Reinforcement Learning with Time-to-Collision Features and Attention-Based Pedestrian Filtering for Crowd Navigation

Mohammadali Rahmati, Mohammadhossein Malmir, and Alois Knoll

***Abstract*—Autonomous navigation in crowded environments is a critical challenge for service robots. Traditional reactive methods suffer from the "freezing robot problem" in dense crowds, while existing deep reinforcement learning approaches often assume holonomic kinematics or use discrete actions, leading to jerky motion. This paper presents a deep reinforcement learning framework for socially-aware navigation of non-holonomic service robots, combining four innovations: (i) the dual-selection attention module with proximal policy optimization for continuous velocity control, (ii) hybrid time-to-collision computation for anticipatory collision avoidance, (iii) high-fidelity training with a realistic 3D robot model in NVIDIA IsaacLab using GPU-accelerated parallel simulation, and (iv) domain randomization over pedestrian count for robust generalization across crowd densities. Comprehensive evaluation demonstrates that our method resolves the freezing robot problem (0% timeout at all tested densities) while achieving the fastest reported navigation time (9.12s) and 4–12× faster training than comparable methods. The policy was validated through sim-to-sim transfer (IsaacLab to Gazebo) and preliminary testing on a physical robot platform.**

I. INTRODUCTION

Service robots operating in airports, hospitals, and shopping malls must navigate safely among humans. This domain, social robot navigation (SRN), demands real-time decision-making in unpredictable, dense crowds under strict kinematic constraints.

Classical methods such as A*, dynamic window approach (DWA), and optimal reciprocal collision avoidance (ORCA) [1] fail in this setting for a fundamental reason: they treat dynamic agents as moving static obstacles rather than interacting agents with predictable trajectories. ORCA additionally assumes full reciprocity from pedestrians, an assumption that fails in practice. The result is the **"freezing robot problem"** [2]: in dense crowds, feasible velocity sets become empty, and the robot stops indefinitely.

Deep reinforcement learning (DRL) overcomes this by enabling the learning of interaction logic, allowing the robot to proactively anticipate crowd dynamics rather than react to them. However, current DRL methods have three key gaps that our method is designed to fill:

- **Hesitation problem:** Attention-based methods such as SARL [3] assign nonzero weights to *all* pedestrians, including irrelevant ones far from the robot's path. This causes oscillation and hesitation in tight crowds.
- **Discrete actions:** MODSRL [4], the state-of-the-art attention method (99.8% success), uses a discrete action space. Discrete control produces jerky, unnatural motion unsuitable for service robots operating in shared spaces with humans.
- **No density generalization:** Most methods train at a fixed pedestrian count. MODSRL's success degrades to 87.4% at $N{=}15$ when trained only at $N{=}5$.

Authors are with the Department of Computer Engineering, School of Computation, Information and Technology, Technical University of Munich, Munich, Germany (email: mohammad.rahmati@tum.de; hossein.malmir@tum.de; k@tum.de). This work was conducted in collaboration with NEURA Robotics.

TABLE I
METHOD COMPARISON AGAINST REQUIREMENTS [12]

Method	Non-Holonomic	Continuous	Anti-Freeze	Success
CADRL	×	×	×	93%
LSTM-RL	×	×	×	98%
SARL	×	×	×	100%*
SafeCrowdNav	×	×	✓	98.6%
MODSRL	✓	×	✓	99.8%
SESN	✓	✓	✓†	96%
Ours	✓	✓	✓	92.77%

*With hesitation. †Requires curriculum learning.

Our method bridges these gaps by integrating the decisive statistical pedestrian filtering of the dual-selection attention module (DSAM) with smooth and continuous velocity control of proximal policy optimization (PPO), a novel hybrid time-to-collision (TTC) feature for anticipatory avoidance, and domain randomization over pedestrian count (0–10 per episode) that yields a single policy robust across all crowd densities without curriculum learning.

II. RELATED WORK

DRL navigation has evolved from pairwise value functions (CADRL [5], 78% success at $N{=}5$) to sequential long short-term memory (LSTM) encoding (LSTM-RL [6]) and, more recently, to crowd-attention models. SARL [3] introduced self-attention over pedestrian states but retained all agents in its context vector, leading to the hesitation problem. Graph-based methods (RGL [7], G2RL [8]) model human–human interactions via a graph convolutional network (GCN) or a graph attention network (GAT), but incur $O(N^2)$ complexity. SafeCrowdNav [9] adds TTC-based safety scores, achieving 98.6%. MODSRL [4] introduced DSAM's dual-threshold statistical filtering, achieving 99.8% at $N{=}5$, but with discrete actions.

On the non-holonomic side, SESN [10] requires curriculum learning (52% without it); Flögel et al. [11] use LSTM-PPO without attention. Critically, *all* prior methods train on simplified 2D circle-robot models using sequential CPU-based simulation (20–72h), and none report 0% timeout across densities. Table I summarizes the gaps our method addresses.

III. METHODOLOGY

A. Problem Formulation

Navigation is formulated as a Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ with state space $\mathcal{S}$, action space $\mathcal{A}$,

state-transition probability matrix $\mathcal{P}$, reward function $\mathcal{R}$, and discount factor γ, learning a policy π^* that maximizes

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \gamma^t R\left(s_t, a_t\right)\right] \ , \tag{1}$$

with episodes terminating at goal achievement, collision, boundary violation, or timeout. The **circle crossing scenario** places the robot at 3.0 m from the center; pedestrians spawn on a 4.5 m-radius perimeter and traverse to diametrically opposite points, creating natural crossing trajectories. Pedestrian counts are uniformly sampled from 0 to 10 per episode.

B. State Representation

The robot state $\boldsymbol{s}_t^{\text{robot}} \in \mathbb{R}^7$ captures kinematic and goal-relative information

$$\boldsymbol{s}_t^{\text{robot}} = [d_g,\ \hat{g}_x,\ \hat{g}_y,\ v_{\text{pref}},\ v_{\text{linear}},\ \omega,\ r]^{\mathsf{T}} \ , \tag{2}$$

where d_g is the goal distance, $(\hat{g}_x, \hat{g}_y)$ is the normalized goal direction in body frame, v_{pref}=1.0 m/s is the preferred forward velocity, and r=0.35 m is the robot radius, while v_{linear} and ω are the current forward and angular velocities.

Each pedestrian i is encoded as an 8-dimensional vector

$$\boldsymbol{h}_t^{(i)} = \left[p_x^{(i)},\ p_y^{(i)},\ v_x^{(i)},\ v_y^{(i)},\ r^{(i)},\ d^{(i)},\ r_{\text{comb}}^{(i)},\ \text{TTC}^{(i)}\right]^{\mathsf{T}} \ , \tag{3}$$

including relative position $(p_x^{(i)}, p_y^{(i)}) \in \mathbb{R}^2$ in body frame, absolute velocity $(v_x^{(i)}, v_y^{(i)}) \in \mathbb{R}^2$ in body frame, pedestrian radius $r^{(i)}$=0.3 m, relative distance $d^{(i)}$, combined collision boundary $r_{\text{comb}}^{(i)} = r^{(i)}+r$, and *normalized TTC*$^{(i)}$. Gaussian noise ($\sigma$=0.1 m) is added to positions during training to improve sim-to-real robustness. Total observation is $\boldsymbol{o}_t \in \mathbb{R}^{127}$ ($7 + 15 \times 8$, zero-padded to $N_{\max}$=15).

C. Hybrid Time-to-Collision

TTC [13] is embedded directly into each pedestrian's state vector, enabling the policy to reason about imminent collisions rather than reacting only to proximity. Two complementary formulations are combined: (i) **closing-speed TTC** capturing head-on approaches

$$\text{TTC}_{\text{closing}}^{(i)} = \frac{d^{(i)} - r_{\text{comb}}^{(i)}}{v_{\text{closing}}^{(i)}} = \frac{d^{(i)} - r_{\text{comb}}^{(i)}}{-\boldsymbol{v}_{\text{rel}}^{(i)} \cdot \frac{\boldsymbol{p}^{(i)}}{d^{(i)}}} \ , \tag{4}$$

where $\boldsymbol{v}_{\text{rel}}^{(i)} = \boldsymbol{v}^{(i)} - \boldsymbol{v}_{\text{robot}}$, and (ii) **geometric TTC** capturing crossing and oblique paths by solving trajectory intersection

$$\left\|\boldsymbol{p}^{(i)} + t^{(i)}\, \boldsymbol{v}_{\text{rel}}^{(i)}\right\|^2 = {r_{\text{comb}}^{(i)}}^2 \ \Rightarrow \ a{t^{(i)}}^2 + bt^{(i)} + c = 0 \ , \tag{5}$$

where $\text{TTC}_{\text{geometric}}^{(i)}$ is the smallest positive root $t^{(i)}$ if the discriminant is positive. The hybrid TTC takes the minimum (most conservative) estimate

$$\text{TTC}_{\text{raw}}^{(i)} = \min\left(\text{TTC}_{\text{closing}}^{(i)},\ \text{TTC}_{\text{geometric}}^{(i)}\right) \ , \tag{6}$$

normalized to $[0, 1]$ over a 5 s horizon, where 0 indicates imminent collision.

D. Action Space and Forward Constraint

The continuous action space matches differential-drive constraints

$$\boldsymbol{a}_t = [v_{\text{linear}},\ \omega]^{\mathsf{T}} \in [0,\ 1.0] \text{ m/s} \times [-0.8,\ 0.8] \text{ rad/s} \ , \tag{7}$$

where forward-only motion ($v_{\text{linear}} \geq 0$) prevents "lazy" backward solutions, forces active turning to face obstacles, and matches real robot sensor geometry.

E. Multi-Objective Reward Function

The reward balances six competing objectives

$$R(s_t, a_t) = \boldsymbol{w}\,[r_{\text{prog}},\ r_{\text{coll}},\ r_{\text{TTC}},\ r_{\text{time}},\ r_{\text{boundary}},\ r_{\text{goal}},\ r_{\text{smooth}},\ r_{\text{fwd}},\ r_{\text{head}}]^{\mathsf{T}} \ , \tag{8}$$

where $\boldsymbol{w} = [30.0, 7.0, 3.0, 1.0, 2.0, 1.0, 0.05, 0.3, 0.5]$ are the weights, and $r_{\text{prog}} = d_{g_{t-1}} - d_{g_t}$ denotes the progress goal-approach signal. Proximity to collision is penalized with a smooth quadratic gradient

$$r_{\text{coll}} = \begin{cases} 0 & d > d_{\text{safe}} \\ -(1-\hat{d})^2 & r_{\text{comb}} < d \leq d_{\text{safe}} \\ -(1-\hat{d})^2 - 1.0 & d \leq r_{\text{comb}} \end{cases} \ , \tag{9}$$

where $d_{\text{safe}} = 2.0$ m, distance to pedestrians $d = \min_i(d^{(i)})$, and normalized proximity $\hat{d} = (d - r_{\text{comb}})/(d_{\text{safe}} - r_{\text{comb}})$. TTC anticipation penalizes imminent threats before proximity alone would trigger a response with $r_{\text{TTC}} = -(0.5 - \text{TTC}_{\min})$ when $\text{TTC}_{\min} < 0.5$, else 0. Time efficiency reward $r_{\text{time}} = 1.0 - 0.1 \cdot (t/T_{\text{limit}})$ is granted only upon goal achievement, incentivizing faster completion while keeping the time penalty small to prioritize reaching the goal over minimizing time. Boundary penalty provides a graduated wall-proximity penalty with r_{boundary} linearly increasing from 0 at $d_{\text{soft}} = 1.5$ m to -0.5 at $d_{\text{hard}} = 0.7$ m, and clamped to -2.0 below d_{hard}, preventing the agent from exploiting room edges. Goal success bonus awards $r_{\text{goal}} = 10.0$ when the robot reaches within 0.5 m of the goal, providing a strong sparse signal for task completion. Smoothness penalty penalizes excessive angular velocity with $r_{\text{smooth}} = \varepsilon \cdot (\omega_{\text{th}} - |\omega_t|)$ when $|\omega_t| > \omega_{\text{th}}$, else 0, where $\varepsilon = 0.01$ and $\omega_{\text{th}} = 0.48$ rad/s. Forward velocity bonus directly counteracts freezing by rewarding forward motion with $r_{\text{fwd}} = \min(v_{\text{linear}}, 0.5)$. Heading reward encourages facing the goal while moving with $r_{\text{head}} = \hat{g}_x \cdot (v_{\text{linear}}/v_{\text{pref}})$. Terminal rewards are $+50$ for reaching the goal and -50 for a collision, wall contact, or a timeout.

F. Dual-Selection Attention Module (DSAM)

DSAM [4] resolves the hesitation problem by statistically filtering pedestrians before aggregating crowd context. Starting from a 127D observation, it produces a fixed 50D crowd vector $\boldsymbol{K}$ through several steps. The first step is to encode robot-pedestrian interactions with a shared embedding network M_e that produces joint robot–pedestrian embeddings $\boldsymbol{e}_i = M_e([\boldsymbol{s}^{robot}; \boldsymbol{h}^{(i)}]) \in \mathbb{R}^{50}$, and an interaction network M_h to further refine these to $\boldsymbol{h}_i' = M_h(\boldsymbol{e}_i) \in \mathbb{R}^{50}$. The next step is to select and weight important pedestrians through two attention

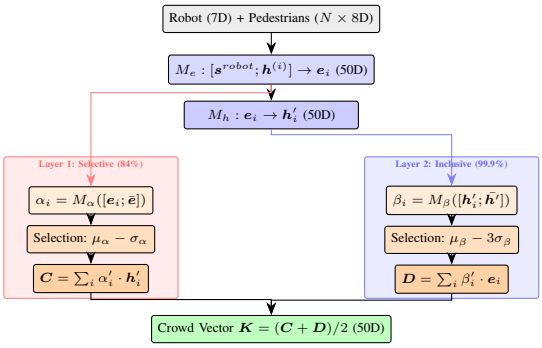

Fig. 1. DSAM architecture: M_e, M_h extract embeddings. Layer 1 scores by $\boldsymbol{e}_i$ ($\mu{-}\sigma$ filter), aggregates $\boldsymbol{h}'_i$. Layer 2 scores by $\boldsymbol{h}'_i$ ($\mu{-}3\sigma$ filter), aggregates $\boldsymbol{e}_i$.

layers. The first layer computes scores $\alpha_i = M_\alpha([\boldsymbol{e}_i; \bar{\boldsymbol{e}}])$ by comparing each pedestrian to the crowd mean $\bar{\boldsymbol{e}}$, then masking agents below $\mu_\alpha - \sigma_\alpha$ via selective filtering, retaining $\sim$84%, and aggregating over the retained agents

$$\boldsymbol{C} = \sum_i \text{softmax}\left(\alpha_i \cdot \text{mask}_\alpha^{(i)}\right) \cdot \boldsymbol{h}'_i \ . \tag{10}$$

The second layer uses a looser inclusive threshold $\mu_\beta - 3\sigma_\beta$ masking the population $\beta_i = M_\beta([\boldsymbol{h}'_i; \bar{\boldsymbol{h}'}])$, retaining $\sim$99.9% of the global crowd context

$$\boldsymbol{D} = \sum_i \text{softmax}\left(\beta_i \cdot \text{mask}_\beta^{(i)}\right) \cdot \boldsymbol{e}_i \ . \tag{11}$$

As illustrated in Figure 1, the key design in DSAM is **cross-aggregation**: layer 1 scores by $\boldsymbol{e}_i$ but aggregates $\boldsymbol{h}'_i$, while layer 2 scores by $\boldsymbol{h}'_i$ but aggregates $\boldsymbol{e}_i$. This ensures that decisive, threat-focused context and broader crowd awareness are both captured in the fused crowd vector

$$\boldsymbol{K} = (\boldsymbol{C} + \boldsymbol{D})/2 \ , \qquad \boldsymbol{x} = \left[\boldsymbol{s}^{robot}; \boldsymbol{K}\right] \in \mathbb{R}^{57} \ , \tag{12}$$

where x is the final representation as input to the RL agent.

G. PPO with Actor-Critic Architecture

The policy is trained with PPO [14] using a shared DSAM encoder. The actor outputs Gaussian parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \pi_\theta(\boldsymbol{x})$ with $\sigma \in [0.1, 1.5]$ for stochastic exploration; the critic estimates $V_\phi(\boldsymbol{x})$. The clipped PPO objective (ϵ=0.2) prevents destabilizing policy updates

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(\rho_t \hat{A}_t,\ \text{clip}(\rho_t, 1{\pm}\epsilon)\hat{A}_t\right)\right] \ , \tag{13}$$

where advantages $\hat{A}_t$ use generalized advantage estimation (GAE) with λ=0.95. Learning rate is low ($5{\times}10^{-5}$) with 8 mini-batches and 8 epochs to stabilize attention patterns; the entropy coefficient is 0.001 to let DSAM provide natural exploration through varying pedestrian importance weights; and the discount is γ=0.995 to extend the horizon to $\approx$3.3 s for anticipatory reasoning.

IV. IMPLEMENTATION

A. System Architecture and Key Design Decisions

The system is built in NVIDIA IsaacLab [15] using a manager-based design, having a top-level RL environment for *MiPA* [16] navigation with the tasks of goal management, base observations, and TTC augmentations, integrated with a crowd manager representing pedestrian dynamics with a social force model (SFM), and controlled with a differential drive mechanism to compute wheel velocities. The simulated environment scene covers the robot, pedestrians, walls, and ground. Three critical implementation choices enable stable large-scale training:

- **Ghost pedestrians:** Physical collision response is disabled to prevent GPU solver instability with 10k environments. Collision avoidance is learned purely via reward signals and distance-based termination.
- **Robot stability:** High angular damping of 5.0 compensates for the MiPA robot's high center of mass (0.69 m), preventing tipping during rapid maneuvers. Linear damping of 1.0 ensures smooth velocity changes.
- **GPU buffer sizing:** Increased contact counts (2^{21}), patch counts (2^{18}), and heap capacity (64 MB) prevent simulation crashes during parallel training.

B. Pedestrian Simulation via Social Force Model

Each pedestrian i follows an SFM [17] governed by

$$m_i \frac{d\boldsymbol{v}_i}{dt} = \boldsymbol{F}_{\text{goal}}^{(i)} + \sum_{j \neq i} \boldsymbol{F}_{\text{ped}}^{(ij)} + \boldsymbol{F}_{\text{robot}}^{(i)} + \boldsymbol{F}_{\text{wall}}^{(i)} \ , \tag{14}$$

where $\boldsymbol{F}_{\text{goal}}^{(i)}$ is the attraction force toward the goal, $\boldsymbol{F}_{\text{ped}}^{(ij)}$ is the repulsion from other pedestrians, $\boldsymbol{F}_{\text{robot}}^{(i)}$ is the repulsion from the robot, and $\boldsymbol{F}_{\text{wall}}^{(i)}$ is the repulsion from walls. Behavioral features increase realism over prior work: (i) each pedestrian has a variable preferred speed $v_{\text{pref}} \in [0.6, 1.2]$ m/s; (ii) with probability p=0.0005/step, pedestrians may stop for a random duration of 1–3 s; (iii) asymmetric robot-awareness where pedestrians react to the robot only within a 200° frontal arc, but react to other pedestrians from all directions. These changes create more challenging training conditions than the cooperative ORCA pedestrians used by all compared methods.

C. Domain Randomization and Training Configuration

Robust density generalization is achieved with an immediate domain randomization on: (i) pedestrian count sampled 0–10 per episode, (ii) robot spawned on a random angle and orientation with 3.0 m from the center, and (iii) goal placed 3–7 m from the robot with an angle randomized within ±90° of the center. At each randomized episode, non-realized pedestrians are added to the observation vector via zero-padding, since DSAM naturally ignores them via attention thresholding. Table II summarizes the training hyperparameters.

TABLE II
TRAINING CONFIGURATION

Parameter	Value	Notes
Parallel environments	10,000	NVIDIA L40S GPU
Physics / Control Hz	120 / 60	Decimation = 2
Episode length	20 s	1200 max steps
Steps per env	32	Rollout length
Transitions/iteration	320,000	$10k \times 32$
Learning rate	5×10^{-5}	Reduced for DSAM
Discount γ	0.995	$\approx$3.3 s horizon
GAE λ	0.95	Advantage estimation
Entropy coefficient	0.001	Reduced for DSAM
Max iterations	3,000	$\approx$6 hours

D. ROS2 Deployment

The trained policy is wrapped in the `mipa_rl_planner` ROS2 package. The controller subscribes to pedestrian data, robot pose, odometry data, and the goal pose; transforms all states to the robot body frame using the ROS transform library TF2; assembles the 127D observation using hybrid TTC and zero-padding; runs the DSAM encoder and actor; and publishes the commanded robot velocity at 60 Hz. Total inference latency is less than 10 ms, and attention weights are published for real-time visualization of pedestrian prioritization.

V. EXPERIMENTAL RESULTS

A. Training Performance and Domain Randomization

Our model was trained for 3,000 iterations across 10,000 parallel environments on an NVIDIA L40S. The best checkpoint (iteration 2505) achieved a 92.45% success rate, 5.44% pedestrian collisions, 1.86% wall collisions, and 0.32% timeouts. Compared to a fixed 7-pedestrian training, this randomized model with 0–10 pedestrians/episode achieved a $+4.4\%$ success rate, a -2.2% collision rate, and a -1.8% timeout. The result shows that a single policy handles all crowd densities without curriculum staging.

B. Evaluation Across Crowd Densities

Table III reports evaluation results across 10,000+ episodes per configuration at certain pedestrian counts not seen as fixed values during training. Three findings are notable. First, 0% timeout at every density directly validates DSAM's resolution to the freezing problem, where the robot always finds a path forward. Second, the success rate decreases linearly with crowd density (92.77% $\rightarrow$ 78.00%), demonstrating predictable scaling behavior. Third, navigation time is quite stable ($\approx$9.1–9.3 s) across densities, confirming that the robot actively replans to take longer routes when needed, rather than freezing or failing to make progress.

C. Comparison with State-of-the-Art

Table IV compares the methods on the standard circle crossing benchmark. Our method achieves the fastest navigation time (9.12 s) and is the only method reporting 0% timeout. Its lower raw success rate compared to MODSRL (92.77% vs. 99.8%) is explained by the more challenging evaluation: non-holonomic kinematics limit maneuverability, and SFM pedestrians with random stopping and asymmetric awareness are significantly harder than the cooperative ORCA pedestrians used by all compared methods.

TABLE III
NAVIGATION PERFORMANCE ACROSS CROWD DENSITIES

Ped.	Success	Ped. Col.	Wall Col.	Timeout	Time (s)
5	92.77%	4.51%	2.80%	0%	9.12
7	88.91%	9.27%	1.83%	0%	9.11
10	78.00%	20.91%	1.09%	0%	9.28

TABLE IV
COMPARISON WITH STATE-OF-THE-ART METHODS

Method	Ped.	Success	Collision	Time	Timeout
CADRL [5]	5	93.0%	4.0%	11.2s	–
LSTM-RL [6]	5	98.0%	2.0%	11.3s	–
SARL [3]	5	100%	0%	10.5s	2.0%
SafeCrowdNav [9]	5	98.6%	1.4%	10.0s	–
MODSRL [4]	5	99.8%	0.2%	11.1s	–
Ours†	5	92.77%	4.51%	9.12s	0%
Ours†	7	88.91%	9.27%	9.11s	0%
Ours†	10	78.00%	20.91%	9.28s	0%

†Harder conditions: non-holonomic kinematics; SFM pedestrians with random stopping, spawn exclusion (2.0 m), and asymmetric robot-awareness (200° arc).

D. Computational Efficiency and Sim-to-Real Transfer

Our method trains in $\approx$6 h, versus 20–72 h for comparable methods (4–12$\times$ faster), and infers in $<$10 ms, versus 400 ms for EmoBipedNav [18], enabling $>$100 Hz control. The policy transfers successfully from IsaacLab to Gazebo with no fine-tuning. Attention weight visualization confirms the learned behavior: pedestrians in the forward path receive high α_i/β_i weights that decrease as they pass. Preliminary physical robot testing on MiPA with 50% velocity scaling demonstrates stable 60 Hz inference and smooth trajectories in indoor corridors.

VI. CONCLUSION

This paper presents a complete DRL framework for socially-aware navigation of non-holonomic service robots, combining DSAM's decisive crowd filtering, PPO's continuous velocity control, hybrid TTC for anticipatory avoidance, GPU-parallel training in IsaacLab, and domain randomization for density generalization. The framework achieves 0% timeout at all tested crowd densities, along with the fastest navigation time and 4–12$\times$ faster training than comparable methods. Successful sim-to-sim transfer and preliminary physical validation on the MiPA robot confirm the readiness of our approach for real deployment. Future work will address real pedestrian evaluation, dense crowd extension with hierarchical attention, static obstacle integration, and safety-constrained optimization with control barrier functions.

REFERENCES

[1] J. Van Den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 1928–1935.

[2] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 797–803.

[3] C. Chen *et al.*, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6015–6022.

[4] C.-L. Cheng, C.-C. Hsu, S. Saeedvand, and J.-H. Jo, "Multi-objective crowd-aware robot navigation system using deep reinforcement learning," *Applied Soft Computing*, vol. 151, p. 111154, 2024.

[5] Y. F. Chen *et al.*, "Socially aware motion planning with deep reinforcement learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1343–1350.

[6] 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.

[7] C. Chen *et al.*, "Relational graph learning for crowd navigation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 007–10 013.

[8] S. Liu *et al.*, "Intention aware robot crowd navigation with attention-based interaction graph," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 015–12 021.

[9] R. Xu *et al.*, "Safecrowdnav: Safety evaluation of robot navigation in crowded environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 7811–7818.

[10] Y. Tian *et al.*, "Sesn: Socially-aware efficient safe navigation via preference-based learning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[11] D. Flögel *et al.*, "Socially integrated navigation: A social force-based deep reinforcement learning approach," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 11 892–11 898.

[12] T. Van Der Meer, A. Garulli, A. Giannitrapani, and R. Quartullo, "A comparative study of human motion models in reinforcement learning algorithms for social robot navigation," *ACM Transactions on Human-Robot Interaction*, vol. 15, no. 1, pp. 1–33, Aug. 2025.

[13] I. Karamouzas, B. Skinner, and S. J. Guy, "Universal power law governing pedestrian interactions," *Physical Review Letters*, vol. 113, no. 23, p. 238701, 2014.

[14] J. Schulman *et al.*, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[15] M. Mittal *et al.*, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[16] NEURA Robotics, "Mipa – my intelligent personal assistant," https://neura-robotics.com/products/mipa/, 2024, accessed: 2026.

[17] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, 1995.

[18] J. Zhu, Z. Wang, and Y. Xiong, "Emobipednav: Emotion-aware bipedal robot navigation in crowded environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.