IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Hydrodynamics Regularization in Reinforcement Learning for Navigating Crowded Scenarios

Pingrui Lai[1], Renjie Pan[1], Jiaqi Yu[1] and Hua Yang[1*], Member, IEEE

***Abstract*—The navigation task in dense crowds is a key research problem in real-world scenarios. It requires an agent to avoid collisions in dynamic environments and reach the agent's destination, ensuring high accuracy and efficiency in its decisions. Existing methods typically treat pedestrians as rigid bodies, detect object bounding boxes, and use rigid body dynamics to guide agent behavior. However, in densely crowded scenarios, this approach may lead to suboptimal path planning solutions, thereby imposing more stringent constraints on the agent's action space. In some real-world navigation scenarios, pedestrians can avoid collisions with minor posture adjustments without having to change their direction. In this letter, we propose Hydrodynamics Regularization to address the challenges posed by the modeling of pedestrians in dense crowd environments. This method treats pedestrians as deformable bodies and leverages fluid dynamics equations to compute socially compliant velocities and postures for the agent in dense crowds. By introducing soft constraints on the action space, it effectively reduces collisions between the agent and surrounding obstacles. We validate the effectiveness of Hydrodynamics Regularization in reinforcement learning navigation problems with partially observable Markov processes. Extensive experiments demonstrate that Hydrodynamics Regularization effectively mitigates collisions caused by the modeling approach, especially in dense crowd environments, allowing the navigation agent to achieve higher success rates, fewer collisions, and shorter completion times. The Hydrodynamics Regularization module is a plug-and-play component that can be seamlessly integrated into any reinforcement learning algorithm, demonstrating excellent generalizability.**

***Index Terms*—Motion and Path Planning, Machine Learning for Robot Control, Reinforcement Learning**

I. Introduction

AGENT navigation [1] has had a profound impact across multiple domains, driving advancements in robotics, autonomous vehicles, virtual reality, and logistics. The development of humanoid robots aims to enable agents to efficiently accomplish tasks in complex crowd environments.

Despite significant progress in navigation methods, navigation in dense crowd scenarios remains a major challenge, characterized by two main features compared to traditional navigation environments. The first feature is **dynamical environment**. In dense crowd scenarios, humans exhibit frequent and dynamic movements, and interactions occur not only between humans but also between humans and robots. To accomplish navigation tasks, robots should engage in close-proximity interactions with humans while simultaneously being influenced by other ongoing interactions [2]. The constraints imposed by these interactions are difficult to predict, making navigation in dense crowd environments highly challenging. The second feature is **pedestrian avoidance**. Ensuring human safety is crucial, and avoiding collisions with pedestrians is a critical aspect of motion planning. Obstacle avoidance is a fundamental component of navigation, however, a major limitation of current avoidance methods is their failure to account for subtle pedestrian posture adjustments, leading to inaccurate collision assessments. As illustrated in Fig. 1, most existing collision detection modules estimate pedestrian bounding boxes and motion spaces based on rigid body mechanics, relying on positional and directional information for collision assessment. However, in real-world scenarios, pedestrians can avoid collisions through minor posture adjustments (such as sidestepping) without significantly altering their intended trajectories. This phenomenon is particularly pronounced in high-density crowds.

The modeling of pedestrian dynamics has progressed from representing individuals as points, to rigid bodies, and as deformable bodies. In high-density crowds, pedestrian flow

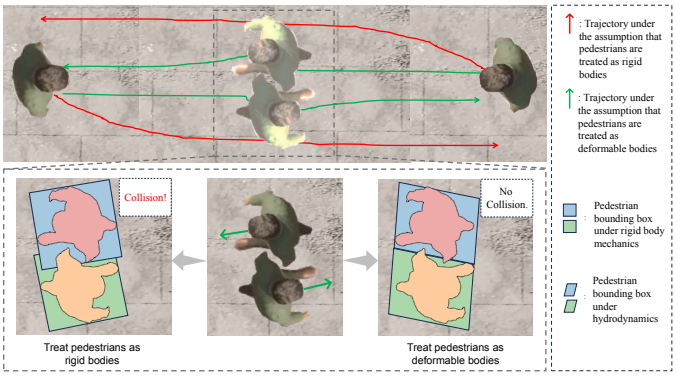

Fig. 1. Different impacts of modeling pedestrians as rigid bodies or deformable bodies. Rigid body mechanics model pedestrians using bounding boxes and determine whether a collision occurs based on the overlap of these boxes. However, this approach has certain limitations. When two pedestrians walk toward each other, they can avoid collisions through slight postural adjustments without significantly altering their walking paths. In such cases, using rigid body mechanics can lead to incorrect collision estimations. This phenomenon is particularly pronounced in dense crowd environments.

Manuscript received: October 11, 2024; Revised April 19, 2025; Accepted May 26, 2025.

This paper was recommended for publication by Editor Jens Kober upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by grants of National Natural Science Foundation of China (NSFC, Grant No. 62171281), Science and Technology Commission of Shanghai Municipality (STCSM, Grant Nos. 20DZ1200203, 2021SHZDZX0102).

[1]The authors are with Faculty of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China (e-mail: laipingrui@sjtu.edu.cn; rjpan21@sjtu.edu.cn; jiaqiyu@sjtu.edu.cn; hyang@sjtu.edu.cn). *Corresponding author.

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

begins to exhibit behaviors analogous to those observed in fluid dynamics [3]. Under such conditions, individual movement patterns are strongly influenced by the surrounding dynamic environment. Pedestrians tend to adapt by making subtle postural adjustments and modifying their trajectories in response to local flow conditions, such as semi-autonomously following the motion of the crowd to avoid collisions. This analogy between dense pedestrian flow and fluid behavior offers a valuable perspective to improve robot navigation in crowded environments. By modeling robots as fluid particles that interact with crowd flow, it becomes possible to more accurately define their action spaces. This perspective facilitates the development of more precise obstacle avoidance methods that are grounded in the motion characteristics of pedestrians.

To address these issues in navigation, we propose a new regularization term for the loss function in Deep Reinforcement Learning (DRL), termed Hydrodynamics Regularization. This method models the humanoid robot navigating through a crowd as a fluid element, predicting the speed and orientation distribution of the agent in densely crowded scenarios using fluid dynamics equations. The predicted values are incorporated into the DRL loss function to achieve soft constraints on the action space, enabling more granular decision-making control. Specifically, we conduct experiments to demonstrate the challenges posed by `Rigid Body Collision` and designed a navigation task for an agent in densely crowded scenarios based on DRL. Hydrodynamics Regularization is a plug-and-play module that can be integrated into any RL algorithm to enhance navigation performance. We explored the impact of different environmental factors, such as road conditions, crowd densities, and the types of algorithms used. Comprehensive comparative studies and ablation experiments validate the feasibility and superiority of our approach. Our contributions can be summarized as follows:

- By modeling pedestrians as deformable bodies, we highlight the limitations of rigid body mechanics in collision avoidance, providing a new paradigm for pedestrian modeling.
- We propose a new DRL loss function regularization term, Hydrodynamics Regularization, which can be integrated into any RL algorithm, improving the accuracy and efficiency of agent navigation in dense crowd environments.
- We design various challenging and realistic dense crowd navigation scenarios and paths in the CARLA [4] simulator, which enables end-to-end training of diverse RL algorithms and leads to a 5% average improvement in navigation success rate.

II. Related Work

A. Navigation with Deep Learning

Many works have applied deep learning to the agent navigation problem and have proven successful in tasks such as end-to-end learning for autonomous driving [5], [6], [7]. Supervised learning is used in [8], [9], and RL is utilized in [10] to learn navigation strategies, but these works do not leverage the agent's current state for prediction. Work in [11] and others provide a framework for navigation using imitation learning, assuming the Social Force Model [12] as the "expert" demonstration, training a social navigation strategy using a generative adversarial imitation learning algorithm. Although assuming the social force model as "expert" demonstration is helpful, they do not consider that there might be more than one strategy for agent navigation. Agents can be trained through a supervised strategy of imitation learning to avoid dynamic obstacles without estimating trajectories or extracting the current state [13].

Research based on DRL [14], [15], [16], [17] intrinsically estimates obstacles by incorporating available obstacle data into their observation space. A multi-agent collision avoidance algorithm using RL is proposed in [18], but their method is limited to specific scenarios. Neural networks are used in [19] to predict the probability of events like collisions, but their reward function design does not ensure that agents reach a safe state. Significant progress has been made by some end-to-end deep learning methods [20], [21].

Currently, some navigation tasks consider the robot's perspective and posture [22], and in many tasks, the camera angle and robot posture do not necessarily align with the direction of movement [23]. However, these works often overlook the changes in the poses of pedestrians, who act as dynamic obstacles in the environment. Avoiding collisions by changing posture is an interactive process. In this letter, we refine the depiction of the phenomenon of avoiding collisions by changing posture by introducing a constraint inspired by principles from fluid mechanics.

B. Pedestrian Modeling

The accuracy of dense crowd scene simulation is crucial for navigation tasks. From the perspective of simulation complexity and realism, existing crowd simulation methods can be re-categorized based on how they treat individuals within the crowd: as the point mass, vectors, rigid bodies, or deformable bodies.

Methods treating individuals as **the point mass** focus on the overall motion and distribution of the crowd, simplifying each individual to a point with mass. This approach is evident in the works of Lashley *et al.* [24], who simulate nonlinear physical phenomena by formulating behavioral criteria based on mass-point dynamics. In methods where individuals are treated as **vectors**, the focus is on the direction and magnitude of movement. Reynolds *et al.* [25] developed a bird-like flock model, a vector-based simulation of collective behavior, defining core rules for directional movement. Considering individuals as **rigid bodies** involves accounting for personal space and collisions, suitable for simulations where these factors are crucial. Helbing *et al.* [12] treated the crowd as a collection of rigid bodies, following rules of social forces and motion mechanics. Subsequent works in this area focus on optimizing exercise rules [26], [27], including adding psychological factors [28], [29], and classifying human behavior based on tasks to provide exercise recommendations for different populations performing various tasks [30], [31].

Methods viewing individuals as **deformable bodies** are the most complex, simulating detailed interactions and movements. This is seen in the work of Van *et al.* [32], who use

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

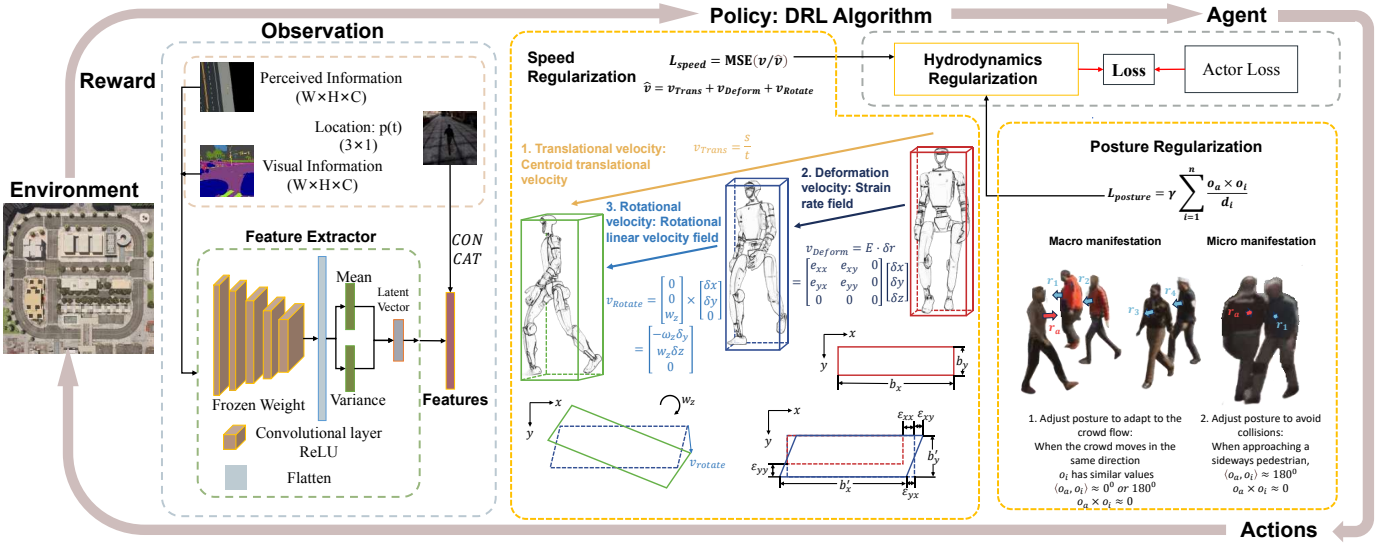

Fig. 2. DRL Navigation System with Hydrodynamics Regularization. The agent's observations include first-person view semantic segmentation images, a local semantic map centered around the agent, and its position. After dimensionality reduction using a pre-trained feature extractor, these observations are utilized for RL. The algorithm receives feedback from the environment at each time step. The loss function, incorporating Hydrodynamics Regularization, includes two penalty terms, L_{speed} and $L_{posture}$, which regulate the agent's speed and posture changes, respectively. These penalties are calculated based on the velocity values predicted by fluid dynamics.

smooth particle hydrodynamics to intuitively control crowd density, considering the dynamic deformation of individuals. Applying the deformable body perspective to navigation tasks is essential for studying agent path and action planning in real-world scenarios [33]. This letter attempts to accurately simulate fine-grained individual behavior and interactions in crowd scenes.

III. Method

A. Problem Formulation

The navigation task in dense crowd scenarios is formulated as a Partially Observable Markov Decision Process (POMDP) in a DRL framework, which is defined by a tuple (S, A, P, R, γ, L).

The agent's state is denoted as $s^t = \{p, v, \omega, o, b\}$, consisting of position p_t, velocity v_t, angular velocity ω_t, orientation o_t and bounding box b_t. We consider b_t as a hexahedron with variability (not always a cube), so the orientation of the agent during movement is not always aligned with the direction of velocity. Therefore, s^t should include the orientation of the agent. The agent's observation of the scene is represented as $g^t \cup m^t$, where g^t denotes the robot's first-person semantic segmentation image, and m^t represents the robot-centric BEV map. Both have dimensions of $W \times H \times C$. Both g^t and m^t are fed into a feature extractor for dimensionality reduction, and then concatenated with p^t to form a new observation feature i^t.

$$i^t = FE_r(g^t) \oplus FE_r(m^t) \oplus p^t \tag{1}$$

where $\oplus$ means concatenation operation, FE_r denotes a Feature Extractor with an output layer of size r. $P : S \times A \to S$ is the state transition function. $\gamma \in (0, 1]$ is the discount factor and L is the maximum episode length. $R : S \times A \to \mathbb{R}$ is the reward function.

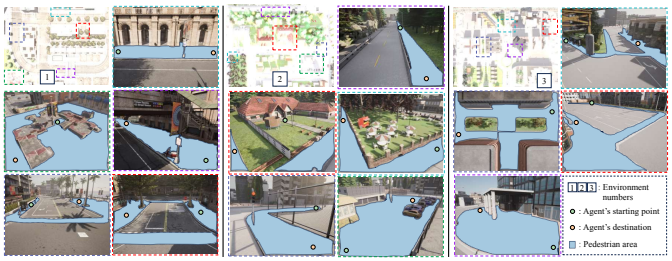

Fig. 3. The navigation environment. Our environment setup comply with the NHTSA typology's scenario specifications [34], the paths in the environment incorporate classic road structures, including SI (Signalized Intersection), USI (Unsignalized Intersection), RAB (Roundabouts), LANE, and EOA (Emergency Obstacle Avoidance).

In each episode, the agent begins at an initial state $s_0 \in S$. At timestep t, the agent selects action $a^t \in A$ according to the navigation policy $\pi(a^t|s^t)$, gaining a corresponding reward r according to the reward function R. The process continues until t exceeds the maximum timestep L, the agent reaches its destination, or the agent collides with pedestrians or obstacles. The reward used for our DRL training is defined as:

$$r = \begin{cases} -2, & \text{if collision with pedestrian} \\ -0.5, & \text{if collision with others} \\ \alpha|v^t - v^*|, & \text{if too fast or too slow} \\ +100, & \text{if success} \end{cases} \tag{2}$$

where $\alpha < 0$ is the weight (We use $\alpha = -0.1$), v^* is the desired velocity of the agent, and the reward function encourages the agent to navigate as quickly as possible while ensuring safety.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

B. Hydrodynamics in Crowd Behavior

When an individual is in a moving crowd, their movement is closely related to the collective motion, analogous to a fluid element in a flow. The theoretical velocity of a robot moving in a pedestrian flow is derived based on hydrodynamics equations. If the current coordinates of the agent are denoted as M, and an adjacent point as M'. As shown in Fig. 2, according to Helmholtz's velocity decomposition theorem, it can be obtained that:

$$\begin{aligned} v_{M'} &= v_{Trans} + v_{Deform} + v_{Rotate} \\ &= v_M + E \cdot \delta r + \omega \times \delta r \end{aligned} \tag{3}$$

where $v_{Trans} = v_M$ represents the translational velocity, $v_{Deform} = E \cdot \delta r$ represents the deformation velocity, and $v_{Rotate} = \omega \times \delta r$ represents the rotational angular velocity, and

$$\delta r = (\delta x, \delta y, \delta z) = \left(\frac{b'_x + b_x}{2}, \frac{b'_y + b_y}{2}, \frac{b'_z + b_z}{2} \right) \tag{4}$$

To compute v_{Deform}, we need to obtain the deformation rate e. the relationship can be expressed as:

$$\begin{aligned} v_{M'} &= v_M + \frac{\partial v_M}{\partial x}\delta x + \frac{\partial v_M}{\partial y}\delta y + \frac{\partial v_M}{\partial z}\delta z \\ &= [u - \omega_z dy + \omega_y dz + e_{xx}dx + e_{xy}dy + e_{xz}dz]\vec{i} \\ &\quad + (...)\vec{j} + (...)\vec{k}. \end{aligned} \tag{5}$$

where $u = \partial b_x/\partial t$, $q = \partial b_y/\partial t$, $e_{xx} = \partial u/\partial x$, $\omega_z = \frac{1}{2}(\partial q/\partial x - \partial u/\partial y)$, $e_{xy} = \frac{1}{2}(\partial u/\partial y + \partial q/\partial x)$, and the omitted parts of the formulas and symbols maintain rotational symmetry. We use the finite difference method to derive the velocity changes of the agent. That means we consider Δt as the smallest time interval. Therefore, we consider the agent's state after Δt as $s^{t+\Delta t} = \{p', v', \omega', o', b'\}$:

$$e_x = \frac{\partial u}{\partial x} dx dt/dx = \frac{\partial u}{\partial x} dt \tag{6}$$

and the linear deformation rates are expressed as follows:

$$e_{xx} = \frac{\partial u}{\partial x} \approx \frac{b'_x - b_x}{\Delta t \frac{b'_x + b_x}{2}} = \frac{2(b'_x - b_x)}{\Delta t (b'_x + b_x)} \tag{7}$$

$$\begin{aligned} e_{xy} = \frac{1}{2}\left(\frac{\partial q}{\partial x} + \frac{\partial u}{\partial y}\right) &\approx \frac{1}{2\Delta t}\left(\frac{b'_x - b_x}{\frac{b'_y + b_y}{2}} + \frac{b'_y - b_y}{\frac{b'_x + b_x}{2}} \right) \\ &= \frac{1}{\Delta t}\left(\frac{b'_x - b_x}{b'_y + b_y} + \frac{b'_y - b_y}{b'_x + b_x} \right) \end{aligned} \tag{8}$$

e_{xy} represents the extent of deformation in the y-direction caused by the movement of the x-boundary (illustrated in Fig. 2, where the bounding box changes from a rectangle to a parallelogram). This results in an additional velocity component. During the agent's navigation process, we provide two simplified conditions based on real-world scenarios: (1) No obstacles apply external force in the robot's z-direction to cause deformation. (2) Only the rotation around the robot's own z-axis is considered.

$$\begin{aligned} E &= \begin{bmatrix} e_{xx} & e_{xy} & e_{xz} \\ e_{yx} & e_{yy} & e_{yz} \\ e_{zx} & e_{zy} & e_{zz} \end{bmatrix} = \begin{bmatrix} e_{xx} & e_{xy} & 0 \\ e_{yx} & e_{yy} & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} \frac{2(b'_x - b_x)}{\Delta t(b'_x + b_x)} & \frac{1}{\Delta t}\left(\frac{b'_x - b_x}{b'_y + b_y} + \frac{b'_y - b_y}{b'_x + b_x} \right) & 0 \\ -\frac{1}{\Delta t}\left(\frac{b'_x - b_x}{b'_y + b_y} + \frac{b'_y - b_y}{b'_x + b_x} \right) & \frac{2(b'_y - b_y)}{\Delta t(b'_y + b_y)} & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{9}$$

According to the constraints of the fluid dynamics, the agent's state transition is given by the following rules:

$$\begin{aligned} v^{t+\Delta t} &= v_{Trans} + v_{Deform} + v_{Rotate} \\ v_{Trans} &= v^t = (v_x, v_y, v_z) \\ v_{Deform} &= \left(\frac{3(b'_x - b_x)(b'_x + b_x) + (b'_y - b_y)(b'_y + b_y)}{2\Delta t(b'_x + b_x)}, \right. \\ &\quad \left. \frac{-(b'_x - b_x)(b'_x + b_x) + (b'_y - b_y)(b'_y + b_y)}{2\Delta t(b'_y + b_y)}, 0 \right) \\ v_{Rotate} &= \frac{1}{2}\left(\omega_z(b_y + b'_y), \omega_z(b_x + b'_x), 0 \right) \end{aligned} \tag{10}$$

C. Reinforcement Learning Regularization

After obtaining the velocity prediction $v^{t+\Delta t}$ for a robot moving in a crowd based on hydrodynamics principles in Section III-B, incorporating the corresponding regularization term into RL can effectively constrain the agent's action space. Since the velocity direction of a humanoid robot does not always align with its orientation (e.g., moving sideways while facing forward), we propose two types of regularization terms: speed regularization and posture regularization. By incorporating hydrodynamics regularization into the RL loss function, the final loss function can be expressed as:

$$Loss = L_{actor} + L_{speed} + L_{posture} \tag{11}$$

where L_{actor} represents the loss function related to the agent part in the original RL algorithm.

1) Speed Regularization: At each time step t, based on Equation 10, we obtain the predicted velocity vector $\hat{v} = v^{t+\Delta t}$ and the actual velocity v of the agent. The loss function for Speed Regularization L_{speed} is obtained by calculating the Mean Squared Error:

$$L_{speed} = MSE(v, \hat{v}) = \frac{1}{3}\|v - \hat{v}\|_2^2 \tag{12}$$

The greater the difference between v and $\hat{v}$, the larger the value of L_{speed}. Therefore, the speed regularization encourages the agent to make speed decisions that are closer to the hydrodynamics rules.

2) Posture Regularization: We transform the states of pedestrians perceived by the agent into an undirected heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that contains 2 types of nodes and 1 type of edge. The nodes $\mathcal{V}$ include one agent node and n pedestrian nodes. Edges $\mathcal{E}$ represent the distance d from the agent to pedestrians, the feature of each node is represented by the agent or pedestrian's orientation $o = (o_x, o_y, o_z)$. The

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

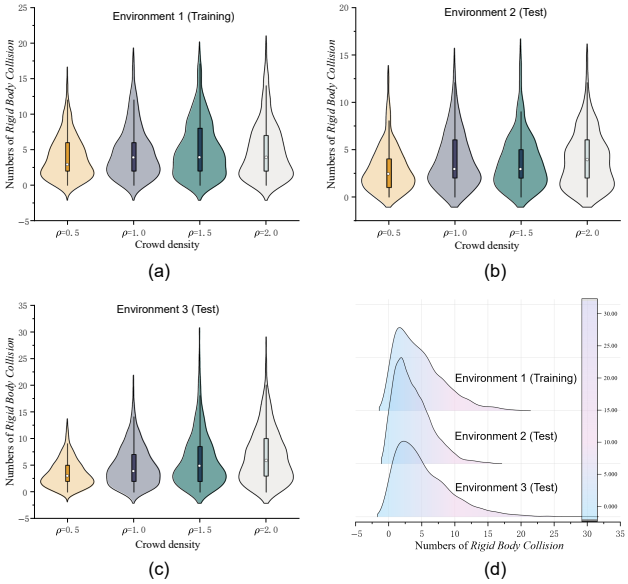

Fig. 4. Occurrences of `Rigid Body Collision` under different crowd densities. The first three figures present box plots and violin plots of `Rigid Body Collision` occurrences under varying crowd densities across three different environments. The fourth figure illustrates the overall distribution of `Rigid Body Collision` in these three environments.

purpose of imposing this constraint is to enable the agent to better adapt to the posture changes of surrounding pedestrians. The agent needs to consider the common orientation of pedestrians in dense crowds. For the graph network centered on the agent, we use eigenvector centrality to design the loss function.

$$L_{posture} = \gamma \sum_{i=1}^{n} \frac{o_a \times o_i}{d_i} \tag{13}$$

where o_a represents the orientation of the agent, and o_i represents the orientation of other pedestrians, γ represents the weight.

The reason for using the cross product of o_i and o_a is that when o_i and o_a are in the same or opposite directions, $o_a \times o_i$ is zero. In real-world scenarios, the agent's posture typically needs to make significant changes in two situations: (1) As shown in Fig. 2, when the crowd is moving towards a direction, o_i tends to align in the same direction. In this case, we expect the agent to adapt to the crowd flow, making o_a and o_i align, resulting in $L_{posture}$ being close to zero. (2) When the agent encounters a very close pedestrian, after the pedestrian sidesteps, the agent needs to adjust its posture so that o_a and o_i are aligned or opposite to avoid collision, resulting in $L_{posture}$ being negligible. Therefore, the Posture Regularization allows the agent to perform collision avoidance tasks with more precise motion planning.

IV. Experiments

A. Experimental Settings

Our experiments leverage the CARLA autonomous driving simulation platform. This setup provides high-precision maps and detailed, realistic rendering capabilities, essential for accurately simulating pedestrian movements in diverse environments. We utilize Gym [45] as our primary platform for implementing and testing RL algorithms. This environment, in conjunction with algorithms from Stable Baselines [46], facilitates the rigorous testing and validation of our models. All simulations were conducted on a high-performance computing setup featuring an NVIDIA GeForce RTX 3090 GPU, ensuring the efficient processing of complex simulations.

B. Collision Differences Induced by Pedestrian Modeling Approaches

Under the same physical constraints, modeling pedestrians as rigid bodies versus deformable bodies leads to differences in collision rates. Therefore, we first design experiments to quantify the increase in the number of collisions caused by using the rigid body model under identical initial conditions, and to determine the proportion of these additional collisions relative to the total number of collisions. We define the types of collisions that occur exclusively when pedestrians are modeled as rigid bodies, rather than deformable bodies, as `Rigid Body Collision`.

We design experiments in three near-realistic simulation scenarios, with partial scenario maps displayed in Fig. 3. The walkable area for pedestrians in each scenario is denoted as S_{ped}, and $n_{pedestrian} = \rho S_{ped}$ pedestrians are randomly generated on the map. The experiment is divided into two groups. In the first group, we allow all pedestrians to move freely for a period of time t_{test}, close to the average time of agent movement in navigation, and record the total number of collisions n_{free}. In the second group, we fix the pedestrians' orientation to always align with their velocity direction and let them move for the same time t_{test}, recording the total number of collisions n_{rigid}. The physical models of pedestrians, average speed of movement, and other variables are kept consistent across both groups of experiments, and each experiment is repeated n_{repeat} times to reduce random error. When n_{repeat} is sufficiently large (We use $n_{\text{repeat}} = 1000$), $\bar{n}_{free}$ and $\bar{n}_{rigid}$ represent the average values of n_{free} and n_{rigid} respectively, and we calculate $n_{rbc} = \bar{n}_{rigid} - \bar{n}_{free}$, considering n_{rbc} as the number of `Rigid Body Collision`.

Fig. 4 presents the quantity and distribution of n_{rbc} under different environments and varying crowd densities. Experimental results show that, in the same environment, the number of `Rigid Body Collision` increases with rising crowd density. However, their proportion relative to the total number of collisions remains stable (within the range of 5% to 10%). The distribution pattern of n_{rbc} is consistent across both training and test environments.

These findings suggest that in dense crowd scenarios, the use of rigid body modeling for pedestrians introduces a non-negligible impact. To effectively navigate such environments, agents should possess more precise posture adjustment capabilities.

C. Navigation Task with Regularization

The experiments are conducted in three large-scale scenarios, where we select 5 classic paths that encompass a variety of challenging situations, as shown in Fig. 3. In each area, we

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I
EXPERIMENTAL RESULTS ON THE TEST SET FOR AGENT NAVIGATION IN DENSE CROWD ENVIRONMENTS. THE VALUES IN ( ) REPRESENT CHANGES COMPARED TO THE ORIGINAL RL ALGORITHM AFTER APPLYING HYDRODYNAMICS REGULARIZATION.

Methods	Environment 2			Environment 3		
	$SR\uparrow$	$AR\uparrow$	$TS\downarrow$	$SR\uparrow$	$AR\uparrow$	$TS\downarrow$
DQN [35]	42% (**+7%**)	68.70 (**+10.12**)	40.97 (**-3.75**)	55% (**+1%**)	63.95 (**+6.52**)	29.72 (**-1.09**)
PPO [36]	36% (**+1%**)	54.59 (**+1.53**)	40.64 (**-3.79**)	56% (**+5%**)	69.47 (**+9.59**)	30.39 (**-0.70**)
SAC [30]	48% (**+7%**)	77.63 (**+14.74**)	43.76 (**-7.66**)	57% (**+7%**)	69.49 (**+3.57**)	29.86 (**-1.56**)
A2C [37]	44% (**+1%**)	61.72 (**+2.02**)	40.30 (**-2.11**)	55% (**+9%**)	66.45 (**+10.89**)	29.42 (**-6.93**)
DDPG [38]	59% (**+8%**)	85.50 (**+18.12**)	36.90 (**-9.74**)	52% (**+5%**)	63.35 (**+5.58**)	30.69 (**-1.57**)
TD3 [39]	58% (**+9%**)	92.49 (**+15.32**)	38.95 (**-3.76**)	54% (**+9%**)	65.93 (**+9.31**)	32.33 (**-5.19**)
QR-DQN [40]	52% (**+9%**)	81.37 (**+13.81**)	39.61 (**-2.33**)	55% (**+8%**)	68.46 (**+9.63**)	30.40 (**-0.20**)
TRPO [41]	33% (**+9%**)	46.05 (**+18.13**)	44.50 (**-2.21**)	62% (**+10%**)	76.58 (**+19.96**)	28.54 (**-8.98**)
TQC [42]	34% (**+4%**)	61.46 (**+10.52**)	45.58 (**-2.63**)	53% (**+1%**)	64.92 (-0.48)	30.53 (**-0.94**)
RecurrentPPO [43]	45% (**+5%**)	65.23 (**+8.07**)	39.98 (+0.21)	65% (**+4%**)	80.05 (**+3.56**)	29.68 (**-0.60**)
MaskablePPO [44]	60% (**+5%**)	92.94 (**+3.59**)	39.04 (**-2.79**)	56% (**+7%**)	69.47 (**+9.17**)	30.39 (**-1.32**)

TABLE II
ABLATION STUDY OF MASKABLE PPO ON THE THREE VELOCITY CONSTRAINTS IN EQ. 10.

Speed Regularization			Posture Regularization	Training Set ($\rho=$)					Test Set ($\rho=$)				
v_{Trans}	v_{Deform}	v_{Rotate}		0	0.5	1.0	1.5	2.0	0	0.5	1.0	1.5	2.0
×	✓	✓	✓	87%	75%	76%	65%	60%	58%	55%	47%	43%	38%
✓	×	✓	✓	93%	88%	83%	78%	69%	71%	63%	61%	56%	43%
✓	✓	×	✓	90%	81%	82%	71%	68%	70%	60%	58%	49%	42%
✓	✓	✓	×	93%	87%	82%	70%	62%	71%	62%	60%	47%	40%
✓	✓	✓	✓	93%	88%	83%	78%	70%	71%	63%	61%	58%	44%

generate crowds with a density of ρ (unit: per square meters), using four configurations: $\rho = 0.5, 1.0, 1.5, 2.0$. Considering that the positions and movement patterns of pedestrians differ in each experiment even on the same path, we repeat each specific scenario 100 times to fully assess the agent's obstacle avoidance capability. The training process was conducted in Environment 1, with the training set comprising $5 \times 4 \times 100 = 2000$ scenarios.

To evaluate the adaptability of the algorithm in different environments, we test the agent in Environments 2 and 3, with $\rho = 1.5$, repeating the experiments 100 times. The test set includes $2 \times 5 \times 100 = 1000$ scenarios. We choose three classic evaluation metrics: Success Rate (SR), Average Return (AR), and average navigation Time Step (TS). SR is the ratio of episodes where the agent successfully reaches the destination. Given the high density of the crowd in these scenarios, the main reason for the agent failing to reach the destination is usually due to collisions (a rare event being the agent wandering in the scene and exceeding the maximum allowed navigation time). Therefore, another classic metric, the Collision Rate (CR), can be derived from SR, calculated as (CR = 1 - SR) . In this task, SR can be used to assess the agent's ability to avoid collisions. AR and TS are widely used to evaluate the efficiency of the agent's navigation.

1) Quantitative Analysis: Table I shows the results on the test set. With the inclusion of Hydrodynamics Regularization, most algorithms saw an increase in their SR metrics by $5\% - 10\%$, which is roughly consistent with the frequency of `Rigid Body Collision`. The experimental results demonstrate that Hydrodynamics Regularization achieves a lower collision rate by avoiding most instances of `Rigid Body Collision`. Additionally, with the incorporation of Hydrodynamics Regularization, there is a noticeable improvement in the AR metric, and a reduction in the TS metric, indicating that the agents are able to complete tasks more efficiently.

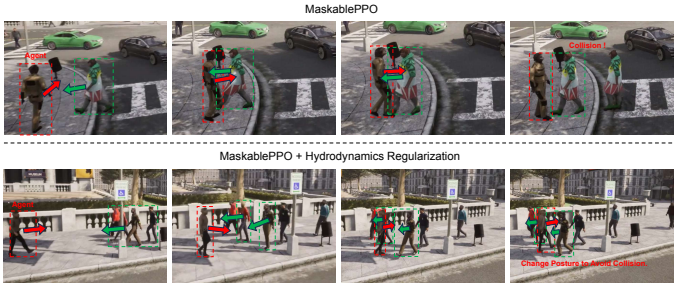

Fig. 5. A qualitative navigation visualization result of the MaskablePPO algorithm after incorporating the Hydrodynamics Regularization module.

2) Qualitative Analysis: Fig. 5 presents a visualization of the navigation performance of the MaskablePPO algorithm after incorporating Hydrodynamics Regularization. Before regularization is applied, the agent primarily avoids obstacles by adjusting its path. However, in scenarios with insufficient space (e.g., navigating through a zebra crossing), the agent struggles to modify its current strategy, leading to a collision. After incorporating regularization, the agent is able to perceive the movement patterns of oncoming crowds and make subtle posture adjustments to continue moving forward. This demonstrates the effectiveness of Hydrodynamics Regularization in optimizing the action space.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

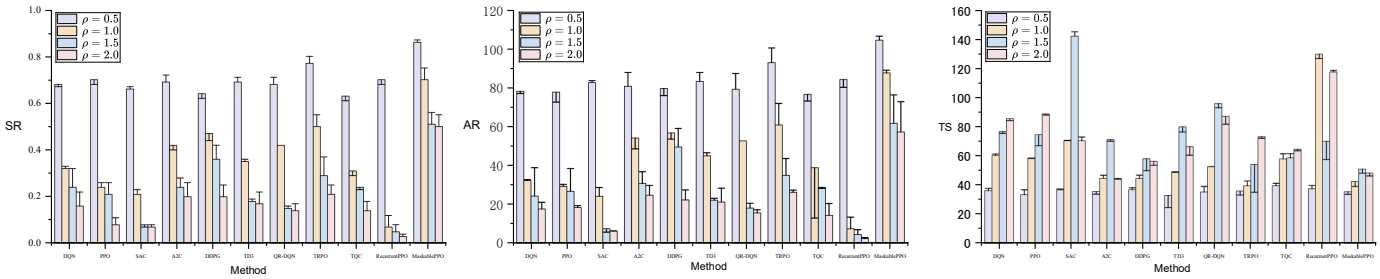

Fig. 6. Experimental results for agent navigation in dense crowd environments with varying crowd densities. The error bars represent the changes in the corresponding metrics after incorporating Hydrodynamics Regularization.

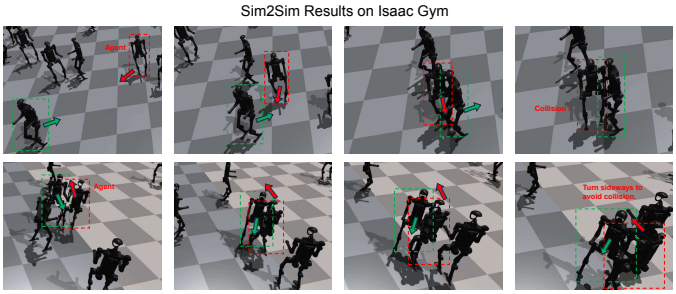

Fig. 7. Qualitative results of Sim2Sim transfer. The figure shows the visualization results on Isaac Gym.

D. Ablation Study

We investigate the effectiveness of speed and posture constraints in Hydrodynamics Regularization. The speed regularization constraints consist of 3 components, defined in Equation 10 as v_{Trans}, v_{Deform}, and v_{Rotate}. In our experiments, we sequentially removed each of these velocity components and incorporated the modified hydrodynamics regularization into a RL algorithm, as presented in Table II. The results indicate that both velocity and posture regularization have a positive impact on obstacle avoidance. While v_{Trans} serves as the primary factor influencing the agent's movement, the inclusion of v_{Deform} and v_{Rotate} enables more precise motion planning.

E. Discussion on Crowd Density

Fig. 6 presents the experimental results under different crowd densities ρ in the training set. When ρ is low, the impact of adding Hydrodynamics Regularization on the SR metric of the algorithms is within 3%. This further demonstrates that in sparse crowd scenarios, the environment provides sufficient space for the agent to adjust its path planning. The primary method for the agent to avoid collisions remains path adjustment, without requiring significant posture adjustments. However, as the crowd density increases, the constraints on the action space imposed by the environment on the agent gradually increase. Posture adjustments become increasingly critical for collision avoidance. At this point, methods incorporating Hydrodynamics Regularization show stable performance improvements, further confirming the effectiveness of our approach in dense crowd scenarios.

F. Sim2Sim and Sim2Real Transfer

We completed the Sim2Sim transfer between simulation platforms. Fig. 7 illustrate the qualitative results. The training of a humanoid robot was conducted on the Isaac Gym simulation platform, where considerations ranged from velocity and posture variations to the coordination of limb movements. To account for these factors, the reward function was modified as follows:

$$r_{humanoid} = r + w_1 r_c + w_2 r_s + w_3 r_a + w_4 r_n + w_5 r_h \quad (14)$$

where r represents the original reward function in Equation 2, r_c encourages correct foot placement based on gait phase, r_s penalizes excessive foot height during the swing phase, r_a is a constant reward for remaining upright, r_n penalizes feet in contact with low velocity, r_h penalizes unnatural hip positions to enforce biomechanical constraints, and w_i are weight coefficients tuned experimentally. We successfully trained the Unitree H1 humanoid robot model in Isaac Gym, and the policy may be directly deployed in real-world environments. Additional experimental details are provided in the supplementary materials.

V. Conclusion

In this letter, we highlight the incorrect judgment in `Rigid Body Collision` detection when using rigid body mechanics to model pedestrians in navigation. We conducted experiments in various environments, demonstrating that these errors become more pronounced as crowd density increases. Consequently, we propose the Hydrodynamics Regularization based on hydrodynamics, treating pedestrians as deformable bodies to further constrain the action space of the agents. Through agent navigation experiments based on the DRL framework, we demonstrated that the Hydrodynamics Regularization achieves efficient navigation decision-making across different environments, paths, and crowd densities. The agents successfully completed tasks with higher success rates and efficiency. Ablation experiments analyzed the performance improvements brought by different types of velocity constraint terms. In the future, we will focus on diverse tasks in embodied intelligence navigation.

References

[1] D. R. Montello, *Navigation.* Cambridge University Press, 2005.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

[2] S. Liu, H. Xia, F. C. Pouria, K. Hong, N. Chakraborty, and K. Driggs-Campbell, "Height: Heterogeneous interaction graph transformer for robot navigation in crowded and constrained environments," *arXiv preprint arXiv:2411.12150*, 2024.

[3] L. Henderson, "The statistics of crowd fluids," *nature*, vol. 229, no. 5284, pp. 381–383, 1971.

[4] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[6] M. Pfeiffer, S. Shukla, M. Turchetta, C. Cadena, A. Krause, R. Siegwart, and J. Nieto, "Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4423–4430, 2018.

[7] Z. Wang, X. Xiao, A. J. Nettekoven, K. Umasankar, A. Singh, S. Bommakanti, U. Topcu, and P. Stone, "From agile ground to aerial navigation: Learning from learned hallucination," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 148–153.

[8] D. Gandhi, L. Pinto, and A. Gupta, "Learning to fly by crashing," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3948–3955.

[9] A. Loquercio, A. I. Maqueda, C. R. Del-Blanco, and D. Scaramuzza, "Dronet: Learning to fly by driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1088–1095, 2018.

[10] F. Sadeghi and S. Levine, "Cad2rl: Real single-image flight without a single real image," *arXiv preprint arXiv:1611.04201*, 2016.

[11] L. Tai, J. Zhang, M. Liu, and W. Burgard, "Socially compliant navigation through raw depth inputs with generative adversarial imitation learning," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 1111–1117.

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

[13] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[14] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6015–6022.

[15] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 285–292.

[16] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3052–3059.

[17] M. Hamandi, M. D'Arcy, and P. Fazli, "Deepmotion: Learning to navigate like humans," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2019, pp. 1–7.

[18] M. Everett, Y. F. Chen, and J. P. How, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," *Ieee Access*, vol. 9, pp. 10 357–10 377, 2021.

[19] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An autonomous self-supervised learning-based navigation system," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1312–1319, 2021.

[20] A. Hu, G. Corrado, N. Griffiths, Z. Murez, C. Gurau, H. Yeo, A. Kendall, R. Cipolla, and J. Shotton, "Model-based imitation learning for urban driving," *Advances in Neural Information Processing Systems*, vol. 35, pp. 20 703–20 716, 2022.

[21] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Advances in neural information processing systems*, vol. 1, 1988.

[22] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[23] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3674–3683.

[24] K. S. Lashley *et al.*, *The problem of serial order in behavior*. Bobbs-Merrill Oxford, 1951, vol. 21.

[25] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 1987, pp. 25–34.

[26] C. W. Reynolds *et al.*, "Steering behaviors for autonomous characters," in *Game developers conference*, vol. 1999. Citeseer, 1999, pp. 763–782.

[27] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical review letters*, vol. 75, no. 6, p. 1226, 1995.

[28] S. J. Guy, S. Kim, M. C. Lin, and D. Manocha, "Simulating heterogeneous crowd behaviors using personality trait theory," in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation*, 2011, pp. 43–52.

[29] A. Sieben, J. Schumann, and A. Seyfried, "Collective phenomena in crowds—where pedestrian dynamics need social psychology," *PLoS one*, vol. 12, no. 6, p. e0177328, 2017.

[30] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. Pmlr, 2018, pp. 1861–1870.

[31] A. Trivedi and S. Rao, "Agent-based modeling of emergency evacuations considering human panic behavior," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 1, pp. 277–288, 2018.

[32] W. van Toll, T. Chatagnon, C. Braga, B. Solenthaler, and J. Pettré, "Sph crowds: Agent-based crowd simulation up to extreme densities using fluid dynamics," *Computers & Graphics*, vol. 98, pp. 306–321, 2021.

[33] D. Dugas, K. Cai, O. Andersson, N. Lawrance, R. Siegwart, and J. J. Chung, "Flowbot: Flow-based modeling for robot navigation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 8799–8805.

[34] W. G. Najm, J. D. Smith, M. Yanagisawa, A. John, *et al.*, "Pre-crash scenario typology for crash avoidance research," United States. Department of Transportation. National Highway Traffic Safety ..., Tech. Rep., 2007.

[35] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

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

[37] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PmLR, 2016, pp. 1928–1937.

[38] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[39] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*. PMLR, 2018, pp. 1587–1596.

[40] W. Dabney, M. Rowland, M. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[41] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[42] A. Kuznetsov, P. Shvechikov, A. Grishin, and D. Vetrov, "Controlling overestimation bias with truncated mixture of continuous distributional quantile critics," in *International conference on machine learning*. PMLR, 2020, pp. 5556–5566.

[43] M. Pleines, M. Pallasch, F. Zimmer, and M. Preuss, "Generalization, mayhems and limits in recurrent proximal policy optimization," *arXiv preprint arXiv:2205.11104*, 2022.

[44] S. Huang and S. Ontañón, "A closer look at invalid action masking in policy gradient algorithms," *arXiv preprint arXiv:2006.14171*, 2020.

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

[46] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of machine learning research*, vol. 22, no. 268, pp. 1–8, 2021.