Attention Graph for Multi-Robot Social Navigation with Deep Reinforcement Learning

Extended Abstract

Erwan Escudie
Univ. Groningen, Bernoulli Institute
CITI Lab., INRIA-INSA Chroma
Univ Lyon, UCBL, LIRIS
e.c.escudie@rug.nl

Laetitia Matignon
Univ Lyon, UCBL, CNRS, INSA Lyon,
LIRIS, UMR5205
F-69622, FRANCE
laetitia.matignon@univ-lyon1.fr

Jacques Saraydaryan
CPE Lyon, CITI Lab., INRIA-INSA
Chroma team,
Villeurbanne, FRANCE
jacques.saraydaryan@cpe.fr

ABSTRACT

In this paper, we present MultiSoc, a new method for learning multi-agent socially aware navigation strategies using RL. Inspired by recent works on multi-agent deep RL, our method leverages graph-based representation of agent interactions, combining the positions and fields of view of entities (pedestrians and agents). Each agent uses a model based on two Graph Neural Networks combined with attention mechanisms. First an edge-selector produces a sparse graph, then a crowd coordinator applies node attention to produce a graph representing the influence of each entity on the others. This is incorporated into a model-free RL framework to learn multi-agent policies. We evaluate our approach on simulation and provide a series of experiments in a set of various conditions that conclude that our method learns faster than social navigation deep RL mono-agent techniques, and enables efficient multi-agent implicit coordination in challenging crowd navigation with multiple heterogeneous humans. A full description and analysis of this work is available in the full paper version [2].

KEYWORDS

Social Robot Navigation; Multi-Agent Reinforcement Learning; Multi-Robot Navigation; Graph Neural Networks; Predictive models

ACM Reference Format:
Erwan Escudie, Laetitia Matignon, and Jacques Saraydaryan. 2024. Attention Graph for Multi-Robot Social Navigation with Deep Reinforcement Learning: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

Robot navigation in crowded spaces has attracted significant attention in recent years given its numerous potential applications, but it still faces many challenges [10]. Especially understanding pedestrian behavior is crucial to develop effective robot navigation strategies that prioritize human safety. But predicting crowd behavior is difficult and most of approaches intend to learn it from experiment or simulation [4, 5, 16].

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

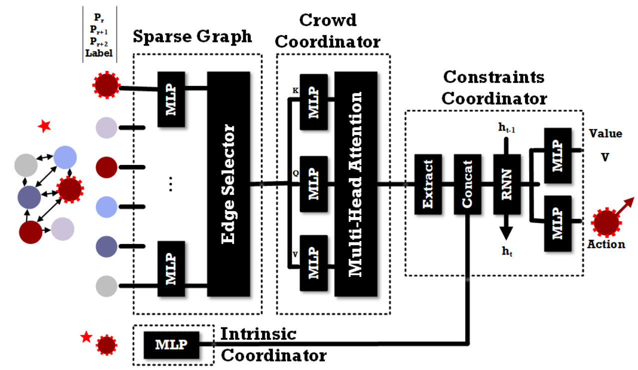

Figure 1: Overview of the MultiSoc architecture. For the agent of interest (surrounded by dotted line), the input is its intrinsic information and a graph limited to its FoV. Each node is composed of the current and consecutive predicted positions of the observed entities, and by a label discriminating entities following their nature.

Recent approaches [1, 8] use deep reinforcement learning (RL) to build social navigation strategies with the help of a simulated crowd. Lately the works of Liu et al. [7] use deep RL combined with attention [13] and graph-based representations [11] of interactions between robot and humans. This achieved very good performance in dense crowds for a single robot. However, as specified by the authors, this model remains difficult to train as it exhibits unstable learning. We propose a model for the learning of multi-robot navigation strategies within crowded environments.

Main challenges compared to the state of the art are learning coordinated and human-safe navigation strategies for the fleet of robots and managing interactions with both controlled (robots) and uncontrolled entities (humans). In our contribution, we highlight the similarity between two approaches that utilize Graph Neural Networks (GNN) [11] to represent, on one hand, human interactions in single-robot social navigation [7], and on the other hand, interactions between agents in multi-robot navigation [14]. Thus GNNs offer a bridge between these two fields which we leverage, combined with attention mechanisms, in our model MultiSoc. MultiSoc follows Centralized Training Decentralized Execution (CTDE) paradigm. During the learning process with MAPPO [15], a multi-agent RL algorithm, the model is shared between robots, taking benefit of each robot experience. But at the execution, each robot processes its input through its MultiSoc model and gets as result its action (commands in velocity). The input is a directed graph with information (current and predicted future poses) concerning the entities (robots and humans) in the field of view (FoV) of the robot.

2 CONTRIBUTION

We introduce MultiSoc, a model for learning multi-agent navigation among humans that can be seen as an homotopy between AttnGraph [7] and MAGE-X [14]. From the former, we keep the attention mechanism and the graph with predicted future positions of the entities. From the latter, we improve edge-selection and take up the graph merging early all the entities. Moreover, unlike AttnGraph where entities ignore each other except for the only robot's consideration of humans, in multi-agent scenarios, interactions and visibilities among controllable agents are crucial for their coordination. That's why in MultiSoc, the computation graph is based on the visibility among entities that is critical. The algorithm has to merge both controllable and uncontrollable entities, all of them interacting with each other. MultiSoc workflow for each agent j is the following (cf. Fig. 1):

- The **Edge-Selector** applies attention on nodes of a graph composed of predicted positions of each entities in the FoV of agent j. This produces a sparse directed graph of the most interesting interactions between entities with adjustable density.
- **The Crowd Coordinator**, a Graph Attention Network with one layer of attention, is applied on the sparse graph to compute node features influenced by neighbors. Meanwhile, the Intrinsic Coordinator produces a broader summary of the constraint applied on the robot (constraint of goal).
- Once the external constraints (Crowd Coordinator) have been correctly represented on the node representing the agent j, this node is extracted and concatenated with the constraint of goal (**Intrinsic Coordinator**).
- Then a GRU, followed by two MLPs, produces both value and action, with respect to the previous hidden state and the information previously computed.

From a technical perspective, the GNNs included in the architecture allows: (i) A flexible computation taking into account as many entities as wanted. It is worth noting that humans and agents are included in the same graph (and not in the architecture itself as does AttnGraph) (ii) A pseudo centralization in the decision acts on each agent in the graph (as node). By extension of the homogeneous paradigm, all other agents with enough information on their own observations, can be accurately understood by the concerned agent. (iii) An extendable receptive field increases the observation space from which the agent takes its action. As in [6], each GNN layer extends the information perceived and allows nodes to be connected indirectly to more nodes.

3 EXPERIMENTATIONS

3.1 Simulation Environment

Simulator. We extend the CrowdNav mono-agent simulator [7] for a multi-agent version. Our multi-agent simulator **MultiCrowd-Nav** is based on multi particle environments (MPE) [9] to facilitate the migration from PPO to MAPPO [15].

Crowd Simulation. Despite their lack of realism, methods such as ORCA [12] or social force (SF) [3] are often used to simulate humans (for learning and testing). Combining the two appears necessary for considering real-life implementation. Thus in our simulator each human can be controlled by ORCA or SF and some experiments will be done with **heterogeneous human policies**.

Scenario. The scenarios are initialized with H humans arranged on a circle with some noise on their positions. Human goals are chosen so that they must cross the circle to reach an opposite point. Humans react only to other humans but not to robots (adversarial crowd). A new random goal is assigned to a human as soon as it reaches its goal. At initialisation, R agents are also laid out randomly with their own goals (positioned and assigned randomly). An episode is over when all agents have either collided (collision) or reached their goal (success). For more details concerning the simulation, the reader can refer to [7].

Metrics. Our metrics include navigation and social metrics traditionally used in multi-robot and social robot navigation.

The **success rate** is the number of agents that reached their goals to the total number of agents, evaluated on all test episodes. Safety is mainly summarized by the **collision rate**, i.e. the number of agents colliding with other entity (humans or not). The proximity of the agent to humans can also be considered to evaluate the social awareness of the agents. We used the **intrusion ratio** as the percentage of time the agent was "too close" to a human averaged over all episodes ("too close" is defined as in [7] with a distance defining the space "close" to an individual). The **reward** is the mean reward obtained by each agent at the end of episode.

3.2 Results

First, the proposed architecture allows a faster training compared to Attngraph (40h for Attngraph, 20h for MultiSoc) under the same conditions.

Second, our **MultiSoc model overcomes AttnGraph especially when several robots are involved.** In the case of 1 robot and 20 humans for both learning and testing, Multisoc has a success rate of 0.96% against 0.92% for AttnGraph. When several robots are involved only in testing (6 R and 6 H) Multisoc obtains a success rate of 0.81% versus only 0.68% for AttnGraph. Finally MultiSoc gets a better result when several robots are involved also the training phase (0.85% of success rate for training with 6 R and 6 H). This shows **a better generalization of our model when dealing with a crowd composed of robots and humans**.

Third our proposed architecture handles the mix of human policies very well even though training was done with homogeneous human policies (ORCA). Indeed tests mixing ORCA and Social Force do not bring any additional difficulties to our model (94% of success rate).

Finally we also demonstrate the **scalability capacities of MultiSoc**. Trainings with different numbers of robots delivered stable performance, and a given training can also adapt to situations with more or less robots. The experiments demonstrate that training with 5 R and 20 H achieves very good results in single-robot (95% success tested with 1 R and 20 H), but also good results with 10 robots (89% success rate, tested with 10 R and 20 H).

REFERENCES

[1] Yuying Chen, Congcong Liu, Bertram E. Shi, and Ming Liu. 2020. Robot Navigation in Crowds by Graph Convolutional Networks With Attention Learned From Human Gaze. IEEE Robotics and Automation Letters 5, 2 (2020), 2754–2761. https://doi.org/10.1109/LRA.2020.2972868

[2] Erwan Escudie, Laetitia Matignon, and Jacques Saraydaryan. 2024. Attention Graph for Multi-Robot Social Navigation with Deep Reinforcement Learning. arXiv:2401.17914 [cs.RO] https://arxiv.org/abs/2401.17914

[3] Gonzalo Ferrer, Anais Garrell, and A. Sanfeliu. 2013. Robot companion: A social-force based approach with human awareness-navigation in crowded environments. IEEE/RSJ International Conference on Intelligent Robots and Systems, 1688–1694.

[4] Yutao Han, Rina Tse, and Mark E. Campbell. 2020. Pedestrian Motion Model Using Non-Parametric Trajectory Clustering and Discrete Transition Points. CoRR (2020).

[5] Zhe Huang, Ruohua Li, Kazuki Shin, and Katherine Driggs-Campbell. 2022. Learning Sparse Interaction Graphs of Partially Detected Pedestrians for Trajectory Prediction. IEEE Robotics and Automation Letters 7, 2 (2022), 1198–1205. https://doi.org/10.1109/LRA.2021.3138547

[6] Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu. 2018. Graph Convolutional Reinforcement Learning. arXiv e-prints, Article arXiv:1810.09202 (Oct. 2018), arXiv:1810.09202 pages. https://doi.org/10.48550/arXiv.1810.09202 arXiv:1810.09202 [cs.LG]

[7] Shuijing Liu, Peixin Chang, Zhe Huang, Neeloy Chakraborty, Kaiwen Hong, Weihang Liang, D. McPherson, Junyi Geng, and Katherine Driggs-Campbell. 2023. Intention Aware Robot Crowd Navigation with Attention-Based Interaction Graph. https://github.com/Shuijing725/CrowdNav_Prediction_AttnGraph. In IEEE International Conference on Robotics and Automation (ICRA). 12015–12021.

[8] Shuijing Liu, Peixin Chang, Weihang Liang, Neeloy Chakraborty, and Katherine Driggs-Campbell. 2021. Decentralized Structural-RNN for Robot Crowd Navigation with Deep Reinforcement Learning. In 2021 IEEE International Conference on Robotics and Automation (ICRA). 3517–3524.

[9] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. arXiv e-prints (2017).

[10] Christoforos Mavrogiannis, Francesca Baldini, Allan Wang, Dapeng Zhao, Pete Trautman, Aaron Steinfeld, and Jean Oh. 2023. Core Challenges of Social Robot Navigation: A Survey. ACM Transactions on Human-Robot Interaction 12, 3 (2023), 1–39.

[11] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. IEEE Transactions on Neural Networks (TNN) 20, 1 (2009), 61–80. https://doi.org/10.1109/TNN.2008.2005605

[12] Jur van den Berg, Stephen J. Guy, Ming Lin, and Dinesh Manocha. 2011. Reciprocal n-Body Collision Avoidance. In Robotics Research. 3–19.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. (2017), 6000–6010.

[14] Xinyi Yang, Shiyu Huang, Yiwen Sun, Yuxiang Yang, Chao Yu, Wei-Wei Tu, Huazhong Yang, and Yu Wang. 2023. Learning Graph-Enhanced Commander-Executor for Multi-Agent Navigation. In Proceedings of AAMAS. 1652–1660.

[15] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre M. Bayen, and Yi Wu. 2021. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In Neural Information Processing Systems.

[16] Bolei Zhou, Xiaoou Tang, and Xiaogang Wang. 2014. Learning Collective Crowd Behaviors with Dynamic Pedestrian-Agents. International Journal of Computer Vision (2014).