2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

A Non-Homogeneity Mapless Navigation Based on Hierarchical Safe Reinforcement Learning in Dynamic Complex Environments

Jianmin Qin, Qingchen Liu*, Qichao Ma, Zipeng Wu, and Jiahu Qin

Abstract— **Addressing safe and efficient navigation in dynamic, realistic, and complex environments stands as a pivotal inquiry within the realm of robotics. Recently, numerous learning-based methods are introduced into the field of navigation, yielding notable outcomes. In this letter, we propose a hierarchical safe reinforcement learning navigation approach (HSRLN) for mapless navigation. It trains mapless navigation policies for non-homogeneous complex scenarios in a hierarchical manner through a kind of three-stage learning, global planning reinforcement learning (RL) + expert imitation learning (IL) + transfer RL (TRL). The innovations of this work are fourfold: a) It effectively reduces the difficulty of training for complex navigation by effectively narrowing the task horizon of RL through a hierarchical framework. b) We designed an imitation learning method based on Relative Driving Safety Index (RDSI) [1] to focus on learning critical expert actions. c) It employs a TRL approach to improve generalization under non-homogeneity assumptions by fine-tuning the policy. d) HSRLN extracts significant features important for navigation decisions from raw observations via velocity obstacle modeling. Experiments indicate that it has it performs better than existing hierarchical RL navigation methods (HDRL [2], SRL-ORCA [3]). Relative to SRL-ORCA, it improves navigation success by 12.1% under the non-homogeneity assumption. Videos are available at https://youtu.be/24h9JmcIfMw.**

I. INTRODUCTION

Efficient and safe navigation systems enhance societal productivity and generate significant economic value. However, there are three challenges in robot navigation: First, performing complex tasks involving global planning and collision avoidance is difficult in the absence of maps or with limited sensor data [4]. Secondly, vehicles and pedestrians typically use different navigation methods and lack pre-established coordinated avoidance protocols, which compromises safety (non-homogeneity assumption) [5]. Thirdly, an additional challenge lies in efficiently extracting features for navigation decisions from the redundant environmental data acquired through sensors [6].

This work was supported in part by the National Key Research and Development Program of China under Grant 2022ZD0120002, in part by the National Natural Science Foundation of China under Grant 62373341, in part by USTC Research Funds of the Double First-Class Initiative under Grant YD2100002013, and in part by Opening Fund of State Key Laboratory of Fire Science (SKLFS) under Grant No. HZ2023-KF01.
(Corresponding author: Qingchen Liu.)

Jianmin Qin, Qingchen Liu, Qichao Ma, and Zipeng Wu are with the Department of Automation, University of Science and Technology of China, Hefei 230027, China (e-mail: qjm@mail.ustc.edu.cn; qingchen_liu@ustc.edu.cn; qcma@ustc.edu.cn; zipengwu@mail.ustc.edu.cn).

Jiahu Qin is with the Department of Automation, University of Science and Technology of China, Hefei 230027, China, and also with the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230088, China (e-mail: jhqin@ustc.edu.cn).

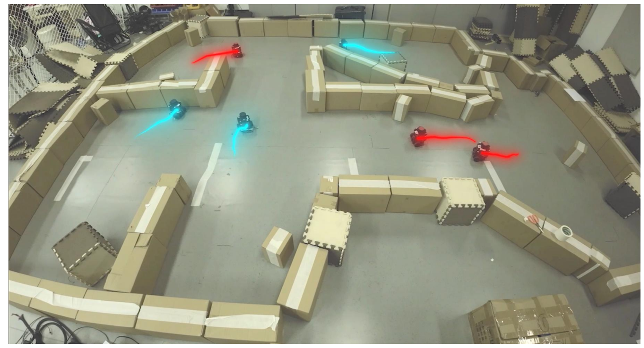


Fig. 1. Robot clusters navigate without maps in non-homogeneous, dynamic, and complex scenarios.

Multi-agent navigation is typically categorized into conventional, learning-based, and their fusion methods. Conventional methods typically entail a fusion of global planning [7], [8] and collision avoidance [9], [10]. They are more stable in operation and are widely used in commercial scenarios. However, conventional collision avoidance methods [9], [10] can guarantee theoretical safety only under homogeneity assumptions. This constraint restricts their applicability in real-world scenarios. These methods [9], [10] exhibit serious safety declines under non-homogeneity assumptions [5]. Learning-based methods [11]–[13] have received extensive attention. Their advantages are the capacity to autonomously acquire navigation policies from real interaction data, obviating the necessity for designing intricate rules. Since the navigation problem is a fusion of different levels of planning, solely relying on end-to-end learning approaches proves inadequate in effectively addressing dynamic environmental challenges. In scenarios characterized by large state and action spaces with long task horizons, standard RL cannot explore sufficiently to obtain better strategies [14]. In contrast, HRL offers a mechanism for tackling challenging tasks by decomposing them into simpler subtasks through a hierarchy of policies.

In our previous work, we proposed SRL-ORCA [3], a "nominal-controller + safetyride" method that uses NH-ORCA [10] as a monitor to ensure safety in RL navigation. However, it inherits NH-ORCA's limitation to homogeneous scenes. This paper proposes a hierarchical safe reinforcement learning approach for mapless navigation, namely HSRLN. It replaces NH-ORCA with learning modules as a safety monitor and extends SRL-ORCA to handle non-homogeneous scenes. To address dynamically complex scenes, it divides the navigation policy into simpler subtasks of global planning and collision avoidance, facilitating hierarchical training. The main contributions are as follows: a) We present a hierarchical training framework for mapless navigation with global planning RL + expert imitation learning + transfer reinforcement learning. By leveraging global or local sub-policy reuse, it effectively reduces the task horizon of RL, thereby reducing the training difficulty of complex tasks and yielding more efficient polices. b) We employ an RDSI-Dagger imitation learning method in HRL based on RDSI threat assessment. It overcomes the state distribution mismatch between IL and expert policy (NH-ORCA), and focuses on learning critical expert actions in complex scenes. c) We designed a Transfer reinforcement learning [15] method for representation reuse. It utilizes a progressive neural network structure [16] to prevent catastrophic forgetting [15] of avoidance policy in IL, and fine-tuning to enhance generalization. It exhibits better avoidance performance under both homogeneity and non-homogeneity assumptions. d) We model dynamic obstacles as velocity obstacles on the Voronoi diagram [9]. This extracts crucial features of the optional action space from the raw observation, thereby improving RL performance.

II. RELATED WORK

A. Conventional navigation methods

Conventional methods are characterized by two notable shortcomings. Firstly, global algorithms [7], [8] typically necessitate highly precise maps to facilitate accurate path planning. This places high demands on both the sensors and the arithmetic power carried by vehicles. They are not deployable in mapless scenarios characterized by inadequate sensing precision. Secondly, ORCA methods [9], [10] are only applicable to collaborative collision avoidance under homogeneity assumption, but not to non-homogeneity assumption [5]. Their avoidance performance degrades dramatically in non-homogeneity environments. homogeneity assumption implies that all agents, whether vehicles or pedestrians, are controlled by the same algorithm. This entails the existence of pre-established, coordinated avoidance protocols among all agents. Evidently, in real-world scenarios, this assumption often does not hold. The behavior of humans or vehicles is frequently influenced by multiple factors such as habits, goals, and emotions. Consequently, they often exhibit diverse movement intentions and do not consistently adhere to standardized avoidance protocols.

B. Learning-based and Fusion navigation methods

Learning-based algorithms commonly employ deep learning (DL) [11], IL [12], RL [13], or TRL [15] to train navigation policies. Moreover, recent studies [6], [17] integrate conventional methods to enhance the efficacy of learning-based approaches. HGAT-DRL [17] employs velocity obstacle (VO) [9] to characterize obstacle data, facilitating more effective collision avoidance learning and enhancing success rates. DRL-VO [6] introduces an innovative RL approach for navigation within pedestrian-dense environments, utilizing VO to design the reward function. However, their [3], [6], [17] common shortcoming is that they jointly execute global and local planning with a single policy, without a reasonable task decomposition. These approaches have excessively long action sequences (task horizons) [18], leading to a high dimensionality of the action space. This challenge makes it difficult for RL to explore and obtain better policies for complex dynamic navigation. Unlike the above approaches, our work reasonably hierarchizes the navigation framework into easily trainable sub-tasks, which effectively improves the performance of learning-based navigation [18].

C. Hierarchical reinforcement learning navigation

HRL methods are a subset of learning-based navigation that has recently received widespread attention. It is well-suitable for learning goal-conditioned behavior in complex tasks and long task horizons [18]. Predominant HRL methods, such as Option Critic [19], Feudal Networks [20], Hierarchical Actor-Critic [21], decompose long-horizon RL tasks into simpler subtasks capable of being reused. This significantly enhances the efficiency of RL exploration. Several new works [2], [22] also introduce HRL into dynamic and complex navigation for robots. HDRL [2] proposes a goal-directed representation of the action space of subgoals based on HRL, enabling fast and safe navigation in various environments. PNSS [22] designs mapless navigation methods on the HRL framework, which obtains a better strategy by predicting neighboring space scores to reduce the observation dimensions. Their shared drawback is a failure to adequately integrate expert knowledge from conventional methods, relying solely on a simplistic hierarchical approach to strategy training. They still exhibit limited efficacy in dynamic obstacle scenarios lacking maps.

III. PROBLEM FORMULATION

The hierarchical safe navigation task is formulated as a semi-Markov decision process (SMDP) [18]. An SMDP is defined as a 7-tuple $(S, A, P, R, T, \Omega, O)$. S is the state space, A is the action space, P is the transfer function, R is the reward, T is the waiting time distribution, Ω is the subtask space, and O is the observation space. Experimental scenarios have M moving agents that act as dynamic obstacles to each other. Next, observation O, subtask space Ω, and action space A are introduced in detail. The state probability transfer model P is unknown.

Observation space: For the agent-i ($1 \leq i \leq M$), its observation at time t consists of four components, $o_{i,t} = [o_{i,t}^{in},\ \tilde{o}_{i,t}^{ro},\ o_{i,t}^{env},\ o_{i,t}^{vo}]$. Here, $o_{i,t}^{in}$ represents the internal data of agent-i, $\tilde{o}_{i,t}^{ro} = [\tilde{o}_{i,t}^{1},\ \tilde{o}_{i,t}^{2}, \cdots,\ \tilde{o}_{i,t}^{k}]$ represents the information of the k nearest mobile agents for agent-i (k is fixed), and $o_{i,t}^{env}$ represents the information of static obstacles around agent-i. We obtain $o_{i,t}^{in} = [\ p_i,\ v_i,\ \phi_i,\ r_i,\ d_i,\ v_{i,pref}]$ from the agent's internal sensors and $\tilde{o}_{i,t}^{ro}$, $o_{i,t}^{env}$ by instruments such as cameras or LIDAR. The position, speed, orientation, safety radius, destination, and preferred speed of the agent-i are indicated as p_i, v_i, ϕ_i, r_i, d_i, and $v_{i,pref}$, respectively. The observation of j-th dynamic agent by agent-i is the position, speed, direction, and safety radius, $\tilde{o}^{j}_{i,t} = [\tilde{p}_j,\ \tilde{v}_j,\ \tilde{\phi}_j,\ \tilde{r}_j]$, while their destination $\tilde{d}_j$ and preferred speed $\tilde{v}_{j,pref}$ are hidden states cannot be observed.

Unlike existing RL methods [4], [13], we extract useful velocity obstacle observations $o^{vo}_{i,t}$ based on $o^{in}_{i,t}, \tilde{o}^{ro}_{i,t}$ to obtain better policies. $o^{vo}_{i,t} = [orca^{t}_{i|1},\ orca^{t}_{i|2}, \ldots,\ orca^{t}_{i|M}]$ can be obtained by performing VO modeling of dynamic obstacles around robot-i on Voronoi diagram, as in Section IV-A. $orca^{t}_{i|j} = [p^{n}_{i|j},\ \vec{n}_{i|j}]$ represents features of the ORCA line [9] for robot-i relative to dynamic obstacle-j ($p^{n}_{i|j}$ is a point on diagram and $\vec{n}_{i|j}$ is a unitized vector of direction). M is the number of ORCA lines.

Subtask space and action space: Similar to [18], we define that hierarchical navigation policy π^{hsrln} consists of policy π^{gp}_{θ} for the global planning (GP) subtask and policies $\pi^{il}_{\xi}, \pi^{trl}_{\epsilon}$ for the collision avoidance (CA) subtask. θ, ξ, ϵ are networks of HSRLN, as in Section IV-B. Based on observation $o_{i,t}$ and policy π^{gp}_{θ}, agent-i outputs the global action $g_{i,t}(o_{i,t}) = \pi^{gp}_{\theta}(o_{i,t})$. Since HSRLN is deployed on differential wheeled robots, the sub-task space Ω is equivalent to the allowed holonomic velocity set $S^{RL}_{AHV_i}$ [10], $g_{i,t} = [\,v_{i,t}, w_{i,t}\,] \in \Omega = S^{RL}_{AHV_i}$. Here, $v_{i,t}$ and $w_{i,t}$ represent linear and rotational velocities. When encountering complex dynamic collision avoidance tasks, policy π^{il}_{ξ} in IL part overrides $g_{i,t}$ based on observation $o_{i,t}$ and global action $g_{i,t}$, outputting action $a^{il}_{i,t} = \pi^{il}_{\xi}(o_{i,t}, g_{i,t})$. Policy π^{trl}_{ϵ} of TRL fine-tunes the action $a^{il}_{i,t}$ of IL in non-homogeneous scenarios to obtain final action $a^{hsrln}_{i,t}$ with better generalization capability, $a^{hsrln}_{i,t} = \pi^{trl}_{\epsilon}(o_{i,t}, a^{il}_{i,t})$. Actions $a^{il}_{i,t}, a^{hsrln}_{i,t}$ also belong to set $S^{RL}_{AHV_i}$.

IV. APPROACH

HSRLN designs GP-module and CA-module to accomplish global planning (GP) and collision avoidance (CA) subtasks hierarchically. GP-module is a standard RL structure, while CA-module is composed of imitation learning (IL) and transfer reinforcement learning (TRL) parts. We use three stages to train these networks: First, as in [3], we activate the NH-ORCA monitor and individually train the GP-module individually to improve its mapless planning policy, as shown in Fig. 2(a). Second, IL part of CA-module is trained to learn actions of the NH-ORCA monitor by using imitation learning, as in Fig. 2(b). We use GP-module as a global planning controller to output action $g_{i,t}$. IL part obtains similar collision avoidance policy as NH-ORCA. Finally, to address NH-ORCA's diminished avoidance efficacy in non-homogeneous scenarios, TRL part of CA-module is trained under such conditions, as in Fig. 2(c). Compared to NH-ORCA, HSRLN fine-tunes the policy of CA-module in real non-homogeneous scenarios, resulting in improved generalization performance. It uses a centralized training, distributed execution approach in which all agents following HSRLN share the same navigation policy π^{hsrln}. This section also discusses modeling dynamic obstacles as velocity obstacles on the Voronoi diagram [9] to extract significant features and thereby improve RL performance.

A. Extract features of dynamic obstacles on VO diagram

In multi-agent RL, it is difficult for agents to learn safety actions when they are directly fed with the position and velocity data of dynamic obstacles from sensors [6]. Even humans sometimes fail to make safe avoidance decisions when surrounded by dense, dynamic obstacles [23]. The reason is that neither humans nor robots can directly know which movement decisions are safe based on a large amount of velocity and position data. Hence, constructing velocity obstacles on Voronoi diagrams from these data allows for easier differentiation of safe actions. We can pre-process observations by VO modeling and extract useful features to build interactive agent-level observations. This enables RL to get useful features directly and learn safe policies.

Based on observation $\tilde{o}^{ro}_{i,t}$, the Voronoi diagram for velocity obstacles in HSRLN is constructed as in Fig. 3. For two robots i and j, the velocity obstacle $VO^{\tau}_{i|j}$ for agent-i is the set of all relative velocities of it (the safety radius is $r_i + \varepsilon_{Hi}$ at p_i, r_i is the physical radius, ε_{Hi} is the tracking error) with respect to agent-j (the safety radius is $r_j + \varepsilon_{Hj}$ at p_j). This results in a collision between agent i and j at some moment before time τ [9]. As shown in the diagram, $VO^{\tau}_{i|j}$ is formally defined as follows:

$$VO^{\tau}_{i|j} = \left\{\vec{v} | \exists t \in [0, \tau]\,, t \cdot \vec{v} \in D\left(\, p_j - \ p_i,\ r_i + \varepsilon_{Hi} + \ r_j + \varepsilon_{Hj}\right)\right\} \tag{1}$$

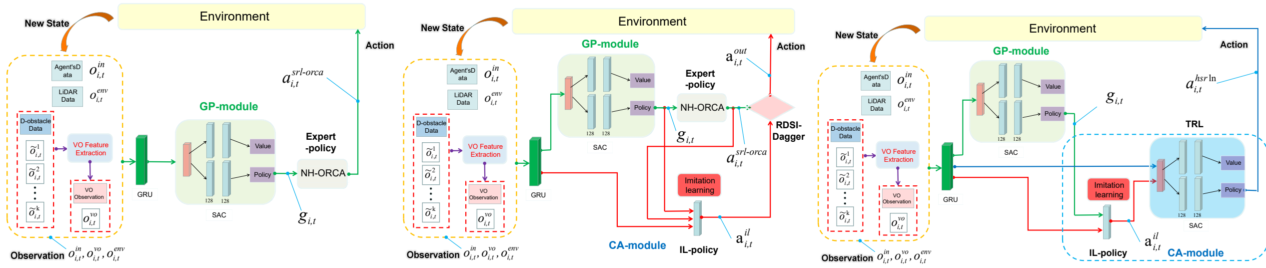
with $D(\mathbf{p}, r) = \{\mathbf{q} | \, \|\mathbf{q} - \mathbf{p}\| < r\}$ the open ball of radius $r = r_i + \varepsilon_{Hi} + r_j + \varepsilon_{Hj}$. Next, the set of collision avoidance velocities $CA^{\tau}_{i|j}\left(V_j\right)$ for agent i (given that robot j selects its velocity from V_j) can be obtained, $CA^{\tau}_{i|j}\left(V_j\right) = \left\{\vec{v} | \vec{v} \notin VO^{\tau}_{i|j} \bigoplus V_j\right\}$. Then we can calculate the minimum variation $\vec{u}$ (the vector from $\left(\vec{v}^{opt}_i - \vec{v}^{opt}_j\right)$ to the closest point on the boundary of velocity obstacle). where $\vec{v}^{opt}_i$ is the optimization velocity and $\vec{n}$ denotes the outward normal of the boundary of $VO^{\tau}_{i|j}$ at point $\left(\vec{v}^{opt}_i - \vec{v}^{opt}_j\right) + \vec{u}$. From the allocation of $\vec{u}$ by priority levels pr^{τ}_{ji} and pr^{τ}_{ij} [3], we can obtain the vector of point $\vec{p}^{n}_{i|j}$ as follows:

$$\vec{p}^{n}_{i|j} = \vec{v}_i + \frac{pr^{\tau}_{ji}}{pr^{\tau}_{ji} + pr^{\tau}_{ij}} \cdot \left[\underset{\vec{v} \in \partial VO^{\tau}_{i|j}}{\operatorname{argmin}} \|\vec{v} - (\vec{v}^{opt}_i - \vec{v}^{opt}_j)\| - (\vec{v}^{opt}_i - \vec{v}^{opt}_j) \right]. \tag{2}$$

The point $\vec{p}^{n}_{i|j}$ and the vector $\vec{u}$ mark the position of the $ORCA^{\tau}_{i|j}$ line, which is a useful feature for robots to avoid each other. They represent which velocity areas are safe on the Voronoi diagram. We feed each $orca_{i|j} = [p^{n}_{i|j},\ \vec{n}_{i|j}]$ into the HSRLN as important observations. Given the higher update frequency of our VO graph, we can approximate that obstacle-j in non-homogeneous scenarios has a large pr^{τ}_{ji}, letting agent-i take the main avoidance responsibility.

B. Hierarchical Safety Reinforcement Learning

(1) RL training of global planning module: Compared to conventional navigation, global planning in RL is less reliable on maps and allows for flexible and dynamic decision-making in non-convex complex scenarios [4]. Similar to SRL-ORCA [3], HSRLN uses the GP-module as the global planning controller in the first stage of training, while the



(a) (b) (c)

Fig. 2. (a)(b)(c) denote three stages of training for the HSRLN algorithm. The yellow box indicates the process of extracting VO features from raw observations. The blue box shows the CA-module, which consists of two parts of networks, IL and TRL. Green lines in (a) demonstrate the process of training global planning in the GP-module. (a) is a standard RL framework. Red lines in (b) represent the IL part learning collision avoidance policy from NH-ORCA under the homogeneity assumption. Blue lines in (c) indicate that TRL part uses transfer reinforcement learning to fine-tune the avoidance policy of IL under the non-homogeneity assumption.

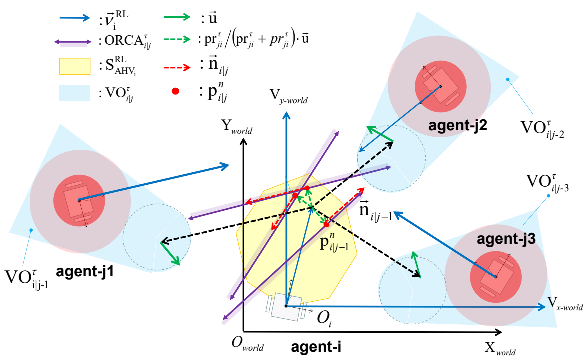


Fig. 3. Extracting features of dynamic obstacles on Voronoi diagrams.

NH-ORCA monitor is turned on to ensure safety of dynamic avoidance, shown by green lines in Fig. 2(a). NH-ORCA overrides unsafe actions output by GP-module, thus GP-module only needs to implement mapless global planning in static environments [3]. Due to the larger space for safe motions in static scenes, it can obtain better navigation performance. We use the SAC algorithm [24] to train policy π_θ^{gp} for GP-module. It contains Q-value network ψ and policy network θ. HSRLN trains agents based on observation $o_{i,t}^{in}, o_{i,t}^{env}$ and VO observation $o_{i,t}^{vo}$. $o_{i,t}^{vo}$ provides more direct and useful feature of dynamic obstacles than the raw observation $o_{i,t}^{ro}$. As in Fig. 2(a), variable-length $o_{i,t}^{vo}$ is encoded by the GRU network as a fixed-length feature. The Q-value function is defined as follows

$$Q_i^{\psi}(o_{i,t}, g_{i,t}) = h(o_{i,t}^{in}, o_{i,t}^{env}, o_{i,t}^{vo}, g_{i,t}), \tag{3}$$

where $h(\cdot)$ represents a two-layer MLP network. The regression loss function of Q-value network ψ can be defined as

$$\begin{aligned} L_Q(\psi) =& E_{(o_{i,t}, g_{i,t}, r_{i,t}^{gp}, o_{i,t+1})\sim R}\left[\frac{1}{2}\left(Q_i^{\psi}(o_{i,t}, g_{i,t}) - y_{i,t}\right)^2\right], \\ y_{i,t} =& r_{i,t}^{gp} + \gamma E_{g_{i,t}\sim\pi_{\bar{\theta}}(o_{i,t+1})}[Q_i^{\bar{\psi}}(o_{i,t+1}, g_{i,t+1}) \\ &- \alpha \log(\pi_{\bar{\theta}}(g_{i,t+1}|o_{i,t+1}))], \end{aligned} \tag{4}$$

where $Q_i^{\bar{\psi}}$ represents another Q-network in SAC to solve the overestimation problem, α is a regularisation factor, $r_{i,t}^{gp}$ is the reward for training in the global planning stage, and $g_{i,t}$ is the global action currently output by agent-i. The loss regression function of policy network θ can be given as

$$L_\pi(\theta) = E_{o_{i,t}\sim R, g_{i,t}\sim\pi_\theta}\left[\alpha\log(\pi_\theta(g_{i,t}|o_{i,t})) - Q_i^{\psi}(o_{i,t}, g_{i,t})\right]. \tag{5}$$

The global planning reward R_t^{gp} consists of two components, goal reward R_t^{goal} and collision penalty R_t^{col}. Collision penalty R_t^{col} is applied when a collision occurs between agent-i and either a dynamic or static obstacle. When the distance between the agent's position $p_{i,t}$ and the position of goal d_i is less than q, it receives an arrival reward R_t^{goal}.

$$R_t^{col} = \begin{cases} g, & \text{if } \forall x \in [1, N] : \left\|p_{i,t} - B_x\right\| \leq r_i \\ h, & \text{if } \forall j \in [1, M], j \neq i : \left\|\mathrm{p}_{i,t} - \mathrm{p}_{j,t}\right\| \leq r_i + \mathrm{r}_j \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

$$R_t^{goal} = \begin{cases} m - z \cdot t, & \text{if } \left\|p_{i,t} - d_i\right\| \leq q \\ 0, & \text{otherwise}. \end{cases} \tag{7}$$

B_x presents x-th static obstacle. Here, g, h, m, q, z are hyper-parameters of reward functions.

(2) Imitation learning training of collision avoidance module: IL offers an advantage over RL by obviating the necessity for exploratory endeavors from the ground up, facilitating swift assimilation of superior expert strategies [13]. Unlike SRL-ORCA [3], HSRLN uses CA-module to replace NH-ORCA as a safety monitor for dynamic collision avoidance, as in Fig. 2(b). CA-module consists of an imitation learning part (network ξ) and a TRL part (same structure as SAC algorithm [24], including Q-value network σ, and policy network ϵ). Network ξ is trained independently with imitation learning. It provides a collision avoidance policy under the homogeneity assumption for the TRL part. In the second stage of training, based on observation $o_{i,t}$, the trained GP-module acts as a controller to provide the global action $g_{i,t}$. Meanwhile, NH-ORCA is turned on to generate the safe action $a_{i,t}^{srl-orca}$, as follows:

$$a_{i,t}^{srl-orca}(o_{i,t}) = \begin{cases} \pi_\theta^{gp}(o_{i,t}), & \text{safe scene}, \\ \pi^{orca}(o_{i,t}, g_{i,t}), & \text{unsafe scene}. \end{cases} \tag{8}$$

This expert policy π^{orca} for collision avoidance provides safe state-action pairs $\left\{(o_{i,t}, g_{i,t}), a_{i,t}^{srl-orca}\right\}$. Unsafe scenarios are delineated in our previous work [3]. However, imitation learning has the following shortcomings: large action errors are accumulated due to a mismatch between the state distributions of expert and learned policies, which leads to a decline in learning performance. In certain high-risk scenarios, the expert policy is too complex, which also leads to learning failure. HSRLN designs an RDSI-Dagger imitation learning method to improve learning performance in dangerous scenarios based on threat assessment. It pre-trains the network ξ based on state-action pairs. This allows network ξ to learn the collision avoidance policy π^{orca} under homogeneous conditions. In contrast, CADRL [13] uses only expert data generated by NH-ORCA for pre-training. It lacks global planning data to cope with complex static obstacles. As for HSRLN, it adopts a hierarchical network architecture to achieve the global planning and collision avoidance tasks separately. CA-module has shorter task horizon and only needs to focus on collision avoidance, which reduces the difficulty of training. We expect policy $\hat{\pi}^{il}_{\xi}$ to learn the avoidance behavior of NH-ORCA:

$$\hat{\pi}^{il}_{\xi} = \underset{\xi}{\operatorname{argmin}}\, E_{o_{i,t}\sim sd(o_{i,t})}\, L_{il}(\xi)(\,\pi^{orca}(o_{i,t}, g_{i,t}), \pi^{il}_{\xi}(o_{i,t}, g_{i,t})\,). \tag{9}$$

Here, if the learner executes the policy π^{il}_{ξ} between step 1 and step $t-1$, $sd(o_{i,t})$ indicates the state distribution at step t. The loss function $L_{il}(\xi)$ of policy $\hat{\pi}^{il}_{\xi}$ can be defined as

$$L_{il}(\xi) = \eta(o_{i,t}, g_{i,t}) \cdot \left\|\pi^{orca}(o_{i,t}, g_{i,t}) - \pi^{il}_{\xi}(o_{i,t}, g_{i,t})\right\|^2, \tag{10}$$

where $\eta(o_{i,t}, g_{i,t})$ is a weighting factor to assess the risk level of the training sample $\left\{(o_{i,t}, g_{i,t}), a^{srl-orca}_{i,t}\right\}$ in the current navigation scenario. According to Relative Driving Safety Index (RDSI) [1], $\eta(o_{i,t}, g_{i,t})$ for agent-i can be defined as:

$$\eta(o_{i,t}, g_{i,t}) = 1 + (\mathbf{SPE}_{R,i} + \sum_{j=1}^{k}\mathbf{SPE}_{V,ji} + + \sum_{j=1}^{k}\dot{\mathbf{SPE}}_{V,ji})/\mathbf{DSI}^{*}, \tag{11}$$

where $\mathbf{SPE}_{R,i}$ represents the safety potential energy (SPE) of potential fields for agent-i. $\mathbf{SPE}_{V,ji}$ is the energy of kinetic field from agent-j to agent-i. $\dot{\mathbf{SPE}}_{V,ji}$ is the rate of change for $\mathbf{SPE}_{V,ji}$, and $\mathbf{DSI}^{*}$ is the standard driving safety index for the current scene [1]. Meanwhile, to ensure that IL part is still able to learn the expert policy effectively in high-risk scenarios, we employ a controlled exploration policy that allows the expert policy to take over the final output action $a^{out}_{i,t}$ proportionally in dangerous scenes. $a^{out}_{i,t}$ is given as

$$a^{out}_{i,t} = A_{cr} \cdot a^{srl-orca}_{i,t}(o_{i,t}, g_{i,t}) + (1 - A_{cr}) \cdot a^{il}_{i,t}(o_{i,t}, g_{i,t}), \tag{12}$$

where $a^{il}_{i,t} = \pi^{il}_{\kappa}(o_{i,t}, g_{i,t})$ is the action of agent-i under the imitation learning policy. $A_{cr} = \max(A^{t}_{p}, \mathrm{sig}(\eta(o_{i,t}, o_{i,t}) - 1))$ is the expert control ratio, which is the maximum of the time t-th power of decay factor A_p and $\eta(o_{i,t}, g_{i,t})$. $\mathrm{sig}(\cdot)$ is the sigmoid function to bound its value within $(0, 1)$. The factor A_p is generally given a number slightly less than 1. This allows the expert policy to take over the training in the early stages so as to provide more demonstration data.

There are three advantages of RDSI-Dagger method: Firstly, when encountering high-threat scenarios, Eq.(10)(11) allows imitation learning to focus on learning the expert's policy as depicted by red lines in Fig. 4. Secondly, output action $a^{out}_{i,t}$ in Eq.(12) employs more safety expert policy in high-threat scenes, which guarantees

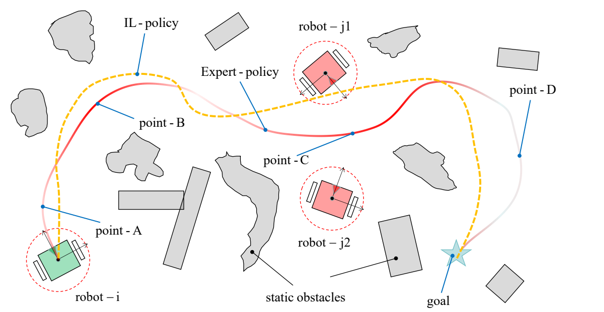


Fig. 4. Training scenes for RDSI-Dagger. The red and yellow lines represent trajectories generated by expert and IL policies, respectively. The depth of the red line for expert policy is evaluated by RDSI to assess the threat level of the current scenario. When there are neighboring static or dynamic obstacles (e.g., point-B,C), IL-policy focuses on learning complex expert strategies at high threat levels, as in Eq. (10)(11). Point-A,D, which have low threat level, do not require special attention by IL learning.

that the training is conducted safely. Third, Eq.(12) augments the training data to overcome the mismatch between the state distributions of IL and expert policies, which increases the efficiency of IL training.

(3) TRL training of collision avoidance module: Compared to pre-training + RL methods, TRL offers the advantage of mitigating catastrophic forgetting of expert policies and yielding better generalization performance [15]. Network ξ of the CA-module acquires a better collision avoidance policy similar to NH-ORCA under the homogeneity assumption through imitation learning. In the third stage of training, GP-module remains acting as a controller outputting the global action $g_{i,t}$ to CA-module. We design a transfer RL method (TRL) [15] for representation reuse to additionally train CA-modules under non-homogeneity assumption. It fixes parameters of the network ξ, which provides the TRL part with the action $a^{il}_{i,t} = \pi^{il}_{\xi}(o_{i,t}, g_{i,t})$. Then we train the TRL part to fine-tune it and finally obtain the action $a^{hsrln}_{i,t}$ of HSRLN, as shown by blue lines in Fig. 2(c).

In this way, network ξ of IL and networks σ, ϵ of TRL compose a progressive neural network [16], which helps to avoid catastrophic forgetting [15] of policy in network ξ. Theoretically, avoidance actions under non-homogeneity assumption are more cautious than avoidance actions under homogeneity assumption (have a narrower Voronoi area), and they are similar tasks [5], [10]. Despite it being unavailable to obtain a completely safe policy, TRL can be used to largely maintain the performance of collision avoidance under the homogeneity assumption in network ξ and effectively improve the navigation performance under the non-homogeneity assumption through fine-tuning. TRL also works better than letting the CA-module directly train the collision avoidance policy under non-homogeneity assumption without pre-training of IL. Based on the action $a^{il}_{i,t}$ of IL, the Q-value network σ In TRL is defined as:

$$Q^{\sigma}_{i}(o_{i,t}, a_{i,t} \mid a^{il}_{i,t}) = m(o^{in}_{i,t}, o^{env}_{i,t}, o^{vo}_{i,t}, a_{i,t} \mid a^{il}_{i,t} = \pi^{il}_{\xi}(o_{i,t}, g_{i,t})\,), \tag{13}$$

where $m(\cdot)$ is a two-layer multi-layer perceptron (MLP). The

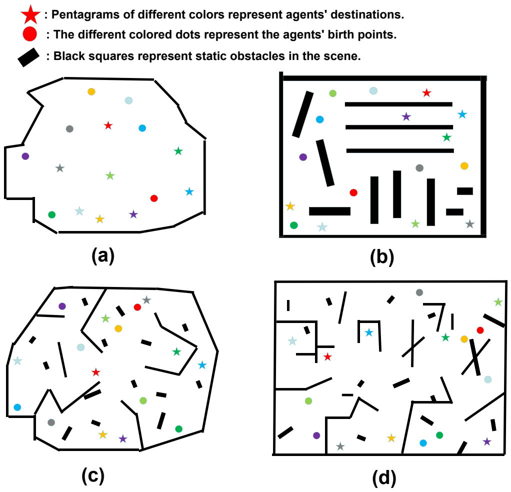


Fig. 5. (a)-(d) represent four scenes employed to train the robot clusters, incorporating corridors, static obstacles, and non-convex structures.

regression loss function of Q-value network σ is given as

$$
\begin{aligned}
L_Q(\sigma) =& E_{(o_{i,t}, a_{i,t}, r^{ca}_{i,t}, o_{i,t+1}, a^{il}_{i,t}) \sim R}\left[\frac{1}{2}\left(Q^{\sigma}_i(o_{i,t}, a_{i,t} \mid a^{il}_{i,t}) - y_{i,t}\right)^2\right], \\
& y_{i,t} = r^{ca}_{i,t} + \gamma E_{a_{i,t} \sim \pi_{\bar{\epsilon}}(o_{i,t+1}, a^{il}_{i,t+1})}[Q^{\bar{\sigma}}_i(o_{i,t+1}, a_{i,t+1} \mid a^{il}_{i,t+1}) \\
& - \alpha \log(\pi_{\bar{\epsilon}}(a_{i,t+1} | o_{i,t+1}, a^{il}_{i,t+1}))].
\end{aligned} \tag{14}
$$

Here, $Q^{\bar{\sigma}}_i$ also denotes another Q-network in SAC to solve the overestimation problem, and the reward $r^{ca}_{i,t}$ is similarly composed by Eq.(6)(7). The loss regression function of policy network ϵ in TRL is defined as follows:

$$
L_{\pi}(\epsilon) = E_{o_{i,t} \sim R, a_{i,t} \sim \pi_{\epsilon}}\left[\alpha \log(\pi_{\epsilon}(a_{i,t} | o_{i,t}, a^{il}_{i,t})) - Q^{\sigma}_i(o_{i,t}, a_{i,t} | a^{il}_{i,t})\right]. \tag{15}
$$

When all three phases of training have been completed, the policy network θ in the GP-module, the ξ network in the IL, and the policy network ϵ in the TRL (the latter two belonging to the CA-module) run in tandem. NH-ORCA is no longer involved in the operation. According to observation [$o^{in}_{i,t}$, $o^{env}_{i,t}$, $o^{vo}_{i,t}$], network θ gets the global planning $g_{i,t} = \pi^{gp}_{\theta}(o_{i,t})$, while network ξ further gets the action $a^{il}_{i,t} = \pi^{il}_{\xi}(o_{i,t}, g_{i,t})$. Finally, network ϵ outputs the final action $a^{hsrln}_{i,t} = \pi^{trl}_{\epsilon}(o_{i,t}, a^{il}_{i,t})$.

V. RESULTS

We assess the efficacy of seven mapless navigation algorithms, HDRL [3], CADRL [4], SRL-ORCA [3], HSRLN, HSRLN(without VO), HSRLN(RL+IL), HSRLN(RL+TRL). These methods share the same environment, realized in PyTorch, trained within Gazebo, and subsequently deployed onto a physical swarm comprising 6-8 robots for experimentation. We execute 1000 simulation trials for each algorithm and obtain results in **Table I**. Robots operate without pre-existing map data, relying solely on LiDAR sensing to acquire environmental observations denoted as $o^{env}_{i,t}$. Additionally, observations $o^{in}_{i,t}, \tilde{o}^{ro}_{i,t}$ can be sourced from Sensors. We set $v_t \in (0.01, 0.20)\, m/s$, $w_t \in (-2.5, 2.5)\, rad/s$, $t = 0.2s$. Hyper-parameters are chosen as $g = -40, h = -15, m = 80, q = 0.15m, z = -0.25$.

The training procedure spanned approximately 60 hours and was executed on a desktop equipped with an Intel Core i9-11900K CPU and an Nvidia GTX 3070Ti GPU. HSRLN is trained by 40,000 episodes of GP-module (RL), 20,000 episodes of imitation learning (IL), and 20,000 episodes of transfer reinforcement learning (TRL). To lower the arithmetic consumption in training, the NH-ORCA module [10] in both SRL-ORCA and HSRLN algorithms only turns on the dynamic collision avoidance function and turns off the static collision avoidance function. Fig. 5(a)-(d) represents four training scenarios for a cluster of robots. Half of the robots use RRT* [8] to simulate dynamic obstacles with non-homogeneity assumptions, and the other half are driven by the algorithm being tested. Fig. 6,7 depict two test scenarios (about 6.75 × 4.5 m), wherein the red, light blue, and green robots are controlled by the algorithm under evaluation, while the black, yellow, and blue robots are governed by RRT*. We employ three evaluation metrics:

1) Success rate ($\bar{\chi}$): The percentage of episodes in which agents successfully reach their target without collisions.
2) Average Time to Goal ($\bar{t}$): The mean duration taken by all successfully arrived robots across episodes. A lower time reflects more efficient navigation. If certain agents fail to reach their destination due to challenges such as non-convex obstacles, or if the success rate is below 80%, "average time" is marked as "fail" in **Table I**.
3) We compare the success rate $\bar{\chi}$ and average time $\bar{t}$ of these algorithms under homogeneity and non-homogeneity assumptions, respectively.

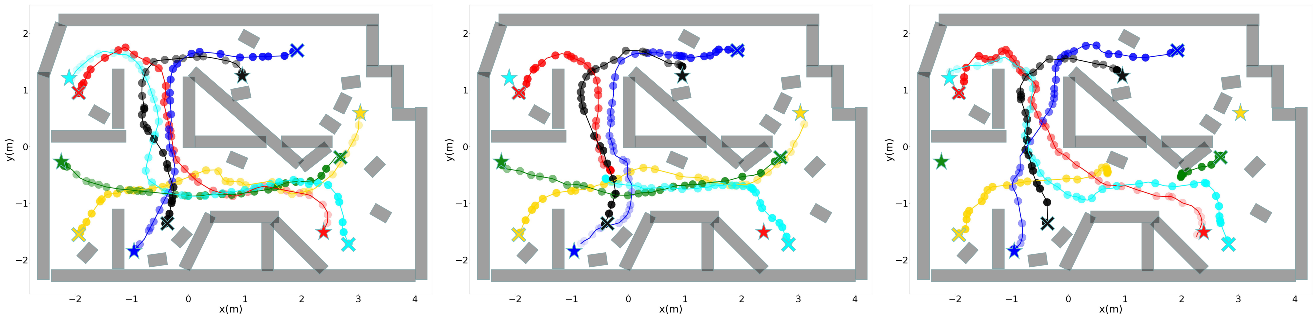
TABLE I

EXPERIMENTAL RESULTS FROM SCENARIO-1 TO SCENARIO-2

Number of robots	Scenario	Assumption	Method	Success Rate(%)	Col-with robots(%)	Col-with obstacles(%)	Time Out(%)	Rotate in Place(%)	Average Time(s)
6	1-Corridor at the intersection	Homogeneity	SRL-ORCA	94.2	0.0	1.5	2.3	2.0	45.30
			HDRL	84.8	9.2	3.2	2.0	0.8	48.54
			CADRL	85.1	4.6	8.0	1.8	0.5	51.07
			HSRLN	92.6	2.8	1.1	1.9	1.6	46.99
		Non-Homogeneity	SRL-ORCA	76.3	14.1	4.2	3.5	1.9	fail
			HDRL	81.0	10.2	4.4	3.7	0.7	50.63
			CADRL	78.6	11.7	6.7	2.0	1.0	fail
			HSRLN	88.4	4.5	3.3	1.4	2.4	47.21
			HSRLN (without VO)	84.5	6.6	5.0	3.0	0.9	48.92
			HSRLN (RL+IL)	72.5	16.7	5.1	3.2	2.5	fail
			HSRLN (RL+TRL)	68.9	15.9	6.6	5.8	2.8	fail
6-8	2-Complex non-convex scenarios	Homogeneity	SRL-ORCA	95.1	0.0	0.9	2.2	1.8	56.27
			HDRL	82.6	10.8	4.9	1.1	0.6	57.42
			CADRL	79.4	5.1	11.0	3.7	0.8	fail
			HSRLN	89.5	3.4	2.7	2.3	2.1	56.86
		Non-Homogeneity	SRL-ORCA	74.7	15.2	5.0	2.8	2.3	fail
			HDRL	81.2	11.0	4.3	2.5	1.0	59.36
			CADRL	73.7	8.5	11.6	5.1	1.1	fail
			HSRLN	85.3	6.3	4.8	2.7	0.9	57.20
			HSRLN (without VO)	80.8	6.8	6.0	3.9	2.5	59.17
			HSRLN (RL+IL)	74.1	15.9	4.5	3.3	2.2	fail
			HSRLN (RL+TRL)	73.2	17.4	5.3	2.0	2.1	fail

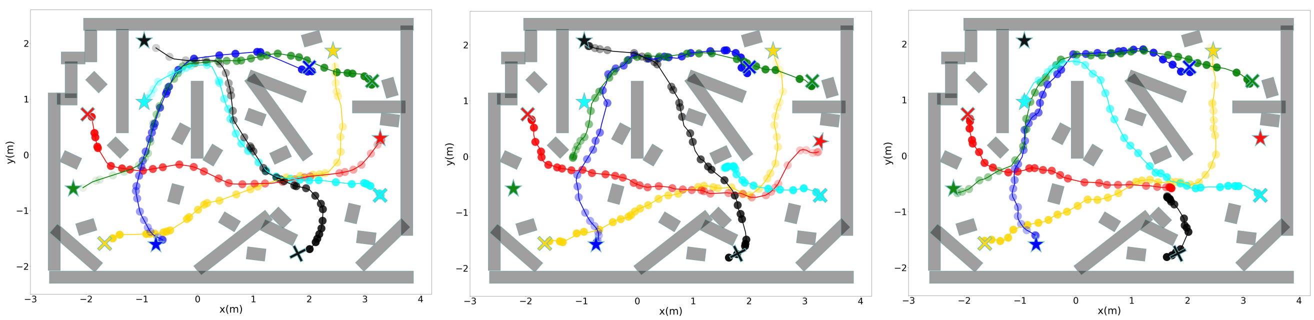
A. Comparative experiments in homogeneous and non-homogeneous scenarios

For Scenario-1,2, SRL-ORCA achieves the highest success rate under the homogeneity assumption ($\bar{\chi}^{srl-orca}$ = 94.2% and 95.1%), as shown in **Table I**. However, under the non-homogeneity condition, its avoidance performance slipped seriously ($\bar{\chi}^{srl-orca}$ = 94.2% to 76.3%, and 95.1% to 74.7% in Scenario-1,2). As for HSRLN, it is more adapted to the non-homogeneity scenario by the fine-tuning of the TRL part. It still maintains a better performance



(a) (b) (c)

Fig. 6. Scenario-1 is an intersection with many complex corridors. (a), (b), and (c) show the results of experimental trajectories for algorithms HSRLN, HDRL, and SRL-ORCA under non-homogeneous conditions, respectively. "X" denotes the starting point, and pentagram indicates the goal.



(a) (b) (c)

Fig. 7. Scenario-2 contains many non-convex obstacles. (a), (b), and (c) show the results of experimental trajectories for algorithms HSRLN, HDRL, and SRL-ORCA under non-homogeneous conditions, respectively.

and achieves the highest success rate among all algorithms ($\bar{\chi}^{hsrln}$ = 88.4% and 85.3%). For HDRL, it does not show a serious performance decline between homogeneous and non-homogeneous scenarios. However, since its collision avoidance module does not have expert knowledge to pre-train the policy, it is difficult to significantly improve the avoidance ability by simply using RL, and thus it cannot achieve a high success rate ($\bar{\chi}^{hdrl}$ = 81.0% and 81.2%). In summary, HSRLN fine-tunes the avoidance policy of IL-part in homogeneous scenarios by TRL, which allows it to achieve better performance in non-homogeneous scenarios as well. Compared to SRL-ORCA, it improves the navigation success rate under the non-homogeneity assumption by 12.1% and 10.6% in Scenario-1,2 respectively. In addition, HSRLN improves average time $\bar{t}$ by 7.2% and 3.8% compared to HDRL in the non-homogeneous Scenario-1,2.

B. Experimental results

Fig. 6,7 show experimental results of three methods, HSRLN, HDRL, and SRL-ORCA. For HDRL, the light blue robot and the red robot collide at the intersection as in Fig. 6b. This illustrates that HDRL cannot efficiently achieve safe dynamic collision avoidance even under the homogeneity assumption. For SRL-ORCA, when the light blue robot meets the yellow robot (RRT*), it obstructs the path of the yellow robot as in Fig. 6(c). Although the light blue robot slows down and resumes traveling, it interfers with the yellow robot's trajectory, resulting in a collision between the yellow one and the static obstacle. This indicates a deficiency in SRL-ORCA's ability to effectively coordinate collision avoidance under non-homogeneous conditions. For HSRLN, although it does not guarantee complete non-collision, Fig. 6(a) shows that it can better coordinate collision avoidance in non-homogeneous scenarios with smooth trajectories for each robot. Furthermore, **Table I** also demonstrates that the performance of HSRLN maintains better from homogeneous to non-homogeneous scenarios ($\bar{\chi}^{hsrln}$ = 92.6% to 88.4%, and 89.5% to 85.3% in Scenario-1,2), with no serious decline like that of SRL-ORCA.

For Scenario-2, it is used to evaluate the performance in complex non-convex scenarios. For HDRL, the light blue robot and the black robot (RRT*) travel in the same direction through a narrow intersection, the light blue one does not decelerate reasonably and collides with a static obstacle, as in Fig. 7(b). The green robot's global planning is inaccurate and it collides with a static obstacle when turning a corner. This suggests that when HDRL learns global planning, its strategy is susceptible to interference from dynamic obstacles, resulting in limited planning performance. For SRL-ORCA, the light blue robot and the black robot (RRT*) do not coordinate their avoidance actions at the intersection and collide with each other, as in Fig. 7(c). This similarly illustrates that the collision avoidance ability of SRL-ORCA declines dramatically in non-homogeneous scenarios. Fig. 7(a) shows that HSRLN has smoother and safer trajectories in non-convex complex scenarios. Our supplementary video also demonstrates that HSRLN has better performance in real scenarios. In summary, HSRLN exhibits better global planning performance than other hierarchical RL methods (HDRL) through the training of GP-module. HSRLN has better collision avoidance performance in non-homogeneous conditions relative to SRL-ORCA by training in both IL and TRL stages.

C. Ablation study

Table I shows the performance of four algorithms, HSRLN, HSRLN(without VO), HSRLN(RL+IL), and HSRLN(RL+TRL), in non-homogeneous Scenario-1,2. For the latter two, either their IL or TRL modules are trained for 40,000 episodes to ensure the same training time. First, the success rate of HSRLN(without VO) has a decline relative to HSRLN with the same number of training episodes (88.4% to 84.5%, and 85.3% to 80.8% in Scenario-1,2). This illustrates that VO module effectively extracts important features from raw observations and improves the efficiency of training. HSRLN(RL+IL) suffers a more severe decline in collision avoidance performance under non-homogeneous conditions (88.4% to 72.5%, and 85.3% to 74.1% in Scenario-1,2). The reason for this situation is that it only imitates the expert policy of NH-ORCA without generalized training in non-homogeneous scenarios. Since it lacks pre-training from expert policy, it is difficult for HSRLN(RL+TRL) to learn efficient avoidance policy directly through reinforcement learning, as in **Table I**. In summary, HSRLN shows the best navigation performance via three stages of learning. It obtains the highest success rate compared to HSRLN(without VO), HSRLN(RL+IL), and HSRLN(RL+TRL). Moreover, ablation experiments demonstrate that HSRLN exhibits graceful degradation, remaining operational even with certain modules missing.

VI. CONCLUSION

This paper proposes a hierarchical safe reinforcement learning approach for mapless navigation, namely HSRLN. It trains mapless navigation policies for non-homogeneous complex scenarios in a hierarchical manner through a kind of three-stage learning, global planning RL + expert imitation learning + transfer reinforcement learning. Experiments indicate that it has a better performance compared to existing hierarchical RL navigation methods (HDRL, SRL-ORCA). Relative to SRL-ORCA, it improves navigation success by 12.1% under the non-homogeneity assumption.

REFERENCES

[1] J. Wang, J. Wu, X. Zheng, D. Ni, and K. Li, "Driving safety field theory modeling and its application in pre-collision warning system," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 306–324, 2016.

[2] W. Zhu and M. Hayashibe, "A hierarchical deep reinforcement learning framework with high efficiency and generalization for fast and safe navigation," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 5, pp. 4962–4971, 2022.

[3] J. Qin, J. Qin, J. Qiu, Q. Liu, M. Li, and Q. Ma, "Srl-orca: A socially aware multi-agent mapless navigation algorithm in complex dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 143–150, 2024.

[4] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 856–892, 2020.

[5] T. Fraichard and V. Levesy, "From crowd simulation to robot navigation in crowds," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 729–735, 2020.

[6] Z. Xie and P. Dames, "Drl-vo: Learning to navigate through crowded dynamic scenes using velocity obstacles," *IEEE Transactions on Robotics*, 2023.

[7] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *in Proceedings of the AAAI conference on artificial intelligence,*, vol. 1001, no. 48105, pp. 18–80, 2008.

[8] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[9] J. v. d. Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics research*. Springer, 2011, pp. 3–19.

[10] J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Distributed Autonomous Robotic Systems*. Springer, 2013, pp. 203–216.

[11] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1527–1533.

[12] C. Yan, J. Qin, Q. Liu, Q. Ma, and Y. Kang, "Mapless navigation with safety-enhanced imitation learning," *IEEE Transactions on Industrial Electronics*, 2022.

[13] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 285–292.

[14] O. Nachum, H. Tang, X. Lu, S. Gu, H. Lee, and S. Levine, "Why does hierarchy (sometimes) work so well in reinforcement learning?" *arXiv preprint arXiv:1909.10618*, 2019.

[15] Z. Zhu, K. Lin, A. K. Jain, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[16] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[17] Z. Zhou, Z. Zeng, L. Lang, W. Yao, H. Lu, Z. Zheng, and Z. Zhou, "Navigating robots in dynamic environment with deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 201–25 211, 2022.

[18] S. Pateria, B. Subagdja, A.-h. Tan, and C. Quek, "Hierarchical reinforcement learning: A comprehensive survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–35, 2021.

[19] P.-L. Bacon, J. Harb, and D. Precup, "The option-critic architecture," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[20] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu, "Feudal networks for hierarchical reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3540–3549.

[21] A. Levy, G. Konidaris, R. Platt, and K. Saenko, "Learning multi-level hierarchies with hindsight," *arXiv preprint arXiv:1712.00948*, 2017.

[22] Y. Gao, J. Wu, X. Yang, and Z. Ji, "Efficient hierarchical reinforcement learning for mapless navigation with predictive neighbouring space scoring," *IEEE Transactions on Automation Science and Engineering*, 2023.

[23] W. Wang, L. Wang, C. Zhang, C. Liu, L. Sun *et al.*, "Social interactions for autonomous driving: A review and perspectives," *Foundations and Trends® in Robotics*, vol. 10, no. 3-4, pp. 198–376, 2022.

[24] 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.