Crowd Navigation for Mobile Robots with Focused Risk Perception

Hafiq Anas [*1] **Owais Ahmed Malik** [2] **Wee Hong Ong** [*1]

Abstract

Current state-of-the-art crowd navigation approaches are mainly deep reinforcement learning (DRL)-based but often struggle to generalize to unseen crowd behaviors and to scale efficiently with crowd density. We propose a DRL method that incorporates risk perception directly into the observation space: collision probability is used to identify the K most hazardous obstacles, whose relative positions and velocities are exposed to the policy, and local waypoints are added to the reward to densify the learning signal. Trained once in the Gazebo simulator with a non-cooperative randomized crowd, our model is evaluated on four crowd-behavior scenarios against a 2D-laser DRL baseline, a human-aware social planner, and two classical planners. Our approach achieves higher success rates and social safety across all settings, generalizes to unseen crowd behaviors without fine-tuning, and is further validated in real-world tests.

1. Introduction

Autonomous mobile robots are increasingly deployed in human living spaces, requiring them to traverse shared space with humans moving at diverse speeds — a problem known as crowd or social navigation. Classical planners such as Artificial Potential Fields (Arambula Cosío & Padilla Castañeda, 2004) and Dynamic Window Approaches (Fox et al., 2002) struggle in dense crowds, often causing the Freezing Robot Problem. They rely on hand-tuned cost functions that are difficult to adapt under stochastic human motion. Model-based socially-aware methods such as the Social Forces Window (SFW) planner (Pérez-Higueras et al., 2023) extend DWA (Fox et al., 2002) with Social Forces Model (Helbing & Molnár, 1995) costs, but over-saturation of repulsive forces in dense scenes and fixed social-cost weights similarly limit adaptability and trigger freezing behavior.

Deep Reinforcement Learning (DRL) (Zhu & Zhang, 2021) circumvents hand-tuning by learning navigation policies directly. Mapless DRL approaches using 2D laser scans have shown strong empirical results for crowd navigation (Jin et al., 2020; Tai et al., 2017; Zhelo et al., 2018; Long et al., 2018; Alyassi et al., 2025). CrowdMove (Long et al., 2018) relied on curriculum diversity for generalization, while Jin et al. (Jin et al., 2020) introduced human-awareness via ego- and social-safety penalty counts in the reward. We take Jin et al. as our representative DRL baseline, but note a key limitation: their risk signal enters only through the reward at training time, so the policy must implicitly infer crowd risk from raw laser-scan observations — demanding extensive training to generalize across scenarios.

In this paper, we propose a DRL-based approach using 2D laser scans that incorporates risk perception of the moving crowd directly into the *observation space*. We use Collision Probability to identify the top-K most hazardous obstacles and include their relative positions and velocities in the observation, giving the policy explicit access to crowd risk at both training and deployment. Our Collision Probability jointly weighs time-to-collision and distance-to-obstacle to capture temporal and spatial risk, and the top-K filtering avoids the search-space collapse that affects SFW's aggregated forces. By studying the limits of K, we further show that the system scales to high crowd density. We evaluate against Jin et al. (Jin et al., 2020), SFW (Pérez-Higueras et al., 2023), and two classical planners (Fox et al., 2002; Rösmann et al., 2015) in simulation, and validate in real-world tests.

2. Approach

Our proposed method builds upon existing DRL-based methods, with the addition of perceived risk in the observation states and prioritizing the K most dangerous obstacles within the robot's field of view (FOV). Figure 1 shows the system overview; the components are described below.

[*]Equal contribution [1]Robotics and Intelligent Systems Lab (Robolab), School of Digital Science, Universiti Brunei Darussalam, Jalan Tungku Link, Brunei [2] Faculty of Engineering & Computing, Atlantic Technological University, Letterkenny, Ireland. Correspondence to: Hafiq Anas <hafiq.anas@gmail.com>, Owais Ahmed Malik <owais.malik@atu.ie>, Wee Hong Ong <weehong.ong@ubd.edu.bn>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

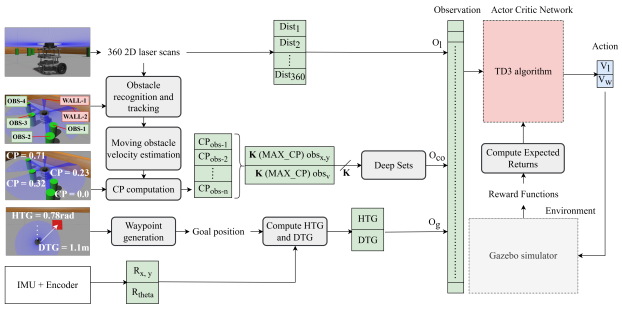

Figure 1. Deep Reinforcement Learning system structure.

2.1. Observation space

The observation $o = [o_l, o_g, o_{co}]$ is a 1D vector composed of: (i) o_l, 360 values from the 2D laser scan, capturing the local dynamic environment; (ii) o_g, the relative distance-to-goal (DTG) and heading-to-goal (HTG) for global navigation; and (iii) o_{co}, the relative positions ($obs_{x,y}$) and velocities (obs_{v_x,v_y}) of the K most dangerous moving obstacles (critical obstacles), giving $K \times 4$ values. Obstacle tracking, velocity estimation, and Collision Probability (CP) computation use privileged obstacle information from the simulator. For the real-world setup, we use YOLOv3 (Redmon & Farhadi, 2018) to detect human positions in the RGB image by extracting the registered point clouds. The tracking, velocity estimation, and CP computation procedures remain the same as those used in the simulation.

We define CP as a weighted sum of two components: the probability of collision based on time-to-collision (P_{c-ttc}) and on distance-to-obstacle (P_{c-dto}). Combining dto and ttc lets the policy distinguish, for example, an obstacle moving slowly near the robot (substantial risk despite low closing speed) from one moving fast toward the robot from far away (lower immediate risk). The two are balanced as in Equation 1.

$$CP = \alpha \cdot P_{c-ttc} + (1-\alpha) \cdot P_{c-dto} \tag{1}$$

where $\alpha \in [0, 1]$ is the parameter that decides the weight of collision probabilities P_{c-dto} and P_{c-ttc}. We performed an ablation over the CP weighting $\alpha \in \{0.0, 0.5, 1.0\}$: using $P_{c\text{-}ttc}$ alone (α=1.0) was too reactive in frontal scenarios, while $P_{c\text{-}dto}$ alone (α=0.0) was overly conservative, and the equal-weight setting (α=0.5) performed best across scenarios. We therefore set $\alpha = 0.5$.

The calculation of collision probabilities uses the Collision Cone (CC) concept in (Sun et al., 2019; Fiorini & Shiller, 1998). P_{c-ttc} is computed based on Time-to-Collision (TTC) as defined in Equation 2. P_{c-dto} is computed based on relative Distance-to-Obstacle (DTO) and defined in Equation 3.

$$P_{c-ttc} = \begin{cases} \min(1, \frac{ts}{t}), & \text{if } V_r' \in CC_{ro} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where t is the time-to-collision (TTC) when the relative velocity V_r' between the robot and the obstacle lies within the Collision Cone CC_{ro}. $V_r' = V_r - V_o$ is the resultant velocity between the robot velocity V_r and obstacle velocity V_o. $t = Dist_o / V_r'$ is the time to collision. CC_{ro} is the collision cone area between the robot and the obstacle. Finally, ts corresponds to the timestep value in seconds for computing the collision probabilities.

$$P_{c-dto} = \begin{cases} \frac{l_{max} - Dist_o}{l_{max} - l_{min}}, & \text{if } Dist_o < l_{max} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where l_{max} and l_{min} are the maximum and minimum range of the laser scan respectively. $Dist_o$ is the distance from the robot to the obstacle of interest.

CP is computed for each obstacle in the list of tracked obstacles, and the position ($obs_{x,y}$) and velocity ($obs_{vx,vy} = V_o$) of the K obstacles with the highest CP are included in the observation space as o_{co}. These K obstacles are seen as most probable to be in collision with the robot.

Due to the unordered list of CP obstacles when $K > 1$, Deep Sets (Zaheer et al., 2017) with mean pooling was used to handle the permutation variance and capture the average context of the K obstacles.

2.2. Action space

An action is defined as $a = [V_l, V_w]$, sampled from a stochastic policy π given observation o: $a \sim \pi(a \mid o)$, where V_l is the linear velocity and V_w is the angular velocity. V_l and V_w were set to ranges of $[0, 0.65]$ ms^{-1} and $[-3.14, 3.14]$ rad·s^{-1} respectively.

2.3. Reward functions

The reward function consists of the following terms:

$$R = R_{step} + R_{dtg} + R_{htg} + R_{goal} + R_{col} + R_{wp} \tag{4}$$

$R_{step} = -2$ is a negative reward given to the robot at every step and serves to discourage oscillation around the goal location without reaching it.

$$R_{dtg} = \begin{cases} +1, & \text{if } \mathrm{d}(r,g)_t < \mathrm{d}(r,g)_{t-1} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

R_{dtg} is a positive reward given whenever the distance from the robot to the target goal location $\mathrm{d}(r, g)$ decreases between consecutive timesteps.

$$R_{htg} = \begin{cases} +1, & \text{if } \theta(r,g)_t < \theta(r,g)_{t-1} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Similarly, R_{htg} is a positive reward given whenever the relative heading $\theta(r, g)$ decreases.

$R_{goal} = +200$ is a large positive reward given to the robot when it reaches the target goal. If a collision occurs, a penalty $R_{col} = -200$ is given. $R_{wp} = +200$ is a large positive reward given when the robot reaches a target local waypoint. The next waypoint is generated every 20 timesteps at the intersection of the circle boundary from the robot center and a straight line towards the target goal position. The radius of the circle region is the maximum laser scan distance of 1.0 m. If the goal is within the boundary, it becomes the waypoint. The inclusion of R_{wp} addresses the problem of sparse goal reward R_{goal} and improves policy convergence as well as the quality of global path planning.

2.4. Deep Reinforcement Learning

We use the Twin Delayed Deep Policy Gradient (TD3) (Fujimoto et al., 2018) algorithm with the default parameters to learn the navigation policy. The robot was trained in the Gazebo simulator using a TurtleBot2 with Kobuki platform equipped with an RPLIDAR A2M8 360-degree 2D laser scanner. The resolution of the laser scanner is 360 with a minimum and maximum range set to 0.2 m and 1.0 m respectively. Training was done once in a 5 m × 5 m space with walls and random moving obstacles at random speeds up to 0.25 m/s in random directions. The moving obstacles were non-cooperative — they ignore the robot's presence and can collide with the robot. The model was trained for 3000 episodes with the stopping criteria of collision with an obstacle or reaching the goal.

3. Evaluation

We evaluated our robot in different crowd behavior environments similar to (Jin et al., 2020): crossing, towards, ahead, and random. For each crowd behavior, the model was tested with a crowd density of twelve moving obstacles, and we computed the average of each metric over 30 runs.

To quantify performance, we used the same evaluation metrics from Jin et al. (Jin et al., 2020): success rate (%), arrival time (s), ego score (0–100), and social score (0–100). Let k be the number of ego-safety violations and N be the total steps to reach the goal; then Ego Score $= (1 - k/N) \times 100$, where a violation is determined when an obstacle comes within the ego-safety radius of the robot used in (Jin et al., 2020). For the social-safety score, we use the same aggregation of Social Score $= (1 - m/N) \times 100$ where m is the number of social-safety violations, but use a collision-probability criterion of $P_{c-ttc} > 0.4$ to register if the robot is on a collision trajectory with an obstacle. This differs from the definition used in (Jin et al., 2020); we report the social score only among methods evaluated under our criterion and do not compare it against the value reported in (Jin et al., 2020). We set this threshold to 0.4 as the point at which collision risk warrants an evasive response; the threshold can be tuned for a more conservative safety margin.

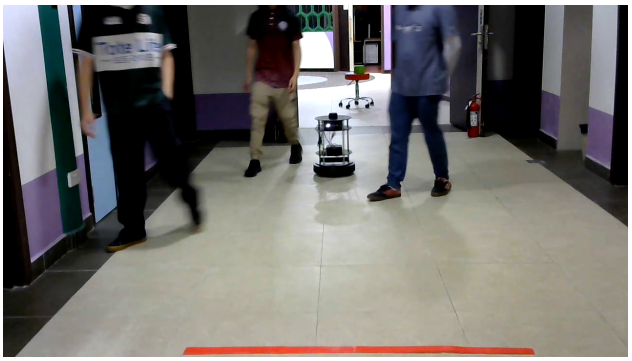

Figure 2. Actual test demonstration footage of the robot navigating within an environment containing three moving humans.

In simulation tests, our proposed method was compared against the *reported* results of Jin et al.'s Ego and Social Policy methods in (Jin et al., 2020), which we do not rerun as their implementation is not publicly available; this comparison covers success rate, arrival time, and ego score under their definitions. We do not compare social scores against (Jin et al., 2020), as our social score violation criteria are not directly comparable, and we cannot recompute either method under the other's definition. To provide condition-matched baselines, additional results were generated by running three navigation approaches — Dynamic Window Approach (DWA) (Fox et al., 2002), Timed Elastic Band (TEB) (Rösmann et al., 2015), and Social Forces Window (SFW) (Pérez-Higueras et al., 2023) — on the simulated robot.

We qualitatively assess sim-to-real transfer with humans as moving obstacles. Figure 2 shows a representative Ahead setup, where the robot navigates to a goal 7 m away among three humans. The robot reproduced the navigation behaviors observed in simulation, though mechanical constraints limited its real-world velocity.

4. Results and Discussion

4.1. Crowd Navigation

Table 1 compares our method against Jin et al. (Jin et al., 2020), SFW (Pérez-Higueras et al., 2023), DWA (Fox et al., 2002), and TEB (Rösmann et al., 2015). The navigation

Table 1. Simulation results across the four crowd-behavior scenarios at a density of 12 moving obstacles. Metrics: SR = success rate (%, ↑), AT = arrival time (s, ↓), ES = ego score (↑), SS = social score (↑). Results for the Ego and Social Policy of (Jin et al., 2020) are reported from their paper at the same crowd density; success rate, arrival time, and ego score follow their definitions. †Social scores for (Jin et al., 2020) use their zone-intrusion definition and are not directly comparable to the collision-probability criterion used for the other methods; they are excluded from the social-score comparison. DWA, TEB, SFW, and our method are run under identical conditions. Best value per metric and scenario in **bold** (social score among comparable methods only). In our method, K was set to 8

	Crossing-12				Towards-12				Ahead-12				Random-12			
Method	SR	AT	ES	SS	SR	AT	ES	SS	SR	AT	ES	SS	SR	AT	ES	SS
Ego Policy (Jin et al., 2020)	60.0	**8.0**	94.0	55.0†	60.0	**12.0**	98.0	60.0†	90.0	**6.0**	**100.0**	32.0†	80.0	**8.0**	**100.0**	70.0†
Social Policy (Jin et al., 2020)	80.0	14.0	**98.0**	74.0†	70.0	**12.0**	**100.0**	70.0†	**100.0**	14.0	**100.0**	85.0†	70.0	12.0	**100.0**	80.0†
DWA (Fox et al., 2002)	30.0	43.9	66.1	83.7	30.0	23.3	61.0	83.2	73.3	23.8	58.6	88.8	40.0	21.7	80.5	94.1
TEB (Rösmann et al., 2015)	53.3	33.0	71.6	86.9	63.3	22.9	71.9	90.9	70.0	19.0	53.8	**98.7**	43.3	23.3	76.7	92.5
SFW (Pérez-Higueras et al., 2023)	56.7	37.6	66.1	87.0	43.3	25.1	81.9	89.1	90.0	22.0	75.4	95.4	60.0	25.9	81.6	92.2
Ours	**83.3**	17.1	75.5	**94.8**	**93.3**	17.1	71.2	**93.8**	**100.0**	17.9	59.5	95.9	**83.3**	17.5	73.8	**95.5**

policy was learned with $K = 8$ and tested with $K = 8$.

Jin et al.'s (Jin et al., 2020) success rates ranged from 60% to 100%, with only 10 of 20 tests reaching 100%. Our arrival times were longer than those reported in (Jin et al., 2020), but their robot moved at roughly twice our maximum speed (1.5 m/s vs. 0.65 m/s). Our robot took approximately 17 s across all environments. Ego- and social-safety violations measure navigation risk rather than collisions; despite occasionally low ego scores, our method maintained the highest success rates among all evaluated methods and higher social scores than condition-matched baselines (DWA, TEB, SFW), indicating strong social compliance.

Among the condition-matched methods (DWA, TEB, SFW, and ours), our method achieved the highest social score in three of the four scenarios (Crossing-12, Towards-12, and Random-12) and remained competitive in Ahead-12 (95.9 vs. TEB's 98.7). Our method also achieved the highest success rate in every scenario. The pattern is consistent: Jin et al.'s Ego Policy (Jin et al., 2020) achieved faster arrival times but at the cost of social-safety violations, while DWA, TEB, and SFW achieved high social scores through conservative behavior that produced low success rates and longer arrival times. Towards-12 and Ahead-12 — the frontal and congestion-heavy scenarios — yielded our largest margins, demonstrating that explicit risk perception is most valuable when crowd geometry penalizes reactive policies. Random-12 confirms robustness to unstructured crowd behavior.

Classical planners and the human-aware planner were consistently prone to the Freezing Robot Problem and reacted slowly to incoming crowds. Although Jin et al.'s methods benefited from higher maximum velocity, our comparable or higher success rate indicate the gain comes from improved crowd interaction rather than speed advantages.

We note that the social score is inversely proportional to total episode steps N and affected by the time spent in free space; this applies equally to all evaluated methods; a denominator where steps with an obstacle within detection range is a refinement we leave to future work. Training with only the sparse goal reward led to poor convergence; adding the waypoint reward densified the signal, improving both convergence speed and path efficiency. In real-world tests, the robot exhibited similar crowd navigation behavior as in simulation, though mechanical constraints prevented it from matching the simulator's velocity profile.

4.2. K Obstacles

The robot was trained and tested with matched K values. $K = 8$ outperformed other settings in success rate while maintaining high social scores. Increasing K from 1 to 8 improved both success rate and arrival time (from ∼22 s to ∼16 s), reflecting richer perception of high-risk obstacles. However, $K = 16$ degraded success rate back to $K = 2$ levels despite faster arrival, suggesting that additional low-impact obstacles introduce redundant risk signals — the policy becomes overconfident and arrives faster without safer behavior. This indicates the usefulness of limiting crowd observation to a small number of obstacles to keep risk perception clear.

5. Conclusions

We developed an improved DRL-based model for crowd navigation from 2D laser scans, incorporating the top K most dangerous obstacles by Collision Probability into the observation space together with local waypoint rewards. Trained in a single crowd environment and tested on four crowd-behavior settings, the model used risk perception to take calculated risks while waypoints guided it to the goal. Real-world tests showed promising sim-to-real transfer, though performance was less dexterous than in simulation. Our social-safety metric uses a collision-probability criterion rather than the zone-intrusion definition of (Jin et al., 2020), so social scores are not directly comparable across the two; a same-definition comparison, expanded real-world evaluation, and further study of risk- and human-awareness in navigation policies remain future work.

References

Alyassi, R., Cadena, C., Riener, R., and Paez-Granados, D. Social robot navigation: a review and benchmarking of learning-based methods. *Frontiers in Robotics and AI*, 12:1658643, 2025.

Arambula Cosío, F. and Padilla Castañeda, M. Autonomous robot navigation using adaptive potential fields. *Mathematical and Computer Modelling*, 40(9):1141–1156, 2004. ISSN 0895-7177. doi: https://doi.org/10.1016/j.mcm.2004.05.001.

Fiorini, P. and Shiller, Z. Motion planning in dynamic environments using velocity obstacles. *The international journal of robotics research*, 17(7):760–772, 1998.

Fox, D., Burgard, W., and Thrun, S. The dynamic window approach to collision avoidance. *IEEE robotics & automation magazine*, 4(1):23–33, 2002.

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

Helbing, D. and Molnár, P. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282, 1995.

Jin, J., Nguyen, N. M., Sakib, N., Graves, D., Yao, H., and Jagersand, M. Mapless navigation among dynamics with social-safety-awareness: a reinforcement learning approach from 2d laser scans. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6979–6985, 2020. doi: 10.1109/ICRA40945.2020.9197148.

Long, P., Fan, T., Liao, X., Liu, W., Zhang, H., and Pan, J. Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6252–6259, 2018. doi: 10.1109/ICRA.2018.8461113.

Pérez-Higueras, N., Otero, R., Caballero, F., and Merino, L. Hunavsim: A ros 2 human navigation simulator for benchmarking human-aware robot navigation. *IEEE robotics and automation letters*, 8(11):7130–7137, 2023.

Redmon, J. and Farhadi, A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

Rösmann, C., Hoffmann, F., and Bertram, T. Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control. In *2015 European Control Conference (ECC)*, pp. 3352–3357, 2015. doi: 10.1109/ECC.2015.7331052.

Sun, L., Zhai, J., and Qin, W. Crowd navigation in an unknown and dynamic environment based on deep reinforcement learning. *IEEE Access*, 7:109544–109554, 2019. doi: 10.1109/ACCESS.2019.2933492.

Tai, L., Paolo, G., and Liu, M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 31–36, 2017. doi: 10.1109/IROS.2017.8202134.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Zhelo, O., Zhang, J., Tai, L., Liu, M., and Burgard, W. Curiosity-driven exploration for mapless navigation with deep reinforcement learning. In *ICRA 2018 Workshop on Machine Learning in Planning and Control of Robot Motion*, May 2018.

Zhu, K. and Zhang, T. Deep reinforcement learning based mobile robot navigation: A review. *Tsinghua Science and Technology*, 26(5):674–691, 2021. doi: 10.26599/TST.2021.9010012.