MAC-ID: Multi-Agent Reinforcement Learning with Local Coordination for Individual Diversity

Hojun Chung[1,3], Jeongwoo Oh[2,3], Jaeseok Heo[2,3], Gunmin Lee[2] and Songhwai Oh[1,2,3]

***Abstract*— With the increase of robots navigating through crowded environments in our daily lives, the demand for designing a socially-aware navigation method considering human-robot interaction has risen. When developing and assessing socially-aware navigation methods, pedestrian motion modeling plays a significant role. However, existing pedestrian models often struggle in complex environments and do not have the capacity to generate diverse pedestrian styles.**

In this paper, we propose multi-agent reinforcement learning with local coordination for individual diversity (MAC-ID), which can synthesize diverse pedestrian motions via local coordination factor (LCF). Our experiments have demonstrated that the manipulation of the LCF induces interpretable changes in pedestrian behaviors, along with a superior performance compared to existing pedestrian motion models. For evaluating socially-aware navigation methods using MAC-ID, we present a novel benchmark called BSON. It offers realistic and diverse social environments with pedestrians modeled via MAC-ID. We have trained and compared various navigation methods in BSON using a newly proposed metric called socially-aware navigation score (SNS). Through BSON, users can evaluate their socially-aware navigation methods and compare them to baselines.

I. Introduction

Autonomous mobile robots (AMRs) that can navigate through crowded environments are becoming increasingly prevalent in our daily lives [1]–[3]. In order to provide effective assistance in everyday situations, it is crucial for such robots to perform socially-aware navigation, which does not disturb nearby pedestrians. However, as pedestrians are dynamic obstacles, traditional static obstacle avoidance methods can confront issues when executed in crowded environments. Also, since pedestrians react to the movement of the robot, the difficulty of considering pedestrian motions for navigation is increased. These characteristics raises the difficulty of developing socially-aware navigation methods.

To adress such issue, previous works [4]–[7] have proposed socially-aware navigation systems considering human-robot interaction. Okal et al. [4] have proposed a method utilizing a graph-based representation, and models pedestrian motions using the social force model [8]. Other works have employed deep reinforcement learning to synthesize a socially-aware policy [5], [6] or have simulated future states of pedestrians to obtain the safest action [7]. These methods have used the ORCA model [9] for modeling pedestrian motions. Although choosing the appropriate pedestrian motion model is crucial for objective evaluation of socially-aware navigation methods, they do not use a unified method for modeling pedestrian motions. [10]

Traditional methods for modeling pedestrian motions have used a rule-based approach [8], [9]. The social force model [8] decides the movement of pedestrians using the attractive force towards the goal and repulsive forces from surrounding pedestrians and the environment. The ORCA model [9] controls pedestrians to avoid collisions with each other by constantly adjusting their velocity to keep a safe distance. However, it is difficult to set appropriate hyperparameters for complicated environments such as crowded environments or rough terrain.

On the other hand, several methods have used multi-agent reinforcement learning (MARL) [11]–[14], which can be adaptable in a complicated environment with an appropriate reward function. Since MARL methods make all agents execute the same policy even though each pedestrian moves in a different manner in the real world, it is difficult to set the motion style for each agent.

To overcome such issues, we propose multi-agent reinforcement learning with local coordination for individual diversity (MAC-ID), which can set each pedestrian style using an interpretable hyperparameter. In MAC-ID, the local coordination factor (LCF) [14] determines whether each agent is cooperative or competitive. During training, a random value is assigned to the LCF of each agent, and the LCF-conditioned policy network is optimized. It enables diverse human behaviors in pedestrian simulation by controlling the LCF of each agent. We evaluate MAC-ID through simulations with different pedestrian densities. Experimental results reveal a significant correlation between LCF and agent behaviors, while pedestrians modelled with MAC-ID achieve 7.8% improved performance in terms of success rate with 29.5% reduced collisions.

To enable the usage of MAC-ID for future research, we introduce a benchmark for socially-aware navigation (BSON) using the Unity 3D game engine [15]. Provided in BSON is the Jackal robot platform [16] equipped with multiple sensors and the goal of navigating towards a target point in crowded environments. As the pedestrians in BSON move under MAC-ID, users can adjust pedestrian motion styles using LCF. Also, users can control the robot via gym [17], which is a widely used framework for reinforcement learning research. Furthermore, we have evaluated several existing navigation methods [18]–[20] in the proposed benchmark. BSON will be made available publicly.[1]

[1]Graduate School of Artificial Intelligence (GSAI) and ASRI, Seoul National University, Seoul, Korea (e-mail: hojun.chung@rllab.snu.ac.kr, songhwai@snu.ac.kr), [2]Department of Electrical and Computer Enginnering and ASRI, Seoul National University, Seoul, Korea (e-mail: {jeongwoo.oh, jaeseok.heo, gunmin.lee}@rllab.snu.ac.kr), [3]Sequor Robotics, Seoul, Korea.

This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01190, [SW Star Lab] Robot Learning: Efficient, Safe, and Socially-Acceptable Machine Learning, 50%) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF2022R1A2C2008239, General-Purpose Deep Reinforcement Learning Using Metaverse for Real World Applications, 50%). (*Corresponding author: Songhwai Oh.*)

[1]https://github.com/rllab-snu/BSON

In summary, the contributions of this paper are as follows:

- We propose a new pedestrian motion model called MAC-ID, which can control the pedestrian motion style using an interpretable parameter.
- We introduce a new benchmark BSON for evaluating socially-aware navigation methods, which utilizes MAC-ID to model pedestrian motions
- We provide baseline results for existing navigation methods in the proposed benchmark.

II. Related Work

The pedestrian motion can be modeled in three different scales [21], [22]: flow-based models [23]–[27], entity-based models [28]–[31], and agent-based models [8], [9]. While some pedestrians in flow-based models and entity-based models do not have any individual personality, pedestrians in agent-based models decide their own behaviors while interacting with others. In this paper, we focus on agent-based models with continuous action spaces. Existing approaches for agent-based models include the social force model [8] and the ORCA model [9]. The social force model designs the local movements using several forces, including an attractive force to the goal and repulsive forces from nearby pedestrians. The ORCA model solves a low-dimensional linear program to obtain the optimal control input for collision avoidance.

In contrast to rule-based approaches, several methods have used reinforcement learning for modeling pedestrian motions. Torrey *et al.* [11] have shown the validity of MARL for simulating pedestrians, and Lee *et al.* [12] utilize the deep reinforcement learning to model pedestrian motions. Recently, independent proximal policy optimization (IPO) [18] and mean field RL [32] are widely used for multi-agent control tasks with continuous action spaces. Based on IPO, Peng *et al.* have proposed coordinated policy optimization (CoPO) [14], which utilizes a single optimized local coordination factor (LCF) distribution to define the degree of cooperativeness of each agent. The definition of LCF is motivated by Schwarting *et al.* [33] and Toghi *et al.* [34], where LCF is employed to consider interactions between agents and solve the dynamic game.

Data-driven methods [35], [36] that utilize human trajectory datasets are also used for modeling pedestrian motion, but this paper focuses on generating diverse motions without relying on data.

Pedestrian motion models can be used for crowd simulation frameworks such as Nomad [37], and Menge [38]. SEAN 2.0 [39] introduces a behavior graph to simulate varying social situations with pedestrians controlled using the social force model. In contrast, the proposed BSON uses MAC-ID, which can generate diverse pedestrian motions.

III. Proposed Method

A. Problem Setting

In the real world, individual pedestrians navigate towards their destinations using only the information about its surrounding environment. Therefore, we can consider the pedestrian motion model as decentralized partially observable Markov decision processes (Dec-POMDPs) [40]. Dec-POMDPs are defined by the tuple $(I, S, A, O, P, R, \rho_0)$. Let $I = [e_1, e_2, ..., e_K]$ be a set of agents, where K is the number of agents, S is the environmental state space, A is the action space of the agents, O is the observation space of the agents, P is the state transition model, $R_{1:K}$ are the reward functions of the agents, and the initial state s_0 is sampled from the initial state distribution ρ_0. Each agent e_k has no access to the environmental state $s_t \in S$ and only uses a history of the individual observation $o_{k,t} \in O$ at environmental time step t. The action of each agent $a_{k,t} \in A$ is determined by a policy $\pi(\cdot|o_{k,1:t})$ to form a joint action $a_t \in A^K$. Then the next state s_{t+1} follows the state transition model $P(s_{t+1}|s_t, a_t)$ and each agent receives a reward $r_{k,t} = R_k(s_t, a_t)$. The goal of each agent is to maximize its discounted reward sum $\sum_{t'=0}^{T} \gamma^{t'} r_{k,t'}$, where γ is a discount factor, and T is the episode length.

B. Local Coordination

The motion of each pedestrian is influenced by the interaction between its surrounding pedestrians. To utilize this property, each agent considers both its individual reward and its neighborhood reward during training MAC-ID. While the individual reward $r^I_{k,t}$ aims to train each agent independently without considering other agents, the neighborhood reward $r^N_{k,t}$ considers nearby agents, and is defined as follows:

$$r^N_{k,t} := \begin{cases} \frac{\sum_{j \in N(k,t)} r^I_{j,t}}{|N(k,t)|}, & \text{if } |N(k,t)| \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $N(k,t) = \{j : e_j \in I \setminus e_k, \text{dist}(e_k, e_j) \leq d_n\}$ is a set of neighborhood, $\text{dist}(e_k, e_j)$ is the Euclidian distance between two agents e_k and e_j, and d_n is the neighborhood threshold.

Then, each agent receives a reward $r_{k,t}(\phi_k) = r^I_{k,t} + \phi_k r^N_{k,t}$, where ϕ_k is the local coordination factor (LCF) of an agent e_k, which indicates the degree of cooperativeness. Intuitively, while $\phi_k = 0$ induces a selfish pedestrian manner, $\phi_k \gg 0$ represents a magnanimous pedestrian manner and $\phi_k \ll 0$ represents an adversarial one. Thus, the degree of cooperativeness of each agent can be customized by adjusting the LCF.

MAC-ID trains the policy $\pi_\theta(\cdot|o_{k,0:t}, \phi_k)$ with parameters θ conditioned on ϕ_k so that an action distribution from the fixed observation depends on the LCF. Furthermore, this means that the expected return varies according to the LCF. Therefore, we use an individual value function $V^I_\eta(o_{k,0:t}|\phi_k) = \mathbb{E}_\pi \left[\sum_{t'=t}^{T} \gamma^{t'-t} r^I_{k,t'}\right]$ and a neighborhood value function $V^N_\psi(o_{k,0:t}|\phi_k) = \mathbb{E}_\pi \left[\sum_{t'=t}^{T} \gamma^{t'-t} r^N_{k,t'}\right]$ conditioned on ϕ_k, with parameters η and ψ, respectively. Then, an individual advantage function $A^I_\eta(o_{k,0:t}|\phi_k)$ and a neighborhood advantage function $A^N_\psi(o_{k,0:t}|\phi_k)$ can be obtained using generalized advantage estimation (GAE) [41]. The advantage function is calculated by the sum of the individual advantage function and the neighborhood advantage function weighted by the LCF as follows:

$$A(o_{k,0:t}|\phi_k) = A^I_\eta(o_{k,0:t}|\phi_k) + \phi_k A^N_\psi(o_{k,0:t}|\phi_k). \tag{2}$$

C. MAC-ID

During the training stage, a random LCF $\phi_k \in (-\bar{\phi}, \bar{\phi})$ is assigned to each agent and fixed within an episode to stabilize the learning process of value networks. This

encourages each agent to output a diverse trajectory, which can be adversarial or cooperative depending on the LCF. After training, users can assign diverse LCFs at each timestep within an episode using the optimized LCF-conditioned policy.

We use a modified version of the clipped objective proposed in proximal policy optimization (PPO) [18] to train the policy. Instead of utilizing an individual advantage function, a coordinated advantage function (2) is used to incorporate local coordination into the training process. After each episode, the policy is updated towards minimizing the coordinated policy loss as follows:

$$J_\pi(\theta) = -\mathbb{E}_{\tau\sim D}\left[\min\left(\rho A, \text{clip}\left(\rho, 1-\epsilon, 1+\epsilon\right) A\right)\right], \quad (3)$$

where $\tau = \left(o_{k,0:t}, a_{k,t}, r^I_{k,t}, r^N_{k,t}, \phi_k\right)$ is the transition tuple with the LCF of the k-th agent at timestep t, D is a rollout buffer, which contains all transition tuples from one episode, $\rho = \pi_\theta(a_{k,t}|o_{k,0:t}, \phi_k)/\pi_{old}(a_{k,t}|o_{k,0:t}, \phi_k)$ is the probability ratio, π_{old} is the policy before update, and ϵ is a hyperparameter.

The individual value network V^I_η and the neighborhood value network V^N_ψ are trained to minimize the mean-squared error between the value estimation and the episode return. The loss function of networks J_{V^I} and J_{V^N} are defined as follows:

$$\begin{aligned} J_{V^I}(\eta) &= \mathbb{E}_{\tau\sim D}\left[\left(V^I_\eta\left(o_{k,0:t}|\phi_k\right) - \sum_{t'=t}^{T}\gamma^{t'-t} r^I_{k,t'}\right)^2\right], \\ J_{V^N}(\psi) &= \mathbb{E}_{\tau\sim D}\left[\left(V^N_\psi\left(o_{k,0:t}|\phi_k\right) - \sum_{t'=t}^{T}\gamma^{t'-t} r^N_{k,t'}\right)^2\right]. \end{aligned} \quad (4)$$

IV. Experiments

A. Environment Setup

We implement a set of crowd environments based on the Unity 3D game engine [15] shown in **Figure** 1. The task of each pedestrian is to reach a given goal location on a map with size $50\ m \times 50\ m$. Each pedestrian is modeled as a cylinder with a radius of $0.35\ m$ and a height of $1.7\ m$. For local observation, each pedestrian receives a vector of 41 dimensions, which is composed of the relative position of the goal, the magnitude of the velocity and the angular velocity, a collision indicator, and information about surrounding obstacles. Information about surrounding obstacles is composed of the distance and relative velocity of nearby objects detected using 12 isotropic rays. If the ray detects nothing, the measurement value when a pedestrian with zero relative velocity locates at the maximum distance, $7.0\ m$, is used. Pedestrians are controlled by setting the velocity and angular velocity, with a maximum velocity of $1.5\ m/s$ and a maximum angular velocity of $1.0\ rad/s$.

We design the individual reward function to motivate each pedestrian to reach the goal in the shortest time while avoiding collisions as follows:

$$r^I = \omega_d * r_d + \omega_g * I_g + \omega_c * I_c, \quad (5)$$

where r_d is the reward proportional to the distance reduced from the pedestrian towards the goal, I_g is an indicator function, which is activated when the Euclidean distance between the goal and the agent is less than $0.5\ m$, I_c is an indicator function, which is activated if a collision is triggered by the Unity 3D engine. ω_d, ω_g, and ω_c are hyperparameters. The defined individual reward function is used to calculate the neighborhood reward (1), and each agent receives the sum of the individual reward and the neighborhood reward weighted by the LCF as described in Section 3.

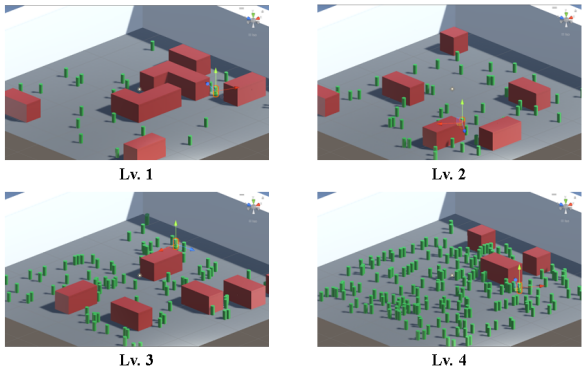

Fig. 1: **Unity pedestrian simulation environment.** This figure is an example scene of the experiment environment. The white elements are walls, the red boxes are static obstacles, and the green cylinders are pedestrians.

To evaluate the trained pedestrian models, we use four maps with static obstacles and a different number of pedestrians shown in **Figure** 1. The number of pedestrians in **Lv.1**, **Lv.2**, **Lv.3**, and **Lv.4** are 25, 50, 100, and 200, respectively. For comparing several pedestrian motion models, the following three metrics are used:

- **Success rate (SR$_n$)**: The percentage of pedestrians who have reached their assigned goals. Only pedestrians who have collided less than n times are counted.
- **Collision penalty (CP)**: The average number of timesteps each pedestrian collides with an obstacle or surrounding pedestrians during a single episode.
- **Path efficiency (PE)**: The ratio between the distance from the initial position to its goal and the total distance traversed by the agent.

In the real world, collisions can occur between pedestrians when navigating through densely crowded environments. Therefore, we have used **SR**$_n$ instead of **SR**$_1$ to assume motions with collisions within a given threshold n as success. In addition, we incorporate **CP** and **PE** into our evaluation, since pedestrians prefer to navigate efficiently while avoiding collisions.

B. Baselines

To evaluate the performance of the MAC-ID, we use social force model and three distinct multi-agent algorithms including IPO [13], MFPO, and CoPO [14] as baselines. IPO optimizes each agent's policy independently to maximize their individual rewards without considering neighborhood rewards. MFPO utilizes the mean states of nearby agents as the additional input to the value functions following the idea of mean field MARL [32]. CoPO optimizes a single local coordination factor to maximize the global reward while training the policy using the coordinated policy loss. It follows trigonometric reward formulation $r_{k,t}(\phi_k) = \cos(\phi_k) r^I_{k,t} + \sin(\phi_k) r^N_{k,t}$ of SVO [33], which differs from

LCF	Lv.1			Lv.2			Lv.3			Lv.4		
	SR$_{10}$(↑)	**CP**(↓)	**PE**(↑)	**SR**$_{10}$(↑)	**CP**(↓)	**PE**(↑)	**SR**$_{10}$(↑)	**CP**(↓)	**PE**(↑)	**SR**$_{10}$(↑)	**CP**(↓)	**PE**(↑)
(-2.00, -1.33)	58.7	38.7	93.3	40.9	57.9	91.1	19.5	83.2	85.3	4.6	157.8	76.4
(-1.33, -0.67)	67.4	24.2	94.2	46.9	40.7	92.6	26.5	56.7	88.4	6.6	119.8	81.0
(-0.67, 0.00)	72.6	15.7	94.9	55.1	22.4	93.4	31.7	38.0	89.8	9.3	90.0	82.7
(0.00, 0.67)	75.8	10.1	**95.3**	58.4	16.2	**93.5**	38.8	23.8	**89.9**	13.4	67.3	**82.9**
(0.67, 1.33)	76.0	7.1	94.6	62.5	10.4	92.9	43.4	19.3	89.1	16.8	55.0	81.6
(1.33, 2.00)	**76.7**	**4.0**	93.7	**63.8**	**8.9**	91.1	**46.3**	**15.0**	86.6	**18.8**	**49.7**	78.2

TABLE I: **Performance of MAC-ID in accordance to the LCF distribution.** The table shows the effect of the distribution of the LCFs on the performance of MAC-ID. We assign the LCF of each pedestrian acquired from the given LCF distribution.

our linear formulation. We have trained the proposed method with the trigonometric reward formulation over an LCF domain of $(-\pi/2, \pi/2)$ to investigate the impact of different reward formulations on our algorithm. SVO [33] seeks Nash equilibrium based on the actions of other agents, but it is difficult to apply as we assume a decentralized POMDP. Finding Nash equilibrium using the proposed algorithm is left as future work.

C. Implementation Details

We have set the length of a single episode T to $1,000$, the total number of episodes M to $1,000$, and the neighborhood threshold d_n to 5.0 m. The hyperparameters for the individual reward function ω_d, ω_g, and ω_c are each set to 10.0, 1.0, and 0.6, respectively. For optimizing the value and policy network as shown in (3), (4), the learning rate is set to 0.001, the clip parameter ϵ is set to 0.2, the discount factor γ is set to 0.999, the parameter λ for estimating the advantage using GAE [41] is set to 0.95, the batch size B is set to $2,048$, and the number of training epochs P is set to 5. $\bar{\phi}$ is set to 2.0 to reduce the search space for LCF, so that the domain of the LCF is $(-2.0, 2.0)$. All agents share the same policy and value functions, which consist of three hidden layers with size $(64, 64, 64)$. PPO is trained with the same parameters except for the batch size B and the training epochs P, which are set to 128 and 10, respectively. We use the current local observation $o_{k,t}$, to decide the control input rather than using a whole history $o_{k,0:t}$. The training machine had 128 GB of RAM, an Intel i9 12900K CPU, and an NVIDIA RTX 3080TI GPU. Each method is trained in **Lv.2** for seven different seeds and evaluated in **Lv.1**, **Lv.2**, **Lv.3**, and **Lv.4** five times for each seed.

V. Results

A. Role of Local Coordination Factor

Table I shows **SR**$_{10}$, **CP**, and **PE** according to the change of the LCF distribution assigned to pedestrians. The domain of the LCF is divided into six sections composed of $(-2.00, -1.33), (-1.33, -0.67), \ldots, (1.33, 2.00)$ to demonstrate the correlation between the pedestrian behavior and the LCF. As LCF increases, **CP** monotonically decreases while **SR**$_{10}$ monotonically increases in all environments. When negative LCFs within $(-2.00, -1.33)$ are assigned to pedestrians, collisions occur more frequently, and it becomes more difficult for each pedestrian to reach goals due to the adversarial behavior of surrounding pedestrians. On the contrary, when positive LCFs within $(1.33, 2.00)$ are assigned, pedestrians cooperate to maximize the neighborhood reward, enabling each pedestrian to reach its goal location due to fewer collisions.

PE shows a tendency to increase as LCF increases until a certain point near zero, then decreases afterwards. If the LCF is higher than zero, pedestrians become cooperative and follow inefficient paths to help others can move easily. On the contrary, when the LCF is lower than zero, collisions occur more frequently as each pedestrian attempts to move more egoistically, degrading efficiency. LCFs within $(0.00, 0.67)$ show the highest **PE** in all environments. In summary, when LCF increases, they move more stably and altruistically, whereas they show a tendency to move aggressively when LCF decreases. It can be seen that the manipulation of the LCF induces interpretable changes in motions styles, and users can assign different LCFs to create various pedestrian motions.

B. Baseline Comparison

In this section, we compare the performance of MAC-ID to existing pedestrian motion models in **Lv.1**, **Lv.2**, **Lv.3**, and **Lv.4**. For a fair comparison, LCFs are sampled from $(0.67, 1.33)$ and assigned to all pedestrians while testing MAC-ID.

The comparison result is summarized in **Table** II. Our method achieves the best **SR**$_{10}$ and **CP** compared to baseline algorithms in all environments. Specifically, MAC-ID shows 7.8% to 33.3% increased **SR**$_{10}$ and 29.5% to 42.2% decreased **CP** compared to baseline methods. Since the adversarial situations in the training stage induce better exploration, it can be inferred that MAC-ID generates safer motions towards reaching the goal while avoiding collisions. In addition, MAC-ID shows the highest **PE** in **Lv.1**, but the performance enhancement diminishes to -2.7% in **Lv.4**. This result highlights how cooperative agents mitigate collisions and provide assistance to other agents by selecting less efficient paths in complex environments.

Among baselines, IPO shows the highest **SR**$_{10}$ with the exception of **Lv.1**, while CoPO shows no performance improvement. MFPO shows better **PE** than IPO, but outputs a higher number of collisions compared to different MARL baselines. To validate MARL methods, we assessed the performance of the classical social force model [8]. In comparison to the social force model, MAC-ID consistently exhibits less than half **CP** across all environments. From this, we can see that the social force model struggles to handle a high volume of pedestrians due to increased repulsive forces from nearby individuals.

MAC-ID (tri) is trained using trigonometric reward formulation $r_{k,t}(\phi_k) = \cos(\phi_k) r^I_{k,t} + \sin(\phi_k) r^N_{k,t}$ to discuss

Method	Lv.1			Lv.2			Lv.3			Lv.4		
	$\mathbf{SR}_{10}(\uparrow)$	$\mathbf{CP}(\downarrow)$	$\mathbf{PE}(\uparrow)$	$\mathbf{SR}_{10}(\uparrow)$	$\mathbf{CP}(\downarrow)$	$\mathbf{PE}(\uparrow)$	$\mathbf{SR}_{10}(\uparrow)$	$\mathbf{CP}(\downarrow)$	$\mathbf{PE}(\uparrow)$	$\mathbf{SR}_{10}(\uparrow)$	$\mathbf{CP}(\downarrow)$	$\mathbf{PE}(\uparrow)$
social force	38.4 ± 0.0	13.3 ± 0.0	76.3 ± 0.0	19.2 ± 0.0	73.8 ± 0.0	77.3 ± 0.0	8.2 ± 0.0	86.5 ± 0.0	71.8 ± 0.0	2.7 ± 0.0	128.4 ± 0.0	54.4 ± 0.0
IPO	69.6 ± 5.5	10.1 ± 4.0	93.7 ± 0.3	53.7 ± 6.2	21.0 ± 6.5	92.4 ± 0.7	34.8 ± 6.3	31.1 ± 9.5	89.2 ± 0.6	12.9 ± 4.4	75.8 ± 21.6	83.1 ± 1.2
MFPO	71.3 ± 2.4	12.8 ± 4.0	93.8 ± 0.7	49.5 ± 2.1	25.9 ± 3.4	92.6 ± 0.7	29.5 ± 3.1	40.7 ± 5.2	$\mathbf{89.6 \pm 1.0}$	10.5 ± 1.9	90.2 ± 9.6	$\mathbf{83.7 \pm 1.4}$
CoPO	69.1 ± 3.2	8.8 ± 2.8	93.9 ± 0.5	50.4 ± 3.8	20.6 ± 5.3	$\mathbf{92.9 \pm 0.5}$	29.8 ± 3.3	31.7 ± 6.4	89.3 ± 1.0	10.4 ± 2.1	76.1 ± 8.9	83.6 ± 2.1
MAC-ID (tri)	$\mathbf{79.0 \pm 3.4}$	$\mathbf{3.4 \pm 3.2}$	92.0 ± 1.1	$\mathbf{68.0 \pm 4.0}$	$\mathbf{10.0 \pm 5.5}$	89.9 ± 1.6	$\mathbf{48.5 \pm 6.9}$	$\mathbf{12.2 \pm 5.2}$	84.1 ± 1.9	$\mathbf{20.3 \pm 7.3}$	$\mathbf{47.1 \pm 17.9}$	76.0 ± 2.3
MAC-ID	$\mathbf{76.9 \pm 4.4}$	$\mathbf{5.1 \pm 3.4}$	$\mathbf{94.6 \pm 0.3}$	$\mathbf{64.5 \pm 1.8}$	$\mathbf{12.3 \pm 6.0}$	$\mathbf{92.9 \pm 0.7}$	$\mathbf{43.7 \pm 4.3}$	$\mathbf{18.6 \pm 7.1}$	89.1 ± 1.4	$\mathbf{17.2 \pm 4.1}$	$\mathbf{53.7 \pm 11.4}$	81.4 ± 2.9

TABLE II: **Pedestrian motion modeling performance comparison.** The table shows the comparison results of different pedestrian motion models. We present mean and standard deviation over seven different seeds.

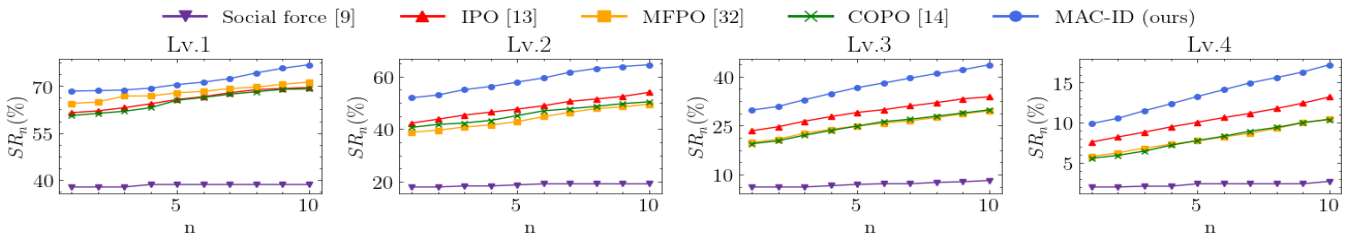

Fig. 2: **Success rate ($\mathbf{SR}_n$) in accordance to the maximum number of tolerated collisions (n)**. The figure shows the results of the success rate of different pedestrian motion models. In each graph, the x-axis represents the maximum number of collisions n, and the y-axis indicates $\mathbf{SR}_n$, whose unit is $\%$. We use MAC-ID with LCFs within $(0.67, 1.33)$.

the impact of different reward formulations on the proposed method. For a fair comparison, LCFs sampled from the fixed interval $(\pi/6, \pi/3)$ are used while evaluating MAC-ID (tri). It outperforms MAC-ID in terms of $\mathbf{SR}_{10}$ and **CP**, but shows the worst **PE** among MARL methods. We have chosen the linear reward formulation for considering overall performance, but the trigonometric reward formulation can be a good option for users less concerned with path efficiency.

To analyze the influence of the maximum number of tolerated collisions n that define $\mathbf{SR}_n$, we provide the experimental result in **Figure** 2. It shows that MAC-ID forms an upper bound on $\mathbf{SR}_n$ for n less or equal to 10. Specifically, the proposed method shows 2.8% to 33.3% increased $\mathbf{SR}_n$ as the pedestrian density increases from **Lv.1** to **Lv.4**. Consequently, our method achieves the highest success rate regardless of the choice of n.

VI. Benchmark for Socially-Aware Navigation

In this section, we introduce the benchmark for socially-aware navigation (BSON). It aims to provide a platform to train and evaluate the socially-aware navigation algorithms in realistic and diverse social environments, based on the Unity 3D game engine [15]. Unlike other benchmarks, we utilize MAC-ID to generate diverse pedestrian behaviors, ranging from adversarial to cooperative styles. It ensures a fair evaluation by preventing socially-aware navigation algorithms specialized in a particular pedestrian model from receiving inflating scores.

A. Benchmark Description

The task of BSON is point goal navigation, which aims to navigate the robot towards a given point goal in a crowded environment. In this work, we have used the Jackal robot platform, bur it is possible to utilize different robots using its model file.

We offer two benchmark modalities, **Local** track and **Global** track. The **Local** track only provides the relative goal position, while the **Global** track provides the global occupancy map and current positions of all pedestrians as additional information. The **Global** track can be used to implement path planning algorithms such as RRT [42], RRT* [43], and CARRT* [44]. Each track includes three distinct sensor setup; using vision sensors including a RGB camera and depth camera, using a 2D LIDAR, and using a 3D LIDAR. Mounting position, spec, and the number of sensors, is all fixed for a fair comparison. We provide a simple python API based on gym [17], which is the standard API for reinforcement learning. It enables users to easily train and evaluate their proposed algoritms.

B. Environments

BSON offers a set of environments to train and test socially-aware navigation algorithm in diverse social situations. Additionally, we provide a functionality to control the density of static obstacles, the density of pedestrians, and the local coordination factor (LCF) of pedestrians. The density of static obstacles is a parameter, which varies between zero and one. As it increases, the number of static obstacles gradually grows, creating a maze-like environment. The density of pedestrians also varies between zero and one, controlling the number of pedestrians whose maximum is decided by the scale of the environment. LCFs of pedestrians control the degree of cooperativeness. Negative LCFs induce an adversarial pedestrian manner, whereas positive LCFs induce a cooperative pedestrian manner. These three variables are employed to control the complexity levels for conducting socially-aware navigation. To support researchers, we have categorized the test maps by the three criteria mentioned above to make it easier to understand under which circumstances the performance of the socially-aware navigation algorithm is poor.

C. Evaluation Metric

To benchmark a socially-aware navigation algorithm, we propose a new score metric called socially-aware navigation score (**SNS**) as follows:

$$\textbf{SNS} = \frac{\sum_{k=1}^{N} \textbf{RC}_k \gamma^{c_k} \times (\textbf{PE}_k \times 50 + \textbf{SP}_k \times 50)}{N}, \quad (6)$$

where N is the number of episodes, c_k is the number of collisions during the k-th episode, and γ is a hyperparameter, which is set to 0.95. The following are the components of **SNS**.

- **Route Completion (RC)**: This metric represents the ratio of the route completed by the robot, which can be calculated by $1 - d_f/d_i$, where d_f is the distance from the final position of the robot to the global goal, and d_i is the distance from the initial position of the robot to the global goal.
- **Path efficiency (PE)**: This metric measures the efficiency of the path, which can be calculated by the ratio of $d_i - d_f$ and the total distance traveled by the robot. This score is clipped between 0 and 1.
- **Speed score (SP)**: This metric represents the score considering the average speed (v) of the robot. The maximum score is achieved when v is above $1.5\ m/s$, and the minimum score is achieved when v is below $0.5\ m/s$. The score is normalized between 0 and 1.

SNS is designed to give a high score to the robot, which gets closer to the global goal, while considering safety, efficiency, and speed. Since safety is crucial in evaluating socially-aware navigation algorithms, **SNS** decreases exponentially as the number of collisions increases.

D. Baseline Comparison

We compare several navigation algorithms in BSON. Each method uses the position of the robot, the 2D LIDAR input, and the goal location as an observation. We use environments composed of a map with size $50\ m \times 50\ m$, 24 pedestrians, and no static obstacle. We assign LCFs within $(-2.00, 2.00)$ to pedestrians so that there exist both adversarial and cooperative pedestrians. The following algorithms are used for the experiment: PPO [18], soft actor-critic (SAC) [19], tsallis actor-critic (TAC) [20], and behavior cloning (BC).

As a reward, ω_d, ω_g, and ω_c in (5) are each set to 1.0, 2.0, and 1.0. For PPO, We use the same network architecture used in section 5. For TAC, SAC, and BC, all networks consist of two hidden layers with size (1024, 1024). TAC utilizes a Tsallis entropy calculated using the q-logarithm with $q = 1.2$. To train a BC policy, the selected data among trajectories of the pre-trained PPO agent are used as expert demonstrations. All methods are trained in BSON for one million environmental steps, with a learning rate of 0.001. Each method is evaluated in one hundred different BSON environments.

The experimental result is shown in **Table** III. TAC has shown the best performance compared to different methods in all metrics except for **RC**, where it has shown comparable results. Since TAC can choose an appropriate entropic index, it can synthesize a safer and more efficient policy compared to other baselines. SAC has shown the second-best performance after TAC. It can be inferred that TAC and SAC have better exploration due to entropy regularizations. BC shows the lowest **SNS** compared to different methods since it is trained with expert demonstrations collected by the pre-trained PPO agent. Although PPO has slightly higher **RC** compared to SAC, it has lower performance in all metrics compared to SAC, which results in lower **SNS**. Through this, it can be seen that **SNS** can reflect the overall performance considering not only route completion, but also safety and path efficiency.

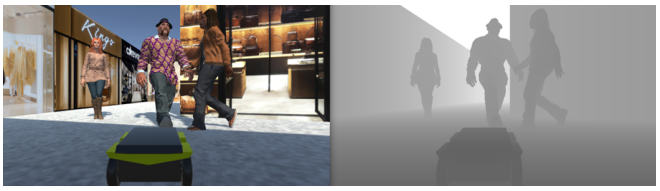

Fig. 3: **BSON**. This figure shows a scene of the BSON. The task of the Jackal robot is to navigate towards the given goal location while avoiding pedestrians.

Method	SNS	RC	Coll	SP	PE
PPO	50.08	**0.995**	12.95	0.603	0.912
SAC	56.57	0.965	12.83	0.657	0.925
TAC	**65.14**	0.993	**8.47**	**0.837**	**0.942**
BC	48.85	0.992	16.32	0.569	0.901

TABLE III: **Navigation methods comparison in BSON.** The table shows the navigation methods comparison results in BSON. The maximum score of **SNS**, **RC**, **SP**, and **PE** are 100, 1, 1, and 1, respectively. **Coll** is the average number of timesteps the robot collides with obstacles in each episode.

VII. Conclusions

In this paper, we have proposed MAC-ID, which can generate diverse pedestrian motions by using the state of each agent and the local coordination factor (LCF) as policy input. We have shown that an increase in LCF leads to pedestrians behaving more cooperatively, since a higher LCF induces an increase in $\textbf{SR}_n$ and a decrease in **CP**. Consequently, we can acquire the desired pedestrian motion property by adjusting the LCF. Also, we have compared MAC-ID to different pedestrian motion models in four different environments. As a result, MAC-ID outperforms baselines in terms of both $\textbf{SR}_n$ and **CP**. For comparing socially-aware navigation methods using MAC-ID, we have proposed a new benchmark called BSON. The pedestrians in BSON move under the MAC-ID policy with different LCFs. We have compared several navigation methods in BSON with a newly proposed metric, **SNS**. Furthermore, future researchers can train and evaluate their algorithms in BSON.

While the proposed method is suitable for simulating social scenarios, there exist several limitations. MAC-ID uses the observation acquired from a limited number of raycasts, which is extremely simple compared to real human perception. Furthermore, additional reward engineering is needed to reduce unnatural behaviors, such as jittering. Regarding BSON, we have focused on pedestrian movements, but real-world human interactions involve a wide range of motions and gestures. In future research, we plan to enhance BSON by incorporating more realistic human behaviors that better capture real-world scenarios.

References

[1] C. I. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *arXiv:2103.05668*, Mar. 2021.

[2] C. Mavrogiannis, P. Alves-Oliveira, W. Thomason, and R. A. Knepper, "Social momentum: Design and evaluation of a framework for socially competent robot navigation," *ACM Transactions on Human-Robot Interaction*, vol. 11, no. 2, pp. 1–37, Jun. 2022.

[3] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, Dec. 2013.

[4] B. Okal and K. O. Arras, "Learning socially normative robot navigation behaviors with Bayesian inverse reinforcement learning," in *Proc. of the IEEE International Conference on Robotics and Automation*, May 2016.

[5] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *Proc. of the IEEE International Conference on Robotics and Automation*, May 2019.

[6] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, "Decentralized structural-RNN for robot crowd navigation with deep reinforcement learning," in *Proc. of the IEEE International Conference on Robotics and Automation*, Oct. 2021.

[7] J. Oh, J. Heo, J. Lee, G. Lee, M. Kang, J. Park, and S. Oh, "Scan: Socially-aware navigation using monte carlo tree search," in *Proc. of the IEEE International Conference on Robotics and Automation*, Jul. 2023.

[8] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, p. 4282, May 1995.

[9] J. v. d. Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Proc. of the International Symposium of Robotics Research*, Aug. 2011.

[10] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, pp. 1–39, Apr. 2023.

[11] L. Torrey, "Crowd simulation via multi-agent reinforcement learning," in *Proc. of the AAAI Artificial Intelligence and Interactive Digital Entertainment*, Jul. 2010.

[12] J. Lee, J. Won, and J. Lee, "Crowd simulation by deep reinforcement learning," in *Proc. of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*, Nov. 2018.

[13] C. S. de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. Torr, M. Sun, and S. Whiteson, "Is independent learning all you need in the Starcraft multi-agent challenge?" *arXiv:2011.09533*, Nov. 2020.

[14] Z. Peng, Q. Li, K. M. Hui, C. Liu, and B. Zhou, "Learning to simulate self-driven particles system with coordinated policy optimization," in *Proc. of the International Conference on Neural Information Processing Systems*, Dec. 2021.

[15] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, and D. Lange, "Unity: A general platform for intelligent agents," *arXiv:1809.02627*, May 2020.

[16] "Jackal UGV - small weatherproof robot - clearpath," Dec. 2020. [Online]. Available: https://clearpathrobotics.com/jackal-small-unmanned-ground-vehicle/

[17] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv:1606.01540*, Jun. 2016.

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, Jul. 2017.

[19] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. of the International Conference on Machine Learning*, Jul. 2018.

[20] K. Lee, S. Kim, S. Lim, S. Choi, M. Hong, J. I. Kim, Y.-L. Park, and S. Oh, "Generalized Tsallis entropy reinforcement learning and its application to soft mobile robots." in *Proc. of the Robotics: Science and Systems*, Jul. 2020.

[21] S. Zhou, D. Chen, W. Cai, L. Luo, M. Y. H. Low, F. Tian, V. S.-H. Tay, D. W. S. Ong, and B. D. Hamilton, "Crowd modeling and simulation technologies," *ACM Transactions on Modeling and Computer Simulation*, vol. 20, no. 4, pp. 1–35, Oct. 2010.

[22] A. Rasouli, "Pedestrian simulation: A review," *arXiv:2102.03289*, Feb. 2021.

[23] N. Bellomo and C. Dogbe, "On the modelling crowd dynamics from scaling to hyperbolic macroscopic models," *Mathematical Models and Methods in Applied Sciences*, vol. 18, pp. 1317–1345, Aug. 2008.

[24] L. Huang, S. Wong, M. Zhang, C.-W. Shu, and W. H. Lam, "Revisiting Hughes' dynamic continuum model for pedestrian flow and the development of an efficient solution algorithm," *Transportation Research Part B: Methodological*, vol. 43, no. 1, pp. 127–141, Jan. 2009.

[25] Y.-Q. Jiang, R.-Y. Guo, F.-B. Tian, and S.-G. Zhou, "Macroscopic modeling of pedestrian flow based on a second-order predictive dynamic model," *Applied Mathematical Modelling*, vol. 40, no. 23-24, pp. 9806–9820, Jun. 2016.

[26] F. S. Hänseler, W. H. Lam, M. Bierlaire, G. Lederrey, and M. Nikolić, "A dynamic network loading model for anisotropic and congested pedestrian flows," *Transportation Research Part B: Methodological*, vol. 95, pp. 149–168, Jan. 2017.

[27] S. P. Hoogendoorn, F. L. van Wageningen-Kessels, W. Daamen, and D. C. Duives, "Continuum modelling of pedestrian flows: From microscopic principles to self-organised macroscopic phenomena," *Physica A: Statistical Mechanics and its Applications*, vol. 416, pp. 684–694, Dec. 2014.

[28] G. G. Løvås, "Modeling and simulation of pedestrian traffic flow," *Transportation Research Part B: Methodological*, vol. 28, no. 6, pp. 429–443, Dec. 1994.

[29] A. Treuille, S. Cooper, and Z. Popović, "Continuum crowds," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 1160–1168, Jul. 2006.

[30] L. Crociani, G. Lämmel, and G. Vizzari, "Multi-scale simulation for crowd management: A case study in an urban scenario," in *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*, May 2016.

[31] A. Tordeux, G. Lämmel, F. S. Hänseler, and B. Steffen, "A mesoscopic model for large-scale simulation of pedestrian dynamics," *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 128–147, Aug. 2018.

[32] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," in *Proc. of the International Conference on Machine Learning*, Jul. 2018.

[33] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 50, pp. 24 972–24 978, Nov. 2019.

[34] B. Toghi, R. Valiente, D. Sadigh, R. Pedarsani, and Y. P. Fallah, "Social coordination and altruism in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 791–24 804, Sep. 2022.

[35] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 2020, pp. 683–700.

[36] R. Korbmacher and A. Tordeux, "Review of pedestrian trajectory prediction methods: Comparing deep learning and knowledge-based approaches," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 125–24 144, Sep. 2022.

[37] M. Campanella, S. P. Hoogendoorn, and W. Daamen, "The nomad model: theory, developments and applications," *Transportation Research Procedia*, vol. 2, pp. 462–467, 2014.

[38] S. Curtis, A. Best, and D. Manocha, "Menge: A modular framework for simulating crowd movement," *Collective Dynamics*, vol. 1, pp. 1–40, Mar. 2016.

[39] N. Tsoi, A. Xiang, P. Yu, S. S. Sohn, G. Schwartz, S. Ramesh, M. Hussein, A. W. Gupta, M. Kapadia, and M. Vázquez, "SEAN 2.0: Formalizing and generating social situations for robot navigation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 047–11 054, Oct. 2022.

[40] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Proc. of the International Conference on Autonomous Agents and Multiagent Systems*, Nov. 2017.

[41] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv:1506.02438*, Jun. 2015.

[42] S. M. LaValle, "Rapidly-exploring random trees: a new tool for path planning," *Technical Report. Computer Science Department, Iowa State University*, Oct. 1998.

[43] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, Jun. 2011.

[44] J. Suh, J. Gong, and S. Oh, "Fast sampling-based cost-aware path planning with nonmyopic extensions using cross entropy," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1313–1326, Dec. 2017.