Predictive World Models for Social Navigation

Goodluck Oguzie[1], Aniko Ekart[1], and Luis J. Manso[1]

Aston University, B4 7ET, Birmingham, UK
https://cs.aston.ac.uk/arp
190212683@aston.ac.uk

Abstract. As robots begin to coexist with humans, the need for efficient and safe social robot navigation becomes increasingly pressing. In this paper we investigate how world models can enhance the effectiveness of reinforcement learning in social navigation tasks. We introduce three approaches that leverage predictive world models, which are then benchmarked against state-of-the-art algorithms. For a comprehensive and reliable evaluation, we employed multiple metrics during the training and testing phases. The key novelty of our approach consists in the integration and evaluation of predictive world models within the context of social navigation, as well as in the models themselves. Based on a diverse set of performance metrics, the experimental results provide evidence that predictive world models help improve reinforcement learning techniques for social navigation.

1 Introduction

With the increased sharing of space between humans and robots, the need for effective robot Social Navigation (SocNav) has become paramount [19]. Most state-of-the-art approaches for SocNav depend on hand-crafted algorithms that are difficult to scale to consider additional variables [7], the most common variables being the goal position, free space, and the 2D poses of humans and robots [7].

Reinforcement Learning (RL) provides a framework to overcome the reliance on hand-crafted algorithms, but current RL algorithms often exhibit prolonged convergence times, requiring extensive interactions with the environment before they can learn a near-optimal policy. Despite the significant success of RL in numerous tasks [2], more research is needed before RL-based SocNav can be successfully applied in complex real-world scenarios [7].

RL approaches using world models capable of predicting future states of the environment have outperformed more traditional approaches in multiple RL environments [4, 8, 15]. In this paper, we propose three methods that integrate predictive world models into an RL algorithm for SocNav tasks. Our methods leverage a world model similar to the one proposed by Ha and Schmidhuber [8], combining a Variational Autoencoder (VAE) [16] and a Long-Short Term Memory network (LSTM) [13]. The first method, termed *2StepAhead*, builds on top of Ha and Schmidhuber [8], but makes the predictions two steps ahead (assuming that the same action is taken twice) and uses Dueling DQN [28] instead

of Covariance Matrix Adaptation (CMA) [12]. Our second method, *MASPM*, also expands upon that of Ha and Schmidhuber [8] by considering multiple actions while performing single-step predictions. The third method, *2StepAhead-MASPM*, combines the ideas of both prior approaches by performing two steps ahead predictions and considering multiple actions.

2 Related Work

Reinforcement Learning is a learning paradigm where an agent learns to interact near-optimally with its environment to maximise a given reward, operating within a Markov Decision Process framework [11, 26]. In the domain of robotics, RL has been leveraged to teach robots complex manipulation tasks [1], and in gaming, it has been employed to develop agents that can play games proficiently [18].

Despite its wide-ranging successes, RL has well-known limitations. Adaptability to novel environments poses a significant challenge [21]. Moreover, RL often requires large volumes of data for training, making it computationally expensive [17]. Specifically in the domain of social navigation, these issues become even more pronounced due to the rich and complex nature of social dynamics [20, 23, 30]. The complex interactions that happen in social settings are difficult to model and predict, making RL agents' learning of optimal policies even more challenging [5].

In response to these challenges, world models have emerged as a promising solution. For instance, MuZero, an RL-based method using world models, has demonstrated its efficacy in learning ATARI game rules using observed image data and action sequences, even with limited computational resources [10, 22]. AlphaGo, another RL-based method using world models, outperformed human experts in the game of Go in 2016 [24].

Unlike traditional RL-based predictions that rely on the current environment state represented by a state-action pair, predictive world models can consider both past and present states to anticipate future ones [8]. Arguably, we can interpret that the inclusion of these models into RL algorithms incorporates into the algorithms the prior that predicting future states is useful. This methodology has been applied successfully in environments such as CarRacing [4] and Doom [15], outperforming traditional RL [8]. Furthermore, world models introduce a predictive component to the RL dynamics, enabling the agent to anticipate future actions. This can lead to faster learning and potentially improved results in fewer episodes [27].

Additionally, world models augment the MDP framework by shifting from reliance solely on current observations and actions to a broader decision optimisation perspective. This allows agents to generate more informed policies based on a predictive understanding of the environment [6]. Notably, Dreamer, a model-based RL agent, has demonstrated the capability of combining world models and policy learning to achieve state-of-the-art performance in various tasks [9].

This study delves deeper into these predictive world models, specifically within RL-based social navigation which leads to our research question: "Can world models help us improve RL-based social navigation?"

3 Methodology

In this paper we explore the use of predictive world models to improve RL-based SocNav using the three aforementioned proposed methods –2StepAhead, MASPM, and 2StepAhead-MASPM; in this section we describe the three approaches and provide experimental details.

Our experiments are conducted in SocNavEnv [14], a configurable environment specifically designed for social navigation scenarios. This environment has the capacity to incorporate a wide range of entities such as humans (static or moving), plants, tables, and laptop computers. For our experiments, SocNavEnv was configured to work with a discrete action space of four actions (stop, move forward, rotate left, and rotate right), three moving humans, and a social navigation reward function [3]. The goal of the agent in SocNavGym is to train the agent to navigate towards the target while (1) avoiding collisions with surrounding entities and (2) minimising the discomfort caused to the humans. A screenshot of SocNavEnv is shown in Fig. 1.

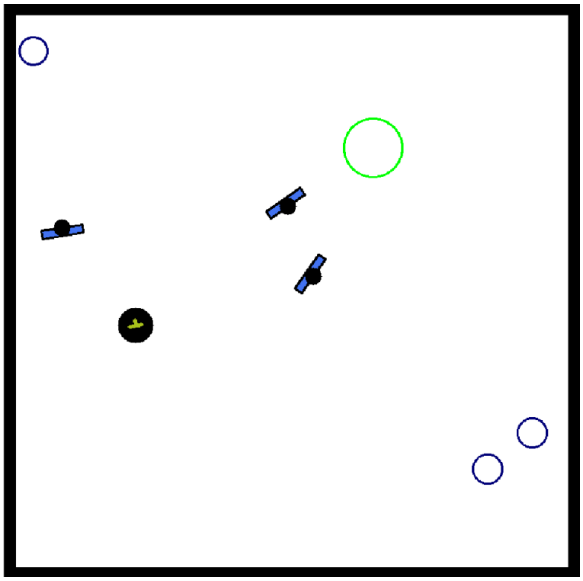

Fig. 1. Screenshot of SocNavEnv, the environment used for the experiments [14]. Blue squares represent humans, blue circles indicate humans' goals (which are unknown to the robot), green circles represent the robot's goals, and black-green circles represent robot agents.

Although we are aware that in real-life settings the number of individuals involved is frequently greater than three, we found that including three humans was sufficient for the experiments to be challenging for the RL algorithm used as a baseline. Dueling DQN was chosen as the baseline, because it is a well known algorithm that performs generally well even when dealing with high-dimensional state spaces and it is suitable for discrete action spaces [28]. Dueling DQN is an evolution of DQN where the final layer of the network is split into two distinct pathways: one computes the state-value function and the other estimates the advantage function for each discrete action [28]. This design allows Dueling DQN to better distinguish between the impact of different actions, thus optimising learning outcomes. To our knowledge, there is no reason to believe that the methods would not be applicable to other RL algorithms.

Our proposed methods build upon the architecture developed by Ha and Schmidhuber [8] (see Fig. 2), where a VAE, parameterised by ϕ, compresses the observation (s) into a latent state (z) (of sizes 23 and 16, respectively), as shown by the relationship $z = \text{VAE}(s; \phi)$. The role of the VAE is to improve the efficiency and performance by compressing important information within this reduced dimensionality. Following this, the LSTM, parameterised by ψ, utilises z and the chosen action (a) to predict the next latent state (z') and hidden state (h') following

$$(z', h') = \text{LSTM}(z, a; \psi).$$

These predicted states are then input into the Dueling DQN, forming the foundation for the predictive world models in our methods.

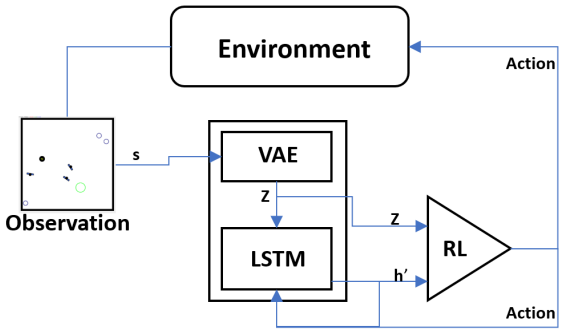

Fig. 2. Predictive World Model as proposed by Ha and Schmidhuber [8].

3.1 Two step Ahead Predictive World Model: 2StepAhead

2StepAhead extends the vanilla approach of Ha and Schmidhuber [8] by predicting the hidden state and the latent state two steps ahead. The number of steps that the model is predicting ahead was empirically determined out of 2, 4, 8,

and 16 steps. Although this number arguably depends on the environment, predicting more than 2 steps ahead did not improve the results in our SocNavGym setup and made training slower.

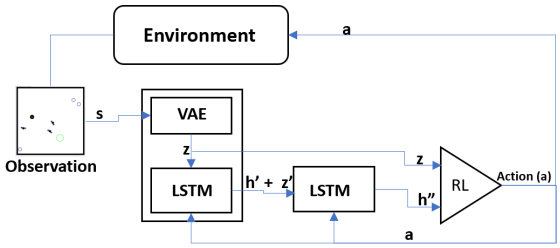

Fig. 3. In 2StepAhead, the same LSTM is used recursively to predict two steps ahead.

As depicted in Fig. 3, our model predicts two steps ahead for the hidden state (h'') and the latent state (z'') by using the predicted next and hidden states (z', h') and the current action (a):

$$(z'', h'') = \text{LSTM}(z', h', a; \psi).$$

Subsequently, the environment's current latent state (z) and the two steps ahead hidden state (h'') are fed into the Dueling DQN to choose the next action (a^*):

$$a^* = \text{Dueling DQN}(z|h''; \xi),$$

where ξ represents the parameters of our Dueling DQN. By predicting the latent state of the environment two steps ahead, we hope to provide to the RL algorithm richer information regarding the future state in case the robot keeps taking the current action, potentially improving performance and robustness in a dynamic environment.

3.2 Multi Action State Predictive Model: MASPM

This model provides the Dueling DQN with a comprehensive view of future state possibilities, encompassing all four available actions, potentially enabling more informed decision-making and thereby improving the model robustness and performance (see Fig. 4).

The latent state (z) along with the action serve as inputs for an LSTM, which predicts the next state and hidden state based on the given action. We denote the action index by i, ranging from 0 to 3 indicating that we only have four possible actions to consider. Thus, for each action i, the latent state and action are input to the LSTM model to predict the subsequent state and hidden state:

$$(z'_i, h'_i) = \text{LSTM}(z|a_i; \psi),$$

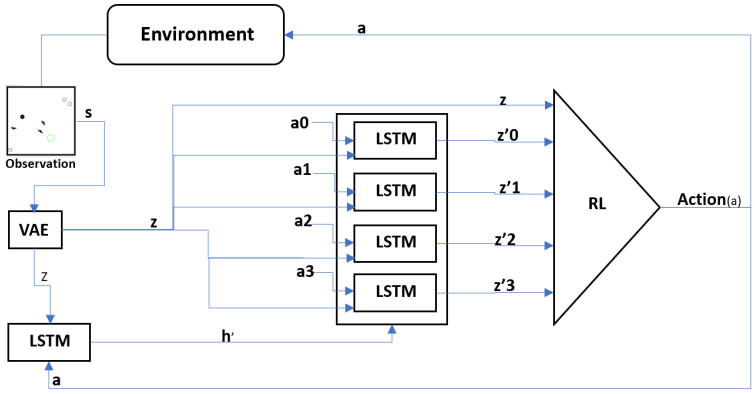

Fig. 4. In MASPM, the LSTM is not used recursively, but it is provided with the four possible actions and all the resulting data are fed into the RL algorithm.

where z is the current latent state, a_i is the i-th action (provided to the network as a one-hot encoding), and ψ represents the LSTM parameters. The four next predicted states, z'_1, z'_2, z'_3, z'_4, together with the current latent state z then serve as inputs for the Dueling DQN to estimate the best action a^*:

$$a^* = \text{Dueling DQN}(z|z'_1|z'_2|z'_3|z'_4; \xi),$$

where ξ represents the Dueling DQN parameters. MASPM provides a broadened perspective of future states across multiple actions, offering the Dueling DQN a richer foundation for decision-making.

3.3 Combining 2StepAhead and MASPM: 2StepAhead-MASPM

The 2StepAhead-MASPM is a combination of MASPM and the 2StepAhead method and aims to combine their advantages. This model provides a two-step-ahead prediction for each potential action. The two-step-ahead prediction horizon facilitates the Dueling DQN algorithm with a more refined decision-making capability. It achieves this by leveraging the current latent state and the predicted two-step-ahead state for each possible action to determine its subsequent action.

Figure 5 illustrates the architecture of the proposed 2StepAhead-MASPM. The latent state (z), coupled with the related action, is fed into the LSTM. The LSTM uses these inputs to predict the next state and the hidden state conditioned on the input action. The action index i can range from 0 to 3, representing the four possible actions. For each action i, the LSTM model processes the latent state and action as input and predicts the corresponding next state and hidden state. The model repeats this process, using the same action and the previously predicted latent state for the second prediction.

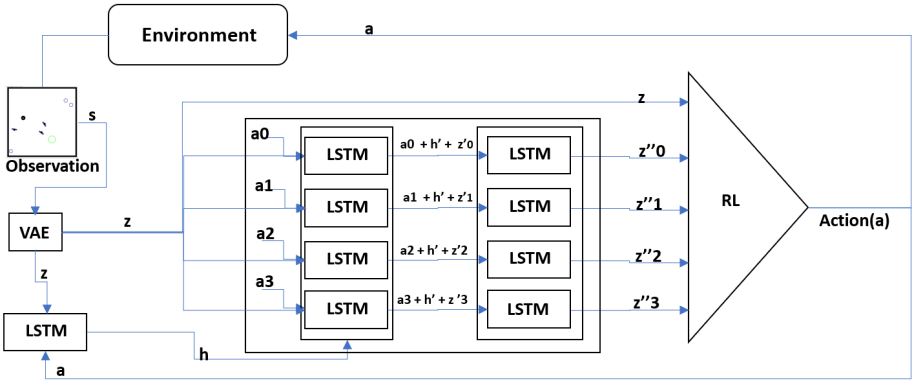

Fig. 5. 2StepAhead-MASPM combines the advantages of 2StepAhead and MASPM. It predicts two steps ahead and considers all actions instead of just the current action.

Given a latent state z and an action a_i at a time t, the LSTM predicts the next state z_{t+1} and hidden state h_{t+1}. The process is repeated using the new latent state z_{t+1} and the same action a_i to predict the next latent state z_{t+2} and hidden state h_{t+2}:

$$(s_{t+1}, h_{t+1}) = LSTM(s_t, a_i, h_t; \phi)$$
$$(s_{t+2}, h_{t+2}) = LSTM(s_{t+1}, a_i, h_{t+1}; \phi)$$

We hypothesise that combining two steps ahead predictions with a coverage of all actions can improve Dueling DQN's decision-making. The next section benchmarks the three proposed methods against the selected baselines to evaluate whether the use of Predictive World Models is beneficial in the context of SocNav.

4 Experimental results

All the developed models are based on the Dueling DQN reinforcement learning algorithm and are trained within the SocNavEnv environment [14]. To ascertain the influence of predictive world models on RL-based social navigation, Dueling DQN is also used as a baseline. The hyperparameters of Dueling DQN, particularly the size of the hidden layers, are critical in determining the agent's learning capabilities [25]. Therefore, we evaluated two Dueling DQN MLP model architectures -one with two hidden layers of size 128 each, and another with layers of sizes 512 and 128, respectively. After 200,000 episodes –the number of episodes required for all experiments to converge in this paper– the model with hidden layers of size 512 and 128 achieved a slightly higher expected cumulative reward for the vanilla Dueling DQN. Therefore, we selected this architecture for the rest of the Dueling DQN-based agents. Subsequently, we integrated predictive

world models into the RL framework according to the three proposed methods in Sec. 3. We evaluated the proposed methods using different metrics [7], each uniquely designed with predictive capabilities, in the context of social navigation tasks.

The novelty of our approach lies in the integration and evaluation of predictive world models –specifically, 2StepAhead, MASP, and 2StepAhead-MASP– within the context of social navigation, which has not been explored in previous work, as well as in the models themselves. For a comprehensive and reliable evaluation, we employed multiple metrics during the training and testing phases.

Using only a single metric can limit the scope of the evaluation and may not fully capture the model's performance due to the multi-faceted nature of social navigation tasks. Metrics such as discomfort counts, human collisions, and personal space compliance are as important as the traditionally employed metrics in RL such as reward or convergence time. Therefore, we use this broad range of metrics to ensure a holistic analysis that comprehensively reflects the performance in a human-robot interactive environment. Furthermore, our comparative analysis extends beyond our baseline Dueling DQN models. For the testing phase, we also include comparisons with other established models in the domain, like the RVO2 and social force model, to provide a broader context for the performance of our models. These benchmarks were chosen due to their widespread use in social navigation tasks.

4.1 Training Phase Metric Evaluation

The training phase is focused on the cumulative reward, training time, and episodes to convergence. The results from this phase showed significant improvements in our proposed models over the baseline Dueling DQN models. The 2StepAhead model was particularly efficient, solving the task in about 3200 episodes, as depicted in Fig. 6. The 2StepAhead-MASPM model outperformed all the other models, achieving the highest average cumulative reward of 0.67.

4.2 Testing Phase Metric Evaluation

In the testing phase, we used a broad range of metrics related to human-robot interactions, navigation efficiency, and overall performance, and measured those metrics for 500 episodes per algorithm. In Figs. 7 and 8 the histograms of the following metrics are shown:

- Human discomfort: Average human discomfort caused to humans, as described in [3].
- Distance travelled: Distance travelled by the agent, per episode (in meters).
- Simulation time: Calculated as the number of steps multiplied by the step time (in seconds).
- Human collisions: Whether the robot collides in a trajectory or not (binary metric).

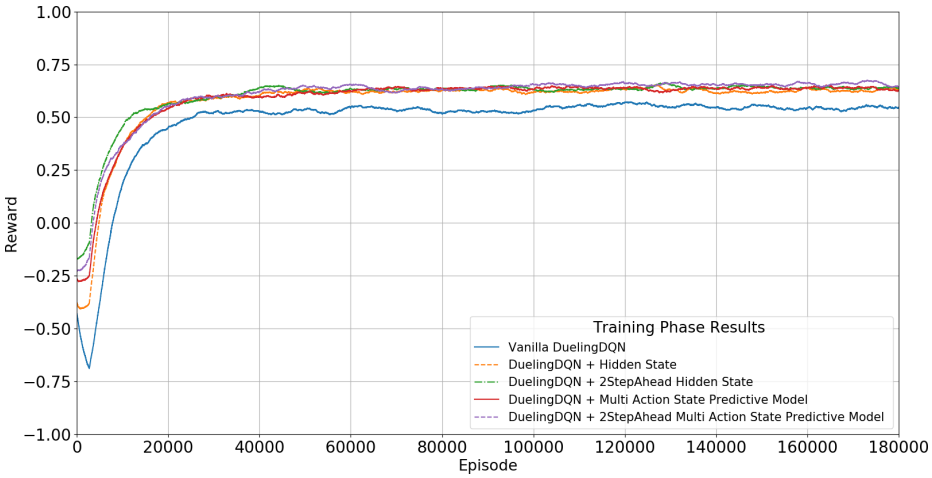

Fig. 6. Smoothed cumulative reward during training.

- Max steps: Whether the agent reaches the maximum number of steps in a particular episode (binary metric).
- Reward: The cumulative reward per episode (scalar).
- Successful run: Whether the agent reaches the goal or not in an episode (binary metric).
- Idle time: Steps where the robot moves less than 0.05m (in seconds).
- Personal space compliance rate: Ratio of the time where robot is further away than 0.5 metres from any human divided by the total time (scalar).

The 2StepAhead-MASPM achieved higher average cumulative reward than the baseline models. Success rate, human collision, and cumulative reward were also improved with our 2StepAhead-MASPM model. Our model performed well overall, achieving the second-best in minimal idleness and ranking third for personal space compliance, simulation time, and distance travelled, respectively. However, it is important to remember that optimising one aspect of social norms may have unintended consequences on others. For example, while reducing the time to reach the goal by finding the shortest path may be desirable, this could compromise human personal space. Therefore, the ideal solution is not to maximise one specific metric but to strike a balance across all metrics.

While our 2StepAhead-MASPM model might not have achieved the highest score in all individual metrics, it excelled in achieving well-rounded results over most metrics used, respecting the Pareto nondomination criterion [29], i.e., no other method performed better across all metrics. It improved critical aspects of social norms such as avoiding collisions with humans and maintaining a high success rate, all without excessively compromising personal space compliance. Moving forward, our aim is to continue refining our models to obtain an even better balance across the multiple dimensions involved, thereby further improving performance in complex, multi-faceted tasks such as social navigation.

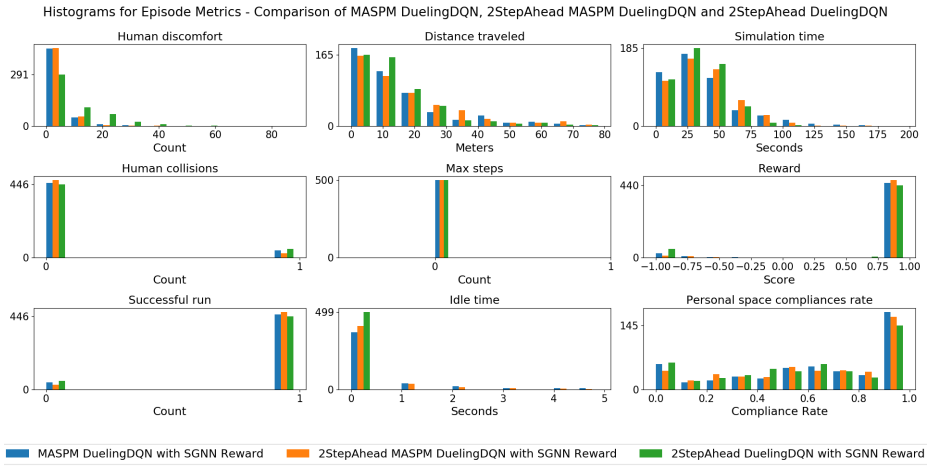

Fig. 7. Histograms of the metrics used for comparison, applied to the three proposed models.

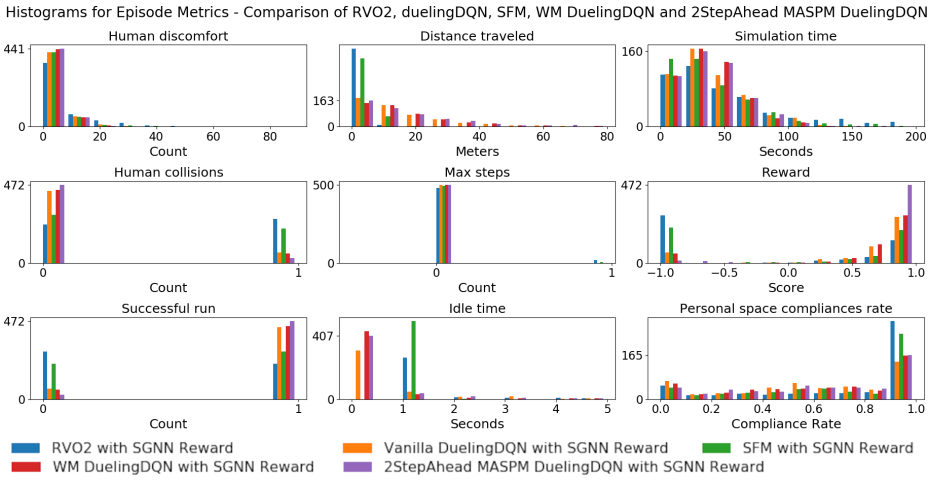

Fig. 8. Histograms of the metrics used for comparison, applied to RVO2, Dueling DQN, SFM, WM Dueling DQN, and 2StepAhead-MASPM Dueling DQN.

5 Conclusions and Future Work

The experimental results confirm the value of integrating world models in RL-based social navigation. We present a novel contribution –the 2StepAhead-MASPM predictive model integrated into the Dueling DQN framework– which demonstrated superior performance over the baseline models across various metrics, particularly in terms of success rate, cumulative reward and human collision. However, our study also revealed areas where improvement can be made, most notably in terms of maintaining personal space –an essential aspect in social navigation. This insight highlights the importance of the Pareto non-domination criterion [29] in dealing with such multi-faceted tasks.

As future work, we are planning to experiment on more complex navigation environments and continuous action spaces. By introducing a range of different obstacles such as tables, chairs, and laptops, and varying the number of humans present in the environment, we aim to simulate more realistic and dynamic scenarios. With these, we want to further test the limits of predictive world models and refine the performance of our models. Ultimately, our goal is to develop an RL agent that not only navigates efficiently through complex social environments but also maintains respect for personal boundaries and pedestrians' comfort.

References

1. Andrychowicz, O.A.M., Baker, B., Chociej, M., Józefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., Schneider, J., Sidor, S., Tobin, J., Welinder, P., Weng, L., Zaremba, W.: Learning dexterous in-hand manipulation. International Journal of Robotics Research **39**(1), 3–20 (2020). https://doi.org/10.1177/0278364919887447
2. Arulkumaran, K., Deisenroth, M.P., Brundage, M., Bharath, A.A.: Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine **34**(6), 26–38 (2017)
3. Bachiller, P., Rodriguez-Criado, D., Jorvekar, R.R., Bustos, P., Faria, D.R., Manso, L.J.: A graph neural network to model disruption in human-aware robot navigation. Multimedia Tools and Applications pp. 1–19 (2021). https://doi.org/https://doi.org/10.1007/s11042-021-11113-6
4. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: Openai gym. arXiv preprint arXiv:1606.01540 (2016)
5. Chen, Y.F., Everett, M., Liu, M., How, J.P.: Socially aware motion planning with deep reinforcement learning. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1343–1350. IEEE (2017)
6. Chua, K., Calandra, R., McAllister, R., Levine, S.: Deep reinforcement learning in a handful of trials using probabilistic dynamics models. Advances in neural information processing systems **31** (2018)
7. Francis, A., Perez-D'Arpino, C., Li, C., Xia, F., Alahi, A., Alami, R., Bera, A., Biswas, A., Biswas, J., Chandra, R., et al.: Principles and guidelines for evaluating social robot navigation algorithms. arXiv preprint arXiv:2306.16740 (2023)
8. Ha, D., Schmidhuber, J.: World models. arXiv preprint arXiv:1803.10122 (2018)
9. Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination. arXiv preprint arXiv:1912.01603 (2019)

10. Hafner, D., Lillicrap, T., Norouzi, M., Ba, J.: Mastering atari with discrete world models. arXiv preprint arXiv:2010.02193 (2020)
11. Han, X.: A mathematical introduction to reinforcement learning. Semantic Scholar pp. 1–4 (2018)
12. Hansen, N.: The cma evolution strategy: a comparing review. Towards a new evolutionary computation: Advances in the estimation of distribution algorithms pp. 75–102 (2006)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
14. Kapoor, A., Swamy, S., Manso, L., Bachiller, P.: Socnavgym: A reinforcement learning gym for social navigation. arXiv preprint arXiv:2304.14102 (2023)
15. Kempka, M., Wydmuch, M., Runc, G., Toczek, J., Jaśkowski, W.: Vizdoom: A doom-based ai research platform for visual reinforcement learning. In: 2016 IEEE conference on computational intelligence and games (CIG). pp. 1–8. IEEE (2016)
16. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
17. Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., et al.: Isaac gym: High performance gpu-based physics simulation for robot learning. arXiv preprint arXiv:2108.10470 (2021)
18. Matsuo, Y., LeCun, Y., Sahani, M., Precup, D., Silver, D., Sugiyama, M., Uchibe, E., Morimoto, J.: Deep learning, reinforcement learning, and world models. Neural Networks **152**, 267–275 (2022). https://doi.org/10.1016/j.neunet.2022.03.037, https://doi.org/10.1016/j.neunet.2022.03.037
19. Mavrogiannis, C., Baldini, F., Wang, A., Zhao, D., Trautman, P., Steinfeld, A., Oh, J.: Core challenges of social robot navigation: A survey. ACM Transactions on Human-Robot Interaction **12**(3), 1–39 (2023)
20. Rao, K., Harris, C., Irpan, A., Levine, S., Ibarz, J., Khansari, M.: RL-CycleGan: Reinforcement learning aware simulation-to-real. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 11154–11163 (2020). https://doi.org/10.1109/CVPR42600.2020.01117
21. Rusu, A.A., Večerík, M., Rothörl, T., Heess, N., Pascanu, R., Hadsell, R.: Sim-to-real robot learning from pixels with progressive nets. In: Conference on robot learning. pp. 262–270. PMLR (2017)
22. Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., Silver, D.: Mastering Atari, Go, chess and shogi by planning with a learned model. Nature **588**(7839), 604–609 (2020). https://doi.org/10.1038/s41586-020-03051-4, http://dx.doi.org/10.1038/s41586-020-03051-4
23. Siekmann, J., Green, K., Warila, J., Fern, A., Hurst, J.: Blind Bipedal Stair Traversal via Sim-to-Real Reinforcement Learning. Robotics: Science and Systems (2021). https://doi.org/10.15607/RSS.2021.XVII.061
24. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. nature **529**(7587), 484–489 (2016)
25. Stathakis, D.: How many hidden layers and nodes? International Journal of Remote Sensing **30**(8), 2133–2147 (2009)
26. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. Robotica **17**(2), 229–235 (1999)

27. Wang, X., Wang, S., Liang, X., Zhao, D., Huang, J., Xu, X., Dai, B., Miao, Q.: Deep Reinforcement Learning: A Survey. IEEE Transactions on Neural Networks and Learning Systems pp. 1–15 (2022). https://doi.org/10.1109/TNNLS.2022.3207346
28. Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., Freitas, N.: Dueling network architectures for deep reinforcement learning. In: International conference on machine learning. pp. 1995–2003. PMLR (2016)
29. Yu, P.L.: Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives. Journal of Optimization Theory and Applications **14**, 319–377 (1974)
30. Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., Finn, C.: COMBO: Conservative Offline Model-Based Policy Optimization. Advances in Neural Information Processing Systems **35**(NeurIPS), 28954–28967 (2021)