning methods need extensive interaction data for training, which is unrealistic and time-consuming to obtain in real world. Plenty of works have achieved good results on the Atari games [17] or robot control tasks [18] [19] with model-based reinforcement learning methods. Oh et al. [20] propose an action-conditional video prediction network to anticipate multi-step future evolution and Leibfried et al. [21] later extend it by including reward prediction, but they use the model for predictive planning instead of policy training. In contrast, Kaiser et al. [17] use the MBPO [22] framework where policies are directly trained on the deep prediction model instead of the real world.

III. Approach

In this paper, we aim to tackle the social navigation problem with a model-based reinforcement learning framework (Fig. 2). The world transition model is trained over real interaction data and then used to provide virtual data for policy training, improving the sample efficiency. Besides, we model the social interactions with a policy-sharing multi-agent environment, which we consider to be more realistic. The following parts give details about the key ingredients of our framework.

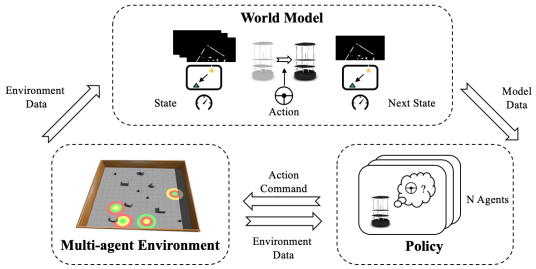


Fig. 2: Framework of our model-based social navigation method. A loop framework is used here with three main parts. 1) Robot interacts with the real world to get environment data. 2) World model gets trained on the real environment data in a self-supervised style. 3) Robot updates its policy on the combination of both real data from robot interaction and virtual data from the world model. The learned policy is then evaluated in the real world.

A. World Transition Model

The world transition model should be able to understand the current state and predict the future state conditioned on the input action. The state is defined as 2D laser reading, relative goal position and current velocity. Among them, future goal and velocity states can be directly estimated with the action command according to the kinematics of robots. While for the prediction of future laser readings, both the motion of dynamic obstacles in the field of view and the ego motions of the robot are involved. By transforming laser readings into obstacle maps using the angle and range information, we can formulate the transition prediction problem as a video prediction problem. A deep transition model is designed to infer the evolution of surrounding dynamics at the pixel-wise level. The last ten consecutive laser readings are utilized to leverage the temporal information of environment dynamics. We represent the laser scans as stacked obstacle maps built by transforming multiple laser scans into one frame based on the odometry, so that the static obstacles are overlaid together, while the dynamic obstacles are highlighted, which fully disentangles the ego motions from the measurement. Besides, this representation also improves the performance of the predictive model.

Inspired by [23], we separately encode the motion and content of sequential obstacle map input. Specifically, we do subtraction between adjacent obstacle maps chronically to get the motion features and encode them with convolutional LSTM to get the hidden representation of dynamics. Then we get the hidden representation of static features by encoding the last obstacle map as content. With both the static and dynamic features, we can get a pixel-level prediction of the next obstacle map by decoding the combined features. In the end, an affine transformation is used on the image to leverage the effect of robot's ego motion and the action command. Together with goal and velocity states, a corresponding reward can also be predicted with fully connected layers. The model architecture is shown in Fig. 3. This network is trained to minimize the combination of image reconstruction loss and reward difference loss.

Considering the epistemic uncertainty caused by scarce data and random model parameter initialization, we use a bootstrapping procedure to better exploit environment data and stabilize policy training.

B. Model-Based Reinforcement Learning

Social navigation problem can be formulated as a Partially-Observable Markov Decision Process in reinforcement learning framework which can be described as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathcal{O})$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{R}$ is the reward function, Ω is the observation space and $\mathcal{O}$ is the observation probability conditioned on the state. Different from model-free methods, $\mathcal{P}$ is the state transition model composed of both the real environment dynamics and the deep predictive model mechanism in our model-based reinforcement learning framework. The following provides a detailed formulation.

Observation space Our observation space only includes laser readings o_l^t, relative goal position o_g^t and robot's own velocity o_v^t which can be directly acquired through sensors mounted on the robot. And the laser readings o_l^t are represented as obstacle map sequence using the method above.

Due to the limited vision of the laser, the robots can obtain various observations even in simple simulation scenarios. Therefore, the learned policy can directly generalize to complex unseen environments with an arbitrary number of dynamic obstacles represented with similar local readings.

Action space The action is composed of the linear and angular velocity of differential robots sampled from policy π in continuous space. Considering the robots' kinematics and average velocity of pedestrians, we set the range of linear velocity $v \in [0, 1]$ and the angular velocity $w \in [-1.5, 1.5]$ to help the agent better integrate into crowds.

Reward setup To guide the policy optimization, we have designed a reward function taking arriving target, collision avoidance and social manners into account:

$$R(s_t) = R_g(s_t) + R_c(s_t) + R_s(s_t)$$

In particular, the agent gets $R_g(s_t)$ for getting closer to its goal:

$$R_g(s_t) = \begin{cases} r_{arrival} & \text{if goal reached} \\ w_1(\left\|p^t - p^*\right\| - \left\|p^{t-1} - p^*\right\|) & \text{otherwise} \end{cases}$$

where p^t refers to the robot position at time step t, and p^* refers to the goal position. To make sure that the robot moves at least in a safe manner, it gets penalized with $R_c(s_t)$ when it gets closer to or collides with the obstacles:

$$R_c(s_t) = \begin{cases} r_{collision} & \text{if collision} \\ w_2(1 - \dfrac{d}{r + 1.0}) & \text{if } d \leq r + 1.0 \\ 0 & \text{otherwise} \end{cases}$$

where r here defines the robot safety radius and d refers to the minimum value of laser distance at the current time step.

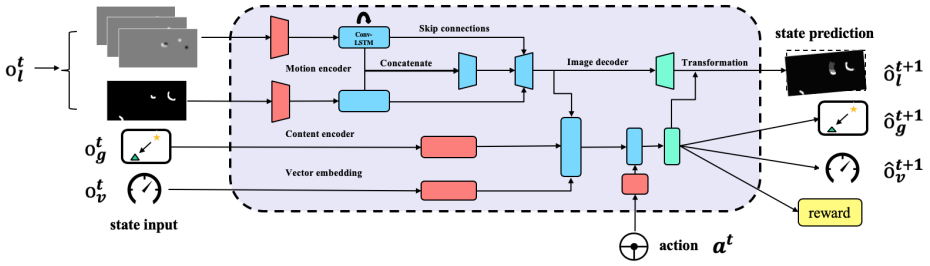


Fig. 3: Architecture of our transition predictive model. The input includes ten consecutive obstacle map representation laser scans, relative goal position and current velocity of robot, while the output includes the future obstacle map transformed according to the velocity change and corresponding reward. All the observation inputs are encoded and combined with action embedded by fully connected layers. Specifically, the sequential obstacle maps are further processed into motion and content, in which motion is composed of differences between adjacent frames and content is represented by the last frame.

Algorithm 1 Social navigation with model-based RL

```
Initialize policy π_φ, predictive model p_θ, environment dataset D_env, model dataset D_model;
Initialize D_env with random exploration and pre-train p_θ on D_env;
for N epochs do
    for R steps do
        Collect data to extend D_env using π_φ;
        Roll out p_θ to extend D_model using π_φ;
        Update π_φ on D_env and D_model
    end for
end for
```

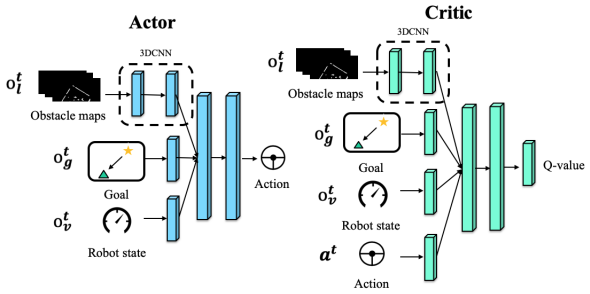


Fig. 4: Architecture of shared policy model. The actor network embeds state inputs and outputs action commands. The critic network embeds states with corresponding actions and outputs Q-values.

As for the socially compliant demands, we believe that social manners emerge from reciprocal interactions where all the agents respect others' scope of activity including the current and future positions. So the key is to keep a safe distance from all the dynamic obstacles in the field of view in the whole process of interaction.

Therefore, we have defined an additional penalty $R_s(s_t)$ to force the robot to be more careful with the dynamic obstacles and respond in advance:

$$R_s(s_t) = \begin{cases} w_3(1 - \dfrac{d_{ped}}{r + 1.25}) & \text{if } d_{ped} \leq r + 1.25 \\ 0 & \text{otherwise} \end{cases}$$

This reward setup will guide the robot to its final goal without collisions especially the ones with dynamic obstacles like pedestrians.

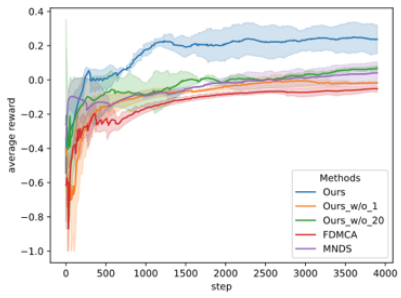
Policy network As for the policy training network, we adopt the framework of TD3 [24]. The policy network is presented in Fig. 4. A 3D-CNN module is used to deal with the sequential obstacle map input. All the state vectors are normalized to the range of $[0, 1]$ before sending into the network.

C. Multi-robot Training with Shared Policy

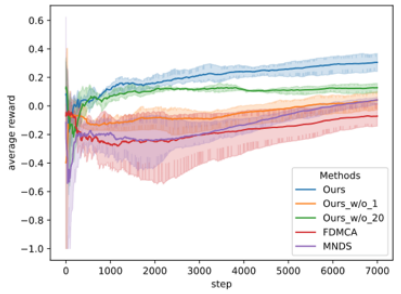
Multi-robot environment We believe that human-robot interaction certainly exists in social navigation tasks and it is unreasonable to let the simulated pedestrians ignore the robots or behave with predefined rules, so we choose to train our policy in a cooperative environment. To expose our agents to complex and crowd-like environments, we adopt the idea of decentralized multi-agent setup proposed in [3] and build a simulation environment in Gazebo with multiple policy sharing agents. All the agents make decisions by sampling from the same policy π without communication, so that they are forced to learn cooperative behaviors when confronting each other.

Multiple scenarios are also designed for training and testing. Both the static obstacles and dynamic agents are randomized in sizes and shapes by setting collision ranges in Gazebo. The agents are also randomized in initialization with different positions, angles and goals. In the training process, only the main agent is used to collect interaction data, so that the sample efficiency improvement from the model can be clearly demonstrated.

Training Algorithm To enhance sample efficiency and provide plenty of data for policy training, we use the model-based framework, summarized in Algorithm 1, to train our social navigation policy. In which, environment samples are collected not only for policy training, but also for world model training. Then the policy network can be trained on both the real environment data and virtual model data.



(a) Mapless Navigation



(b) Social Navigation

Fig. 5: Comparison between model-based method and the baselines. We refer to our full version method as 'Ours' and the version without using world model as 'Ours_w/o'. The number behind means the iterations of policy updates. The solid curves and shaded regions depict the mean and the standard deviation of average reward among three trials. Our model-based method can arrive at a similar average reward level with much less data than other method, which proves the sample efficiency. In the training process, we also observe that our method can converge with higher stability between different trials.

IV. Experiments

A. Ablation Study

Laser Scan Representation: To prove the merits of our obstacle map representation laser input, we test both obstacle map representation and angle range representation laser input on state transition prediction task. We implement an LSTM-based predictive network using angle range representation laser as input to compare with our predictive model. Both of these two inputs are disentangled from robot's ego motion and normalized to $[0, 1]$ before sending into the networks. And both networks are trained thoroughly until convergence.

We use a five-step rollout to evaluate the performance of prediction, results are shown in obstacle map representation for clear contrast in Fig. 6. We also compare the reconstruction error of these two methods in Tab. I.

It turns out that our model can predict the dynamics of the surrounding environment with both high accuracy and validity. The dynamic obstacles' motions are captured and predicted by our model in the obstacle maps, while other obstacles are regarded as static content. But the prediction from angle range representation laser input can hardly reconstruct the size or shape of obstacles and the motion predictions also deviate a lot from the true positions.

Quantitatively, Tab. II demonstrates that obstacle map representation of laser input can also help the policy achieve a better performance, meaning that this representation of laser input can help the robots distinguish dynamic obstacles much more easily and act accordingly, namely socially aware.

(a) history trajectory (b) our prediction (c) array prediction

Fig. 6: Comparison between obstacle map and angle range prediction. To make a clear comparison, we demonstrate the predictions of these two methods both in the representation of the local obstacle map, where the robot's position is at the middle of the bottom edge. Consecutive obstacle maps are stacked in chronological order with different color depths. We mark the historical trajectory with dark grey, four-step prediction with light grey, and the fifth prediction with white. The moving parts can be correctly captured and predicted with our method, while the angle range representation fails.

	Obstacle Map	Angle Range
Mean Square Error	**0.0032**	0.0805

TABLE I: Reconstruction error. We calculate the mean square error between the predictions and labels on multiple random samples. The obstacle map representation can achieve lower reconstruction error.

Sample efficiency: We first evaluate the sample efficiency of our model-based framework on the mapless navigation task in static environments. We compare policy training with and without using our predictive model in the same environments multiple times. Results in Fig. 5 show that model-based learning achieves a higher level of average reward with notably fewer environment interactions.

We further test our framework on social navigation tasks in an initially randomized environment with multiple agents. All agents are decision makers with the same policy, but only one agent is used to collect interaction data, so that fair comparison can be made with other baselines. As shown in Fig. 5, our model-based learned policy still achieves the best average reward with the least environment interactions.

Besides, the training data includes much more exploration of the environment, so that the policy can be trained multiple times on every environment step with less risk of getting overfitting. To ensure that our predictive model actually helps, we also test the version without using our predictive model with additional iterations of policy updating. Results show that increasing the iterations of policy updating does help convergence but it still can not match the speed of our full version method.

B. Comparative Study

We further provide a thorough evaluation of the performance of our social navigation framework with model-based reinforcement learning by comparing it with existing approaches. In particular, we choose the following two methods as baselines to compare:

	Passing				Towards				Crossing				Random			
	Success Rate	Arriving Time	Ego Score	Social Score	Success Rate	Arriving Time	Ego Score	Social Score	Success Rate	Arriving Time	Ego Score	Social Score	Success Rate	Arriving Time	Ego Score	Social Score
FDMCA	100 %	**17.6 s**	97	95	95 %	18.0 s	94	93	70 %	15.1s	75	73	85 %	21.3 s	89	86
MNDS	80 %	18.6 s	93	91	90 %	**16.3 s**	89	88	70 %	**13.2 s**	88	86	80 %	19.5 s	90	89
OURS	**100** %	18.1 s	**100**	**100**	**100** %	16.5 s	**96**	**96**	**85** %	15.9 s	**93**	**93**	**100** %	**19.1 s**	**99**	**98**

TABLE II: Comparative results. We evaluate the three methods in four different scenarios. Four policy-sharing agents are involved in those scenarios but initialized with different goal distributions, so that different social interactions emerge. Each setup is tested for 20 rounds. The results show that our method can achieve higher scores and success rates in most cases with similar time needed.

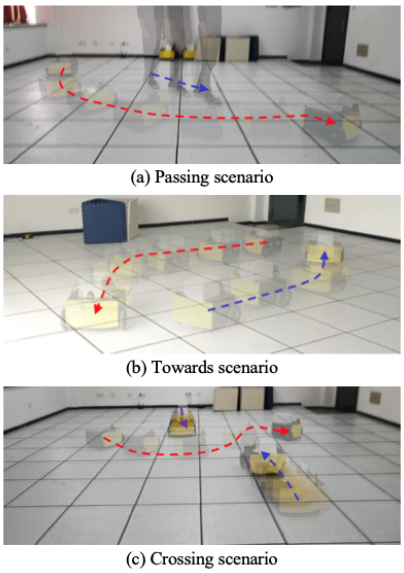
- ***FDMCA*** [3] is a RL-based dynamic collision avoidance algorithm using only 2D laser scans. It uses 3 consecutive angle range lasers as input. The policy is trained in a decentralized multi-agent environment.
- ***MNDS*** [2] is the state-of-the-art socially aware mapless navigation approach based on RL. It uses laser scans with disentangled angle range representation as network input. The policy is trained for one agent in environments where obstacles move with predefined rules.

Sample efficiency: To make a fair comparison, we implement these methods in similar parameter quantities and train them under the same hyper parameters. All these networks are trained in a four-agent environment. In the training process, only one agent is used to collect data and the other policy-sharing agents are used to simulate social interactions. As we can see in Fig. 5, our method not only converges fast, but also achieves a higher level of average reward. It proves that 1) the predictive model can help policy converge with limited data and 2) the obstacle map representation helps agents make decisions.

Tasks performance: We test our learned policy with baselines in crossing, passing, towards and random settings in Gazebo. To compare the performance of these two baselines with our method, we use the following metrics:

- ***Success Rate:*** The ratio of arriving goals without collision within the time allowed after 20 runs.
- ***Arriving Time:*** Average time for reaching the goals.
- ***Ego Score:*** The ratio of steps keeping a safe distance from obstacles. Let m be the ego safe steps and N be total steps, $Ego\ Score = m/N * 100$.
- ***Social Score:*** The ratio of steps keeping a comfortable distance from dynamic obstacles like pedestrians, representing the policy's ability of avoiding others' social comfort zone and acting in advance. Let n be the socially compliant steps and N be total steps, $Social\ Score = n/N * 100$.

The evaluation results are shown in Tab. II. It turns out that our method can achieve better performance in most cases. The learned policy can navigate through crowds in a cooperative way considering both ego safety and human comfort. We have observed that MNDS can also avoid dynamic obstacles accordingly but usually makes over-optimistic expectations to the environment that lead to collisions. FDMCA always makes short-sighted decisions that result in shaky trajectories when confronting dynamic obstacles. It also tends to negatively slow down and make a detour in open areas.



(a) Passing scenario

(b) Towards scenario

(c) Crossing scenario

Fig. 7: Real robot evaluation. The learned policy is evaluated in multiple real world scenarios, including passing, towards and crossing. Figures above show the trajectories of real robots in these scenarios.

C. Real World Experiments

We evaluate the performance of learned policy on the real differential robots. With on-board LiDAR and IMU, the robot can get its own velocity and relative goal position. We test our learned policy in a variety of scenarios with multiple robots and pedestrians. The robot keeps a safe distance from walking pedestrians and makes cooperative decisions to avoid collision with other robots in the experiments as shown in Fig. 7.

V. Conclusions

This paper presents a model-based reinforcement learning approach for social navigation task using ego motion disentangled obstacle map. By using the virtual data generated from the model to train the policy, sample efficiency can be significantly improved. Experiments show that the proposed method can achieve better convergence and higher average reward in training, and successful deployment in real world.

VI. Acknowledgement

This work was supported in part by the National Nature Science Foundation of China (61903332), in part by the Science and Technology Project of Zhejiang Province (2019C01043).

REFERENCES

[1] A. J. Sathyamoorthy, U. Patel, T. Guan, and D. Manocha, "Frozone: Freezing-free, pedestrian-friendly navigation in human crowds," *IEEE Robotics and Automation Letters*, 2020.

[2] J. Jin, N. M. Nguyen, N. Sakib, D. Graves, H. Yao, and M. Jagersand, "Mapless navigation among dynamics with social-safety-awareness: a reinforcement learning approach from 2d laser scans," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6979–6985, IEEE, 2020.

[3] T. Fan, P. Long, W. Liu, and J. Pan, "Fully distributed multi-robot collision avoidance via deep reinforcement learning for safe and efficient navigation in complex scenarios," *arXiv preprint arXiv:1808.03841*, 2018.

[4] L. Tai, J. Zhang, M. Liu, and W. Burgard, "Socially compliant navigation through raw depth inputs with generative adversarial imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1111–1117, IEEE, 2018.

[5] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[6] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*, pp. 1928–1935, IEEE, 2008.

[7] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*, pp. 3–19, Springer, 2011.

[8] T. Randhavane, A. Bera, E. Kubin, A. Wang, K. Gray, and D. Manocha, "Pedestrian dominance modeling for socially-aware robot navigation," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5621–5628, IEEE, 2019.

[9] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation.," in *Robotics: science and systems*, 2012.

[10] 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.

[11] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 285–292, IEEE, 2017.

[12] 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)*, pp. 3052–3059, IEEE, 2018.

[13] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1343–1350, IEEE, 2017.

[14] 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)*, pp. 6015–6022, IEEE, 2019.

[15] 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, 2017.

[16] 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)*, pp. 31–36, IEEE, 2017.

[17] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, *et al.*, "Model-based reinforcement learning for atari," *arXiv preprint arXiv:1903.00374*, 2019.

[18] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International Conference on Machine Learning*, pp. 2555–2565, PMLR, 2019.

[19] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control," *arXiv preprint arXiv:1812.00568*, 2018.

[20] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in atari games," in *Advances in neural information processing systems*, pp. 2863–2871, 2015.

[21] F. Leibfried, N. Kushman, and K. Hofmann, "A deep learning approach for joint video frame and reward prediction in atari games," *arXiv preprint arXiv:1611.07078*, 2016.

[22] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," in *Advances in Neural Information Processing Systems*, pp. 12519–12530, 2019.

[23] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," *arXiv preprint arXiv:1706.08033*, 2017.

[24] S. Fujimoto, H. Van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," *arXiv preprint arXiv:1802.09477*, 2018.