Socially Compliant Navigation through Raw Depth Inputs with Generative Adversarial Imitation Learning

Lei Tai[1] Jingwei Zhang[2] Ming Liu[1] Wolfram Burgard[2]

***Abstract*— We present an approach for mobile robots to learn to navigate in dynamic environments with pedestrians via raw depth inputs, in a socially compliant manner. To achieve this, we adopt a generative adversarial imitation learning (GAIL) strategy, which improves upon a pre-trained behavior cloning policy. Our approach overcomes the disadvantages of previous methods, as they heavily depend on the full knowledge of the location and velocity information of nearby pedestrians, which not only requires specific sensors, but also the extraction of such state information from raw sensory input could consume much computation time. In this paper, our proposed GAIL-based model performs directly on raw depth inputs and plans in real-time. Experiments show that our GAIL-based approach greatly improves the safety and efficiency of the behavior of mobile robots from pure behavior cloning. The real-world deployment also shows that our method is capable of guiding autonomous vehicles to navigate in a socially compliant manner directly through raw depth inputs. In addition, we release a simulation plugin for modeling pedestrian behaviors based on the social force model.**

I. Introduction

1) Socially compliant navigation: The ability to cope with dynamic pedestrian environments are crucial for autonomous ground vehicles. In static environments, mobile robots are required to reliably avoid collision with static objects and plan feasible paths to their target locations; while in dynamic environments with pedestrians, they are additionally required to behave in socially compliant manners, where they need to understand the dynamic human behaviors and react accordingly under specific socially acceptable rules.

Traditional solutions can be classified into two categories: *model-based* and *learning-based*. Model-based methods aim to extend the multi-robot navigation solutions with socially compliant constraints [1]. However, force parameters need to be carefully tuned for each specific scenario. Learning-based methods, on the other hand, aim to directly recover the expert policy through direct supervised learning (e.g., behavior cloning), or through inverse reinforcement learning to recover the inner cost function.

Yet, all previous approaches require the knowledge of the precise localization and velocity information of nearby pedestrians. This limitation restricts these methods to be only applicable for robots equipped with high precision sensors, like 3D Lidars [2], [3]. Moreover, the estimation of the state information of pedestrians is generally time-consuming. Developing effective navigation strategies directly from raw sensor inputs is still of great importance.

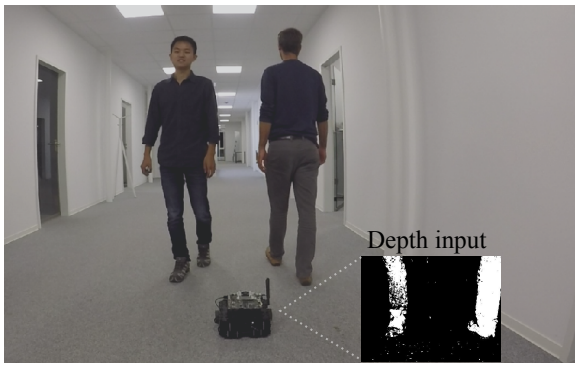


Fig. 1: A mobile robot navigating in a socially compliant manner in an indoor environment. Our approach tackles this scenario through raw depth images.

2) Perception from visual input: Compared with 3D Lidars, vision sensors come at affordable prices, making them more suitable to equip mobile agents that are beginning to populate our social life. Thus, navigation solutions that can directly operate on raw visual inputs are more feasible than those depending on expensive Lidars.

In coping with visual information, deep learning approaches have become imperative due to their ability to extract hierarchically more abstract feature representations. Also, with the improvement of the computation power of mobile platforms, such approaches have been utilized in various robotics tasks such as obstacle avoidance [4]. In this paper, we deploy neural networks for extracting useful features from raw depth visual inputs, captured by onboard depth cameras of the mobile robot.

We use depth inputs over RGB images because the visual fidelity of the simulated color images are much worse than that of the depth images. This bigger deviation of color images from synthetic environments to real-world scenarios makes it more challenging to transfer the model trained on simulated images directly to the real world. The simulated depth images, on the other hand, are more consistent with the real domain and can greatly ease the transfer phase.

3) Learning-based navigation: Deep reinforcement learning have recently gained much attention, and have been successfully extended to learn autonomous navigation

*This paper is supported by Shenzhen Science, Technology and Innovation Commission (SZSTI) JCYJ20160428154842603 and JCYJ20160401100022706; also supported by the Research Grant Council of Hong Kong SAR Government, China, under Project No. 11210017 and No. 16212815 and No. 21202816 awarded to Prof. Ming Liu.

[1]Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology; {ltai, eelium}@ust.hk

[2]Department of Computer Science, Albert Ludwig University of Freiburg; {zhang, burgard}@informatik.uni-freiburg.de

from raw sensory inputs [5], [6], [7], [8].

In terms of socially compliant navigation, behavior cloning methods are easy to deploy since they treat the policy learning as a pure supervised learning task. Yet, the learned model of behavior cloning completely ignores the temporal correlation between samples in subsequent frames, thus it cannot generalize well to scenarios that deviate too much from the training data. Agents are also able to learn from expert demonstrations via an intermediate step of learning the latent cost or the reward function [2], [9], [10] through inverse reinforcement learning [11], [12]. Chen *et al.* [3] defined a complex reward to train the socially aware planning policies (e.g. passing by the right/left side) through reinforcement learning. However, all of these methods depend on accurate pedestrian information and expensive sensors as mentioned before.

Generative adversarial imitation learning (GAIL) is an effective alternative for learning from demonstrations [13]. InfoGAIL [14] successfully solved a simulated autonomous driving task with raw visual input. However, only simulated experiments are presented in those two papers.

In this paper, we effectively deploy behavior cloning for learning an initial policy, Then, we apply GAIL on the basis of this initial policy, to benefit the policy model by taking the temporal correlations in the dataset into account.

Particularly, this paper presents the following contributions:

- We introduce an effective GAIL-based approach that is able to learn and improve socially compliant navigation policies through raw depth inputs.
- We release a plugin for simulating pedestrians behaving in socially compliant manners, as well as a dataset, where 10,000 socially compliant navigation state-action pairs are recorded, based on the social force model [1], in various social scenarios.

II. BACKGROUND

We consider a *Markov decision process* (MDP), where an agent interacts with the environment through a sequence of observations, actions and reward signals. The agent executes an action $a_t \in \mathcal{A}$ at time step t from its current state $s_t \in \mathcal{S}$, according to its policy $\pi : \mathcal{S} \to \mathcal{A}$. It then receives a reward signal $r_t : \mathcal{S} \to \mathbb{R}$ and transits to the next state s_{t+1} according to the dynamics of the environment. We use π_E to denote the expert policy.

A. Imitation learning

Imitation learning aims to learn policies directly from experts, whose demonstrations are only provided in the form of samples of trajectories. Main approaches for imitation learning can be categorized into behavior cloning (BC) and inverse reinforcement learning (IRL) Behavior cloning tackles this problem in a supervised manner, by directly learning the mapping from the states in the recorded trajectories to their corresponding labels: the expert policy. It is conceptually simple and is able to work well with large amounts of training data, but suffers from the compounding error caused by covariate shift, due to its ignorance of the temporal correlation in the recorded trajectories. Thus pure behavior cloning tends to over-fit and is difficult to generalize to unseen scenarios that deviate much from the recorded dataset. Inverse reinforcement learning aims to extract the latent reward or cost function under the optimal expert demonstrations. It imitates by taking the underlying MDP into account, learning from entire trajectories instead of single frames. However, the requirement of running reinforcement learning in an inner loop makes IRL extremely expensive to run [2], [9], [10].

B. Generitive adversarial imitation learning

Inspired by Generative adversarial networks (GAN) [15], Ho and Ermon proposed generative adversarial imitation learning (GAIL) [13], which surpasses the intermediate step of learning a reward function, but is able to directly learn a policy from expert demonstrations. In the GAIL model, the generator π_θ is forced to generate state-action $(\mathcal{S}\times\mathcal{A})$ pairs matching that from the expert demonstrations, while the discriminator D_ω learns to tell the generated policy π_θ (θ denotes the parameters of the generator of the GAIL model) apart from the expert policy π_E. The objective of GAIL is to optimize the function below ($H(\pi)$ represents the causal entropy of the policy):

$$\mathbb{E}_{\pi_\theta}[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] - \lambda H(\pi_\theta) \tag{1}$$

Following this objective, the learning procedure of GAIL interleaves between updating the parameters ω of the discriminator D_ω to maximize Eq. 1, and performing trust region policy optimization (TRPO) [16] to minimize Eq. 1 with respect to θ, which parameterizes the policy generator π_θ. Here, the discrimination scores of the generated samples are regarded as costs (can be viewed as the negative counterpart of rewards) of the state-action pairs in the learning process of TRPO. As a state-of-the-art on policy reinforcement learning method, TRPO constraints the deviation of the updated policy from the original policy according to their KL divergence.

We extend the GAIL framework with Wasserstein GAN (WGAN) [17]. In WGAN, the classification formulation of the discriminator network is substituted with a regression problem. By eliminating the usage of *softmax* in traditional GAN, WGAN directly maximizes the score of the real data and minimizes the score of the generated data. It is proved to improve the training stability of InfoGAIL [14] The objective function of WGAN is:

$$\mathbb{E}_{\pi_\theta}[D(s,a)] - \mathbb{E}_{\pi_E}[D(s,a))] \tag{2}$$

C. Social force model

Helbing and Molnar [1] proposed the social force model for pedestrian dynamics, which is broadly applied to socially compliant navigation scenarios. This model (Eq. 3) computes the acceleration of pedestrians according to the sum of various forces applied to them.

Fig. 2: The simulated environment used in this paper. Each pedestrian is behaving under the social force model [1]. The *Gazebo* plugin is released[1].

$$\frac{d\vec{v}_t}{dt} = \vec{F}_{\text{desired}} + \vec{F}_{\text{social}} + \vec{F}_{\text{obs}} + \vec{F}_{\text{fluct}} \tag{3}$$

In Eq. 3, $\vec{F}_{\text{desired}}$ represents the *desired force*, which drives the vehicle towards its navigation goal; $\vec{F}_{\text{social}}$ is the *social force* to measure the influence of nearby pedestrians; $\vec{F}_{\text{obs}}$ is the *obstacle force* to keep the agent from colliding with static obstacles in the environment; and $\vec{F}_{\text{fluct}}$ is the force caused by *fluctuations*, which comes from the random variations of the environment and the stochastic pedestrian behaviors. Among them, the *desired force* can be simply represented as $\lambda(p_{\text{desired}} - p_t)$, where p_t and p_{desired} represent the pose of the agent and the target respectively. It is similar to the concept of *prefered velocity* in other planning methods [18]. The calculation of the *obstacle force* $\vec{F}_{\text{obs}}$ is based on [1] and we will not present the details here. We note that both the estimation of $\vec{F}_{\text{obs}}$ and $\vec{F}_{\text{fluct}}$ are omitted in this paper to prioritize the social aspects.

To successfully train our model in an interactive manner with all considered social scenarios, an efficient simulation environment that is capable of modeling pedestrians in socially acceptable behaviors is crucial. The existing robotics simulation environments for pedestrians typically do not come with embedded social force model, that the simulated pedestrians are not able to behave in social compliant manner. As an additional contribution of this paper, we release a plugin[1] for simulating socially compliant pedestrians under the framework of *Gazebo*, as is shown in Fig. 2.

III. Methods

In this paper, we formulate the problem of navigation in pedestrian environment as a *Markov decision process* (MDP). The state s_t is composed of the depth image x_t and the force towards the desired target $\vec{F}_{t_{\text{desired}}}$. Each action a_t corresponds to a moving command u_t to be executed by the mobile robot. Two parameterised networks, the generator (policy) network π_θ and the discriminator network D_ω are updated in an interleaving manner in the training phase.

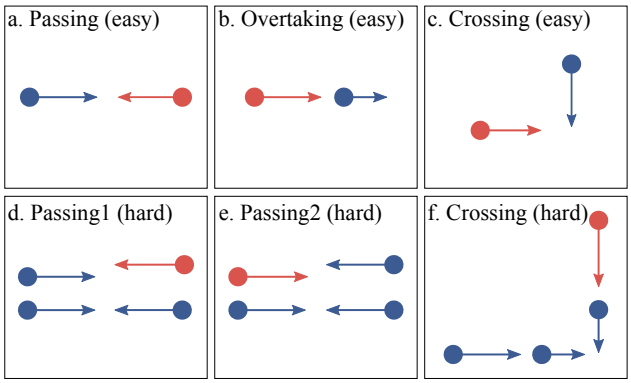


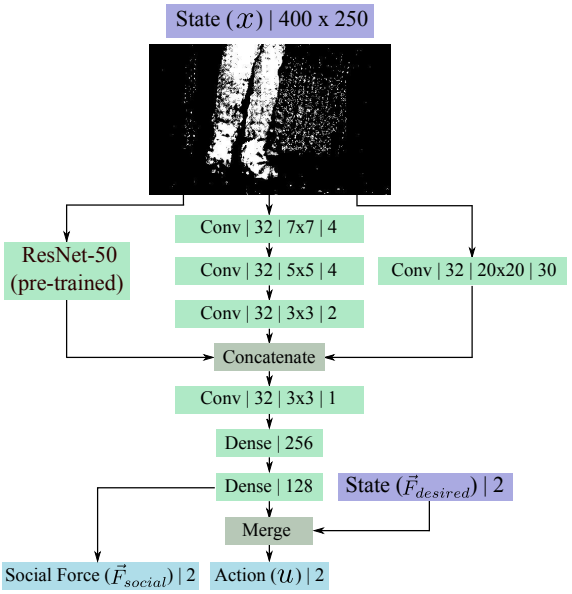
Fig. 3: Six basic scenarios considered in this paper to compare the navigation performance of the behavior cloning policy and the GAIL policy. We test the algorithm on a mobile robot from the view of the red agent.

A. Behavior cloning

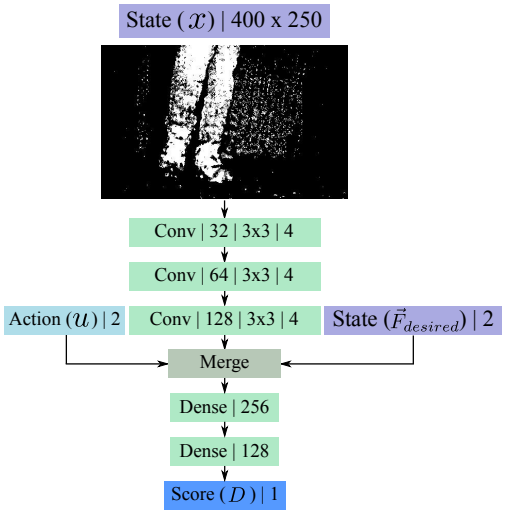
We initialize the policy generator network π_θ with the weights pre-trained via behavior cloning. To collect the training dataset and test the trained model, we identify the following most commonly encountered and typical social scenarios [3], which are shown in Fig. 3. Specifically, we collect trajectories from three relatively easy scenarios: (a) passing, (b) overtaking, (c) crossing, as well as three difficult ones: (d) passing out of a group of pedestrians, (e) passing between a group of pedestrians, and (f) crossing with a group of pedestrians. The differences between scenarios (d) and (e) are intended to motivate socially aware navigation strategies [2] that pedestrians walking close to each other should be regarded as a group, and the agent is expected to decide whether there is enough space between them for it to make a crossing. In our socially compliant pedestrian simulator, we collect data by mounting a depth sensor onto one of the pedestrians, to the height matching that of real-world setups. Then, the social force model, as described by Eq. 3, is used to label each incoming depth image with their corresponding social force. We note that only the pedestrians within the field of view (FOV) of the depth sensor (with a sensing range of $3.5m$, and a vertical sensing angles of $\pm 35^\circ$), are considered in the social force calculation. Desired force, represented by the normalized direction vector to the navigation target, are also collected as another input source for the model. We assign target locations for each episode, ensuring that the agent would have to encounter with pedestrians in the environment before reaching its targets.

As an additional contribution of this paper, we release the collected dataset[2] for further benchmarking for the literature. The first person view depth image, RGB image, social force, desired force and velocity of the agent are all recorded and included in the dataset. The final dataset contains 10,000 state-action pairs collected from social scenario shown in Fig. 3. The initial configuration (the

[1]https://github.com/onlytailei/gym_ped_sim
[2]https://ram-lab.com/file/tailei/tai_18icra_dataset.html



(a) Generator (policy) network.



(b) Discriminator network.

Fig. 4: Network structures for the generator and the discriminator network. Every convolutional layer is represented by its type, channel size, kernel size and stride size. Other layers are represented by their types and dimonsions.

position and velocity information) of the pedestrians and the agent are perturbed by random noise for each episode.

The model for behavior cloning is the same as the generator network of our GAIL model, which is depicted in Fig. 4a. The model takes the depth image and the *desired force* as input. The depth image is passed through three steams of feature extraction layers, to benefit the resulting representation from the residual learning of skip connection structures [19]. The extracted features are merged with the desired force and then used to predict the expert policies, as well as for performing a subtask: social force prediction. Learning on multiple tasks using the same set of features enforces the extracted features to be an effective and compact representation, which can greatly improve

Algorithm 1 Asynchronous GAIL

Collect expert trajectories $\mathcal{T}_E$.
Initialize π_θ with the behavior cloning policy θ_0.
Randomly initialize D_ω with ω_0.
for iteration $i = 0, 1, \ldots$ **do**
  for iteration $k = 1, K$ **do**
    Randomly choose a social scenario simulation.
    Sample one trajectory: $\mathcal{T}_{ik} \sim \pi_{\theta_i}$
  end for
  Ascending gradients of ω_i on mini-batches ($\kappa_i \sim \mathcal{T}_i$, $\kappa_E \sim \mathcal{T}_E$):
    $\Delta_\omega = \mathbb{E}_{\kappa_i}[\nabla_\omega D_\omega(s,a)] - \mathbb{E}_{\kappa_E}[\nabla_\omega D_\omega(s,a)]$
  Update to ω_{i+1} after clipping the weights to $(0.01, 0.01)$.
  Update θ_i to θ_{i+1} with the cost function $D_{\omega_{i+1}}(s,a)$ under the TRPO rule.
end for

the generalization ability of the model and prevent overfitting [20]. This setup also makes it possible to transfer our trained model naturally to different motion planners, as the social force prediction part can stay intact regardless of the mobile platforms in use.

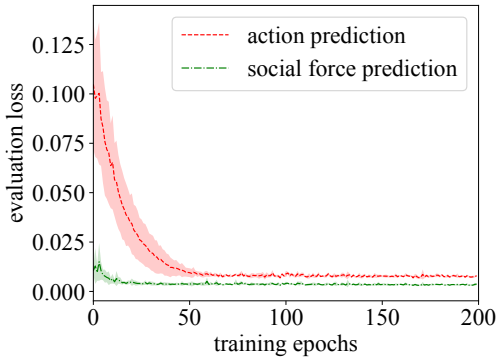


Fig. 5: Average loss $\pm$ one standard deviation for supervised behavior cloning of social force and action predition in 200 epochs. It is used as the initial policy.

B. Algorithm

The original GAIL [13] requires training one model for each specific task. The trajectories sampled from the environment are quite similar across episodes, which greatly limits the generalization of the trained model. Inspired by various asynchronous methods in the deep reinforcement learning literature (Async-DRL) [21], we propose to perform generative adversarial imitation learning in a modified training procedure. Specifically, during training, our model interacts with several different social scenario simulations in an interleaving manner. The resulting model thus converges to a generalized policy across all social scenarios and is able to perform well on all considered tasks. We note that our proposed method differs from the other Async-DRL methods in that, those previous approaches have multiple instances of the environment of

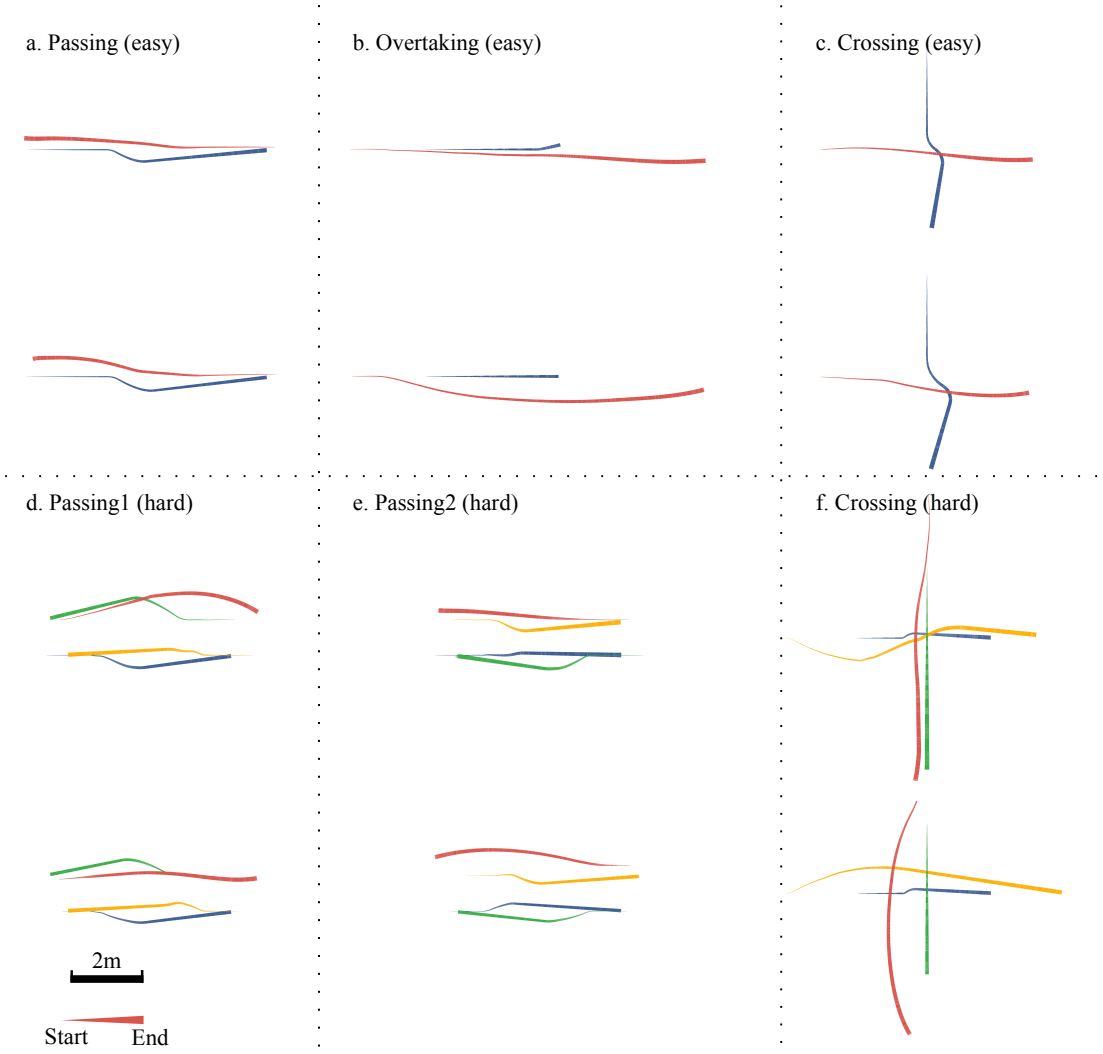


Fig. 6: Navigation trajectories of test episodes. In every scenario, the upper part is the performance of the model purely trained from behavior cloning and the lower part is the performance of the model optimized by GAIL. The thickness of the path increases from the starting position of the agent to its ending point. Red paths are the trajectories executed by the mobile robot under the trained policy. The other agents are moving under the social force model [1].

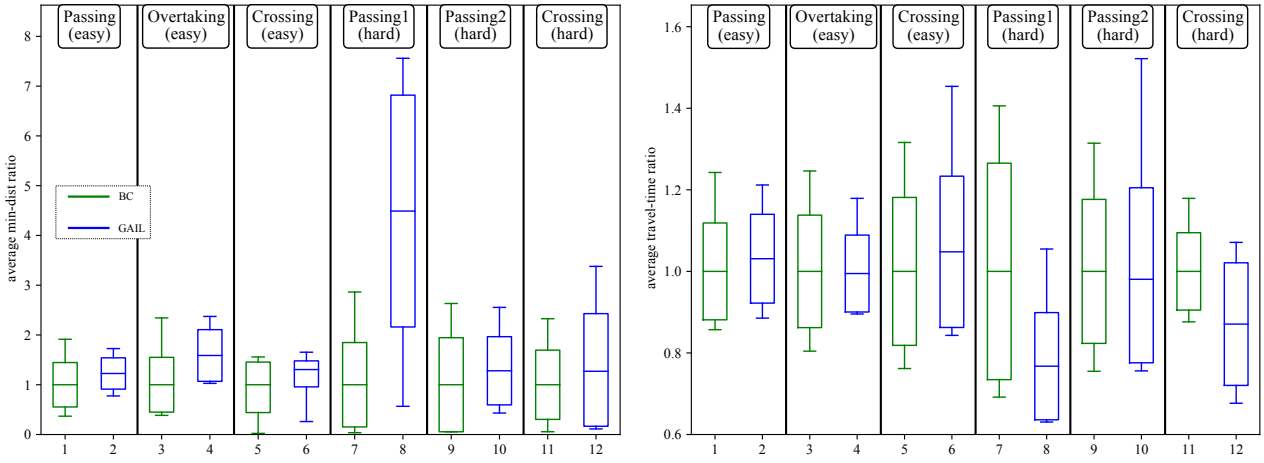
the same task, while for our approach, each instance of the environment corresponds to a different social task.

We describe our approach in detail in Algorithm 1. We initialize the generator (policy) network of our GAIL model by the pre-trained weights θ_0 from behavior cloning. The discriminator network is initialized randomly with ω_0. In every training step, the trajectories τ_i are sampled from different social scenario simulations. The sampled trajectories τ_i and the sampled expert trajectories τ_E are then fed into the discriminator network. The weight of the discriminator network is clipped between $(-0.01, 0.01)$ to update to ω_{i+1}, to fulfill the constraint of WGAN [17]. After that, θ_i is updated to θ_{i+1}, by following the TRPO rule for updating the policy parameters. The discrimination score of the state action pair $D_{\omega_{i+1}}(s, a)$ are used as the cost for policy gradient optimization. As [14], in the trajectories sampling procedure, an augmented cost punishing the collision with pedestrians is added, in case that demonstrations under social force model are not optimal enough.

As mentioned before, the generator model has the same architecture as the model used for conducting behavior cloning, which is shown in Fig. 4a. The discriminator, on the other hand, contains only one stream for feature extraction, the resulting features are concatenated with the *desired force* of the corresponding step. Then, they are merged with the selected action, which is then fed to several fully-connected layers to estimate a score to tell apart generated policies from expert policies. The details of the discriminator architecture are shown in Fig. 4b.

IV. Experiments

We begin our experiment by training the behavior cloning model. First, we collect training data from different social scenario simulations, as shown in Fig. 2. To add to the variations of the training dataset, we collect sample trajectories from the perspective of both the agent (shown in red in Fig. 3) and the pedestrians (shown in blue in Fig. 3). The social force model, as is described in Sec. II-C, is



(a) *min-dist*: average minimum distance ratio

(b) *travel time*: average travel time ratio

Fig. 7: Average minimum distance to pedestrians and the average travel time of 10 episodes in all test scenarios. Green boxes represent results from the BC policy, and the blue ones show results from the GAIL policy. In every scenario, all the statistics are normalized by dividing the mean of the BC policy.

utilized to generate the expert policy. In total, 10,000 state-action pairs are collected, containing trajectories from all the considered social scenarios, with randomized starting configurations. 2000 samples are separated for evaluating the pre-trained model.

To train the behavior cloning model, we use *RMSprop* with a learning rate of $1e{-}4$, and decay it by a factor of 0.9 after every 20 epochs. We experiment with different weight ratios for the two supervised tasks (social force prediction, action prediction, as shown in Fig. 4a). We can observe from Fig. 5 that those tasks are effectively learned.

Then we start the training of our model. As mentioned before, we initialize the generator network of the GAIL policy with the pre-trained behavior cloning model. In every training step, 3 trajectories are sampled from 6 randomly chosen scenarios as Algorithm 1, which add up to a total number of 1000 samples approximately. This training strategy is highly effective for learning a generalized policy over all considered tasks, and greatly improves the generalization capability of our trained model across the most commonly encountered social scenes.

We train GAIL for 300 iterations, in a duration of 10 hours, on a Tesla P100 graphics processing unit. The discriminator is optimized through *RMSprop* with a learning rate of $5e{-}5$. The generator (policy) network is optimized following the TRPO rule.

In the following, we present both quantitative and qualitative results, which clearly show the effectiveness our approach. In particular, we can observe that the pre-trained policy via behavior cloning is only able to perform some reactive actions, since it completely ignores the temporal correlations in the sample trajectories, thus is only able to make frame-wise decisions; while the improved policy via GAIL is able to drive the agent to behave in a socially-compliant manner, due to its underlying formulation that takes the underlying MDP of the trajectories into account, which enables it to make planning decisions on the horizon of a whole trajectory.

A. Evaluation in simulated environments

To ensure a fair comparison between test runs, we remove all the randomness that is included in the training setups. A simulated *Turtlebot3 waffle* is used to execute 10 test episodes in each scenario, under both BC policy and GAIL policy, taking the depth image and the *desired force* as input. The other pedestrian agents in the environment are navigating under the social force model, taking in also the social force from the mobile robot.

1) Qualitative evaluation: Both the trajectories of BC policy and GAIL policy are shown in Fig. 6. For each scenario, we show the BC policy performance in the upper part and the GAIL policy in the lower part. We increase the thickness of the trajectory from the starting point to the ending point. The trajectories of the robot are shown in red, the trajectories of pedestrians are shown blue. We can clearly observe that agent performing under the optimized policy by GAIL is able to navigate in a more socially-compliant manner in all considered scenarios. In particular, in scenarios (a), (b), (c) and (f), the GAIL policy guides the robot further away from the pedestrians. Also, the GAIL policy successfully guides the mobile robot to pass in between (d) and travel out of (e) a group of pedestrians, which the BC policy fails to accomplish.

We note that limited by the small FOV, the robot is only able to perceive the pedestrians when they are fairly nearby, especially in the crossing scenario (c). This leads to some sudden turnings shown in the trajectories.

2) Quantitative evaluation: We choose two metrics to perform quantitative evaluation: (1) *min-dist*: the minimal distance from the robot to other pedestrians in one episode and (2) *travel-time*: the time taken by the robot to travel from the starting location to the target. Those statistics are shown in Fig. 7.

From Fig. 7, we can observe that our GAIL model performs much better than the behavior cloning baseline. Fig. 7a shows that it clearly converges to a much safer navigation strategy as its average *min-dist* to pedestrians is larger than that of the baseline in all six social scenarios; as for the *travel-time*, it plans more efficient paths in most scenarios as can be seen in Fig. 7b, but performs slightly slower than that of behavior cloning in two easy scenarios. Those two statistics show that the GAIL model learns to navigate in a much more socially acceptable way, and is able to plan paths that are both safe and efficient.

B. Real world experiments

We also conduct real-world experiments to test the performance of our approach in realistic scenarios. We use a *Turtlebot waffle* platform, which is shown in Fig. 8. It navigates autonomously in an indoor office environment as shown in Fig. 1, under the improved GAIL policy. A low-cost laser range sensor is used to localize the robot in the environment. Navigation targets are chosen randomly on the collision-free areas of the map. The depth image, captured by an obboard Intel Realsense R200, is cropped to 400×250 and fed into the trained model. A Nvidia Jetson TX2 is mounted for real-time neural network processing. The model control cycle runs in real-time at 15Hz. A video showing the real-world performance of our trained policy can be found in `https://goo.gl/42yf6f`.

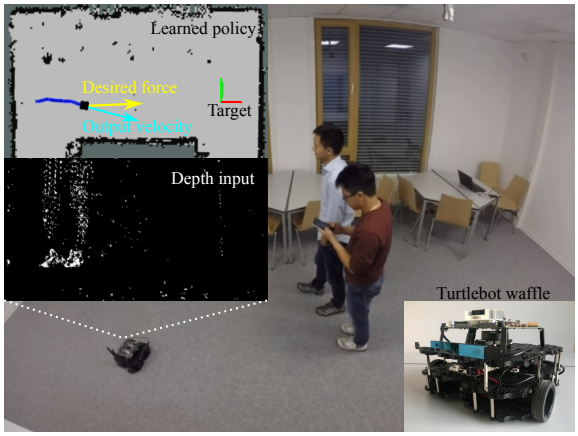


Fig. 8: Real world experiments in an indoor office environment, through a *Turtlebot waffle* platform by taking raw depth images and desired forces as inputs.

V. Conclusion

We presented an approach for agents to learn to navigate in a socially compliant manner, from raw depth images. Our algorithm is able to guide the agent to perform socially compliant behaviors, as well as plan efficient paths to reach its goal location. We validated our approach in both simulated and real world experiments; moreover, we release a plugin for simulating pedestrians under social force model, as well as a dataset collected from our simulation environment.

The performance of the GAIL policy in real-world environments is influenced by the limited maximum speed and the FOV of the *Turtlebot waffle*. We leave it as future work to evaluate the proposed algorithm on more compatible platforms. Recent work [22] of real-to-sim domain adaptation for visual control also makes it possible to tackle this problem through raw RGB images.

References

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

[2] M. Pfeiffer, U. Schwesinger, H. Sommer, E. Galceran, and R. Siegwart, "Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2096–2101.

[3] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *IROS*, Sept 2017, pp. 1343–1350.

[4] L. Tai, S. Li, and M. Liu, "A deep-network solution towards model-less obstacle avoidance," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 2759–2764.

[5] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, *et al.*, "Learning to navigate in complex environments," in *ICLR*, 2017.

[6] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2017, pp. 31–36.

[7] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard, "Deep reinforcement learning with successor features for navigation across similar environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2017, pp. 2371–2378.

[8] J. Zhang, L. Tai, J. Boedecker, W. Burgard, and M. Liu, "Neural slam: Learning to explore with external memory," *arXiv preprint arXiv:1706.09520*, 2017.

[9] 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.

[10] B. Okal and K. O. Arras, "Learning socially normative robot navigation behaviors with bayesian inverse reinforcement learning," in *ICRA*, May 2016, pp. 2889–2895.

[11] A. Y. Ng, S. J. Russell, *et al.*, "Algorithms for inverse reinforcement learning." in *ICML*, 2000, pp. 663–670.

[12] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *ICML*, 2004, p. 1.

[13] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *NIPS*, 2016, pp. 4565–4573.

[14] Y. Li, J. Song, and S. Ermon, "Infogail: Interpretable imitation learning from visual demonstrations," in *NIPS*, 2017.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[16] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *ICML*, 2015, pp. 1889–1897.

[17] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[18] 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*, May 2008, pp. 1928–1935.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, June 2016.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, vol. 1.

[21] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *ICML*, 2016.

[22] J. Zhang, L. Tai, Y. Xiong, M. Liu, J. Boedecker, and W. Burgard, "Vr goggles for robots: Real-to-sim domain adaptation for visual control," *arXiv preprint arXiv:1802.00265*, 2018.