2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

An Environmental-Complexity-Based Navigation Method Based on Hierarchical Deep Reinforcement Learning

Pengbin Chen†, Qi Liu†, Yanjie Li*, and Shuaikang Ma

Abstract— **Navigation methods based on deep reinforcement learning (RL) have recently exhibited superior performance, particularly for navigation in dynamic environments. However, most existing methods solely rely on deep neural network feature encoders to extract features from raw LiDAR data, lacking an explicit representation of environmental structure. This limitation hinders effective environmental representation and interpretability, constraining navigation performance improvement. To solve this problem, we propose two quantitative metrics based on laser scans, which explicitly represent environmental complexity and show great interpretability. Furthermore, we propose an environmental-complexity-based navigation method based on hierarchical deep RL with the proposed metrics. Experimental results show that the proposed method achieves better navigation performance than baselines, especially in challenging scenarios with corners and dynamic obstacles.**

I. Introduction

Mobile robots have widespread application across various industrial domains, including logistics, warehousing, and home automation. Autonomous navigation in unknown and highly dynamic environments emerges as a foundational challenge within these diverse contexts [1].

Traditional navigation methods involve two key stages: path planning and trajectory tracking. Path planning entails the generation of a collision-free route using a known static map. The subsequent stage requires the robot to adhere to kinematic constraints while following the generated trajectory. Trajectories crafted from static maps provide an optimality assurance. However, although existing methods for local planning can achieve obstacle avoidance, they generally compromise global optimality. These methods rely heavily on manually tuned parameters, posing challenges for seamless adaptation to unforeseen scenarios. Furthermore, parameter tuning demands significant expertise and time, even in a fixed scenario.

Deep reinforcement learning (RL) [2] has great potential for decision-making problems in various complex environments [3], [4]. Recently, navigation methods based on deep RL have attracted extensive research to overcome the limitations of traditional methods. Deep RL-based navigation methods [5]–[8] employ an end-to-end training approach to learn a policy that directly maps raw sensor inputs to actions and has shown excellent performance compared to traditional methods. However, despite the notable achievements of these methods, they still struggle in some complex situations such as corners, dead-ends and dynamic obstacle avoidance [9], [10].

The performance of deep RL navigation largely depends on the ability to perceive the environment. Existing methods primarily employ artificial neural networks [11] as encoders to extract features from raw laser data. However, the feature encoders based on artificial neural networks lack an explicit representation of environmental structures such as corners and dynamic obstacles. Moreover, these encoders struggle to ensure that the extracted features effectively represent crucial information of the environment due to the inherent randomness of neural networks. Furthermore, comprehending the physical significance of these features remains elusive, hindering the derivation of guidance for performance improvement from the trained encoders.

To address this problem, we propose two quantitative metrics: variation rate and entropy of environment structure based on laser scans. Compared to features extracted by neural networks, these two metrics can effectively provide explicit expressions of environmental complexity and offer better interpretability. Moreover, we propose an environmental-complexity-based navigation method using a hierarchical deep RL framework. In this hierarchical framework, the high-level policy gains a deep understanding of the surrounding environment with the explicit features provided by the proposed metrics. Our method demonstrates improved navigation performance with a clear advantage in challenging scenarios with corners and dynamic obstacles.

The main contributions of this work can be summarized as follows:

- In this paper, we propose two quantitative metrics based on laser scans, which explicitly represent the complexity of the environment.
- We propose an environmental-complexity-based hierarchical deep RL navigation method with the proposed metrics.
- Experimental results show that the navigation performance of the proposed method outperforms the baseline, particularly in testing environments with corners and dynamic obstacles.

The rest of the paper is organized as follows. Section II discusses related works. Section III described the proposed method. Section IV provides the detailed experimental re-

This work was supported by National Natural Science Foundation of China [61977019, U1813206] and Shenzhen Fundamental Research Program [JCYJ20180507183837726, JCYJ20220818102415033, JSGG20201103093802006]. *(Corresponding author: Yanjie Li, autolyj@hit.edu.cn)*

The authors are with the Guangdong Key Laboratory of Intelligent Morphing Mechanisms and Adaptive Robotics and School of Mechanical Engineering and Automation, the Harbin Institute of Technology Shenzhen, 518055, China.

† These authors contributed equally to this work.

sults. Finally, Section V presents the conclusions and future work.

II. Related Work

A. End-to-end Deep RL-based Navigation Methods

Navigation methods based on deep RL have been extensively studied and have demonstrated superior performance. Tai et al. [12] first trained an exploring strategy on a discrete action space using Deep Q-Networks (DQN) in simulation. Then, they [13] used Asynchronous Deep Deterministic Policy Gradients (DDPG) to train a continuous controller taking single-line 2D LiDAR data as input, which is able to implement navigation in the real environment. Long et al. [5] introduced a multi-agent framework to train an obstacle avoidance policy based on 2D LiDAR data. Jin et al. [7] addressed the issue of social safety awareness in navigation purely based on 2D laser scans. To improve navigation efficiency in crowded environments, Chen et al. [14] introduced the attention mechanism to extract a compact crowd representation. Liu et al. [15] designed models for static and dynamic obstacles separately to obtain better features. However, these methods rely on precise perception of dynamic obstacles. Despite the notable achievements of the existing methods, obstacle avoidance in complex environments is still an open frontier.

B. Hybrid Framework for Navigation

To further enhance the effectiveness of navigation strategies, various researchers adopt a hybrid control framework to combine the strengths of different controllers. Within this framework, the primary objective of high-level policy is to comprehend the navigation environment and subsequently determine which lower-level policy to employ. Jin et al. [16] proposed a set of manually designed switching rules to combine goal navigation and obstacle avoidance policies. Shucker et al. [17] proposed specific switching rules tailored to scenarios where addressing collisions and noise pose challenges. In a different vein, Zhang et al. [18] proposed a heuristic method to switch between deep RL-based policies to address the problem of local-minimum areas. However, these methods exhibit limited flexibility due to their dependence on manually designed rules and parameters. To address this inflexibility, Kastner et al. [9] trained a deep RL agent as the high-level planner in a hierarchical framework to integrate both model-based and learning-based controllers. Furthermore, instead of utilizing a specific set of lower-level policies, Lee et al. [19] trained a family of policies adaptable to a wide range of reward functions via hierarchical reinforcement learning (HRL).

In the hierarchical framework for navigation, a deep RL agent as the high-level planner determines the deployment of low-level policy based on sensor observations. The decision-making ability of the high-level planner depends on its ability to extract environmental features from sensor data. Nevertheless, even for artificial neural networks, extracting vital information from raw laser data remains a formidable challenge. To address this challenge, we propose an environmental-complexity-based navigation method with two quantitative metrics explicitly representing environmental complexity.

III. Environmental-Complexity-Based Navigation Method

A. The Complexity of the Environment

Fig. 1 shows diverse factors of the complexity of the environment, including topographic structures, the presence of dynamic obstacles, and the difficulty associated with perceiving and comprehending the environment. These factors greatly influence the decisions of the agent. However, it is difficult for encoders based on artificial neural networks to guarantee that the extracted features effectively represent the information of environmental complexity due to the inherent randomness of neural networks. To explicitly represent the complexity of the environment, we propose two metrics based on LiDAR measurements: variation rate of environment structure and entropy of environment structure.

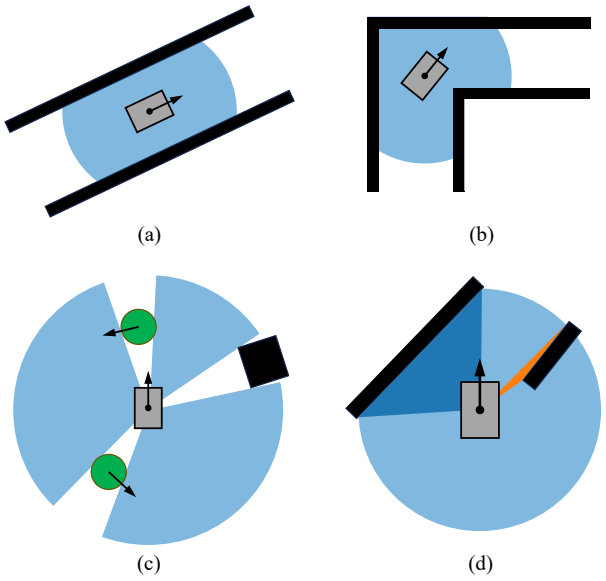


Fig. 1. Diverse factors of environmental complexity. (a), (b) Different topographic structures. (c) Dynamic obstacles. (d) Difficulty of perception. Only a small amount of data detects the planar surfaces roughly parallel to the laser beams. Extracting features from a limited data sample (orange) presents a more formidable challenge than a larger one (deep blue).

1) Variation Rate of Environment Structure: Let $\boldsymbol{P_i}$ denote the coordinates of a specific data point detected in a scan. $\mathcal{N}_i$ denotes the set of consecutive points of $\boldsymbol{P_i}$ in the same scan. $\mathcal{N}_i$ contains $\boldsymbol{P_i}$ and chosen points in pairs in the clockwise and counterclockwise directions from $\boldsymbol{P_i}$. Then, we formally define the variation rate of the environment structure:

$$v_i = \frac{1}{|\mathcal{N}_i|\|\boldsymbol{P_i}\|} \| \sum_{j \in \mathcal{N}_i, j \neq i} (\boldsymbol{P_i} - \boldsymbol{P_j}) \| \tag{1}$$

$|\mathcal{N}_i|$ denotes the number of data points in $\mathcal{N}_i$. $\|\boldsymbol{P_i}\|$ represents the magnitude of $\boldsymbol{P_i}$.

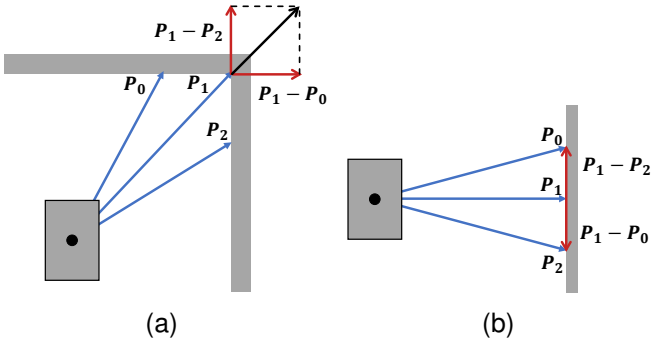


Fig. 2. Calculation process of variation rate v_1 of (a) a corner and (b) a planar surface. v_1 is the variation rate in the direction of $\boldsymbol{P_1}$. Set $\mathcal{N}_1$ contains $\boldsymbol{P_0}$, $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$. The black arrow in (a) represents the vector sum of $\boldsymbol{P_1} - \boldsymbol{P_2}$ and $\boldsymbol{P_1} - \boldsymbol{P_0}$. The vector sum is a zero vector in (b) a planar surface.

Two illustrative examples in Fig. 2 visually demonstrate the calculation procedure for the variation rate. To simplify the problem, we use 2D LiDAR data as an example and only choose a pair of data points for calculation. Note that this calculation process can be easily extended to 3D LiDAR data. $\boldsymbol{P_1}$ in Fig. 2 is the chosen direction to calculate the variation rate. The angle between $\boldsymbol{P_0}$ and $\boldsymbol{P_1}$ is equal to the angle between $\boldsymbol{P_2}$ and $\boldsymbol{P_1}$. v_i represents the variation rate of the environment structure in a particular direction. This calculation methodology can be applied across all directions.

2) Entropy of Environment Structure: Normalize the elements in $\mathcal{N}_i$ and denote the processed set as $\overline{\mathcal{N}}_i$. The element in the $\overline{\mathcal{N}}_i$ is derived through the following calculation:

$$\overline{P_j} = \frac{\|\boldsymbol{P_j}\|}{\sum_{\boldsymbol{P_k} \in \mathcal{N}_i} \|\boldsymbol{P_k}\|}, \boldsymbol{P_j} \in \mathcal{N}_i \tag{2}$$

The information entropy of the environment structure in a direction is defined as:

$$h_i = \mathbb{H}(\overline{\mathcal{N}}_i) = \sum_{\overline{P_k} \in \overline{\mathcal{N}}_i} -\overline{P_k} \log_2 \overline{P_k} \tag{3}$$

Higher entropy arises when the data in $\overline{\mathcal{N}}_i$ are closely clustered. Lower entropy signifies substantial differences between adjacent data points.

The experiments in Section IV-B suggest that variation rate and entropy of environment structure explicitly represent the environmental complexity and possess different sensitivities to diverse factors.

B. Environmental-Complexity-Based Hierarchical Deep RL Navigation Method

Fig. 3 presents the comprehensive architecture of our hierarchical deep RL method. The low-level agent acquires necessary navigation skills. The high-level agent comprehends the environment from the laser scans and determines the deployment of these acquired skills. The environmental-complexity-based high-level agent is constructed based on the proposed metrics: variation rate of environment structure and entropy of environment structure.

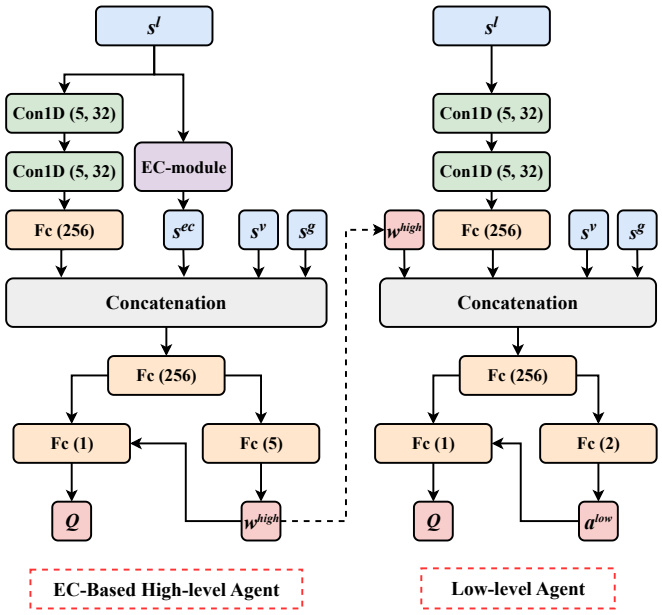


Fig. 3. The network architecture of our method. Environmental-complexity-based (EC-based) high-level agent incorporates the environmental complexity with the EC-module we designed, rather than relying solely on the network-based encoder. Fc denotes a fully-connected layer, and Conv denotes a convolutional layer.

1) Low-level Agent: We train a low-level agent adapting to a diverse array of reward functions following the methodology in research [19]. The details are comprehensively described below.

State space: State of low-level agent $\boldsymbol{s}_t^{low}$ is composed of four components: the raw 2D LiDAR data $\boldsymbol{s}_t^l$, the linear and angular velocity of the robot $\boldsymbol{s}_t^v \in \mathbb{R}^2$, the relative position of the target point in polar coordinate $\boldsymbol{s}_t^g \in \mathbb{R}^2$, and the action of high-level agent $\boldsymbol{w}_t^{high} \in \mathbb{R}^5$.

Action space: Action $\boldsymbol{a}_t^{low} \in \mathbb{R}^2$ is consisting of linear velocity and angular velocity.

Reward: The reward of the low-level agent r_t^{low} is partially determined by a row vector $\boldsymbol{w}_t^{high}$ output from the high-level agent. The specific setting is as follows:

$$r_t^{low} = r_{\text{success}} + \boldsymbol{w}_t^{high}[r_{\text{collision}}, r_{\text{progress}}, r_v, r_w, r_{\text{safety}}]^{\mathbf{T}} \tag{4}$$

where r_{success} is the sparse reward of the navigation task, which is 1 when reaching the goal and 0 otherwise. $r_{\text{collision}}$ is a penalty with a value of -1 when the robot collides with obstacles. r_{progress} is set to encourage the robot to approach the goal. Its value is the difference between the previous and the current distance to the goal. r_v positively correlates with the linear speed, encouraging the robot to move rapidly. The trajectory smoothness reward r_w is negatively correlated with the angular speed. r_{safety} is a penalty when the distance between the robot and the closest obstacle is smaller than a certain threshold. The value of r_{safety} is $1 - \frac{d_t}{z+0.5}$, where d_t denotes the distance to the closest obstacle, and z denotes the radius of the safety zone.

Network architecture: The right part of Fig. 3 illustrates the specific neural network architecture.

2) Environmental-Complexity-Based High-level Agent: We propose an environmental-complexity-based high-level agent based on two explicit metrics that quantify environmental complexity.

State space: Similar to the low-level agent, $\boldsymbol{s}_t^l$, $\boldsymbol{s}_t^v$ and $\boldsymbol{s}_t^g$ are components of the state of the high-level agent. Significantly different from the previous work, the high-level agent incorporates the features of environmental complexity $\boldsymbol{s}_t^{ec}$. $\boldsymbol{s}_t^{ec}$ is consisting of the metrics proposed above: $\boldsymbol{s}_t^{ec} = [\boldsymbol{V}_t, \boldsymbol{H}_t]$. $\boldsymbol{V}_t = [v_0, v_1, ..., v_{n-1}]$ denotes the vector containing variation rates of the environment structure. The parameter n determines the angle interval between adjacent v_i. That is to say, the angular difference between v_i and v_{i+1} is $2\pi/n$. The entropy of the environment structure denoted as $\boldsymbol{H}_t$ follow the same principle.

Action space: Action $\boldsymbol{w}_t^{high} \in \mathbb{R}^5$ is a skill weight. $\boldsymbol{w}_t^{high} \in \mathbb{R}^5$ is part of the input of the low-agent, and decides which kind of skill is finally deployed.

Reward: To encourage the agent to complete the navigation task in the shortest possible time, the reward is set as follows:

$$\begin{cases} r_t^{high} = 0; & \text{reaching the goal} \\ r_t^{high} = -1; & \text{otherwise} \end{cases} \tag{5}$$

Network architecture: The left part of Fig. 3 shows the details. In addition to the encoder based on the neural network, we design a module based on Eqs. (1) and (3) for calculating the environmental complexity features, called EC-module in Fig. 3.

IV. Experimental Results

A. Simulation Setup

We deploy the proposed environmental-complexity-based agent in the nav-gym environment [19]. Fig. 4 shows randomized training environment varying in complexity levels. In each new episode, the positions of static obstacles and the trajectories of dynamic obstacles are randomly generated. Moreover, random noise is introduced into the sensor data. In a single episode, the terminal state is reached under the following conditions: the agent successfully reaches the goal, collides with an obstacle, or exceeds 1000 time steps.

TABLE I

Hyper-parameter

Parameter	Value
Dimension of $\boldsymbol{s}_t^{ec}$ (high-level)	72
Size of $\mathcal{N}_i$ (high-level)	7
Learning Rate (low-level)	1e-4
Learning Rate (high-level)	3e-4
Target Smoothing Coefficient (low-level, high-level)	5e-4
Buffer Size (low-level, high-level)	5e6
Batch Size (low-level, high-level)	256
Discount (low-level, high-level)	0.99

The training consists of two phases. First, the low-level agent is trained with $\boldsymbol{w}_t^{high}$ sampled from a predefined distribution. After the low-level agent completes the learning of navigation skills, the high-level agent is trained to deploy these skills. In the second phase, the actions output by the high-level agent serve as an additional input to the trained low-level agent, thus determining the velocity of the robot. Both the low-level agent and high-level agent are trained using soft actor-critic (SAC) [20] algorithm with the Adam [21] optimizer. Table I shows specific hyper-parameter settings. We evenly selected 36 directions and calculated the variation rate and entropy of the environment structure for each direction, meaning that $\boldsymbol{s}_t^{ec}$ is a 72-dimension feature vector. Furthermore, $\mathcal{N}_i$ is set to 7, meaning that adjacent three pairs of data points in the same scan are selected to calculate v_i and h_i.

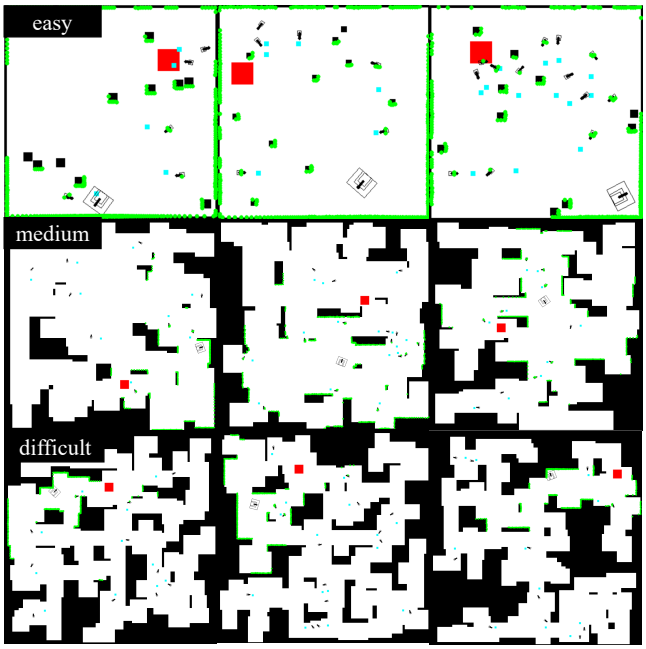


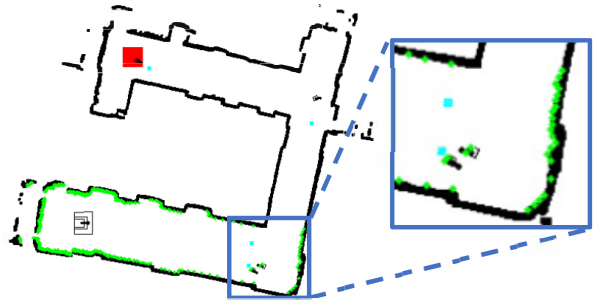
Fig. 4. Randomized environments used in the training phase, where the complexity is jointly determined by a number of static and dynamic obstacles, as well as the size of the maps.

Fig. 5 shows testing environments with different levels of complexity. We evaluated the performance of the proposed method on these maps and compared it with the baseline methods. Baseline methods include a traditional dynamic window approach (DWA) [22] and three learning-based methods LONG [5], GRING [23] and HRL [19].

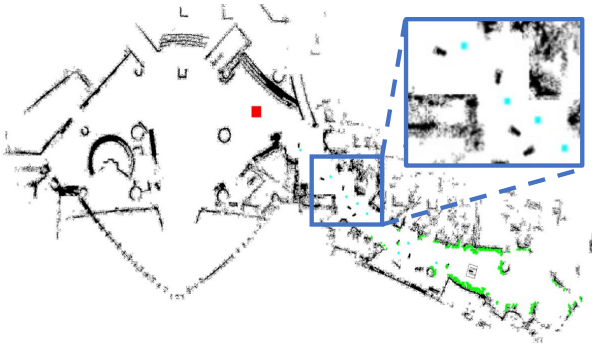
B. Qualitative Evaluation

To validate the effectiveness of the proposed method, we selected two typical scenarios to analyze the ability of metrics qualitatively. In the experiments, five specific directions were selected to calculate the variation rate and entropy of the environment structure.

In Fig. 6 (a), the environment on the left and right sides of the robot's forward direction is almost symmetrical except for the dynamic obstacles in direction 1. The dynamic obstacle is the most complex part of the environment in this frame. The metrics designed both effectively represent this situation similarly. In the scenario depicted in Fig. 6 (b), direction 5 corresponds to a corner, while direction 4 corresponds to a surface roughly parallel to the laser beams.

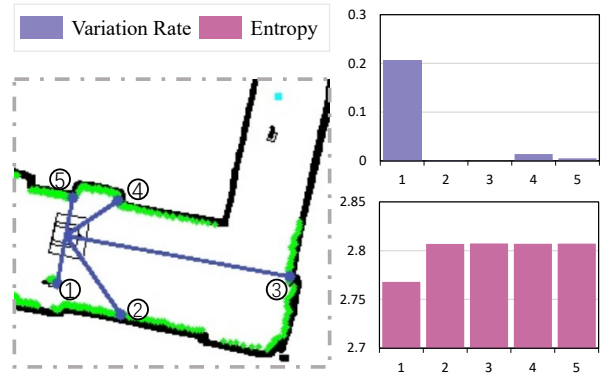


(a) corridor with four dynamic obstacles

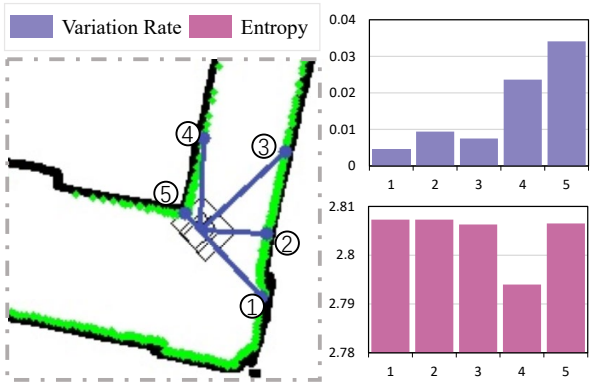


(b) shopping mall with eight dynamic obstacles

Fig. 5. Testing environments used in the evaluation. The red area in the map is the target point of the navigation task.



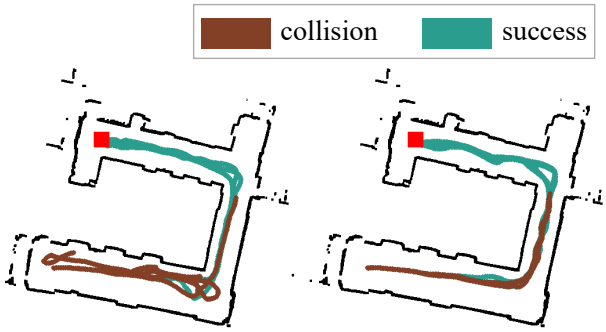
(a)



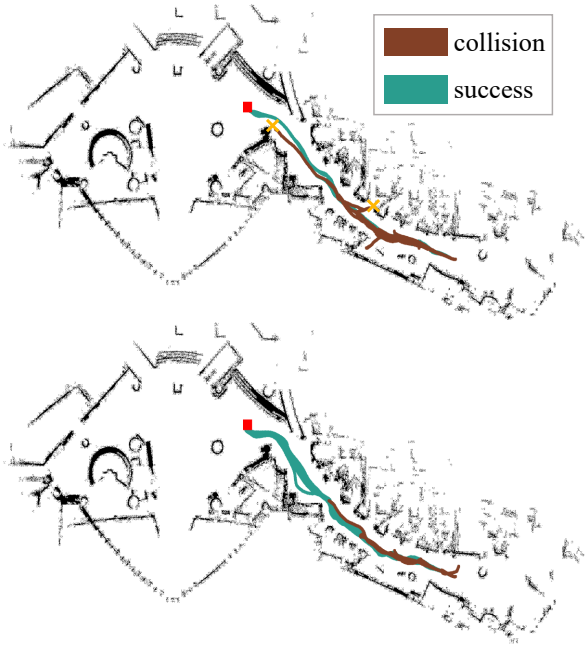
(b)

Fig. 6. Two scenarios for the qualitative evaluation. The histogram on the right shows the entropy and variation rate of the environment structure in five selected directions.

The corner is a crucial aspect of complexity information because it poses a collision risk and serves as a turning point for the robot. The data detecting the planar surface roughly parallel to the laser beams is very limited, which poses challenges to feature extraction and diminishes its reliability. The entropy of the environment structure emphasizes the complexity in directions 4. The variation rate represents the complexity information in directions 4 and 5, with a higher sensitivity to the presence of the corner. In summary, the two metrics designed to characterize the complexity of the environment are reasonable and practical.



(a) Trajectories in corridor



(b) Trajectories in shopping mall

Fig. 7. Trajectories of HRL ((a) left and (b) upper) and EC-HRL (ours) in two test environments. The yellow marker in (b) highlights the situation where HRL collides with the corner.

Moreover, Fig. 7 displays ten trajectories for the baseline HRL and our method in the testing environments. As shown in Fig. 7 (a), in the instance where HRL failed, it exhibits a propensity for making wrong decisions at corners, resulting in subsequent deviations from the target point. As a comparison, the proposed method exhibits more stable decisions at corners. Fig. 7 shows the trajectories in the densely

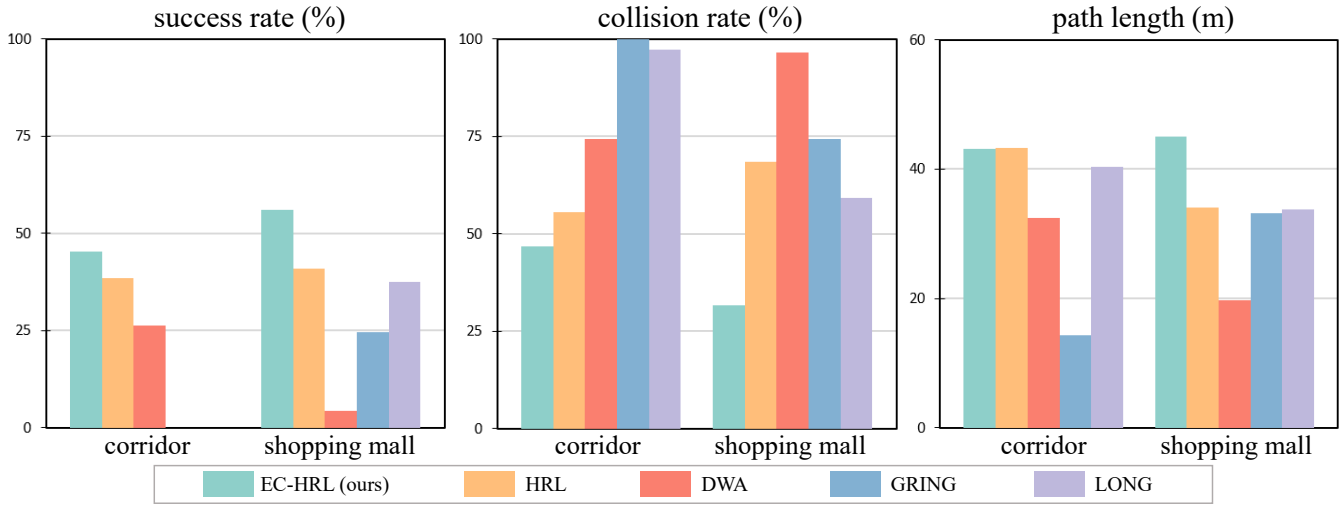


Fig. 8. Quantitative results in two test environments: corridor and shopping mall. Three commonly concerned metrics: success rate, collision rate, and path length are shown in the figure. Our method (EC-HRL) demonstrates excellent performance compared to the baselines.

crowded environment. HRL collides with the wall corners in this scenario, but the proposed method does not. These trajectories intuitively demonstrate that the proposed method enhances the agent's environmental perception, improving its navigation effectiveness.

C. Quantitative Evaluation

We evaluated the performance of the trained policy in two environments unseen during the training phase. Fig. 5 shows the testing environments, including (a) a corridor in the size of 35m*39m with four dynamic obstacles and (b) a shopping mall in the size of 50m*100m with eight dynamic obstacles. During the training phase, the global planner is unavailable to all methods, so the global planning ability is evaluated in the mapless task. Both testing environments demand robots to have sufficient ability for global planning and local obstacle avoidance. Moreover, (a) corridor focuses more on global planning while (b) shopping mall focuses more on local obstacle avoidance in dense crowds scenarios.

The proposed method and baselines are evaluated under identical experimental conditions. To mitigate the impact of randomness stemming from dynamic obstacles, we carried out 100 test repetitions for each scenario. Three commonly assessed metrics containing success rate, collision rate, and path length were recorded for quantitative evaluation.

TABLE II

QUANTITATIVE RESULTS OF CORRIDOR

Method	Success ↑	Collision ↓	Path Length
GRING	0.00%	100.00%	14.27m
LONG	0.00%	97.22%	40.40m
DWA	26.25%	74.44%	32.40m
HRL	38.50%	55.56%	43.33m
EC-HRL (ours)	**45.33%**	**46.67%**	43.15m

Fig. 8 and Tables II-III present the experimental results. Table II and Table III show the success rates, collision rates, and path length in two testing environments. The traditional method DWA performs poorly among the baseline methods, especially in densely crowded environments. LONG and GRING exhibit competence in local obstacle avoidance within crowded environments but fail in the corridor scenario without a global planner. HRL demonstrates the best overall performance among the baseline methods. The performance of the proposed method outperforms baselines across all evaluations. Furthermore, the proposed method significantly reduces collision occurrences compared to other methods. The improvement in collision rate robustly supports the significance of the metrics proposed for characterizing the surrounding environment because local obstacle avoidance in environments with dynamic obstacles emphasizes the demand for a strong perception ability.

TABLE III

QUANTITATIVE RESULTS OF SHOPPING MALL

Method	Success ↑	Collision ↓	Path Length
DWA	4.38%	96.51%	19.72m
GRING	24.58%	74.42%	33.23m
LONG	37.50%	59.30%	33.84m
HRL	41.00%	68.60%	34.08m
EC-HRL (ours)	**56.00%**	**31.67%**	44.96m

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed two quantitative metrics based on laser scans, which explicitly represent environmental complexity and have great interpretability. Furthermore, we proposed an environmental-complexity-based hierarchical deep RL navigation method that incorporates the designed features. Experimental results in complex scenarios with dynamic obstacles demonstrated that the proposed method outperforms the baselines. In the future, we will deploy the proposed method in the real world. For further research, extracting explicit features with physical significance from continuous frames of sensor data is an attractive direction.

REFERENCES

[1] G. Fragapane, D. Ivanov, M. Peron, F. Sgarbossa, and J. O. Strandhagen, "Increasing flexibility and productivity in industry 4.0 production networks with autonomous mobile robots and smart intralogistics," *Annals of Operations Research*, vol. 308, no. 1-2, pp. 125–143, 2022.

[2] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

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

[4] 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.

[5] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, and J. Pan, "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6252–6259.

[6] 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.

[7] J. Jin, N. M. Nguyen, N. Sakib, D. Graves, H. Yao, and M. Jagersand, "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)*. IEEE, 2020, pp. 6979–6985.

[8] D. Dugas, J. Nieto, R. Siegwart, and J. J. Chung, "Navrep: Unsupervised representations for reinforcement learning of robot navigation in dynamic human environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7829–7835.

[9] K. Linh, J. Cox, T. Buiyan, J. Lambrecht *et al.*, "All-in-one: A drl-based control switch combining state-of-the-art navigation planners," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2861–2867.

[10] Q. Liu, Y. Li, and L. Liu, "A 3d simulation environment and navigation approach for robot navigation via deep reinforcement learning in dense pedestrian environment," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2020, pp. 1514–1519.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[12] L. Tai and M. Liu, "A robot exploration strategy based on q-learning network," in *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 2016, pp. 57–62.

[13] L. Tai, G. Paolo, and M. Liu, "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)*. IEEE, 2017, pp. 31–36.

[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] L. Liu, D. Dugas, G. Cesari, R. Siegwart, and R. Dubé, "Robot navigation in crowded environments using deep reinforcement learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5671–5677.

[16] J. Jin, Y. Kim, S. Wee, D. Lee, and N. Gans, "A stable switched-system approach to collision-free wheeled mobile robot navigation," *Journal of Intelligent & Robotic Systems*, vol. 86, pp. 599–616, 2017.

[17] B. Shucker, T. Murphey, and J. K. Bennett, "Switching rules for decentralized control with simple control laws," in *2007 American Control Conference*. IEEE, 2007, pp. 1485–1492.

[18] W. Zhang and Y. Zhang, "Behavior switch for drl-based robot navigation," in *2019 IEEE 15th International Conference on Control and Automation (ICCA)*. IEEE, 2019, pp. 284–288.

[19] K. Lee, S. Kim, and J. Choi, "Adaptive and explainable deployment of navigation skills via hierarchical deep reinforcement learning," *arXiv preprint arXiv:2305.19746*, 2023.

[20] 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.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[23] R. Guldenring, M. Görner, N. Hendrich, N. J. Jacobsen, and J. Zhang, "Learning local planners for human-aware navigation in indoor environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 6053–6060.