Reinforcement Learning Based Shared Control Navigation

Bingqing Zhang
GDI hub/Computer Science
University College London
London, United Kingdom
bingqing.zhang.18@ucl.ac.uk

Catherine Holloway
GDI hub/Computer Science
University College London
London, United Kingdom
c.holloway@ucl.ac.uk

Tom Carlson
Aspire Create
University College London
London, United Kingdom
t.carlson@ucl.ac.uk

***Abstract*—Shared control is a mode where the user input is combined with a planned motion to achieve a common goal. In navigation, a shared control approach could provide a potential mobility solution for people who have a mobility impairment and find traditional powered wheelchairs unsuitable. While state-of-the-art work explored shared control navigation in simple environments, it is still challenging to solve for dynamic, crowded scenarios, in a way that is acceptable to users. Learning from recent advances in robot navigation, we present a reinforcement learning based framework, which allows navigation to be achieved in a shared controlled way. Our approach was trained and tested in a Unity3D based simulator. It achieved fewer collisions, comparable completion time and relatively high user agreement when compared with other state-of-the-art methods.**

***Index Terms*—shared-control, reinforcement learning, wheelchair**

I. Introduction

A smart wheelchair is a type of robot that is normally built on a standard powered wheelchair, with a collection of sensors for perception and navigation purposes. Based on the level of autonomy and the amount of assistance, the work in this area can be classified into: fully autonomous [1] and semi-autonomous [2] (including shared control [3], [4]). Shared control is generally preferred by many users due to their ability to maintain high user authority [5]. While previous research in shared control navigation has focused on simple environments, the challenge of crowds has recently attracted attention. Traditional methods treat pedestrians as obstacles, leading to "freezing robot" issues [6]. To address this problem, it is important to understand the interactions between pedestrians and the robot (or wheelchair). To this end, deep reinforcement learning (RL) approaches have been actively studied due to their reported high performance and robustness to changes in the environment.

While prior work in crowd navigation has focused on fully autonomous robots, we propose a shared control framework that incorporates both imitation learning for generating diverse driving policies and deep RL for navigation. Our approach is evaluated through simulations, and we contribute both the driving policy generation and shared control framework as solutions to the problem.

Funded by EU H2020 Crowdbot project

II. Related Work

A. Navigation in Crowded Environments

Navigation of robots in crowded environments has been a popular research topic for decades. Early approaches employed 'social force' models to capture attractive and repulsive interactions between humans. However, these models did not take into account potential human-robot cooperation. Recent works have approached social-aware navigation in highly dynamic human environments using either model-based [6] or learning-based approaches [7]–[9].

Model-based methods initially used proxemic potential functions to model human-robot interactions [10], [11], but ignored human-robot cooperation. To address this limitation, Trautman (2015) proposed Interacting Gaussian Process (IGP), which modelled the robot as one of the agents, and subsequently modelled a joint distribution describing their interaction [6]. However, hand-crafting the interaction function can be challenging. On the other hand, learning-based methods have started gaining more attention due to their ability in capturing complex natural human-human and human-robot interaction directly [7]–[9]. Tai et al. (2018) used imitation learning to learn a direct map from sensor inputs or map data to the motion command [12]. Similarly, in [13], the interaction features and the cost function are learned from demonstration by using inverse reinforcement learning (IRL). The learning outcomes for these methods are highly dependent on the scale and quality of the demonstration, and it is normally difficult to then generalize to other scenarios. Some other approaches, which used RL and leverage agent-level information in presenting the crowd structure, showed promising performance in both simulation and real-world tests [7], [14].

B. Learning-based Shared Autonomy

While fully autonomous robots have their place, collaborative work between humans and robots may be preferred in many scenarios. However, there is limited research on using deep RL in a control sharing setting. Reddy et al. (2018) addressed this issue by decomposing the reward function into two parts: one part captures general requirements such as collision avoidance, while the other captures user-generated feedback [15]. The control sharing is achieved by involving

user feedback in the reward function, and deep neural networks are used to discover arbitrary relationships between user controls and observations of the physical environment directly. However, this method requires discrete human input during training, which could be impractical and problematic when continuous input is required.

To address this limitation, Schaff et al. (2020) proposed a model-free, residual policy learning algorithm for shared autonomy in a continuous action space [16]. They created a surrogate user by behavior cloning and augmented this user policy with a learnt residual policy. This approach eliminates the need for continuous human input during training. Their method was evaluated on continuous gaming tasks, such as Lunar Lander, Lunar Reacher, and Drone Reacher, and showed significant improvement in the performance of human operators. However, as far as we know, such a method has not been applied in shared-control wheelchair navigation.

III. Background

A. Imitation Learning

Imitation learning techniques aim to mimic human behavior in a given task [17]. It is normally achieved by training a model from a fixed set of observation-action samples (or trajectories) obtained from some expert. One of the most popular imitation learning techniques is behavior cloning (BC). BC uses supervised learning to directly learn a mapping between observations and actions. It has been widely explored in autonomous driving [18], [19] and its performance depends on the amount and quality of the training data due to compounding error. Alternatively, researchers have been using IRL to learn a cost function that prioritizes entire trajectories over others. While these methods have been applied successfully in areas such as robot navigation [13], it is computationally expensive as it requires RL as the inner loop. In addition, IRL only provides the cost function, which requires further techniques to generate the policy.

Generative Adversarial Imitation Learning (GAIL) was proposed in [20]. It introduces a framework that directly learns policies from data, bypassing any intermediate IRL step. This model-free imitation learning algorithm that is able to handle complex and high-dimensional environments. Its working mechanism is similar to Generative Adversarial Network (GAN) where a generator aims to confuse a discriminator D that learns to discriminate between the true data distribution and the one being generated. Specifically, GAIL achieves this by finding a saddle point of this expression:

$$E_{\pi_\theta}[log(D(s,a))] + E_{\pi_E}[log(1 - D(s,a))] - \lambda H(\pi_\theta) \quad (1)$$

where π_E is the expert policy and π_θ is the policy we want to learn, characterized by θ. $H(\pi_\theta)$ is the causal entropy of the policy π_E. During learning, GAIL uses Adam gradient-based optimization to update the parameter w for the discriminator D that increases equation (1), while performing trust region policy optimization (TRPO) [21] with respect to θ to decreases equation (1).

In general, GAIL is sample efficient for the expert data, while it may requires heavy environmental interaction during training. A typical way to improve the learning speed is to use BC for initializing the policy parameters [20].

B. Reinforcement learning via PPO

Robot navigation can be considered as a sequential decision problem, which can be modelled as Markov Decision Process (MDP) or Partially Observed Markov Decision Process (POMDP). The main components of MDP include states S, actions A, transitions T, reward function R, and a discount factor $\gamma \in [0, 1]$, where the optimal policy maximizes the expected future discounted return. However, in real life navigation situations, many states such as the pedestrians' goal are not fully observable. In this case, the problem can be modelled as POMDP by adding an additional set of possible observations Ω and observation function O : $S \times \Omega \rightarrow$ [0, 1]. While POMDP is normally not tractable, an estimation can be made through RL.

One of the most widely used policy-based algorithms, proximal policy optimisation (PPO), has demonstrated promising results [22] and has become a popular RL algorithm. In this work, we decided to use PPO as our main RL algorithm as it handles continuous action spaces, and directly modifies the policy during training, which is suitable for navigation applications. While many RL methods suffer from stability issues, PPO guarantees stability during training by setting a trust region [21].

C. Residual Policy Learning

Shared control navigation can also be formed as a POMDP problem where the user's goal (or short term intention) is partially observed by the agent (wheelchair). State-of-the-art data driven approaches infer the user's goal using IRL [23], [24] or hindsight optimization [25]. However, these methods normally require a known user goal space or a transition function which can be difficult to obtain. Recently, Silver et al., (2019) proposed Residual Policy Learning (RPL) [26], which aims to improve non-differentiable policies using model-free deep RL. The main idea of RPL is to learn a residual policy $\pi_\theta(s)$ from a residual function $f_\theta(s)$ and augment on top of some arbitrary initial policy $\pi(s)$.

$$\pi_\theta(s) = \pi(s) + f_\theta(s) \quad (2)$$

By evaluating the performance on a robot picking task, with a hand designed policy and a model predictive controller (MPC) as initial policies, [26] showed that RPL not only improves on initial policies but is also more data-efficient than learning from scratch. While this method could help with imperfect controllers, it can also be combined with imitation learning, where the user's demonstration is imperfect. Schaff et al., (2020) has applied residual policy learning in continuous action space and evaluated it in simple gaming tasks [16]. Our work was inspired by their work, while we used imitation learning for obtaining the user policy and extended the application scenario to shared control navigation in crowds.

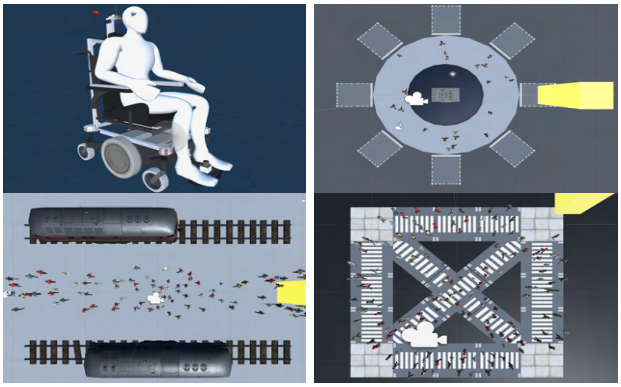

Fig. 1. (a) The simulated wheelchair (b) Crowds Scenario 1 (c) Crowds Scenario 2 (d) Crowds Scenario 3

IV. Method

A. Problem formulation and setup

We tackle the challenge of providing personalized assistance in shared control wheelchair navigation in crowds by dividing the problem into two sub-problems: learning the user's driving style and achieving shared control navigation in real-time. To evaluate the generalizability of our approach, we designed three scenarios in Unity3D with realistic pedestrian dynamics and collision avoidance. In these scenarios, we also accounted for the human tendency to get distracted, by designing a subset of pedestrians to ignore the wheelchair. Our simulated wheelchair is governed by a differential PID controller that considers non-holonomic constraints, with maximum linear velocity of 1.3 m/s and maximum angular velocity of 0.785 rad/s. These values are set to be comparable with typical human walking speed to observe potential interaction behaviors [27].

B. Learning user driving style

Ethics approval for the study was granted by UCL Ethics Committee with ID 6860/011. 20 healthy participants from the UCL Psychology participant pool were recruited, all of them are over 18, with unimpaired vision and wrist mobility. Each of them was given 10 minutes to get familiar with the setup before performing the actual experiment. They were then instructed to drive a simulated wheelchair with a joystick for 45 minutes in three simulated scenarios (15 minutes for each scenario). Participants were asked to drive the wheelchair in their own style and reach a goal (highlighted in yellow) in each scenario (Fig. 1). Once the goal was reached, they were brought to a new scenario generated in a random sequence at a random starting position. During the data collection, participants have the first-person view of the wheelchair. We collected all the wheelchair data and environment data during the experiment, at the frequency of 10Hz. A trial is defined when the user completed all three scenes. In total, 32 valid trials have been completed which gave us 640 valid trajectories.

We performed filter-based feature selection on the data. By iteratively evaluating the silhouette value for K-means clustering using different feature combinations, we found that "The average distance to the nearest pedestrian", "The average linear velocity", "The average angular acceleration" and "The average collision number" gave us the best description of the driving data. Principal component analysis was used to further reduce the data's dimensionality and produced two principal components with explained variance $> 99.5\%$. While one user could potentially exhibit different driving styles, we assume their driving is consistent throughout this experiment. This allowed us to categorize the user data into two driving styles "aggressive" and "non-aggressive" using K-means clustering, with the result verified by silhouette coefficient > 0.6.

In the literature, behaviour cloning has been used to learn user policies from human demonstrations [28]. When it comes to the crowd navigation scenario, the high-dimensional states make it impracticable to directly learn a motion output from limited and potentially imperfect human demonstrations. As a result, we use Generative Adversarial Imitation Learning (GAIL) [20] to learn a user policy, which not only leverages the information in the demonstration but also allows exploration of other unvisited states. As pure GAIL could be sample-inefficient during training, we initialize the parameters with behaviour cloning of the recorded user data.

C. Shared control via reinforcement learning

For shared control navigation, continuous user input is required throughout the whole process. This characteristic differs from other works, where discrete user input is provided as feedback [15]. In addition, continuous user input will require extensive user interaction during training, the amount of training time may take tens of hours based on the difficulty of the task and the complexity of the scenario. It is simply impractical for every wheelchair user to be supervised for that long before they can actually use the wheelchair. Therefore, we approached the problem using residual policy learning, where the initial policy comes from a *surrogate* user that is pre-trained offline. During training, the user input is sampled from the surrogate user policy, which is augmented with the robot states. The control sharing is then achieved by shaping the rewards into two parts – one that takes care of user inputs and the one that deals with motion planning.

In our work, the wheelchair must be able to assist the user in avoiding static obstacles, navigating through crowds and reaching their final goal. To form this as a RL problem, the states S, action A and reward function R should be designed carefully. While the states could be the raw sensor information, this would create very high-dimensional states and require high computational resources. Inspired by previous work in crowd navigation [7], we used agent-level information in our state. For static obstacles, we detected them using a Ray module provided by Unity. For simplicity, only three rays were used, with one pointing to the front of the wheelchair and the other two pointing to each side. The angle between each ray is 60 degrees. Each ray is associated with three states:"hit the tagged

obstacle", "no hit", and "hit fraction". This gives us 9 states s_o in total.

Consequently, the final state of the robotic wheelchair is set as $[s_w, s_o, s_u]$, where $s_w = [g_x, g_y, v_r, w_r, p_{ix}, p_{iy}, p_{ivx}, p_{ivy}]$. g_x, g_y is the position of the goal, v_r, w_r represents the velocity of the wheelchair. p stands for the pedestrian-related information. In this paper, i takes range from 0 to 9 which includes the 10 nearest pedestrians within 5 m of the wheelchair. This value is set by considering the range of the people tracker which gives us a local density about $1p/m^2$. If fewer than 10 people are detected around the wheelchair, the position values are padded with the maximum range and the velocity values are filled with 0. All values are wheelchair-centric. During training, the wheelchair and environment related states are determined by the simulator, while $s_u = [v_u, w_u]$ are inferred from the surrogate user model. To better capture the human-robot interaction and the user intention, we used a window of 3 time-steps (0.3 s) for the states (i.e. state input at time t will be observations at time $t-2, t-1, t$) and a recurrent neural network. The final action is the desired linear and angular velocities for the next time step.

The key to achieving shared control via RL is shaping the reward. Inspired by [15], [16], we divided the reward function R^t into two parts, one part R_r^t solves basic robot navigation requirements such as: avoid collisions (R_c^t); reach the final goal (R_g^t); and encourage smoother trajectories by penalizing sudden large changes in angular velocity (R_{com}^t); while the other part (R_u^t) rewards solutions that most closely follow the user's intention. Fig. 2 shows a high-level summary of our proposed approach.

$$R^t = (R_r^t) + (R_u^t) \tag{3}$$

$$R_r^t = R_g^t + R_c^t + R_{com}^t \tag{4}$$

$$R_g^t = \begin{cases} r_d * (\left\|\mathbf{p_r^{t-1}} - \mathbf{g}\right\| - \left\|\mathbf{p_r^t} - \mathbf{g}\right\|) & \text{Otherwise} \\ r_g & \text{if } \left\|\mathbf{p_r^t} - \mathbf{g}\right\| \leq 2 \end{cases} \tag{5}$$

$$R_c^t = \begin{cases} 0 & \text{Otherwise} \\ r_c & \text{if } \left\|\mathbf{p_r^t} - \mathbf{p_i^t}\right\| \leq 0.66 \\ r_{cc} * \left\|\mathbf{p_r^t} - \mathbf{p_i^t}\right\| & \text{if } \left\|\mathbf{p_r^t} - \mathbf{p_i^t}\right\| \leq 1.5 \end{cases} \tag{6}$$

$$R_{com}^t = -0.01 * |w^t| \tag{7}$$

Where $\mathbf{p_r^t}$, $\mathbf{p_i^t}$ are the position vectors of the robot and pedestrians at time t, and $\mathbf{g}$ represents the position of the goal. In our implementation, r_d = 5, r_g = 50, r_c = -20,r_{cc} = -2.

In terms of the user, we use a function to evaluate the consistency of the wheelchair motion candidate with the surrogate user policy output.

$$R_u = r_a * \exp^{\lambda * (\left\|\mathbf{v_r^t} - \mathbf{v_u^t}\right\|^2 + \left\|\mathbf{w_r^t} - \mathbf{w_u^t}\right\|^2)} \tag{8}$$

Parameter λ controls how closely the user input and the final motion are correlated. Through trial and error, we found λ=-0.5 gives us a reasonable trade-off. r_a is set to 0.5.

V. Experiment Validation

A. Implementation details

All training and testing in the simulator are implemented in Unity 3D 2019.3.14f, with *mlagents* version 15. The laptop has Intel® Core™ i7-9750H CPU @ 2.60GHz × 12, with graphics card GeForce RTX 2070 with Max-Q Design/PCIe/SSE2. All the training hyperparameters are listed in Table I and II.

Parameter	Value
Algorithm	GAIL
Gamma	0.99
Num_layers	2
Normalize	true
Hidden_units	128
Learning_rate	0.003
Use_actions	False
Use_vail	False
behaviour Cloning	10000 steps

TABLE I
IL Hyperparameters

Parameter	Value
Algorithm	PPO
Time_horizon	128
Batch_size	1024
Beta	0.005
Buffer_size	4096
Epsilon	0.2
Lambda	0.95
Learning_rate	0.0005
Num_layers	2
Normalize	true
Hidden_units	256
Gamma	0.99

TABLE II
RL Hyperparameters

The imitation learning was performed using GAIL without extrinsic rewards in all scenarios. The user policies converge after approximately 60k steps. In terms of the final shared control policy, we adopted a curriculum learning strategy due to the complexity of the task. During some initial trials, we noticed that populating all the human agents at the very beginning of the training may result in undesirable behaviours, which include the wheelchair wandering around in the free area or colliding with obstacles immediately to end the episode, as it expects more negative cumulative reward during the trip to reach the goal. Therefore, we break down the learning objectives by first training the wheelchair to reach the goal in a human-free environment. After that, we keep populating crowds until the local density reaches $1p/m^2$. During training, the wheelchair starting position was random generated. For each scenario, we used 5 random seeds and report a 95% confidence interval for the final averaged cumulative return.

The final policy converged after about 400k, 200k, and 600k steps for the three scenarios. Training S3 requires the longest time as it consisted of the most crowd patterns.

B. Qualitative Evaluation

Through initial user policy training using GAIL, we obtained two surrogate user models with different driving styles (aggressive and non-aggressive). We evaluated the user policy generated by these models, and compared it with the final shared policy given actual user input from a self-identified non-aggressive user. Fig. 3 shows the robot and pedestrian trajectories in a simple passing and crossing cases (which is a part of S3). We can see that the non-aggressive surrogate user model well represented the actual user driving style, while differ itself from the aggressive user model.

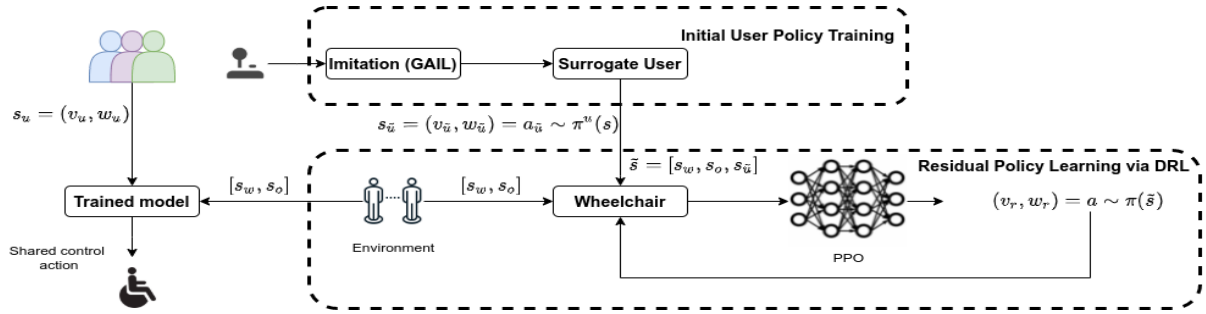

Fig. 2. A summary of our proposed approach. The state input $\tilde{s}$ to the neural network is a combination of the robot(wheelchair) information, environment(crowds) information and the predicted input comes from the surrogate user. The final shared control policy for the wheelchair is achieved by residual policy learning and is guided by the initial user policy obtained from imitation learning.

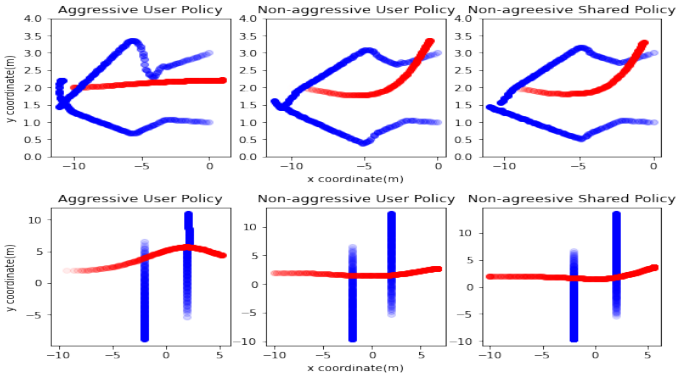

Fig. 3. Trajectories for simple passing and crossing cases. Color alpha shows the moving direction (start from transparent). Red represents the robot while blue represents the pedestrians. (a) Passing (b) Crossing

C. Quantitative Evaluation

1) Metrics: We used three basic metrics to evaluate the safety and assistance for the proposed RL-based shared control navigation design. These are defined as:

- C: Number of collisions (with pedestrians). This metric is just the count of collisions that occurred in the scenario and was reported by the simulator.
- T_c : Task completion time. The time that user required to reach the goal position from the starting position.

$$T_c = t_{end} - t_{start} \tag{9}$$

- A: Agreement. We define agreement in terms of the deviation of the direction of the user's command from the direction of the final shared control's command. Mathematically, it is calculated as:

$$a_i = 1 - \frac{\theta(z_u^i) \ominus \theta(z_{sc}^i)}{\pi} \tag{10}$$

$$A = \frac{\sum_{i=0}^{N} a_i \cdot \Delta t_i}{\sum_{i=0}^{N} \Delta t_i} \tag{11}$$

where $\theta(z) = tan^{-1}(\frac{v}{w})$, v and w are the translational and rotational velocities, a_i is the normalised agreement at time step t_i. N is the number of samples available in which data from the measured user input z_u^i coincide in time with the final shared control output z_{sc}^i, and Δt_i is the duration of the user's input command.

We tested the performance of our approach (RLPSC) with one user giving input through a joystick, in all three simulated crowd scenarios (See Fig. 1), with local crowd density $\geq 1p/m^2$. It was compared with our previous work which instead uses a velocity-based probabilistic shared control (GVDWAPSC) [29]. In addition, direct user input without any assistance was used as a baseline. The user was first given 10 minutes to get familiar with the setup, and was asked to self-identify their preferred driving style. Then the user drove the wheelchair (no assistance) in three simulated scenarios for 10 minutes. The collected data was used to determine the suitable assistance model, which turned out to be "non-aggressive" and was consistent with the user's self-identification. During testing, the wheelchair started at a random position which was kept the same across different methods. The user drove the wheelchair to the goal and complete one trial. Each method was tested for 5 trials in each scenario and Fig.4 gives a summary of the evaluation results.

It can be seen that although both RLPSC and GVDWAPSC had one collision, they reduced the number of collisions with pedestrians substantially compared to the one without assistance. While the learning-based methods achieve collision avoidance by setting negative rewards, they do not guarantee collision-free behaviour, especially in challenging scenarios. On the other hand, while the velocity-based PSC required longer to reach the goal, potentially due to "freezing" when the crowd density increases, the RL-based method had a similar completion time to the no-assist one. This implies that the proposed method is promising in learning crowd interactions, and moves the wheelchair in a safe and socially-compliant manner. In terms of the agreement, both RLPSC and GVDWAPSC had a similar value at about 0.88, which shows good control sharing performance in general.

VI. Conclusion and Future Work

In this paper, we proposed a RL based shared control approach for wheelchair navigation in crowds. To address the challenge of involving humans in the training loop, a

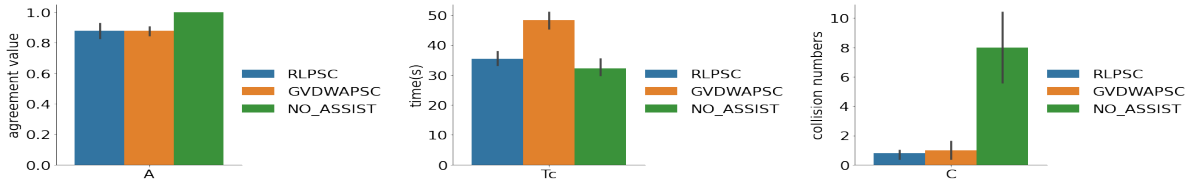

Fig. 4. Performance results for RLPSC, GVDWAPSC and NO_ASSIST: (a) User-wheelchair agreement (b) Time to complete (c) Number of collision.

surrogate user was trained by GAIL that learns the driving style from the collected user data. The final shared control policy combined the predicted user input, the wheelchair state, as well as agent-level crowd information, and was trained to provide suitable driving assistance. The performance of our proposed RL method has been evaluated in simulated circular crowds, 1D crowds and 2D crowds scenarios and showed promising navigation performance, while obtaining relatively high user agreement when compared with other state-of-the-art approaches. In future, we would like to collect data, evaluate the model performance and its usability in various challenging crowd scenarios on actual wheelchair users. In addition, while we assumed agent-level state input for simplicity, we are interested in exploring end-to-end RL, where the raw sensor inputs could be used directly to map to the navigation decision, and test our approach in real world.

REFERENCES

[1] E. Prassler, J. Scholz, and P. Fiorini, "A robotic wheelchair roaming in a railway station," in *Proc. Int. Conf. Field and Service Robotics, Pittsburgh*, pp. 31–36, 1999.

[2] A. Escobedo, A. Spalanzani, and C. Laugier, "Multimodal control of a robotic wheelchair: Using contextual information for usability improvement," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4262–4267, IEEE, 2013.

[3] Q. Li, W. Chen, and J. Wang, "Dynamic shared control for human-wheelchair cooperation," in *2011 IEEE International Conference on Robotics and Automation*, pp. 4278–4283, IEEE, 2011.

[4] C. Ezeh, P. Trautman, L. Devigne, V. Bureau, M. Babel, and T. Carlson, "Probabilistic vs linear blending approaches to shared control for wheelchair driving," in *2017 International Conference on Rehabilitation Robotics (ICORR)*, pp. 835–840, IEEE, 2017.

[5] E. A. Biddiss and T. T. Chau, "Upper limb prosthesis use and abandonment: A survey of the last 25 years," *Prosthetics and Orthotics International*, vol. 31, no. 3, pp. 236–257, 2007. PMID: 17979010.

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

[7] 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, 2019.

[8] L. Liu, D. Dugas, G. Cesari, R. Siegwart, and R. Dubé, "Robot navigation in crowded environments using deep reinforcement learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5671–5677, 2020.

[9] Z. Zhou, P. Zhu, Z. Zeng, J. Xiao, H. Lu, and Z. Zhou, "Robot navigation in a crowd by integrating deep reinforcement learning and online planning," *Applied Intelligence*, vol. 52, pp. 15600–15616, mar 2022.

[10] M. Svenstrup, T. Bak, and H. J. Andersen, "Trajectory planning for robots in dynamic human environments," *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4293–4298, 2010.

[11] N. Pradhan, T. Burg, and S. Birchfield, "Robot crowd navigation using predictive position fields in the potential function framework," in *Proceedings of the 2011 American Control Conference*, pp. 4628–4633, 2011.

[12] 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)*, IEEE, may 2018.

[13] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, oct 2009.

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

[15] S. Reddy, A. Dragan, and S. Levine, "Shared autonomy via deep reinforcement learning," in *Robotics: Science and Systems XIV*, Robotics: Science and Systems Foundation, jun 2018.

[16] C. Schaff and M. Walter, "Residual policy learning for shared autonomy," in *Robotics: Science and Systems XVI*, Robotics: Science and Systems Foundation, jul 2020.

[17] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Comput. Surv.*, vol. 50, apr 2017.

[18] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *NIPS*, 1988.

[19] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, 2016.

[20] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems* (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.

[21] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 1889–1897, PMLR, 07–09 Jul 2015.

[22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms.," 2017.

[23] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, (New York, NY, USA), p. 1, Association for Computing Machinery, 2004.

[24] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, *et al.*, "Maximum entropy inverse reinforcement learning.," in *Aaai*, vol. 8, pp. 1433–1438, Chicago, IL, USA, 2008.

[25] S. Javdani, S. Srinivasa, and A. Bagnell, "Shared autonomy via hindsight optimization," in *Robotics: Science and Systems XI*, Robotics: Science and Systems Foundation, jul 2015.

[26] T. Silver, K. Allen, J. Tenenbaum, and L. Kaelbling, "Residual policy learning," 2019.

[27] R. V. Levine and A. Norenzayan, "The pace of life in 31 countries," *Journal of Cross-Cultural Psychology*, vol. 30, no. 2, pp. 178–205, 1999.

[28] B. Cèsar-Tondreau, G. Warnell, E. Stump, K. Kochersberger, and N. R. Waytowich, "Improving autonomous robotic navigation using imitation learning," *Frontiers in Robotics and AI*, vol. 8, 2021.

[29] B. Zhang, C. Holloway, and T. Carlson, "A hierarchical design for shared-control wheelchair navigation in dynamic environments," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 4439–4446, 2020.