# Deep Reinforcement Learning-Based Navigation Method for Mobile Robots in Dense and Dynamic Pedestrian Environments

Hongyu Zhou, Runhua Wang, and Xuebo Zhang, *Member, IEEE*

***Abstract*—Addressing the challenges of mobile robot navigation in dense and dynamic pedestrian environments, this paper proposes a deep reinforcement learning framework that integrates pedestrian trajectory prediction with social feature understanding. The core contributions of this method are as follows: First, a spatio-temporal probability density map is designed, which encodes Kalman filter-based pedestrian trajectory predictions into structured inputs, enabling the robot to explicitly reason about the future position distribution of pedestrians. Second, a DBSCAN clustering-based social feature extraction mechanism is proposed, combined with a bio-inspired attention network, to model group interactions among pedestrians. Finally, a novel reward function incorporating Time-To-Collision and social potential fields is constructed to synergistically optimize both goal-directed navigation and social compliance. Simulation results in Gazebo demonstrate that, in dense dynamic scenarios, the proposed method achieves an 8% improvement in navigation success rate compared to existing mainstream approaches, validating its comprehensive advantages in safety, efficiency, and social rationality.**

## I. Introduction

With the increasing deployment of mobile robots in dense human-populated scenarios such as shopping malls, train stations, and hospitals, their reliable navigation within dynamic pedestrian environments has become a critical challenge. Traditional path planning methods often underperform due to the high uncertainty and stochasticity of pedestrian motion. Deep Reinforcement Learning (DRL), which enables agents to learn optimal policies through interaction with the environment, offers a promising solution to this problem. Building upon existing works such as DRL-VO [1], this paper introduces the following key contributions:

- The use of a Kalman filter model to predict pedestrian trajectories, which are encoded as a spatio-temporal probability density map serving as input to the deep reinforcement learning model.
- The design of a DBSCAN clustering-based social feature extraction mechanism, coupled with a bio-inspired social attention network, to enhance the model's capability to understand pedestrian group behaviors.
- The formulation of a novel reward function that integrates Time-To-Collision and social potential fields, aimed at improving the system's proactive obstacle avoidance performance.

H. Zhou, R. Wang, and X. Zhang are with the Institute of Robotics and Information Automation, School of Artificial Intelligence, Nankai University, Tianjin, China, and also with the Tianjin Key Laboratory of Intelligent Robotics.

## II. Design of Prediction-Based Observation Space

### A. Trajectory Encoding via Spatio-Temporal Probability Density Map

This study employs a Kalman filter model for pedestrian trajectory prediction. The output of this model is encoded into a spatio-temporal probability density map, generated through the following process: A 20 m × 20 m area surrounding the robot is discretized into an 80 × 80 grid. For each predicted trajectory point, its position within the grid is calculated, and a Gaussian kernel is applied for spatial diffusion across its 3×3 neighborhood. Concurrently, an exponential time decay factor is introduced for prediction points across future time steps. Finally, the contributions from all trajectory points are superimposed and normalized, resulting in a probability density map that reflects the future position distribution of pedestrians. This method effectively characterizes the probabilistic distribution of future pedestrian positions.

### B. LiDAR Data Processing

The LiDAR performs 10 scans within 0.5 seconds, covering a 180-degree frontal field of view of the robot. The 720 data points obtained from each scan are processed as follows: every 9 consecutive data points are grouped, and their minimum and average values are computed separately, thereby dividing the scanning area into 80 sectors. By stacking recent historical data, an 80×80 LiDAR data map is ultimately formed.

### C. Social Feature Extraction Mechanism

The DBSCAN clustering algorithm [2] is utilized to identify pedestrian groups and individual pedestrians. For each entity (group or individual), the following seven-dimensional features are computed: group size, density, average velocity, velocity direction, positional angle (relative to the robot), velocity variance, and distance to the robot. All features are normalized. Finally, the 6 entities closest to the robot are selected to generate a 6×7 social feature matrix.

### D. Sub-goal Optimization and Selection Strategy

This study improves upon the Pure Pursuit algorithm [3] by dynamically adjusting the path segment index to select sub-goals. When it is detected that the line connecting the robot and a candidate sub-goal intersects an obstacle, the algorithm automatically backtracks and selects a closer point on the path as the sub-goal, iterating until a safe navigation point is identified.

## III. Deep Reinforcement Learning Network Architecture

This study employs two distinct network frameworks to process different input features:

### *A. Feature Extraction Network*

A Deep Neural Network (DNN) is adopted to represent the parametric model π, leveraging its exceptional function approximation capability. The network utilizes an early fusion strategy, concatenating the LiDAR data and the pedestrian trajectory prediction map (both formatted as 80×80 arrays) along the depth dimension. The concatenated data is subsequently passed through one 2D convolutional layer, six bottleneck residual blocks [4], and two 2D pooling layers. At the terminus of the feature extraction network, a fully connected layer integrates the extracted high-level features with the sub-goal information and the social features processed by the bio-inspired network, thereby comprehensively capturing environmental context. Ultimately, the network outputs a 256-dimensional high-level feature vector.

### *B. Bio-inspired Social Attention Network*

To achieve a deeper understanding of social navigation scenarios, we propose a bio-inspired social attention network. Its core consists of three components: First, a four-head self-attention mechanism is responsible for modeling the complex interactions among multiple pedestrians. Second, we design a set of learnable bio-inspired prior weights, grounded in human behavior studies. These weights are initialized to assign higher importance to critical features such as distance and motion velocity, guiding the network to rapidly focus on potential collision risks. Finally, a spatial attention sub-module is incorporated to simulate the human visual preference for paying greater attention to the frontal area during navigation. This design aims to learn obstacle avoidance strategies that align with human social conventions from data.

### *C. Actor-Critic based Reinforcement Learning Network*

The Proximal Policy Optimization algorithm (PPO) [5] is employed to train the network. The network structure comprises two parts: The Actor network is responsible for outputting continuous actions, namely the robot's linear and angular velocities. The Critic network is used to estimate the value function V(s) of the current state. The action output range is constrained as follows: linear velocity [0, 0.5] m/s, angular velocity [-2, 2] rad/s.

### *D. Reward Function Design*

The reward function is designed to synergistically optimize goal-directed navigation, safety, and social compliance. It consists of the following components:

1) Basic Reward Terms

- Goal reward: $r_g^t$ :Encourages the robot to move towards the goal. This includes a sparse reward for successful goal arrival and a dense reward proportional to the reduction in distance to the goal.
- Collision Avoidance Reward $r_c^t$ : Penalizes proximity to obstacles based on LiDAR scan data.
- Smoothness Reward $r_w^t$: Penalizes abrupt changes in angular velocity to encourage motion smoothness.

2) Time-To-Collision (TTC) Reward

The Time-To-Collision is calculated based on the relative position and velocity between the robot and pedestrians. The TTC concept, formally introduced from a visual control perspective by Lee et al. [6], provides a direct metric for assessing collision imminence. In this study, the TTC reward is designed as:

$$r_{ttc}^t = -w_{ttc} \cdot (1 - \frac{ttc}{ttc_{threshold}})$$

Here, the $ttc_{threshold}$ is set to 3.0 seconds. This function penalizes potential collision risks.

3) Social Potential Field Reward

Inspired by the repulsive force formula in the Social Force Model [7], a social potential field reward function is designed:

$$r_{social}^t = -w_{social} \cdot \frac{\log(1 + group_size)}{dist} \cdot \exp(-\frac{dist}{social_radius})$$

This function comprehensively considers pedestrian group characteristics and penalizes behaviors that intrude upon social comfort zones.

## IV. Experiments and Results Analysis

To validate the effectiveness of the proposed algorithm in practical scenarios, this section presents a comprehensive evaluation of the navigation policy within a simulation environment built in Gazebo. The performance of the proposed algorithm is validated by comparing it against traditional local planners (e.g., DWA [8], TEB [9], E$^3$MoP [10]) and a learning-based method (DRL-VO) across metrics such as navigation success rate, average path length, average navigation time, and average speed. Furthermore, comparative experiments conducted in environments with varying pedestrian densities are used to further examine the generalization capability of the proposed method.

### *A. Simulation Setup*

The experimental arena in the Gazebo simulation environment is configured to be 20 m × 10 m, populated with irregularly distributed static obstacles and 35 dynamic pedestrians moving along different paths. The robot platform utilizes a simulated Turtlebot2 model equipped with a ZED stereo camera and a Hokuyo UTM-30LX LiDAR, with its maximum speed limited to 0.5 m/s. The ZED camera's depth range is set to [0.3, 20] m with a field of view (FOV) of 90°, while the LiDAR's measurement range is set to [0.1, 30] m with a FOV of 270°.

Training was conducted on a high-performance server equipped with 256 logical CPU cores and three NVIDIA GeForce GPUs. The testing phase utilized a laptop with an AMD Ryzen 7 5800H processor and 16 GB of RAM. All software and hardware environments ran Ubuntu 20.04, ROS Noetic, and Gazebo 11.

### *B. Comparative Results*

The proposed method, which incorporates pedestrian trajectory prediction, is compared against traditional methods

(DWA, TEB, E$^3$MoP) and the learning-based method DRL-VO. The evaluation metrics include success rate, average navigation time, average path length, and average speed. The simulation results are presented in Table I. To validate the algorithm's generalization capability, the performance of the prediction-enhanced model is compared against baseline methods under different pedestrian densities, with the results shown in Table II.

TABLE I. COMPARISON OF SUCCESS RATES AMONG LOCAL PLANNING ALGORITHMS IN DYNAMIC ENVIRONMENTS

| Method | Success Rate | Average Time(s) | Average Length(m) | Average Speed (m/s) |
|---|---|---|---|---|
| DWA | 0.68 | 11.03 | 4.89 | 0.44 |
| TEB | 0.72 | 20.37 | 6.38 | 0.31 |
| E$^3$MoP | 0.66 | 15.80 | 5.02 | 0.32 |
| DRL-VO | 0.76 | 14.11 | 6.36 | 0.45 |
| Ours | 0.84 | 15.08 | 6.83 | 0.45 |

TABLE II. COMPARISON OF ROBOT GOAL-REACHING SUCCESS RATES IN DYNAMIC ENVIRONMENTS

| Number of Pedestrians | Method | Success Rate | Average Time(s) | Average Length (m) | Average Speed (m/s) |
|---|---|---|---|---|---|
| 15 | DRL-VO | 0.87 | 15.05 | 6.56 | 0.44 |
| | Ours | 0.92 | 15.26 | 6.64 | 0.44 |
| 25 | DRL-VO | 0.81 | 14.58 | 6.61 | 0.45 |
| | Ours | 0.88 | 15.86 | 7.02 | 0.44 |
| 35 | DRL-VO | 0.76 | 14.11 | 6.36 | 0.45 |
| | Ours | 0.84 | 15.08 | 6.83 | 0.45 |

### C. Results Analysis

The E$^3$MoP algorithm, which focuses on path smoothness and safety, is primarily designed for static environments. It does not fully account for the dynamic nature of pedestrians, and its safety distance parameters require tuning for specific scenarios, leading to its relatively low success rate. The DWA method achieves the best performance in terms of average time and average path length. This likely stems from its relatively simple obstacle avoidance strategy when encountering obstacles, which shortens the planned path but simultaneously increases collision risk. The TEB method exhibits the longest average navigation time, attributable to increased computational overhead caused by frequent path replanning in dynamic environments.

By employing the algorithm proposed in this study, the robot's obstacle avoidance capability in dynamic environments is significantly enhanced. Taking the high-density scenario with 35 pedestrians as an example, the success rate of our method, which integrates prediction information, reaches 0.84, representing an 8% improvement over methods using only the current pedestrian positions and velocities. It is noteworthy that our method results in an increased average path length. This is primarily because the robot, while balancing goal-directed movement and pedestrian avoidance, demonstrates enhanced compliance with pedestrian social spaces.

## V. CONCLUSION

This paper proposes a deep reinforcement learning-based navigation method for mobile robots. By introducing a spatio-temporal probability density map, a social feature extraction mechanism, and a multi-modality-informed reward function, the method effectively enhances navigation performance in dense and dynamic pedestrian environments. Experimental results demonstrate the favorable practicality and robustness of the proposed approach in complex crowd scenarios.

Although this study has achieved certain outcomes, there remains room for further improvement. The simulation environment assumes perfect access to ground-truth pedestrian positions and velocities, whereas in real-world settings, acquiring this information involves uncertainty. The impact of such perceptual uncertainties on navigation success rates warrants further investigation. Future work will focus on validating the algorithm's performance under more realistic perceptual conditions and deploying it on a physical robot platform.

## REFERENCES


[1] Z. Xie and P. Dames, "DRL-VO: Learning to navigate through crowded dynamic scenes using velocity obstacles," IEEE Trans. Robot., vol. 39, no. 4, pp. 2700–2719, 2023...

[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discov. Data Min.*, 1996, pp. 226–231.

[3] R. C. Coulter, *Implementation of the Pure Pursuit Path Tracking Algorithm*, Carnegie Mellon University, Robotics Institute, Tech. Rep. CMU-RI-TR-92-01, 1992.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 1–12, 2017.

[6] D. N. Lee, "A theory of visual control of braking based on information about time-to-collision," *Perception*, vol. 5, no. 4, pp. 437–459, 1976.

[7] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, May 1995.

[8] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Automat. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[9] C. Rösmann, F. Hoffmann, and T. Bertram, "Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control," in *Proc. Eur. Control Conf.*, Linz, Austria, 2015, pp. 3352–3357.

[10] J. Wen, X. Zhang, X. Gao, H. Yu, and X. Li, "E$^3$MoP: Efficient motion planning based on heuristic-guided motion primitives pruning and path optimization with sparse-banded structure," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, pp. 2762–2775, 2022.