Towards Visual Social Navigation in Photo-realistic Indoor Scenes

Feng Gao[1*] Hengshuang Zhao[2,3] Yu Wang[1]
[1]Tsinghua University [2]The University of Hong Kong [3]Massachusetts Institute of Technology

***Abstract*—As reinforcement learning (RL) has shown great potential in games and robotics, real-world robotic services are drawing more and more attention, such as home-serving robots. For home service, robots should move in a socially compliant way around human beings. However, current state-of-the-art RL-based visual navigation methods for static environments cannot be directly applied to the social setting, which will lead to a high risk of dangerous collisions. In this paper, we use an end-to-end model to address the visual social navigation (VSN) task. We tackle VSN as a non-communication multi-agent problem for which we allow the robot to have the ability of safe navigation through centralized training. To enhance spatial perception, we further apply a goal-aware geometric mapping module before the visual encoder. Besides, to allow our method to adapt to variational environmental settings, we propose a potential function-based self-tuning method from the perspective of uncertainty. We conduct extensive experiments to compare our method with existing baselines, both planning-based and learning-based. The results demonstrate that our agent can surpass baselines on both point-goal navigation and visual social navigation and show better robustness.**

I. Introduction

Navigation is one of the most fundamental skills for autonomous robots, where a robot needs to automatically find a path to reach a specific position within a time limit. If the robot makes decisions on visual observations, it's dubbed visual navigation. A classic approach is to build a map-building-based navigation system with some handcrafted modules, while another emerging way is to directly map the visual inputs to the action via an end-to-end model [1, 2, 3, 4], namely mapless navigation. And they usually use convolutional neural networks to process visual observations and train their agents with reinforcement learning (RL) algorithms. As shown in DD-PPO [3], an RL-based mapless agent can achieve near-perfect performance on point-goal navigation using billions of training samples and distributed RL training.

Unlike navigation in static environments, social navigation requires the robot to move around human beings in a socially compliant way and put more requirements, as shown in Fig. 1. Kruse et al. [5] summarized these requirements into three aspects: comfort, naturalness, and sociability. For comfort, robots should maintain a comfortable distance from humans and move without annoyance and stress beyond safety. Naturalness needs the robot to have similar low-level behavior patterns to humans. And sociability requires the robot to adhere to high-level cultural conventions.

*Corresponding author. e-mail: gao-f19@mails.tsinghua.edu.cn

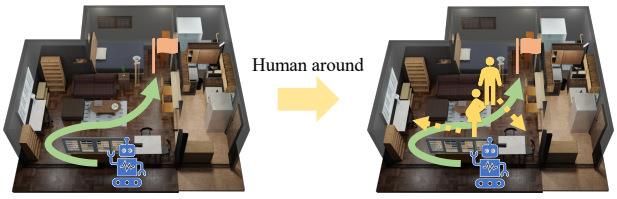

Fig. 1: Comparison between point-goal navigation and social navigation: the latter adds moving pedestrians upon the former along with social constraints.

Since the robot should take into account the status of surrounding pedestrians in social navigation, many prior methods [6, 7, 8, 9, 10] directly fed the ground-truth human states, such as positions and velocities, into their models, which is difficult and infeasible to achieve in the real world. They focused on modeling the human-robot interaction [6, 7] and/or designing novel reward functions [8, 10], and conducted experiments on abstract particle environments, such as CrowdNav [7]. Some recent work has also attempted to perform social navigation in more realistic 3D simulators, most of which adopted lidar sensors to acquire highly-accurate perception [11, 12]. Although great success, the ground-truth human states and expensive lidar sensors are hard to access and therefore limit their real-world usage. Inspired by the recent success of RL-based visual point-goal navigation, we aim to build a vision-based social navigation agent that can perform well to navigate while being aware of social rules. In the following sections, we will step towards visual social navigation (VSN) in photo-realistic indoor environments.

II. Methodology

In this section, we first introduce the definition of visual social navigation and the reward functions for training. Next, we elaborate on our proposed method.

A. Visual Social Navigation

1) Task definition: Visual social navigation refers to the socially constrained point-goal navigation. The agent should navigate to the goal specified by a coordinate while keeping a personal distance away from pedestrians. If the agent reaches the destination safely and timely, the task ends successfully. Otherwise, if the robot collides with a pedestrian or exceeds the time limit, the mission fails. Following [13], we set the (center-to-center) personal distance threshold to 1.5m.

2) Reward functions: According to the task definition, we can divide the task requirements into two aspects: navigation and social compliance. Therein, we adopt a common reward structure for navigation, which contains three parts: the success reward, the decreased geodesic distance $F_{goal}(s_t, a_t)$ as a potential reward, and a slack penalty. Then, the reward function for navigation is formed as

$$R_{nav}(s_t, a_t) = \alpha \mathbb{I}_{Success} + F_{goal}(s_t, a_t) + \gamma \tag{1}$$

where s_t and a_t are the current state and the corresponding selected action respectively, α is set to +10 for awarding success, and $\gamma = -0.01$ is a slack penalty.

For social compliance, we introduce two social constraint rewards similar to some prior work [8, 10]. One is a terminal reward meaning the task fails for colliding with a pedestrian, and the other is to punish the agent for getting too close to pedestrians. We denote the latter penalty as

$$F_{ped}(s_t, a_t) \triangleq \omega \sum_{i=1}^{N} \max\left(0, 1.5 - d(\mathcal{P}(s_t, a_t), z_t^i)\right) \tag{2}$$

, where N is the total number of pedestrians, $\mathcal{P}(\cdot)$ denotes the transition function, z_t^i is the current position of i-th pedestrian, and $d(\cdot, \cdot)$ is the geodesic distance. ω is a hyper-parameter multiple to adjust the scale of punishment. By default, we set ω to -0.05. Then the reward function for social compliance is

$$R_{social}(s_t, a_t) = \beta \mathbb{I}_{Danger} + F_{ped}(s_t, a_t) \tag{3}$$

with β set to -10 for penalizing the agent when it collides with a pedestrian.

Then, our total reward function is

$$R(s_t, a_t) = R_{nav}(s_t, a_t) + R_{social}(s_t, a_t) \tag{4}$$

. For brevity, we will use F_g and F_p to denote $F_{goal}(s_t, a_t)$ and $F_{ped}(s_t, a_t)$ in the remaining paper.

B. Agent Architecture

The overview of our agent architecture is diagrammed in Fig. 2. We adopt the best configuration in a simplified version [14] of DD-PPO [3] as our baseline, i.e., the depth-only agent with GPS+Compass, upon which we build our agent with several improvements. Besides, we also extend it to a depth-semantic setting with an extra semantic segmentation sensor. Sax et al. [15] showed that this mid-level vision could benefit the navigation. Upon the baseline architecture, as we highlight in Fig. 2, we make two main structural improvements, i.e., the goal-aware geometric mapping module and the global-view critic.

1) Goal-aware Geometric Mapping: Many times, it has been proven that the depth-only agent can achieve better navigation performance than those with RGB sensors [16, 3, 14]. An intuitive guess is that the perception of the spatial distance is the most important for the navigation task due to the benefit to obstacle avoidance and local planning. Inspired by this hypothesis, we propose to augment its spatial perception by projecting the ego-view depth and semantic segmentation into the top-down view via geometric mapping, which is commonly used for local mapping in map-build-based systems [17]. Here we adopt it for spatial enhancement instead of cumulatively

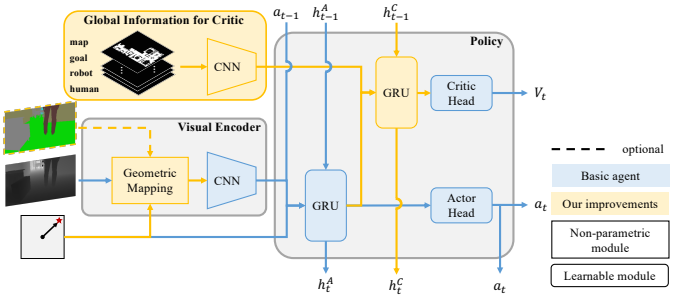

Fig. 2: Overiew of our agent architecture. We highlight our contributions in yellow for clarity.

mapping. Meanwhile, we can also project the target position into this top-down view for goal-aware perception.

For visual observations, we perform the geometric mapping with a two-stage projection, from the front view to the point cloud and then from the point cloud to the top-down view. For the goal projection, we can directly draw it at the top-down view according to its coordinates. If it is located inside the local agent-centric map, we assign the value of the grid point closest to it as one. Otherwise, we take the nearest point p_{map} to the intersection of the line from the robot to the goal and the map and assign its value as the distance ratio $d(p_{map}, robot)/d(goal, robot)$. The rest part of the goal map is zero. By this way, the agent can be better aware of goal-conditioned perception. After projection, we concatenate these augmented observations in the channel wise and feed them into the visual encoder.

2) Global-view Critic for Centralized Training: We use the centralized training and distributed execution (CTDE) to train our agent, which is a popular technology in the multi-agent RL. In other words, we keep a local-view actor to maintain the ego observation while feeding additional privileged information to a global-view critic to enable better training.

It's essential for social navigation, because the agent's intrinsic value function cannot be decided by only immediate observations due to the lack of dynamic environmental information. Taking the privileged information into account, the global-view critic can fairly assess the agent's states and guide its action selection. In detail, the privileged information includes: (i) the global map; (ii) the location of the goal; (iii) the oracle states of the agent, containing its trajectory, location, and future shortest path; (iv) the oracle states of the pedestrians, containing their trajectories, locations and future waypoints. We feed these oracle observations into a CNN encoder to extract global features. Then, the global features are combined with the local features from the actor, and fed into a one-layer GRU followed by a fully-connected layer. Finally, the critic outputs the estimate of the value function. If the uncertainty-aware potential function proposed next is in use, the critic also outputs the estimates of two potential functions, forming a three-head critic.

C. Uncertainty-aware Potential Function

Beyond structural improvements above, we further dig into the reward shaping problem that is also crucial for social

TABLE I: Experimental results on visual social navigation where human density is $\frac{1}{5m^2}$. $\downarrow$ indicates that lower is better while $\uparrow$ indicates that higher is better. Scores in **bold** denote **the best** under the corresponding setting.

Method	Backbone	Reward	Depth	SemSeg	Danger$\downarrow$	Success$\uparrow$	SPL$\uparrow$	STL$\uparrow$	PSC$\uparrow$
GT-SLAM+RRT	-	-	✓	×	0.139	0.379	0.179	0.245	0.378
Wijmans et al.	vanilla CNN	R_{nav}	✓	×	0.191	0.594	0.332	0.544	0.422
Wijmans et al.	ResNet18	R_{nav}	✓	×	0.201	0.582	0.326	0.529	0.420
Wijmans et al.	vanilla CNN	R_{nav}+R_{social}	✓	×	0.128	0.549	0.302	0.500	0.433
Wijmans et al.	ResNet18	R_{nav}+R_{social}	✓	×	0.122	0.426	0.230	0.379	**0.436**
Ours	vanilla CNN	R_{nav}+R_{social}	✓	×	**0.105**	**0.642**	**0.352**	**0.588**	0.435
Wijmans et al.	vanilla CNN	R_{nav}+R_{social}	✓	✓	0.100	0.511	0.279	0.461	**0.443**
Wijmans et al.	ResNet18	R_{nav}+R_{social}	✓	✓	0.110	0.510	0.275	0.460	0.441
Ours	vanilla CNN	R_{nav}+R_{social}	✓	✓	**0.092**	**0.617**	**0.338**	**0.563**	0.430

navigation. As described in Sec. II-A2, our reward function contains two parts (R_{nav} and R_{social}) with two potential rewards (F_g and F_p) inside. Unlike point-goal navigation, approaching the goal along the shortest path may not be optimal in social navigation. For instance, when someone moves in front of the agent, the agent should detour or wait rather than move forward. Nevertheless, excessive penalties for approaching pedestrians will make the robot fail to learn navigation, which is not what we want as well. Therefore, we need to take a balance between navigation and social compliance, namely a reward shaping problem.

In our default setting, we use a hyper-parameter ω (Eq. 2) to measure the degree of penalty for uncomfortable violations, and we set it to a cherry-picked value (-0.05). However, when the environment changes, such as higher pedestrian density, we need to re-pick a proper value of ω; otherwise, the performance will drop or even fail (Fig. 3). To avoid this laborious reward shaping process, we propose a potential function-based self-tuning method based on our centralized training paradigm.

First, to alleviate the side effect of improperly shaped rewards, we recover the potential rewards F_g and F_p by the difference of the corresponding potential functions Φ_g and Φ_p and use the global-view critic to learn these potential functions. We replace the original single-head critic with a three-head one which estimates the original value function in one head and estimates two potential functions Φ_g and Φ_p in the other two.

Formally, as defined in [18], let $s, s' \in S$ be two successive states in a Markov Decision Process (MDP) and $\Phi : S \to \mathbb{R}$ be a real-valued potential function, the potential reward is

$$\mathbf{F}(s, s') = \lambda\Phi(s') - \Phi(s) \tag{5}$$

, where λ is a discount factor.

Then, to achieve self-adaptive tuning, we consider this problem from the perspective of uncertainty, which has been successful in multi-task learning [19]. We regard learning from potential rewards as a regression problem and use a Gaussian distribution to model the likelihood, in which the mean is the multi-head outputs of the critic. The variance σ^2 is a learnable parameter to automatically balance multiple potential functions and the value function. Then we can compute a new value function with the awareness of uncertainty, called uncertainty-aware potential function, by

$$\hat{V} = \frac{1}{\sigma_v^2}V + \frac{1}{\sigma_g^2}\Phi_g + \frac{1}{\sigma_p^2}\Phi_p + \log\sigma_v\sigma_g\sigma_p \tag{6}$$

with σ_v, σ_g, and σ_p for denoising and balancing the estimates of V, Φ_g, and Φ_p respectively, all initialized as $e^{0.01}$.

III. Experiments

A. Experimental Setup

To illustrate the effect of our method, we conduct extensive experiments on a photo-realistic indoor simulator, iGibson [20], in which pedestrians are controlled by an ORCA engine [21]. Following the iGibson challenge [13], we use 8 scenes for training and 7 scenes for evaluation. We train each agent for 10M frames across 8 scenes with the PPO algorithm [22]. For evaluation, we run 100 episodes for each scene and collect the average score of total episodes. We conduct three independent experiments with different random seeds for each agent and take the average as the final results.

In each episode, we set the time limit to 500 steps and initialize the agent and the goal randomly. Only if the agent can reach the goal without any collision with pedestrian and within the time limit, the task is done successfully. Otherwise, the task is failed. Specifically, when the distance between the agent and a pedestrian is less than 0.3m, we terminate the task due to a dangerous collision. The distance threshold for reaching is 0.36m. The human density is $1/(5m^2)$, and there are no more than 10 pedestrians per scene.

For evaluation, we use the following metrics to assess the performance of each method: **(i) Danger:** the percentage of episodes that dangerous collisions occur between the agent and a pedestrian; **(ii) Success:** success rate; **(iii) SPL:** success weighted by (normalized inverse) path length; **(iv) STL:** success weighted by time length, proposed by Li et al. [13]; **(v) PSC:** personal space compliance, proposed by Li et al. [13]. PSC is defined as the percentage of timesteps that the agent maintains a personal distance from humans. Following [13], we set the threshold of PSC to 1.5m.

Let l_{min} be the length of shortest path between the start and the goal, l_{robot} be the length of the robot's actual path, T_{robot} be the time that the agent costs to reach the goal and T_{ORCA} be the time that an oracle ORCA agent costs to reach the same goal, then SPL and STL are as followings:

$$SPL = Success \times \frac{l_{min}}{max(l_{min}, l_{robot})} \tag{7}$$

$$STL = \min\left(Success \times \frac{T_{ORCA}}{T_{robot}}, 1.0\right) \tag{8}$$

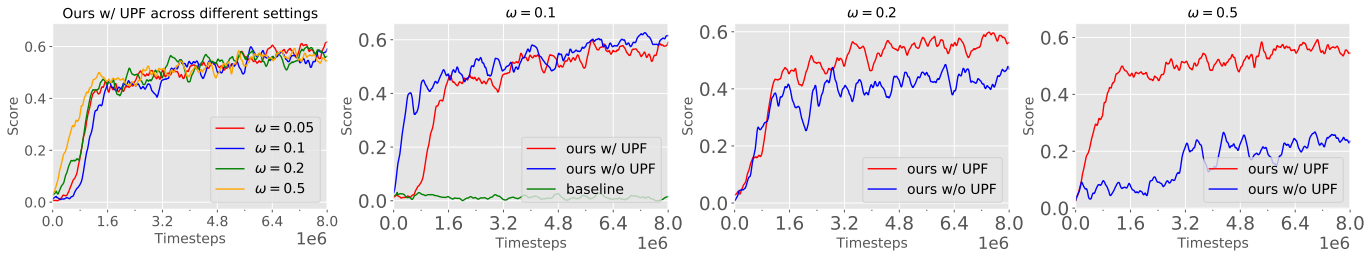

Fig. 3: Effect of our uncertainty-aware potential function method with variational reward weights. The first plot shows that our UPF can adapt to different settings while others shows the advantage of our UPF.

TABLE II: Evaluation on point-goal navigation.

Method	Success	SPL
Wijmans et al.	0.737	0.420
Wijmans et al.+ GM (ours)	0.782	0.445
Wijmans et al.+ GM + GC (ours)	**0.828**	**0.470**

TABLE III: Ablative experiments for depth-only agent.

Method	Danger↓	Success↑	SPL↑	STL↑	PSC↑
Ours	**0.105**	**0.642**	**0.352**	**0.588**	0.435
Ours w/o GM	0.146	0.571	0.310	0.519	**0.443**
Ours w/o GC	0.126	0.606	0.334	0.553	0.428

B. Evaluation

1) Visual social navigation: We compare our agent with a classic SLAM+RRT agent and the best agent from Wijmans et al. [14] under two settings, depth-only agents and depth-semantic agents. For depth-semantic agents, we concatenate visual observations in channel-wise as inputs for baselines. And to show the effect of our introduced social constraint rewards R_{social}, we also show two baseline agents trained with only navigation rewards R_{nav}. Results are shown in Table I.

From these results, we can find that classic planning-based agent will work badly on visual social navigation task, since it don't take the unpredictable dynamics of the environment into account. As for learning-based methods, as discussed earlier, baseline agents trained with only R_{nav} will cause a high risk of dangerous collisions, which will make the home-serving robot unacceptable. On the other hand, baseline agents trained with $R_{nav} + R_{social}$ will work worse on the navigation task although more safely and compliantly. On the contrary, our method can achieve better performance and safer navigation simultaneously for both depth-only and depth-semantic agents.

2) Point-goal navigation.: Beyond visual social navigation, we also conduct experiments on the point-goal navigation. The agents are trained with only navigation rewards R_{nav}. And we keep the same episode data as used in visual social navigation for evaluation, except that there are no pedestrians in the environment. We denote Goal-aware Geometric Mapping as GM and Global-view Critic as GC for simplicity. Since Wijmans et al. [14] has shown that vanilla CNN is comparable with or even better than ResNet18 for point-goal navigation with depth-only agent, we only compare ours with the vanilla CNN here. We add our two structural improvements based on the vanilla CNN agent step by step for a clearer comparison. The results in Table II demonstrate that our structural improvements can also benefit the point-goal navigation.

C. Ablation Study

The previous experiments have demonstrated the overall performance of our agent. Here, we conduct some ablative experiments on visual social navigation under the depth-only setting to illustrate the impact of each structural improvement we propose. The results are shown in Table III, which demonstrate the contribution of each component to the final performance.

D. Effect of Uncertainty-aware Potential Function

In this part, we will analyze the effect of our proposed uncertainty-aware potential function (UPF) to demonstrate the adaptability of our method.

In the RL field, the setting of the reward function is often related to the task setting. When the task setting changes, the reward function needs to be adjusted to adapt to the new setting. For example, in our task, when the human density in the environment increases, the original high social penalty will make the agent not dare to explore new areas, resulting in failure to learn the correct navigation performance, and vice versa. Fig. 3 shows the performance of our method under different reward function settings. As the penalty level (indicated by ω) increases, the performance of the methods without UPF (both baseline and ours) gradually deteriorates, while our method with UPF enabled can achieve almost the same effect under different reward settings. It is worth noting that the baseline method has totally failed at $\omega = 0.1$, while our method can always guarantee a certain learning ability even without using UPF, which shows that our centralized training is also beneficial for the adaptability.

IV. Conclusion

In this paper, we propose an RL-based end-to-end model for visual social navigation. We propose two structural improvements and an uncertainty-aware potential function method upon a point-goal navigation baseline. Extensive experiments show that our method can significantly improve both point-goal navigation and visual social navigation and can automatically adapt to variational task settings due to the uncertainty-aware potential function.

References

[1] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, 2017.

[2] Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. In *ICLR*, 2018.

[3] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *ICLR*, 2019.

[4] Linhai Xie, Andrew Markham, and Niki Trigoni. Snapnav: Learning mapless visual navigation with sparse directional guidance and visual reference. In *ICRA*, 2020.

[5] Thibault Kruse, Amit Kumar Pandey, Rachid Alami, and Alexandra Kirsch. Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 2013.

[6] Yu Fan Chen, Miao Liu, Michael Everett, and Jonathan P How. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In *ICRA*, 2017.

[7] Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *ICRA*, 2019.

[8] Tessa van der Heiden, Florian Mirus, and Herke van Hoof. Social navigation with human empowerment driven deep reinforcement learning. In *ICANN*, 2020.

[9] Kapil Katyal, Yuxiang Gao, Jared Markowitz, I Wang, Chien-Ming Huang, et al. Group-aware robot navigation in crowded environments. *arXiv:2012.12291*, 2020.

[10] Takato Okudo and Seiji Yamada. Reward shaping with subgoals for social navigation. *arXiv:2104.06410*, 2021.

[11] Jing Liang, Utsav Patel, Adarsh Jagan Sathyamoorthy, and Dinesh Manocha. Crowd-steer: Realtime smooth and collision-free robot navigation in densely crowded scenarios trained using high-fidelity simulation. In *IJCAI*, 2020.

[12] Claudia Pérez-D'Arpino, Can Liu, Patrick Goebel, Roberto Martín-Martín, and Silvio Savarese. Robot navigation in constrained pedestrian environments using reinforcement learning. In *ICRA*, 2021.

[13] Chengshu Li, Jaewoo Jang, Fei Xia, Roberto Martín-Martín, Claudia D'Arpino, Alexander Toshev, Anthony Francis, Edward Lee, and Silvio Savarese. igibson challenge 2021: Interactive and social navigation in indoor environments. http://svl.stanford.edu/igibson/challenge.html, 2021.

[14] Erik Wijmans, Irfan Essa, and Dhruv Batra. How to train pointgoal navigation agents on a (sample and compute) budget. In *AAMAS*, 2022.

[15] Alexander Sax, Jeffrey O Zhang, Bradley Emi, Amir Zamir, Silvio Savarese, Leonidas Guibas, and Jitendra Malik. Learning to navigate using mid-level visual priors. In *CoRL*, 2020.

[16] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *ICCV*, 2019.

[17] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *NeurIPS*, 2020.

[18] Andrew Y Ng, Daishi Harada, and Stuart J Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, 1999.

[19] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018.

[20] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne P. Tchapmi, Micael E. Tchapmi, Kent Vainio, Josiah Wong, Li Fei-Fei, and Silvio Savarese. igibson 1.0: a simulation environment for interactive tasks in large realistic scenes. *arXiv:2012.02924*, 2021.

[21] Javier Alonso-Mora, Andreas Breitenmoser, Martin Rufli, Paul Beardsley, and Roland Siegwart. Optimal reciprocal collision avoidance for multiple non-holonomic robots. In *Distributed autonomous robotic systems*. 2013.

[22] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.