Navigating the Practical Pitfalls of Reinforcement Learning for Social Robot Navigation

Dhimiter Pikuli, Jordan Cosio, Xavier Alameda-Pineda, Thierry Fraichard, Pierre-Brice Wieber

Navigating the Practical Pitfalls of Reinforcement Learning for Social Robot Navigation

Dhimiter Pikuli[†], Jordan Cosio[‡], Xavier Alameda-Pineda, Thierry Fraichard and Pierre-Brice Wieber
[†]dhimiter.pikuli@inria.fr [‡]jordan.cosio@inria.fr
Centre INRIA de l'Université Grenoble Alpes

***Abstract*—Navigation is one of the essential tasks in order for robots to be deployed in environments shared with humans. The problem becomes increasingly complex when taking in consideration that the robot's behaviour should be suitable to humans. This is referred to as social navigation and it is a cognitive task that us humans pay little attention to as it comes naturally. Since crafting a model of the environment dynamics that faithfully characterises how humans navigate seems an impossible task, we look on the side of learning-based approaches and especially reinforcement learning. In this paper we are interested in drawing conclusions on the vast number of design choices when training a navigation agent using reinforcement learning. To make this educated decisions, we offer a short survey on recent papers addressing the social navigation problem using learning-based algorithms. Additionally, we take note of what worked best in our testing.**

I. Introduction

Despite decades of research in social robot navigation, basic issues such as robots freezing unnecessarily [50] or exhibiting socially unwelcome behaviors remain largely unsolved. Machine learning methods have recently matured to the point of providing new, interesting solutions to various perception, decision-making and control problems in robotics. These methods could potentially contribute therefore to tackle those unsolved problems in social robot navigation.

Reinforcement Learning (RL), in particular, provides a means to approach problems such as social navigation where the desired behaviour is difficult to define formally and explicitly. Social navigation adds an additional layer of complexity due to the need to interpret and respond to nuanced human behaviors and dynamic environments. The problem is, existing RL algorithms are still very sensitive to design choices, which can make the difference between a very effective and a completely ineffective solution. And these practical aspects are largely overlooked in existing literature reviews on the topic.

The goal of this paper is to start filling this gap by providing an initial review of these practical aspects, together with some simple simulation experiments to compare their effects on the convergence of state-of-the-art RL algorithms, and on the applicability of the resulting solution in simple navigation scenarios. In order to gain additional insights on existing approaches, we decided to widen the scope of our literature review to include nonsocial robot navigation scenarios.

In Section II we offer a formal introduction to the problem and discuss various design choices. In Section III we survey various RL papers that relate to navigation. We present a taxonomy over different applications and offer analysis and cross-referencing between different problem formulations, environment design options and learning configurations. We conclude by reporting on experiments conducted in our own testing scenario and analysis of the most effective strategies in Section IV.

II. Problem Statement

The navigation problem can be mathematically formulated as a *Markov decision process* (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$, $\mathcal{A}$ are the *state space* and *action space* respectively, $\mathcal{P}(s, a, s')$ is the *transition probability* from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ via action $a \in \mathcal{A}$ and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the *reward function*. The manner in which the components of the MDP are defined, describe a specific task. The transition probabilities are not considered explicitly in this work, however they can have a major effect on the complexity of the task. Generally, the navigation task involves an agent moving in a crowded environment with a given goal position. We use the term *agent* to refer to the artificial system and the term *crowd* to refer to people and obstacles in the environment.

A. State Space

In navigation, the state $s \in \mathcal{S}$ includes the environment and its components. It can be fully described by positions and velocities of the agent $\vec{p} = \begin{bmatrix} p_x, p_y \end{bmatrix}, \dot{\vec{p}} = \begin{bmatrix} \dot{p}_x, \dot{p}_y \end{bmatrix}$, goal $\vec{g} = \begin{bmatrix} x_g, y_g \end{bmatrix}$, static obstacles and members of the crowd $\vec{c}_i = \begin{bmatrix} x_i, y_i \end{bmatrix}, \dot{\vec{c}}_i = \begin{bmatrix} \dot{x}_i, \dot{y}_i \end{bmatrix}$. However, it is possible to introduce another level of abstraction in terms of *observation space* $\mathcal{O}$ which is an arbitrary form of representing $\mathcal{S}$. For navigation, $\mathcal{O}$ can be implemented as relative Cartesian coordinates for crowd and goal. It is possible to convert this into polar coordinates, $\begin{bmatrix} x_g, y_g \end{bmatrix} \to \begin{bmatrix} r_g, \theta_g \end{bmatrix}$ meaning distance and orientation. A third option would be to use Light Detection and Ranging (LiDAR) observation that expresses the distance of the agent to a number of preset directions. Assuming a 360° FOV and N LiDAR rays the observation could look like the following $o = \{d_{i\frac{360}{N}\circ} \mid i \in \{1, ..., N\}\} \in \mathbb{R}_+^N$. The information of the goal position and agent velocity would be appended to this kind of observation. Finally, a general representation of the state space comes in form of images with arbitrary resolution, sometimes also as maps.

TABLE I
PAPERS BY APPLICATION

Application	References
UAV	[22], [40], [24], [66], [21], [55], [9], [33], [11]
USV	[61], [53], [59], [58], [28], [34] [7], [41], [35], [32]
UGV	[6], [8], [29], [19], [31], [26], [42], [67], [4], [23], [15], [14], [60], [27], [57] [38], [52], [64] [48], [37], [62], [2], [54], [12] [13], [3], [18], [43], [49], [17], [10], [63]

B. Action Space

The action space $\mathcal{A}$ refers to the motion control applied to the agent. Naturally, the change in position is described either by the velocity or acceleration. Cartesian acceleration and velocity are given as $\vec{p} = \left[\ddot{p}_x, \ddot{p}_y\right]$ and $\vec{p} = \left[\dot{p}_x, \dot{p}_y\right]$ respectively. Another option is linear velocity and angular change (LVAC) $\left[\dot{r}, \theta\right]$. Often, their values are bounded by maximum speed and acceleration, and $\mathcal{A}$ can be discrete and finite.

C. Reward Function

The reward function $\mathcal{R}$ mainly reflects how close the agent is to the goal and to a potential collision. Different definitions can have surprisingly different outcomes when training a RL policy. We elaborate on the engineering of $\mathcal{R}$ when discussing the the surveyed papers in Section III-D and while presenting our own environment in Section IV-A.

III. RELATED WORK

We survey recent papers starting from 2020 and concentrate on learning-based methods. Firstly, we classify in terms of application. As mentioned beforehand, our initial purpose is to make educated guesses on practical aspects of the problem, in Singamaneni et al. [47] a more general and extensive survey spanning a wider range of navigation algorithms can be found. We continue the discussion on a general overview of environment settings like the observation space, action space and reward function. Finally, we iterate the employed learning algorithms and the expected performance.

A. Application-based Taxonomy

While we are predominantly interested in robot navigation with pedestrians, the term navigation can relate potentially to non-social Unmanned Ground Vehicle (UGV), Unmanned Aerial Vehicle (UAV) or Unmanned Surface Vehicle (USV). In Table I a quantitative distribution of the different papers by application is shown. We further partition UGV papers into methods that address navigation in a crowd and multi-robot systems as seen in Table II. A more practical division distinguishes between systems tested in the real world, methods trained with the Gazebo simulation [25, 1] and a few papers integrating classic approaches, namely path finding and Dynamics Window Approach (DWA). It is of note, that papers tested in the real world are first trained inside a simulation.

Most of deep learning navigation papers usually construct the problem as an agent bounded in a room with a goal and various obstacles in the room [8, 29, 19, 6, 31, 42, 67, 14]. This makes away with the classical separation in navigation between global and local planning. There are methods that directly combine global path finding to learned fine-grained control [3, 27, 26]. Arguably, all deep learning methods for navigation can be adapted to use waypoints from a global planner. Moreover, it is possible to address such problems strictly with data-based approaches. A practical application is learning the layout of any home. In [5], an algorithm is proposed where an implicit map is learned and stored to pinpoint and navigate to different objects inside an apartment. This is potentially extensible to all kinds of navigation but highly ineffective where prior knowledge of the layout of the environment is known. Of our interest are problems for which such knowledge exists and if necessary a global planner relays the waypoints as reference for the control task.

TABLE II
FURTHER CHARACTERIZATION OF UGV PAPERS.

Tested IRL	[6], [8], [29], [19], [31], [42], [23], [14], [38], [12], [3], [18], [10]
Gazebo	[6], [8], [42], [23], [57], [38], [48], [37], [2], [13], [17], [10]
Classic	[26], [27], [42]
Multi-robot	[4], [15], [57], [37], [2], [54], [18], [49], [17]

We remark that even though many of the papers investigate crowd navigation, some simplifications are often made in the environment. For example, in [23] the initial position of the robot and obstacles remain the same during all experiments. In [10, 19], multiple scenarios are generated with randomization of obstacle position and shape which encapsulates a more general problem. Using plausible, hand-crafted, social scenarios also adds more realism to the situations and can facilitate the simulation-to-reality (sim2real) transfer. For example, narrow passages, intersections, and crowded areas are all relevant instances. To construct such scenarios the use of social-centric crowd models, e.g. [51, 44, 20], would be crucial.

B. Observation Space

LiDAR observations are used in the majority of the literature with 21 occurrences. This is due to the fact, that LiDAR is a very efficient way to get a 360-degree view of the environment. Compared to images, the information conveyed is less detailed but much lighter to process. While using positions and velocities is a simpler representation, it is not as informative as LiDAR especially w.r.t. obstacle shape and size. It is used in almost all 13 environments implemented in the real world and sometimes used in combination with other observations, especially RGB images.

Most of UAV and USV papers make use of images, given that LiDAR sensors are rarely applied for such long distances. In [66, 28] it is mentioned the use of equidistant angles from which distances are computed as observation but LiDAR sensors are not explicitly mentioned. Images were used in multiple occasions for ground robots in the form of maps [52, 54, 67] and from the robot's POV [60, 8]. Commonly, fusion is implemented between RGB-D images and LiDAR observation [13, 29, 19]. The advantage of employing a RGB

images with a depth channel is the ability to detect thin obstacles as well as reducing the number of sensor needed.

Position and velocity are used predominantly in multi-robot systems [17, 49, 18, 2], a few UAV/USV papers and only one crowd navigation paper [14]. It is the lightest and most efficient form of representation, which is why it may be more suitable for demanding applications like multi-robot tasks where control is applied over an entire fleet of agents. On the other hand, it might scale inefficiently as the observation size is directly linked to the number of obstacles in the environment.

C. Action Space

Most commonly LVAC were used, with 36 occurrences from which 9 are discretized and only 9 papers employ Cartesian control. This choice is partially explained by the use of non-holonomic robots (including ships and aircrafts). In context of social navigation, the majority of papers use continuous actions for higher flexibility and expressiveness. Discrete actions are more restrictive but can be learned more easily by the agent. In some cases, the change in velocity is unbounded, resulting in unnatural trajectories. Intuitively, in cases where Cartesian coordinates are used for the observation, a Cartesian control representation is used and analogously LVAC define control for polar coordinates representation. We uphold this relation in our environment implementation as well.

D. Reward Function

In this study, we observe that each surveyed paper uses its own definition of the reward function. However we find the following common components:

1) A terminal task completion positive reward (all papers).
2) A terminal collision negative reward (all papers).
3) A penalty for obstacle proxemics (15 papers).
4) A positive reward for the goal distance (17 papers).
5) An idleness penalty to encourage movement (14 papers).
6) A penalty for big angular changes (14 papers).

These patterns are not exhaustive and in some cases more complex. Woo and Kim [58] add a path following reward that helps the agent follow guideline positions, USV papers [28, 61, 34, 7] implement COLREG (Convention on the International Regulations for Preventing Collisions at Sea) specific rewards.

E. Learning Algorithms

In this study we observe a dominance of Proximal Policy Optimization (PPO) [46] algorithm with 16 occurrences over 53 papers. Suited for continuous and discrete actions and often used with images as observation space. Next are Deep Q-Networks (DQN) [36], Deep Deterministic Policy Gradients (DDPG) [30] and DuelingDQN [56] with 7, 6 and 5 occurrences respectively. Differently from PPO, these algorithms are off-policy. DQN, DoubleDQN and DuelingDQN make use of Convolutional Neural Network (CNN) policies and since they are based on Q-learning, they implement discrete actions. Finally, Soft actor-critic (SAC) [16], is the most recent algorithm used and implemented in 3 papers.

The policy configuration often remains that of the popular libraries. Some papers implement a custom, more complex architecture. In Yuan et al. [65], a Long Short-Term Memory (LSTM) is used in conjunction with DoubleDQN. Results show that the LSTM learns smoother and more natural trajectories compared to vanilla DoubleDeep Q-Networks (DQN) but convergence is slower due to the added complexity. Han et al. [19] present an architecture that combines RGB and LiDAR inputs in order to recreate minimalist 2D depth data. This data is then passed into a self-state-attention unit able to handle partially accurate data. The resulting 2D representation of the data is further processed by a CNN. Naturally, all methods using image observations make use of CNNs or vision Transformers to extract features. Moreover, in some cases, 1D CNN architectures are used to handle 2D LiDAR data.

Papers that extends the problem to moving obstacles have to integrate memory in their architecture, either with a recurrent cell (Recurrent Neural Network (RNN), Gated Recurrent Units (GRU), LSTM) or by stacking past observations as input.

F. Performance

Different units are used to reflect the training curve. Some papers use the conventional environment steps while some use the number of episodes which does not explicitly indicate the number of training transitions. In some cases hardware dependent or configuration dependent metrics are used. As mentioned previously a few papers simplify the task resulting in faster learning, whereas other approaches combine more complex observations for which pre-trained models are used. A rough estimate indicates that around a few million transitions are necessary to solve the task with on-policy algorithms converging faster.

IV. Experiments

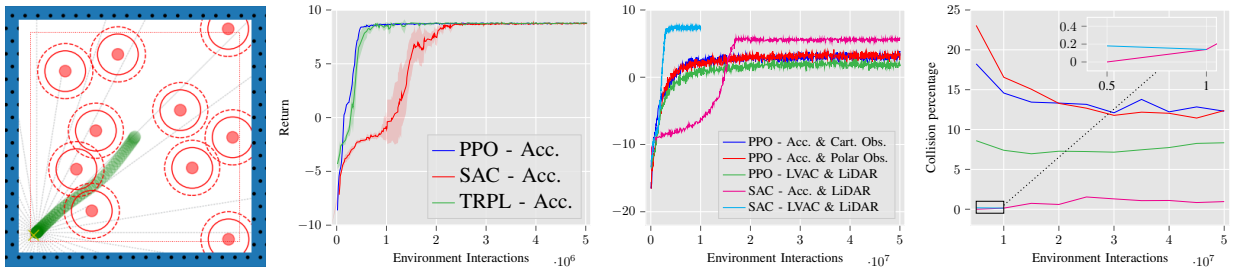
The purpose of the experiments is an initial evaluation on the different design choices of the environment and the learning algorithm.

A. Environment

The environment is strictly defined based on the problem formulation in Section II, including the various options for $\mathcal{O}$ and $\mathcal{A}$. Initially, we test for a simple navigation task without a crowd, then implement a static crowd and finally a blind crowd manifesting constant linear motion. Figure 1(a) presents an example of a trajectory involving 10 static crowd members. This environment does not account for obstacles with arbitrary shape and is limited to non socially-aware scenarios. The goal, crowd and wall reward functions, r_g, r_{c_i} and r_w, depend respectively on the distance to the goal d_g, to the i-th crowd member d_{c_i} and to the wall d_w, and are defined as:

$$r_g = \begin{cases} 10, & d_g < \tau_{\text{P}} \wedge \|\vec{p}\| < a_{\text{MAX}} T, \\ -C_g \max(d_g^2, 1), & \text{otherwise}, \end{cases}$$

$$r_{c_i} = \begin{cases} 0, & d_{c_i} > \tau_{\text{S}}, \\ -10, & d_{c_i} < 2\tau_{\text{P}}, \\ 1 - \exp(C_c / d_{c_i}), & \text{otherwise}, \end{cases}$$



(a) (b) (c) (d)

Fig. 1. (a) Example of successful trajectory from a SAC policy with LiDAR, (b) returns obtained without obstacles, (c) returns obtained in a static crowd environment, and (d) evolution of the collision rate in the same environment. The legend for (c) and (d) indicates the model's control and observation types.

where $\tau_{\text{P}} = 0.4$ m and $\tau_{\text{S}} = 1.9$ m are the physical and social distance thresholds, $a_{\text{MAX}} = 1.5$ m/s^2 is the maximum acceleration, and $T = 0.1$ s is the time step. In addition, r_w is defined similarly to r_{c_i} using d_w, C_g and C_c are set appropriately so the sum of all rewards is comparable to the terminal reward for the goal and collision respectively. The complete reward function is the sum of the components above:

$$R = r_g + \sum_i r_{c_i} + r_w. \tag{1}$$

B. Results

As a baseline RL method we use PPO and SAC tested using the `stable-baselines3` [45] library. We test with another library implemented for developing and evaluating the TRPL [39] algorithm against PPO and SAC. SAC solves a maximum entropy RL problem which means that on average, training is more stable whilst algorithms like PPO are strongly conditioned by hyperparameters making training less robust.

1) Observation: By default we use Cartesian observations and acceleration control. The simple navigation task is learned in less than 1M steps with longer training only improving control smoothness. For this task PPO learns much faster than SAC as shown in Figure 1(b). We see a clear hurdle when adding a static crowd to the environment. In this case, the agent does not learn a perfectly safe behaviour without collisions as depicted in Figure 1(d). Quantitatively, the PPO agent crashes around 12% of times whereas SAC LiDAR models collide only 1% of the time yielding the highest performance as shown in Figure 1(c). In contrast, we report no difference in performance or learning trend for Cartesian and polar observations. PPO and TRPL converge to an under-performing agent when using LiDAR observations with Cartesian representation for goal and velocity. We observe that the increase in number of LiDAR rays from the default value of 40, leads to a decrease in convergence speed and global performance.

2) Control: From testing it is clear that action space has an influence in training speed. Across different parameters, using velocity control speeds up training by at least a factor of 3 compared to acceleration control, as represented in Figure 1(c) for SAC with LiDAR observations. Additionally, PPO LiDAR works exclusively with LVAC. However, velocity control is not constrained in terms of acceleration at the moment which results in unnatural movement. This brings higher returns as the agent can immediately accelerate to maximum velocity and stop as it reaches the goal.

3) Reward: The reward function in Equation (1) was developed incrementally based on performance results. Idleness feedback was added in conjunction to Equation (1) but only slowed down training. A negative reward for large angular changes is irrelevant when acceleration is limited. All other rewards described in Section III-D were implemented in Equation (1). In this function we keep the values close to 0 and give negative rewards consistently based on d_g. Using a positive reward for the distance of the agent to the goal, encouraged a stalling behaviour close to the goal in order to maximize the return. Using a linear penalty based on the goal distance delayed convergence and prevented learning for PPO. The change of obstacles proxemics from linear $-C_c d_{c_i}$ to quadratic $C_c d_{c_i}^2$ and exponential $1 - \exp(C_c/d_{c_i})$ has little influence. However, appropriate weighting of this reward against the goal distance reward is vital. Finally, the task completion reward and the collision reward were set to the inverse of each other. A lower absolute task completion reward compared to the absolute collision reward slowed down learning. On the other hand, making the task completion larger than the collision reward in absolute value increased the number of collisions.

Overall, SAC generalizes well to all variously complex navigation tasks, whereas PPO is more sensitive to environment design choices. For more complex task formulations, PPO becomes increasingly challenging to tune. The same trends hold true for TRPL, but these details have been omitted from the graphs for better legibility. Initial tests with a non-social blind crowd suggest robustness in dynamic settings.

V. Conclusion

In this short paper we take note of some of the important and practical aspects of developing a RL method for the navigation task. We conclude from our testing that for our broad environment definition, LiDAR state representation with velocity control and a reward function based on points 1,2,3,4 iterated in Section III-D, the SAC algorithm is able to converge to a desirable behaviour. In future work, we hope to extend the analysis for socially-aware RL agents.

Acknowledgments

This work has been partially supported by MIAI@Grenoble Alpes, (ANR-19-P3IA-0003).

References

[1] C.E. Aguero, N. Koenig, I. Chen, H. Boyer, S. Peters, J. Hsu, B. Gerkey, S. Paepcke, J.L. Rivero, J. Manzo, E. Krotkov, and G. Pratt. Inside the virtual robotics challenge: Simulating real-time robotic disaster response. *Automation Science and Engineering, IEEE Transactions on*, 12(2):494–506, April 2015. ISSN 1545-5955. doi: 10.1109/TASE.2014.2368997.

[2] Chengchao Bai, Peng Yan, Wei Pan, and Jifeng Guo. Learning-Based Multi-Robot Formation Control With Obstacle Avoidance. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):11811–11822, August 2022. ISSN 1558-0016. doi: 10.1109/TITS.2021.3107336. URL https://ieeexplore.ieee.org/abstract/document/9527169.

[3] Runqi Chai, Hanlin Niu, Joaquin Carrasco, Farshad Arvin, Hujun Yin, and Barry Lennox. Design and Experimental Validation of Deep Reinforcement Learning-Based Fast Trajectory Planning and Control for Mobile Robot in Unknown Environment. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):5778–5792, April 2024. ISSN 2162-2388. doi: 10.1109/TNNLS.2022.3209154. URL https://ieeexplore.ieee.org/abstract/document/9913936.

[4] Guangda Chen, Shunyi Yao, Jun Ma, Lifan Pan, Yu'an Chen, Pei Xu, Jianmin Ji, and Xiaoping Chen. Distributed Non-Communicating Multi-Robot Collision Avoidance via Map-Based Deep Reinforcement Learning. *Sensors*, 20(17):4836, January 2020. ISSN 1424-8220. doi: 10.3390/s20174836. URL https://www.mdpi.com/1424-8220/20/17/4836.

[5] Shizhe Chen, Thomas Chabal, Ivan Laptev, and Cordelia Schmid. Object goal navigation with recursive implicit maps. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[6] Jaewan Choi, Geonhee Lee, and Chibum Lee. Reinforcement learning-based dynamic obstacle avoidance and integration of path planning. *Intelligent Service Robotics*, 14(5):663–677, November 2021. ISSN 1861-2784. doi: 10.1007/s11370-021-00387-2. URL https://doi.org/10.1007/s11370-021-00387-2.

[7] Do-Hyun Chun, Myung-Il Roh, Hye-Won Lee, Jisang Ha, and Donghun Yu. Deep reinforcement learning-based collision avoidance for an autonomous ship. *Ocean Engineering*, 234:109216, August 2021. ISSN 0029-8018. doi: 10.1016/j.oceaneng.2021.109216. URL https://www.sciencedirect.com/science/article/pii/S0029801821006466.

[8] Reinis Cimurs, Jin Han Lee, and Il Hong Suh. Goal-Oriented Obstacle Avoidance with Deep Reinforcement Learning in Continuous Action Space. *Electronics*, 9 (3):411, March 2020. ISSN 2079-9292. doi: 10.3390/electronics9030411. URL https://www.mdpi.com/2079-9292/9/3/411.

[9] Zhengyang Cui and Yong Wang. UAV Path Planning Based on Multi-Layer Reinforcement Learning Technique. *IEEE Access*, 9:59486–59497, 2021. ISSN 2169-3536. doi: 10.1109/ACCESS.2021.3073704. URL https://ieeexplore.ieee.org/abstract/document/9406006.

[10] Matej Dobrevski and Danijel Skočaj. Deep reinforcement learning for map-less goal-driven robot navigation. *International Journal of Advanced Robotic Systems*, 18 (1):172988142199262, January 2021. ISSN 1729-8814, 1729-8814. doi: 10.1177/1729881421992621. URL http://journals.sagepub.com/doi/10.1177/1729881421992621.

[11] Oualid Doukhi and Deok-Jin Lee. Deep Reinforcement Learning for End-to-End Local Motion Planning of Autonomous Aerial Robots in Unknown Outdoor Environments: Real-Time Flight Experiments. *Sensors*, 21(7):2534, January 2021. ISSN 1424-8220. doi: 10.3390/s21072534. URL https://www.mdpi.com/1424-8220/21/7/2534.

[12] Daniel Dugas, Juan Nieto, Roland Siegwart, and Jen Jen Chung. NavRep: Unsupervised Representations for Reinforcement Learning of Robot Navigation in Dynamic Human Environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7829–7835, May 2021. doi: 10.1109/ICRA48506.2021.9560951. URL https://ieeexplore.ieee.org/abstract/document/9560951. ISSN: 2577-087X.

[13] Muhammad Mudassir Ejaz, Tong Boon Tang, and Cheng-Kai Lu. Vision-Based Autonomous Navigation Approach for a Tracked Robot Using Deep Reinforcement Learning. *IEEE Sensors Journal*, 21(2):2230–2240, January 2021. ISSN 1558-1748. doi: 10.1109/JSEN.2020.3016299. URL https://ieeexplore.ieee.org/abstract/document/9166560.

[14] Michael Everett, Yu Fan Chen, and Jonathan P. How. Collision Avoidance in Pedestrian-Rich Environments With Deep Reinforcement Learning. *IEEE Access*, 9:10357–10377, 2021. ISSN 2169-3536. doi: 10.1109/ACCESS.2021.3050338. URL https://ieeexplore.ieee.org/abstract/document/9317723.

[15] Tingxiang Fan, Pinxin Long, Wenxi Liu, and Jia Pan. Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *The International Journal of Robotics Research*, 39(7):856–892, June 2020. ISSN 0278-3649, 1741-3176. doi: 10.1177/0278364920916531. URL http://journals.sagepub.com/doi/10.1177/0278364920916531.

[16] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018.

[17] Ruihua Han, Shengduo Chen, and Qi Hao. Cooperative Multi-Robot Navigation in Dynamic Environment with Deep Reinforcement Learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 448–454, May 2020. doi: 10.1109/ICRA40945.2020.9197209. URL https:

//ieeexplore.ieee.org/abstract/document/9197209. ISSN: 2577-087X.

[18] Ruihua Han, Shengduo Chen, Shuaijun Wang, Zeqing Zhang, Rui Gao, Qi Hao, and Jia Pan. Reinforcement Learned Distributed Multi-Robot Navigation With Reciprocal Velocity Obstacle Shaped Rewards. *IEEE Robotics and Automation Letters*, 7(3): 5896–5903, July 2022. ISSN 2377-3766. doi: 10.1109/LRA.2022.3161699. URL https://ieeexplore.ieee.org/abstract/document/9740403.

[19] Yiheng Han, Irvin Haozhe Zhan, Wang Zhao, Jia Pan, Ziyang Zhang, Yaoyuan Wang, and Yong-Jin Liu. Deep Reinforcement Learning for Robot Collision Avoidance With Self-State-Attention and Sensor Fusion. *IEEE Robotics and Automation Letters*, 7(3): 6886–6893, July 2022. ISSN 2377-3766. doi: 10.1109/LRA.2022.3178791. URL https://ieeexplore.ieee.org/abstract/document/9789512.

[20] Olivier Hauterville, Camino Fernández, Phani Teja Singamaneni, Anthony Favier, Vicente Matellán, and Rachid Alami. Interactive Social Agents Simulation Tool for Designing Choreographies for Human-Robot-Interaction Research. In Danilo Tardioli, Vicente Matellán, Guillermo Heredia, Manuel F. Silva, and Lino Marques, editors, *ROBOT2022: Fifth Iberian Robotics Conference*, volume 590, pages 514–527. Springer International Publishing, Cham, 2023. ISBN 978-3-031-21061-7 978-3-031-21062-4. doi: 10.1007/978-3-031-21062-4_42. URL https://link.springer.com/10.1007/978-3-031-21062-4_42. Series Title: Lecture Notes in Networks and Systems.

[21] Lei He, Nabil Aouf, James F. Whidborne, and Bifeng Song. Integrated moment-based LGMD and deep reinforcement learning for UAV obstacle avoidance. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7491–7497, May 2020. doi: 10.1109/ICRA40945.2020.9197152. URL https://ieeexplore.ieee.org/abstract/document/9197152. ISSN: 2577-087X.

[22] Jueming Hu, Xuxi Yang, Weichang Wang, Peng Wei, Lei Ying, and Yongming Liu. Obstacle Avoidance for UAS in Continuous Action Space Using Deep Reinforcement Learning, November 2021. URL http://arxiv.org/abs/2111.07037. arXiv:2111.07037 [cs].

[23] Jun Jin, Nhat M. Nguyen, Nazmus Sakib, Daniel Graves, Hengshuai Yao, and Martin 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)*, pages 6979–6985, May 2020. doi: 10.1109/ICRA40945.2020.9197148. URL https://ieeexplore.ieee.org/abstract/document/9197148. ISSN: 2577-087X.

[24] Sanghyun Kim, Jongmin Park, Jae-Kwan Yun, and Jiwon Seo. Motion Planning by Reinforcement Learning for an Unmanned Aerial Vehicle in Virtual Open Space with Static Obstacles. In *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, pages 784–787, October 2020. doi: 10.23919/ICCAS50221.2020.9268253. URL https://ieeexplore.ieee.org/abstract/document/9268253. ISSN: 2642-3901.

[25] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2149–2154, Sendai, Japan, Sep 2004.

[26] Linh Kästner, Teham Buiyan, Lei Jiao, Tuan Anh Le, Xinlin Zhao, Zhengcheng Shen, and Jens Lambrecht. Arena-Rosnav: Towards Deployment of Deep-Reinforcement-Learning-Based Obstacle Avoidance into Conventional Autonomous Navigation Systems. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6456–6463, September 2021. doi: 10.1109/IROS51168.2021.9636226. URL https://ieeexplore.ieee.org/abstract/document/9636226. ISSN: 2153-0866.

[27] Linh Kästner, Xinlin Zhao, Teham Buiyan, Junhui Li, Zhengcheng Shen, Jens Lambrecht, and Cornelius Marx. Connecting Deep-Reinforcement-Learning-based Obstacle Avoidance with Conventional Global Planners using Waypoint Generators. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1213–1220, September 2021. doi: 10.1109/IROS51168.2021.9636039. URL https://ieeexplore.ieee.org/abstract/document/9636039. ISSN: 2153-0866.

[28] Lingyu Li, Defeng Wu, Youqiang Huang, and Zhi-Ming Yuan. A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field. *Applied Ocean Research*, 113:102759, August 2021. ISSN 0141-1187. doi: 10.1016/j.apor.2021.102759. URL https://www.sciencedirect.com/science/article/pii/S0141118721002352.

[29] Jing Liang, Utsav Patel, Adarsh Jagan Sathyamoorthy, and Dinesh Manocha. Realtime Collision Avoidance for Mobile Robots in Dense Crowds using Implicit Multi-sensor Fusion and Deep Reinforcement Learning, April 2020. URL http://arxiv.org/abs/2004.03089. arXiv:2004.03089 [cs].

[30] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019.

[31] Lucia Liu, Daniel Dugas, Gianluca Cesari, Roland Siegwart, and Renaud Dubé. Robot Navigation in Crowded Environments Using Deep Reinforcement Learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5671–5677, October 2020. doi: 10.1109/IROS45743.2020.9341540. URL

https://ieeexplore.ieee.org/abstract/document/9341540. ISSN: 2153-0866.

[32] Xiongqing Liu and Yan Jin. Reinforcement learning-based collision avoidance: impact of reward function and knowledge transfer. *AI EDAM*, 34 (2):207–222, May 2020. ISSN 0890-0604, 1469-1760. doi: 10.1017/S0890060420000141. URL https://www.cambridge.org/core/journals/ai-edam/article/abs/reinforcement-learningbased-collision-avoidance-impact-of-reward-function-and-knowledge-transfer/DCB77383336C3CFE6F254D0D34FB6B65.

[33] Aye Aye Maw, Maxim Tyan, Tuan Anh Nguyen, and Jae-Woo Lee. iADA*-RL: Anytime Graph-Based Path Planning with Deep Reinforcement Learning for an Autonomous UAV. *Applied Sciences*, 11(9):3948, January 2021. ISSN 2076-3417. doi: 10.3390/app11093948. URL https://www.mdpi.com/2076-3417/11/9/3948.

[34] Eivind Meyer, Amalie Heiberg, Adil Rasheed, and Omer San. COLREG-Compliant Collision Avoidance for Unmanned Surface Vehicle Using Deep Reinforcement Learning. *IEEE Access*, 8:165344–165364, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.3022600. URL https://ieeexplore.ieee.org/abstract/document/9187823.

[35] Eivind Meyer, Haakon Robinson, Adil Rasheed, and Omer San. Taming an Autonomous Surface Vehicle for Path Following and Collision Avoidance Using Deep Reinforcement Learning. *IEEE Access*, 8:41466–41481, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.2976586. URL https://ieeexplore.ieee.org/abstract/document/9016254.

[36] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

[37] Seongin Na, Hanlin Niu, Barry Lennox, and Farshad Arvin. Bio-Inspired Collision Avoidance in Swarm Systems via Deep Reinforcement Learning. *IEEE Transactions on Vehicular Technology*, 71(3):2511–2526, March 2022. ISSN 1939-9359. doi: 10.1109/TVT.2022.3145346. URL https://ieeexplore.ieee.org/abstract/document/9693174.

[38] Hanlin Niu, Ze Ji, Farshad Arvin, Barry Lennox, Hujun Yin, and Joaquin Carrasco. Accelerated Sim-to-Real Deep Reinforcement Learning: Learning Collision Avoidance from Human Player. In *2021 IEEE/SICE International Symposium on System Integration (SII)*, pages 144–149, January 2021. doi: 10.1109/IEEECONF49454.2021.9382693. URL https://ieeexplore.ieee.org/abstract/document/9382693. ISSN: 2474-2325.

[39] Fabian Otto, Philipp Becker, Vien Anh Ngo, Hanna Carolin Maria Ziesche, and Gerhard Neumann. Differentiable Trust Region Layers for Deep Reinforcement Learning. October 2020. URL https://openreview.net/pdf?id=qYZD-AO1Vn.

[40] Sihem Ouahouah, Miloud Bagaa, Jonathan Prados-Garzon, and Tarik Taleb. Deep-Reinforcement-Learning-Based Collision Avoidance in UAV Environment. *IEEE Internet of Things Journal*, 9(6):4015–4030, March 2022. ISSN 2327-4662. doi: 10.1109/JIOT.2021.3118949. URL https://ieeexplore.ieee.org/abstract/document/9564258.

[41] Chao Pan, Zhouhua Peng, Lu Liu, and Dan Wang. Data-driven distributed formation control of under-actuated unmanned surface vehicles with collision avoidance via model-based deep reinforcement learning. *Ocean Engineering*, 267:113166, January 2023. ISSN 0029-8018. doi: 10.1016/j.oceaneng.2022.113166. URL https://www.sciencedirect.com/science/article/pii/S0029801822024490.

[42] Utsav Patel, Nithish K Sanjeev Kumar, Adarsh Jagan Sathyamoorthy, and Dinesh Manocha. DWA-RL: Dynamically Feasible Deep Reinforcement Learning Policy for Robot Navigation among Mobile Obstacles. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6057–6063, May 2021. doi: 10.1109/ICRA48506.2021.9561462. URL https://ieeexplore.ieee.org/abstract/document/9561462. ISSN: 2577-087X.

[43] Claudia Pérez-D'Arpino, Can Liu, Patrick Goebel, Roberto Martín-Martín, and Silvio Savarese. Robot Navigation in Constrained Pedestrian Environments using Reinforcement Learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1140–1146, May 2021. doi: 10.1109/ICRA48506.2021.9560893. URL https://ieeexplore.ieee.org/abstract/document/9560893. ISSN: 2577-087X.

[44] Noé Pérez-Higueras, Roberto Otero, Fernando Caballero, and Luis Merino. HuNavSim: A ROS 2 Human Navigation Simulator for Benchmarking Human-Aware Robot Navigation. *IEEE Robotics and Automation Letters*, 8(11):7130–7137, November 2023. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2023.3316072. URL https://ieeexplore.ieee.org/document/10252030/.

[45] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

[46] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[47] Phani Teja Singamaneni, Pilar Bachiller-Burgos, Luis J. Manso, Anaís Garrell, Alberto Sanfeliu, Anne Spalanzani, and Rachid Alami. A Survey on Socially Aware Robot Navigation: Taxonomy and Future Challenges, November 2023. URL http://arxiv.org/abs/2311.06922. arXiv:2311.06922 [cs].

[48] Hailuo Song, Ao Li, Tong Wang, and Minghui Wang. Multimodal Deep Reinforcement Learning with Aux-

iliary Task for Obstacle Avoidance of Indoor Mobile Robot. *Sensors*, 21(4):1363, January 2021. ISSN 1424-8220. doi: 10.3390/s21041363. URL https://www.mdpi.com/1424-8220/21/4/1363.

[49] Qingyang Tan, Tingxiang Fan, Jia Pan, and Dinesh Manocha. DeepMNavigate: Deep Reinforced Multi-Robot Navigation Unifying Local & Global Collision Avoidance. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6952–6959, October 2020. doi: 10.1109/IROS45743.2020.9341805. URL https://ieeexplore.ieee.org/abstract/document/9341805. ISSN: 2153-0866.

[50] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 797–803, 2010. doi: 10.1109/IROS.2010.5654369.

[51] Nathan Tsoi, Alec Xiang, Peter Yu, Samuel S. Sohn, Greg Schwartz, Subashri Ramesh, Mohamed Hussein, Anjali W. Gupta, Mubbasir Kapadia, and Marynel Vazquez. SEAN 2.0: Formalizing and Generating Social Situations for Robot Navigation. *IEEE Robotics and Automation Letters*, 7(4):11047–11054, October 2022. ISSN 2377-3766, 2377-3774. doi: 10.1109/LRA.2022.3196783. URL https://ieeexplore.ieee.org/document/9851501/.

[52] Binyu Wang, Zhe Liu, Qingbiao Li, and Amanda Prorok. Mobile Robot Path Planning in Dynamic Environments Through Globally Guided Reinforcement Learning. *IEEE Robotics and Automation Letters*, 5(4):6932–6939, October 2020. ISSN 2377-3766. doi: 10.1109/LRA.2020.3026638. URL https://ieeexplore.ieee.org/abstract/document/9205217.

[53] Chengbo Wang, Xinyu Zhang, Zaili Yang, Musa Bashir, and Kwangil Lee. Collision avoidance for autonomous ship using deep reinforcement learning and prior-knowledge-based approximate representation. *Frontiers in Marine Science*, 9, January 2023. ISSN 2296-7745. doi: 10.3389/fmars.2022.1084763. URL https://www.frontiersin.org/articles/10.3389/fmars.2022.1084763.

[54] Di Wang, Hongbin Deng, and Zhenhua Pan. MRCDRL: Multi-robot coordination with deep reinforcement learning. *Neurocomputing*, 406:68–76, September 2020. ISSN 0925-2312. doi: 10.1016/j.neucom.2020.04.028. URL https://www.sciencedirect.com/science/article/pii/S0925231220305932.

[55] Fei Wang, Xiaoping Zhu, Zhou Zhou, and Yang Tang. Deep-reinforcement-learning-based UAV autonomous navigation and collision avoidance in unknown environments. *Chinese Journal of Aeronautics*, 37(3):237–257, March 2024. ISSN 1000-9361. doi: 10.1016/j.cja.2023.09.033. URL https://www.sciencedirect.com/science/article/pii/S1000936123003448.

[56] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning, 2016.

[57] Shuhuan Wen, Zeteng Wen, Di Zhang, Hong Zhang, and Tao Wang. A multi-robot path-planning algorithm for autonomous navigation using meta-reinforcement learning based on transfer learning. *Applied Soft Computing*, 110:107605, October 2021. ISSN 1568-4946. doi: 10.1016/j.asoc.2021.107605. URL https://www.sciencedirect.com/science/article/pii/S1568494621005263.

[58] Joohyun Woo and Nakwan Kim. Collision avoidance for an unmanned surface vehicle using deep reinforcement learning. *Ocean Engineering*, 199:107001, March 2020. ISSN 0029-8018. doi: 10.1016/j.oceaneng.2020.107001. URL https://www.sciencedirect.com/science/article/pii/S0029801820300792.

[59] Xing Wu, Haolei Chen, Changgu Chen, Mingyu Zhong, Shaorong Xie, Yike Guo, and Hamido Fujita. The autonomous navigation and obstacle avoidance for USVs with ANOA deep reinforcement learning method. *Knowledge-Based Systems*, 196:105201, May 2020. ISSN 0950-7051. doi: 10.1016/j.knosys.2019.105201. URL https://www.sciencedirect.com/science/article/pii/S0950705119305350.

[60] Wendong Xiao, Liang Yuan, Li He, Teng Ran, Jianbo Zhang, and Jianping Cui. Multigoal Visual Navigation With Collision Avoidance via Deep Reinforcement Learning. *IEEE Transactions on Instrumentation and Measurement*, 71:1–9, 2022. ISSN 1557-9662. doi: 10.1109/TIM.2022.3158384. URL https://ieeexplore.ieee.org/abstract/document/9733281.

[61] Xinli Xu, Yu Lu, Gang Liu, Peng Cai, and Weidong Zhang. COLREGs-abiding hybrid collision avoidance algorithm based on deep reinforcement learning for USVs. *Ocean Engineering*, 247:110749, March 2022. ISSN 0029-8018. doi: 10.1016/j.oceaneng.2022.110749. URL https://www.sciencedirect.com/science/article/pii/S0029801822001998.

[62] Jiachen Yang, Jingfei Ni, Yang Li, Jiabao Wen, and Desheng Chen. The Intelligent Path Planning System of Agricultural Robot via Reinforcement Learning. *Sensors*, 22(12):4316, January 2022. ISSN 1424-8220. doi: 10.3390/s22124316. URL https://www.mdpi.com/1424-8220/22/12/4316.

[63] S. M. Yang and Y. A. Lin. Development of an Improved Rapidly Exploring Random Trees Algorithm for Static Obstacle Avoidance in Autonomous Vehicles. *Sensors*, 21(6):2244, January 2021. ISSN 1424-8220. doi: 10.3390/s21062244. URL https://www.mdpi.com/1424-8220/21/6/2244.

[64] Qingfeng Yao, Zeyu Zheng, Liang Qi, Haitao Yuan, Xiwang Guo, Ming Zhao, Zhi Liu, and Tianji Yang. Path Planning Method With Improved Artificial Potential Field—A Reinforcement Learning Perspective. *IEEE Access*, 8:135513–135523, 2020. ISSN 2169-

3536. doi: 10.1109/ACCESS.2020.3011211. URL https://ieeexplore.ieee.org/abstract/document/9146273.

[65] Jianya Yuan, Hongjian Wang, Honghan Zhang, Changjian Lin, Dan Yu, and Chengfeng Li. AUV Obstacle Avoidance Planning Based on Deep Reinforcement Learning. *Journal of Marine Science and Engineering*, 9(11):1166, November 2021. ISSN 2077-1312. doi: 10.3390/jmse9111166. URL https://www.mdpi.com/2077-1312/9/11/1166.

[66] Sitong Zhang, Yibing Li, and Qianhui Dong. Autonomous navigation of UAV in multi-obstacle environments based on a Deep Reinforcement Learning approach. *Applied Soft Computing*, 115:108194, January 2022. ISSN 1568-4946. doi: 10.1016/j.asoc.2021.108194. URL https://www.sciencedirect.com/science/article/pii/S1568494621010383.

[67] Yihao Zhang, Zhaojie Chai, and George Lykotrafitis. Deep reinforcement learning with a particle dynamics environment applied to emergency evacuation of a room with obstacles. *Physica A: Statistical Mechanics and its Applications*, 571:125845, June 2021. ISSN 0378-4371. doi: 10.1016/j.physa.2021.125845. URL https://www.sciencedirect.com/science/article/pii/S0378437121001175.