Adversarial Reinforcement Learning for Enhanced Decision-Making of Evacuation Guidance Robots in Intelligent Fire Scenarios

Hantao Zhao, Zhihao Liang, Tianxing Ma, Xiaomeng Shi, Mubbasir Kapadia, Tyler Thrash, Christoph Hoelscher, Jinyuan Jia, Bo Liu, and Jiuxin Cao

***Abstract*—In the context of rapid urbanization, traditional manual guidance and static evacuation signs are increasingly inadequate for addressing complex and dynamic emergencies. This study proposes an innovative emergency evacuation framework that optimizes the crowd evacuation by integrating multiagent reinforcement learning (MARL) with adversarial reinforcement learning (ARL). The developed simulation environment models realistic human behavior in complex buildings and incorporates robotic navigation and intelligent path planning. A novel simulated human behavior model was integrated, capable of complex human–robot interaction, independent escape route searching, and exhibiting herd mentality and memory mechanisms. We also proposed a multiagent framework that combines MARL and ARL to enhance overall evacuation efficiency and robustness. Additionally, we developed a new ARL evaluation framework that provides a novel method for quantifying agents' performance. Various experiments of differing difficulty levels were conducted, and the results demonstrate that the proposed framework exhibits advantages in emergency evacuation scenarios. Specifically, our ARLR approach increased survival rates by 1.8% points in low-difficulty evacuation tasks compared to the RLR approach using only MARL algorithms. In high-difficulty evacuation tasks, the ARLR approach raised survival rates from 46.7% without robots to 64.4%, exceeding the RLR approach by 1.7% points. This study aims to enhance the efficiency and safety of human–robot collaborative fire evacuations and provides theoretical support for evaluating and improving the performance and robustness of ARL agents.**

***Index Terms*—Adversarial reinforcement learning (ARL), human–robot interaction, multiagent reinforcement learning (MARL), simulation frameworks.**

I. Introduction

THE accelerated urbanization process has led to a significant increase in the density and complexity of public buildings, posing severe challenges to the effectiveness of traditional evacuation methods during emergencies like fires, such as manual guidance and static signs [1], [2]. This, in turn, constitutes a major threat to public safety. In response, researchers are actively developing new technologies and methods that integrate human behavior models [3], simulation tools [4], [5], and optimization algorithms [6], [7]—such as adaptive signage systems and evacuation-assistance robots [8]—to enhance evacuation efficiency and safety. These innovative approaches aim to meet the urgent evacuation demands in complex environments and provide scientific support for improving public safety levels [9], [10].

Accurate guide information and the visibility of emergency evacuation signs are key factors for optimizing crowd evacuation efficiency [11], [12]. Studies have shown that dynamic signage systems [13], [14] and intelligent interactive human–robot collaborative evacuation systems [15], [16], such as evacuation robots utilizing sensors and map data, can effectively guide crowds to safety during emergencies [17], [18]. Moreover, the behavior of robots within the crowd has been proven to significantly enhance overall evacuation performance [19].

Despite some progress, numerous challenges remain in practical application, such as ethical constraints and

Received 29 April 2024; revised 22 August 2024 and 10 October 2024; accepted 12 November 2024. This work was supported in part by the Ministry of Education in China (MOE) Project of Humanities and Social Sciences under Project 24YJCZH436 and in part by the National Natural Science Foundation of China under Grant 62472092, Grant 62172089, Grant 5243000365, and Grant 71971056. *(Corresponding author: Hantao Zhao.)*

Hantao Zhao and Jiuxin Cao are with the School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China, and also with Purple Mountain Laboratories, Nanjing 211111, China (e-mail: htzhao@seu.edu.cn; jx.cao@seu.edu.cn).

Zhihao Liang is with the School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China (e-mail: zhliang@seu.edu.cn).

Tianxing Ma is with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China (e-mail: matianxing@iie.ac.cn).

Xiaomeng Shi is with the School of Transportation, Southeast University, Nanjing 211189, China (e-mail: shixiaomeng@seu.edu.cn).

Mubbasir Kapadia is with the Department of Computer Science, Rutgers University, Newark, NJ 07102 USA (e-mail: mk1353@cs.rutgers.edu).

Tyler Thrash is with the Department of Biology, Saint Louis University, St. Louis, MO 63103 USA (e-mail: tyler.thrash@slu.edu).

Christoph Hoelscher is with the Chair of Cognitive Science, ETH Zurich, 8092 Zurich, Switzerland (e-mail: choelsch@ethz.ch).

Jinyuan Jia is with The Hong Kong University of Science and Technology, Guangzhou 511453, China, and also with the Game School, Jilin Animation Institute, Changchun 130012, China (e-mail: jinyuanjia@hkust-gz.edu.cn).

Bo Liu is with Purple Mountain Laboratories, Nanjing 211111, China, and also with the School of Computer Science and Engineering, Southeast University, Nanjing 211189, China (e-mail: bliu@seu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TCSS.2024.3502420, provided by the authors.

Digital Object Identifier 10.1109/TCSS.2024.3502420

experimental limitations. The adaptability of traditional robotic algorithms in complex dynamic environments still needs improvement [20], [21]. This involves complex environment simulation, robot navigation, intelligent path planning, and balancing human–robot collaborative evacuation strategies. To overcome these obstacles, researchers have turned to novel digital methods such as computer simulation and robotics to create realistic environments for in-depth study of related issues [22], [23], [24], [25]. Studies have also introduced reinforcement learning (RL) and adaptive evacuation route optimization systems, attempting to address these challenges using reward and punishment mechanisms [26], [27].

Existing research primarily focuses on developing efficient and adaptive evacuation frameworks using RL techniques to simulate and optimize crowd evacuation in emergencies such as fires. However, this approach still faces the following challenges: 1) the unpredictability and dynamic nature of real-world scenarios present significant challenges to traditional RL algorithms, especially when being aligned with the need for real-time decision-making; 2) traditional RL methods lack robustness and are prone to overfitting; and 3) integrated systems of multiagent reinforcement learning (MARL) and adversarial reinforcement learning (ARL) suffer from insufficient perturbation optimization, unstable ARL training environments, and a lack of effective evaluation methods.

To address these issues, this study proposes an ARL framework for enhanced decision-making of evacuation guidance robots in intelligent fire scenarios. First, a transferable simulation environment and human behavior model were constructed. Multiple reusable prefabs ensure the variability and scalability of the simulation environment. A microscopic human behavior model was employed, allowing each individual to independently search for escape routes with a memory mechanism. A comprehensive framework combining MARL and ARL was proposed to enhance agent decision-making efficiency and robustness in complex, dynamic, and uncertain environments. Specifically, MARL employs a multiagent algorithm with experience sharing to achieve data interoperability and distributed execution for observations and actions. Additionally, a hybrid game-theoretic adversarial agent was introduced, reducing the training impact and mitigating overfitting risks. In this framework, the adversarial agent acts as a generator responsible for constructing challenging environments, while the evacuation agent group serves as the solver to obtain higher rewards in the generated environments. Both sides compete and improve, enabling the evacuation agent group to learn more from random scenarios and prevent overfitting. Finally, based on ARL characteristics, environmental difficulty was organically combined with rewards to propose new evaluation criteria that quantify performance in ARL scenarios, with multiple experiments conducted to validate their effectiveness. This comprehensive approach aims to overcome the limitations of existing technologies and improve the adaptability and efficiency of multiagent evacuation systems in complex dynamic environments.

Our novelty and key contributions are summarized as follows.

1) Developed a novel human behavior model capable of replicating the ability of real humans to observe and make decisions in complex buildings without prior knowledge. This model not only interacts with the surrounding environment and other simulated humans but also receives and executes robot commands.
2) Proposed a new multiagent framework that integrates MARL and ARL methods. By using distributed agents, overall efficiency is improved, and adversarial training enhances system robustness and generalization capabilities.
3) Introduced a general ARL evaluation framework that addresses the significant impact of adversarial perturbations on deep reinforcement learning (DRL) agents, particularly regarding changes in observational input during training. This helps to accurately assess the effectiveness of ARL algorithms.

Our framework can improve evacuation efficiency and safety in multiagent human–robot collaborative scenarios and provide theoretical tools for evaluating and enhancing the performance and robustness of ARL algorithms. The rest of this article is organized as follows. Section II reviews related work; Section III systematically describes the model and its construction; Section IV covers the framework training process and relevant parameters; Section V presents the experimental design and data analysis; and Section VI summarizes the main findings and outlines future research directions.

II. Prior Work

Through this literature review, we present the development trend of intelligent evacuation systems, from simple to complex and from single-agent to multiagent systems, while highlighting the critical role of simulation and adversarial training in optimizing decision-making and improving adaptability. This section is divided into three sections, each based on the relevant research. Section II-A introduces the application of RL in the field of robotics, focusing on optimizing behavior strategies through interaction with the environment. Section II-B discusses research on simulating crowd evacuation in emergencies using computer simulation techniques and human behavior models, allowing for the testing and study of evacuation strategies without directly involving real environments. Section II-C describes the literature on the application of adversarial training in multiagent systems, with particular emphasis on how adversarial environments enhance agents' adaptability and decision-making diversity.

A. RL Methods for Robots

RL is a branch of machine learning that differs from traditional supervised learning, which relies on large amounts of prelabeled data. Instead, RL involves an agent gradually developing strategies through a process of trial and error in simulated or real environments. In this process, the agent observes the state of the environment and takes actions based on these observations [28]. In RL, the environment reacts to the agent's actions and provides rewards or penalties, and the agent adjusts its behavior based on this feedback. The goal of the agent is to learn how to choose the optimal actions in a given state to maximize long-term cumulative rewards [29]. Researchers have explored the application of RL algorithms to enhance the decision-making capabilities of robots, aiming to improve operational efficiency and effectiveness [30], [31]. The research primarily focuses on simulating virtual environments and using DRL algorithms to train robot behaviors, thereby improving task performance [32], [33], [34]. Additionally, researchers have developed an end-to-end learning framework that combines attention with value network training, enabling robots to learn optimal actions in real time [35]. This study extends RL to the multiagent domain and introduces adversarial agents to enhance overall performance.

B. Simulated Evacuation Studies

Due to the hazardous nature of emergency events, conducting evacuation experiments in reality is impractical. Researchers often construct virtual evacuation environments using computer technology and human behavior models to simulate emergency crowd evacuation [23]. These environments incorporate realistic architectural models, precise simulations of environmental dynamics, and detailed simulations of crowd behavior [24].

Current human behavior models are divided into macroscopic and microscopic categories. Macroscopic models offer a simpler understanding of human movement, viewing crowd behavior as fluid motion within environments, and providing high-computational efficiency [36] but overlooking individual interactions. Microscopic models treat each human as an independent entity, with popular models including the cellular automata model [37] and the social force model [38]. The cellular automata model describes the evacuation environment as discrete cells, with humans occupying cells and deciding their actions based on surrounding cells' states. The social force model accounts for various subjective psychological phenomena in humans, with each individual influenced by forces encapsulating movement, avoidance, and following behaviors. We applied a microscopic human behavior model, treating each human as an independent entity, integrating concepts from cellular automata and social force models. The designed human model is limited by vision and lacks prior knowledge of the building structure.

Simulated environments consider factors such as building structures, exit layouts, environmental changes, and crowd dynamics to provide a realistic simulation environment. This allows researchers and emergency teams to study and train intelligent decision-making programs safely [39]. Huang et al. proposed an evacuation model based on information transfer and rerouting, effectively guiding the process of making evacuation plans for metro station halls [25]. However, these frameworks may face challenges in dynamic or unpredictable environments. Some studies have incorporated RL approaches, such as adaptive evacuation route optimization systems [26], which guide agents to learn the best strategies through rewards and punishments but increase complexity and training difficulty. To address these challenges, this study introduces MARL and ARL to construct an adaptive, transferable, and practical evacuation simulation framework. Utilizing the Unity engine [40], the framework autonomously develops human behavior models aligned with real-world scenarios, providing stable and effective intelligent decision-making solutions and contributing to intelligent decision-making technology in simulated environments.

C. Adversarial Training in Multiagent Systems

The development of MARL has provided new perspectives for agent decision-making in complex interactive environments. Traditional RL algorithms, such as proximal policy optimization (PPO) [41] and soft actor–critic (SAC) [42], often assume that agents in the environment are singular or independent of each other. However, this assumption is no longer applicable in multiagent environments where agents' behaviors affect the state of the environment, and each agent has its own strategy and value network, making the state-value function of the entire environment a function of all agents' network parameters. The interaction between agents manifests as complex game relationships, which, according to research by Zhang et al. [43], can be cooperative, competitive, or a mix of both.

Within the MARL framework, this study improves upon the multiagent posthumous credit assignment (MA-POCA) algorithm [44], adopting a centralized training with a decentralized execution mode to enhance the adaptability and decision efficiency of agents in dynamic environments. Moreover, ARL, as a special form of RL, enhances the robustness and strategy diversity of agents by training them in environments that include adversaries or competitors. The essence of ARL lies in improving their performance and adaptability through competition between agents, making this approach particularly effective in games, simulated environments, or any scenarios requiring strategic decision-making.

The purpose of adversarial training is to enhance the overall capabilities and adaptive robustness of agents. For instance, Wu et al. [45] have used adversarial training to improve the grasping capabilities of robotic arms on moving objects. This study draws inspiration from the "generator–solver–evaluator" structure proposed by Bontrager and Togelius [46], modeling ARL tasks in a "generator–solver" mode and integrating the "evaluator" within the environment to assess learning outcomes. This structure allows agents to compete and improve by generating challenging environments and seeking high rewards within them.

However, optimizing and evaluating ARL faces challenges such as the instability of training environments and the effects of adversarial perturbations. Research by Pinto et al. [47] has highlighted the possibility of enhancing agent robustness in adversarial environments while also emphasizing the challenges of environmental instability. Therefore, this study aims to establish an evaluation metric to assess the actual effects of training and hopes to generalize it to all similar situations. Through carefully designed training and evaluation strategies, this study seeks to effectively enhance agent performance and monitor training outcomes, even in complex and uncertain ARL environments.

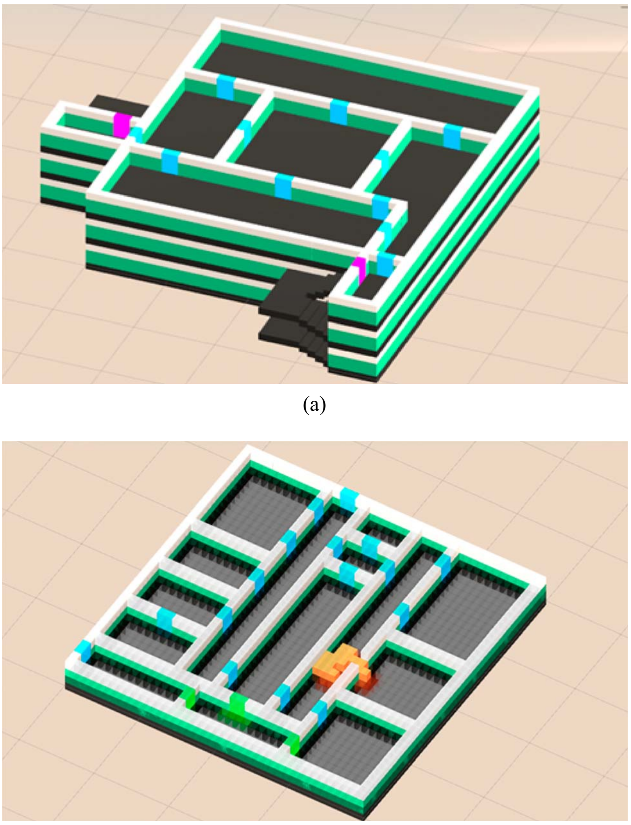
III. Methodology

The proposed method addresses the shortcomings of traditional evacuation robots, such as constrained decision-making capacity, poor robustness, and inability to handle complex environments and emergencies. To tackle these issues, we employed a RL mechanism to enhance the robots' autonomous guidance ability and robustness. Additionally, we introduced ARL to improve their decision-making capability in complex fire environments. The Unity engine [40] was used to construct a virtual environment for studying building fire scenarios, which utilizes the NavMesh system [48] for controlling human and robot movement. The machine learning (ML)-Agents Toolkit [49] was employed for environment construction, enabling precise and efficient simulation and emulation. The goal is to develop an evacuation robot control algorithm that can plan routes, issue instructions, and collaboratively guide humans to evacuate quickly in complex, rapidly changing environments such as multistorybuilding fires. Based on this, we established an independently developed ARL framework within the environment and improved upon potential issues.

A. Simulation Environment

Our simulation environment consists of three stories, each comprising several rooms, with two independent staircases connecting the floors. In the event of a fire, each floor has its own evacuation robot guiding humans to evacuate, with each robot responsible solely for the evacuation tasks on its floor. Robots on different floors can collaborate to some extent. If a human can pass through the "exit," which is located on the first floor of the environment, it is considered to be a successful escape. The specific simulation environment is depicted in Fig. 1(a).

In our study, the layout of the three-story building was specifically designed to test the robustness of evacuation robots. Each floor presents varying levels of complexity, with the most complex layout on the first floor to increase the difficulty of navigation tasks. As the floor level ascends, the number and complexity of rooms decrease. With limited visibility in fire scenarios, humans on higher floors face the challenge of navigating in open environments with reduced visual cues, while those on lower floors encounter the issue of getting lost due to the complexity of the surroundings. This configuration is not intended to represent typical building designs but is instead an intentional configuration to evaluate how well our algorithms handle various challenges. Specific overhead views of each floor's layout can be found in Appendix A.



(a)

(b)

Fig. 1. Schematic diagram of the virtual environment constructed in this study. (a) Oblique overhead view of the virtual environment of the building. (b) Grid sensor deployed on the first floor, marking the grids near walls, flames, and exits.

B. Human Behavior Model

To meet the requirements of the RL training environment, this study designed a simulated human behavior model with limited vision and no prior knowledge of the building structure. In this model, humans independently search for escape routes under emergency conditions, with their ultimate goal being the exit. Human memory was simulated through a finite-length memory queue to recall previously traversed paths. When an unexplored direction appears within their field of view, the modeled humans attempt to "explore." Additionally, to more accurately simulate real crowd behavior, this study incorporates the herd effect by introducing the concept of teams. Individuals seek and follow suitable leaders within their visual range, such as evacuation robots and robot-led evacuation teams, thus becoming part of a treelike team structure. Humans, as part of the team, unconditionally trust and follow the guidance of the robot.

Within the Unity engine, the perception of the environmental state is implemented using the RayCast method, which functions similar to the perception methods in the cellular automata model. Furthermore, by leveraging the NavMesh navigation system, basic behaviors such as movement, avoidance, and following from the social force model [38] are simulated. This setup ensures that humans in the virtual environment can autonomously navigate within a fire-stricken building, and their behavior patterns are consistent with real human actions.

The human behavior model is designed with two modes: "explorer" and "follower." The switching between these two modes is based on the surrounding environmental conditions. The model determines whether to join a group or act independently by considering the distance and line of sight between itself and other entities (humans or robots). Through this mechanism, we ensure that the ratio of "explorers" to "followers" dynamically adjusts, which, to some extent,

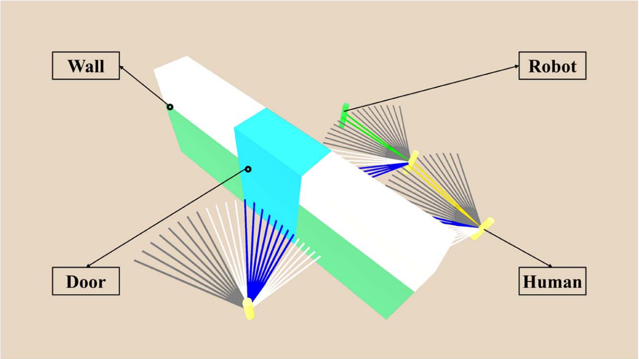


Fig. 2. Schematic diagram of the human–robot interaction mechanism, where the green capsule represents a robot, the yellow capsule represents a human, the green and white squares represent walls (obstacles), and the blue square represents an accessible regular door.

simulates the impact of crowd density in real-life scenarios. Additionally, we focus on human–robot interaction, with a key feature of our human behavior model being the efficient reception and execution of robot commands. When following an evacuation robot, the model takes into account the instructions given by the robot. If a target door that matches the robot's instructions appears within its field of vision ("upstairs" corresponds to an upward staircase; "downstairs" corresponds to a downward staircase; and if no specific command is given, it corresponds to an exit), the human would voluntarily disengage from the robot's leadership, switch its behavior mode back to "Explorer," and move toward the target door. It is important to note that each individual receives instructions directly from the robot, ensuring that even in conditions of high- or low-crowd density, the robot's guidance is effectively communicated to everyone.

A schematic of this operation is shown in Fig. 2. We assumed a human field of vision of 120°, using the RayCast method to observe the surrounding environment and make decisions. In this schematic, gray lines represent unknown areas, blue represents observed doors, white represents observed obstacles, green represents observed robots, and yellow represents other humans.

Specifically, to enhance the simulation of fire scenarios, we incorporated the detrimental effects of toxic fumes on human health during fire emergencies. To simulate smoke exposure in a fire, each simulated individual is assigned a health counter with an initial value of 100. For every frame that a human remains in the fire environment, their health decreases by 0.01 (0.5 health per second). Once a human exits through an escape, their health no longer decreases. If a human's health drops to zero or they come into contact with flames, they are considered dead. The game ends when all humans have either evacuated or died. The total duration does not exceed 5 min, which closely aligns with the expected survival time of humans in real-life situations [50].

Through the methods described, this study has constructed an individualized and realistic virtual human behavior model. This not only provides an effective tool for researching emergency evacuation behavior but also offers important theoretical references for the design and application of intelligent evacuation robots. For additional information on human behavior models, please refer to Appendix B.

C. Evacuation Robot Model

In our simulated environment, we deployed one evacuation robot on each floor, restricting its movement to the main areas and stairwells of that specific floor, without the ability to traverse between floors. However, the robot can guide humans by issuing instructions to ascend or descend stairs. The individual goal of each floor's robot was to minimize errors and expedite the evacuation of all occupants from that floor, while the collective objective of all robots was to ensure the safe evacuation of all individuals within the building. We based the decision to deploy one robot per floor on several key considerations. First, our test-run experiments demonstrated that multiple robots working collaboratively in a single-floor environment significantly enhance evacuation efficiency compared to a single robot operating independently. This finding underscored the importance of leveraging multirobot collaboration in complex environments. Second, recognizing that real-world buildings are often multistory, we designed the system to equip each floor with at least one robot to maximize evacuation efficiency while optimizing resource allocation. In a multifloor building, the robots were not limited to simply handing off tasks at stairwells; they could dynamically adjust their positions to respond to needs on other floors. This cross-floor collaboration is particularly crucial in addressing scenarios where certain exits or pathways might be blocked due to emergencies, thereby ensuring a more efficient and adaptive evacuation process.

D. Collaborative Multirobot Agents for Evacuation

In this study, we employed the MA-POCA algorithm to drive the agent group. MA-POCA integrated a framework for centralized training with decentralized execution and a counterfactual baseline based on the self-attention mechanism, effectively rewarding agents that terminate early. It utilized self-attention to learn a centralized baseline value function

$$Q_\psi(\text{RSA}(g_j(o_j^t), f_i(o_i^t, a_i^t)_{1\leq i\leq k^t, i\neq j})) \qquad (1)$$

here g_j and f_i encode individual observations and observation-action pairs, respectively, while the residual self-attention (RSA) module performs self-attention computation, and Q_ψ represents the learned baseline action-value function. This structure can learn the contribution baseline of each agent to the team reward based on individual information and efficiently handle a variable number of agents without requiring fixed-sized inputs. As shown in Fig. 3, our model uses three 40 × 40 grid sensors to observe the static elements on each floor, namely walls, flames, and exits, with inputs taken every two frames. Compared to traditional vector observations, grid sensors significantly reduce the complexity of the RL network at the cost of some precision. After obtaining the grid observations, the process is similar to feature extraction in images, where the grid

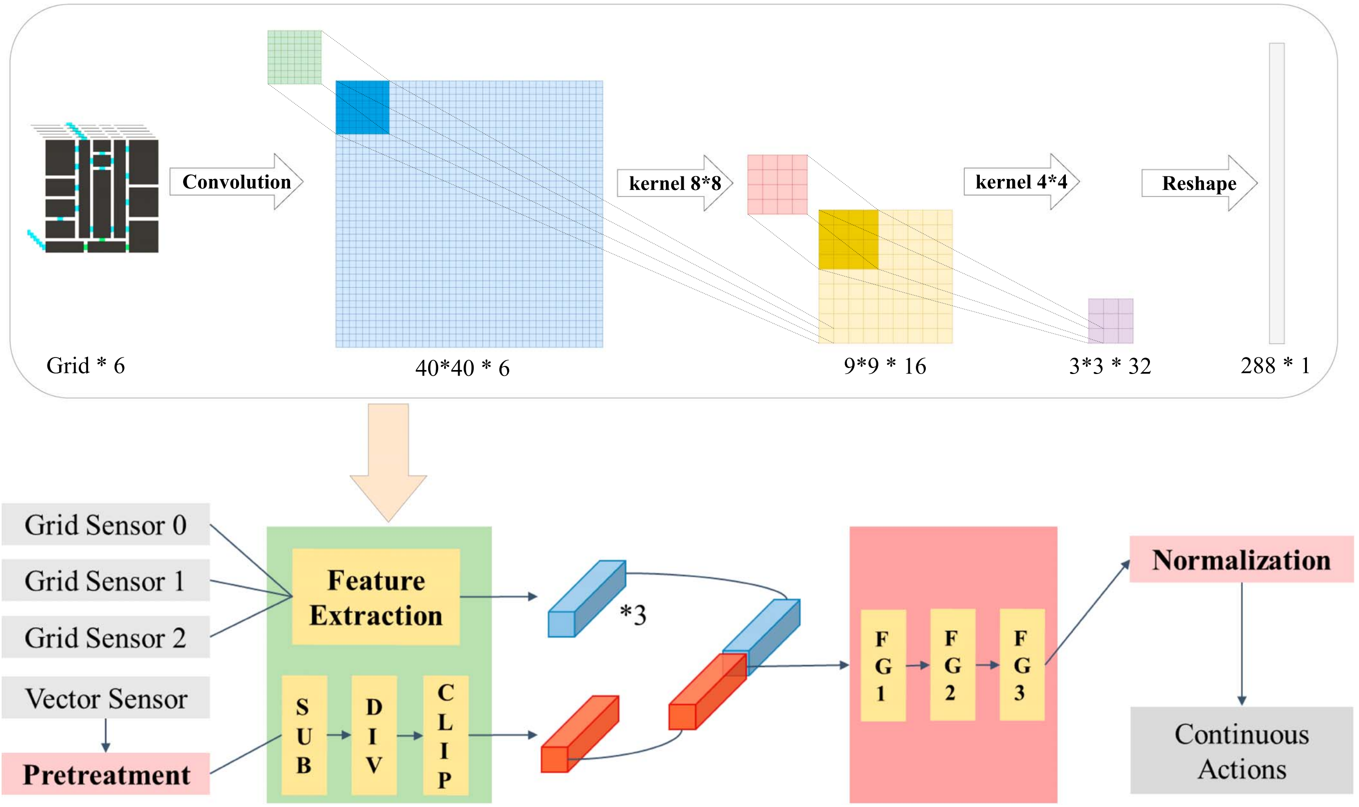


Fig. 3. Overall model structure of our evacuation robot group would be described in this figure. First, convolution and feature extraction are performed on the grid observation inputs, then concatenated with the obtained vector observation inputs. This is followed by processing through a forgetting mechanism, ultimately resulting in a set of actions. Here, *FG* represents the forget gate, mainly used to control which information from the previous time step needs to be forgotten or retained.

observations of each floor are individually processed through convolution to ultimately form three feature vectors. For dynamic elements, traditional vector observations are used, and all observation data are concatenated and passed through a forget gate to create memory. Finally, after normalization, two continuous actions representing the robot's destination coordinates are produced.

In RL, an agent receives an observation from the environment at each timestep and selects an action based on this observation. For our model, a grid sensor observes each floor's situation and performs feature extraction to form a feature vector. This vector is then concatenated with vector observation data to determine an action. The main elements of our RL agent system include the observation space, action space, and reward function.

1) Observations: In the proposed framework, an observation represents the current state of the environment and is provided as input to the agent. Observations can include various types of data, such as visual images, numerical values, and discrete categories, depending on the nature of the environment. In this article, agents on each floor have identical observations to maximize the sharing of various information within the shared environment. Evacuation robot agents have two types of observations: vector observations and grid sensor observations.

Vector observations are the conventional observation method for RL agents, directly using each observation value as input to the agent's neural network. The vector observations implemented for the evacuation robots include the 3-D coordinates of all humans in the environment, the 3-D coordinates of all evacuation robots, the size of the evacuation robot team on the floor, and the elapsed time since the start of the evacuation. Considering the lethal nature of fire, we also incorporated the 3-D coordinates of all fire locations and the specific number of fires on each floor into the vector observations.

Grid sensor observations are a novel observation method. This method divides a specified planar area into a grid of specified dimensions and detects whether each grid contains entities of a specified type. In this article, each evacuation robot agent uses a grid sensor to observe walls, flames, and exits across the three floors, as shown in Fig. 1(b). The grid sensor encodes all static elements that need to be observed into an image format, which serves as input to a convolutional neural network for feature extraction. Compared to vector observations, grid sensor observations significantly reduce the complexity of the RL environment at the cost of losing some precision.

2) Action Space: To simplify the model, each evacuation robot's action decision consists of only two floating-point numbers, representing the coordinates (X_r, Y_r) of its planned next horizontal position. However, since the stairwells on the north and south sides are located outside the main building structure, there is a certain adjustment between the robot's actual movement target and the decision outcome

$$(X_r', Y_r') = \begin{cases} (X_a, Y_a), & \|(X_r, Y_r) - (X_a, Y_a)\| < 20 \\ (X_b, Y_b), & \|(X_r, Y_r) - (X_b, Y_b)\| < 20 \\ (X_r, Y_r), & \text{Others} \end{cases} \quad (2)$$

where (X_a, Y_a) and (X_b, Y_b) represent the center positions of the stairwells on the south and north sides of the floor, respectively, and (X_r', Y_r') represents the actual movement target position of the evacuation robot. This means that when an evacuation robot sets its destination near a stairwell, it automatically proceeds to the interior of the stairwell.

3) Reward Function: To iteratively train the RL model, the agent updates its neural network based on behaviors and receives rewards. The group rewards and individual rewards are set up separately. In this section, where health represents the value of human life, rate represents the rate at which human life decreases due to toxic fumes (0.01), humanAmount is the total number of evacuees on this floor, and Follower is the number of followers. In the context of RL, agents update their neural network parameters based on the rewards received from their executed actions. In this study, we have assigned three evacuation robots to the same MA-POCA algorithm group [44], with both collective and individual reward mechanisms designed for them. It is important to emphasize that our research focuses primarily on collective rewards, and all subsequent experiments and analyses are based on the collective rewards of the MA-POCA agent group.

However, in methods based on DRL, the randomness of exploration in unstructured environments poses a significant problem. This approach often leads to inefficiency and lack of robustness in trajectory planning tasks. To address these limitations, we propose the use of a novel dense reward function to replace the conventional sparse reward function. Dense reward functions provide more feedback information after each action, although they are more challenging to design. Traditional sparse reward functions are zero in most cases, except in a few specific situations [51]. Such reward functions often lead to a large amount of ineffective exploration, significantly reducing algorithm efficiency [52], [53], [54]. To tackle this problem, we introduced an optimized reward function method for robots.

For our training method 1, considering its rationality, we named it adversarial reinforcement learning robot, basic (ARLR-B). The collective reward setting of ARLR-B is presented as follows:

$$\text{Group Reward} \begin{cases} +\text{health} + 300, & \text{Human evacuation} \\ -200, & \text{Human fatality.} \end{cases} \quad (3)$$

This group reward represents the shared objective of the three evacuation robots: to evacuate all humans as quickly as possible and prevent any human casualties in the environment. For the "human evacuation" task, whenever a human successfully exits the environment, the group reward for that round is increased by the remaining health value of the evacuated human plus a fixed reward of 200. The shorter the time a human remains in the fire environment, the higher their remaining health, and consequently, the greater the reward given to the robot agent group. However, the survival of humans is of utmost importance, so we included a fixed reward value in this task. For the "human fatality" task, to teach the evacuation robots to prioritize rescuing humans in more dangerous situations, any human fatality is treated as negative feedback, resulting in a group penalty of −200. The individual reward is determined based on the behavior of each evacuation robot on its respective floor, as shown in (4) at the bottom of the page.

For the "reward per frame" task, when there are still humans on a floor who have not yet evacuated, the evacuation robot responsible for that floor receives a certain negative reward over time. The more humans that remain on the floor, the higher the penalty per frame. This individual reward function is designed to encourage robots to "clear" the floor of humans as quickly as possible, thereby reducing evacuation time. Regarding the "robot encounters fire" task, the robot should recognize the danger of approaching flames, as in our setting, human contact with fire leads to death. We used individual rewards as a supplement to the group rewards, helping the evacuation robots on each floor better learn how to evacuate humans as quickly as possible.

RL models often struggle to maintain robustness in dynamic environments [55], which poses challenges for these models

$$\text{Individual Reward} \begin{cases} -rate \times \text{HumanAmount}, & \text{Reward per frame} \\ -100, & \text{Robot encounters fire} \end{cases} \quad (4)$$

$$\text{Group Reward} \begin{cases} +\text{health} + 200, & \text{Human evacuation} \\ -300, & \text{Human fatality} \\ +\text{health}, & \text{Successful human navigation} \end{cases} \quad (5)$$

$$\text{Individual Reward} \begin{cases} -\text{rate} * \text{humanAmount}, & \text{Reward per frame} \\ -20, & \text{Robot destroyed by fire} \\ +10 * \text{health} * \text{Follower}, & \text{Successful human navigation} \\ -100, & \text{Robot encounters fire} \end{cases} \quad (6)$$

to adapt to rapidly evolving and complex fire environments. To address this issue, we optimized the reward mechanism mentioned earlier to enhance model training in such complex dynamic environments. This optimization enables the models to better handle dynamic evacuation tasks, thereby improving the agent's overall performance. To facilitate more effective learning, we introduced additional rewards at key waypoints along the route to mitigate the impact of our task's sparse reward space on training (successful human navigation). To emphasize the importance of human survival and maintain the overall reward balance introduced by these modifications, we increased the penalty for human fatalities (human fatality). For our training method 2, which has been specifically optimized, we refer to it as adversarial reinforcement learning robot, optimized (ARLR-O). The adjusted collective reward setting for ARLR-O is shown in (5) at the bottom of the previous page.

We further optimized the reward function for individual agents to align more closely with the group goal. The individual reward for *ARLR-O* is depicted in (6), shown at the bottom of the previous page.

After considering the overall situation and the reasonableness of the function, this study decided to keep the first and fourth reward items unchanged. For the second reward item, it is triggered only when the robot is surrounded by flames. The intention behind this punishment mechanism is twofold: on one hand, when the robot is in a state of being completely trapped, frequent determinations can cause system lag, and the robot is powerless to change the environment; on the other hand, terminating the robot's operations can fully leverage the computational resource advantages of the MA-POCA algorithm. Furthermore, based on real-world considerations, robots may be damaged under extreme temperatures, and ensuring human life safety is the top priority. However, the survival of the robot is crucial for guiding more humans to safety during evacuations. Therefore, this study expects robots to stay in relatively safe areas after completing evacuation tasks to facilitate their reuse. Experimental results have also validated the effectiveness of this practice, with current robots able to stand by on corridor platforms between tasks, ready for redeployment. As for the third reward item, it is a new addition aimed at addressing the sparse and delayed reward problem. For individual robots, the lack of effective positive incentives throughout the environment is detrimental to model training and learning.

E. ARL Framework

The objectives of this study necessitate a task environment characterized by high dynamism and complexity, thereby requiring a more efficient RL framework. According to research by Uther and Veloso [56], ARL shows higher work efficiency and robustness compared to traditional RL methods. Therefore, we leverage the characteristics of ARL, which are more closely aligned with real-world application scenarios, as well as its enhanced robustness and generalization capability, to improve agents' adaptability to different environments. Our ultimate goal is to develop an ARL model capable of efficiently completing complex tasks under various conditions.

To achieve efficient task environment simulation and enhance agents' adaptability to different environments, this study adopts a semisupervised ARL model based on a "generator–solver–evaluator" structure, namely the generative adaptive framework. This structure, inspired by the research findings of Bontrager and Togelius [46], aims to enhance the model's robustness and generalization capability through the interaction between simulated environment generation and agent behavior optimization.

We integrated MARL and ARL methods into our framework. Both methods have been proven [32], [44], [47] to enhance the decision-making efficiency and robustness of agents in complex, dynamic, and uncertain environments. By combining and optimizing these methods, we aim to leverage their strengths while mitigating their weaknesses. For MARL, we employed a multiagent algorithm with experience sharing, which enables the exchange of data such as observations and actions and facilitates distributed execution. Additionally, we introduced a hybrid game-theoretic adversarial agent into the environment, which helps to reduce training impact and mitigate the risk of overfitting.

The specific working principle involves the adversarial agent acting as a generator responsible for creating challenging environments, while our evacuation agent group serves as the solver tasked with obtaining higher rewards in the environments generated by the former. This competitive interaction between the two enhances both parties, allowing the evacuation agent group to learn from more varied and random situations. Furthermore, based on the characteristics of ARL, we organically combined environmental difficulty with rewards, forming a new evaluation standard to quantify performance in ARL scenarios. This evaluator is integrated into the environment, enabling real-time assessment of the agents' learning outcomes. We conducted multiple experiments to test the effectiveness of this approach.

1) Adversarial Opponent—Fire Source Agent: The fire source agent is capable of determining the location of the fires. It has access to the coordinates of all humans and robots, as well as grid observations of the status of each floor. The agent's actions include selecting the fire generation area by determining the horizontal coordinate of the next spreading fire location. The PPO algorithm [57] is employed for policy optimization. This algorithm constrains the magnitude of policy updates by utilizing a clipped surrogate objective

$$L_{\mathrm{CLIP}}(\theta) = \hat{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \mathrm{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t \right) \right] \tag{7}$$

where $r_t(\theta)$ represents the probability ratio and A_t denotes the estimation of the advantage function. Moreover, PPO enhances data efficiency through multiround minibatch updates. This straightforward yet effective policy optimization technique allows PPO to achieve satisfactory performance in complex control tasks.

a) Observations: The observations for the fire source generator are similar to those for evacuation robots, including the 3-D coordinates of all humans and evacuation robots, as well as grid sensors observing the state of each floor of the building.

b) Action space: The action space of the fire source generator consists of one discrete action and two continuous actions. The discrete action has three branch options, representing the choice of floor level for generating the fire; the continuous actions, similar to those of the evacuation robots, represent the horizontal coordinates (X_f, Y_f) of the fire source's generation position. Since the entire fire building environment is constructed in the form of cells, the actual generation position of the fire source also needs to be adjusted to the center of the cell closest to the decision

$$X'_f = \begin{cases} \lfloor X_f \rfloor - 0.5, & \left| \lfloor X_f \rfloor - X_f + 0.5 \right| > \left| \lfloor X_f \rfloor - X_f - 0.5 \right| \\ \lfloor X_f \rfloor + 0.5, & \left| \lfloor X_f \rfloor - X_f + 0.5 \right| \leq \left| \lfloor X_f \rfloor - X_f - 0.5 \right| \end{cases} \tag{8}$$

$$Y'_f = \begin{cases} \lfloor Y_f \rfloor - 0.5, & \left| \lfloor Y_f \rfloor - X_f + 0.5 \right| > \left| \lfloor Y_f \rfloor - Y_f - 0.5 \right| \\ \lfloor Y_f \rfloor + 0.5, & \left| \lfloor Y_f \rfloor - Z_f + 0.5 \right| \leq \left| \lfloor Y_f \rfloor - Y_f - 0.5 \right| . \end{cases} \tag{9}$$

(X'_f, Y'_f) represents the actual generation position of the next fire source controlled by the fire source generator. During training, if the area at (X'_f, Y'_f) is occupied by walls or other entities, the fire source would not be generated and a certain reward would be deducted; in nontraining mode, if an occupied situation occurs, a fire source would be generated at another random location in the environment.

c) Reward function: The fire source generator is responsible for creating challenging fire environments that impede evacuation robots from accomplishing their goal. However, it is important to note that the game between the fire source generator and the evacuation robots is not a zero-sum game [58]. The fire source generator does not aim for all humans to perish in the fire environment. Instead, its objective is to create a "difficult yet solvable" game environment for the evacuation robots. We want the fire source generator to prolong the time that humans spend in the environment. The reward function is designed based on these principles, with the maximum reward obtained when humans evacuate with 20% of their life value remaining. The specific design of the reward structure is as shown in (10) at the bottom of the page.

2) Dynamic Evaluation of ARL: In ARL, effectively evaluating the training results is crucial. DRL agents are vulnerable to adversarial perturbations, which can be attributed to the inherent instability of RL training [59]. Generally, in ARL scenarios, the most significant impact on agent training stems from changes in observational inputs [60]. In our fire evacuation scenario, the fire source agent acts as an adversary and has control over the generation locations of fire sources. Consequently, the environment perceived by the group of agents is constantly changing. Fluctuations or even drops in the reward curve during training are to be expected. However, temporary drops in reward do not necessarily indicate a decline in actual performance or training failure, as they are likely due to sudden environmental changes and increased difficulty [61]. Therefore, it is crucial to establish an evaluation metric that accurately assesses the actual effects of training. This enables us to fairly evaluate the performance of the agents in the ARL scenario.

TABLE I
RELATIONSHIP BETWEEN FLOOR NUMBER AND DIFFICULTY COEFFICIENT

Floor Number	Difficulty Coefficient
1	1.5
2	5.5
3	9.5

To achieve an ideal balance between environmental difficulty and rewards, we need to establish a relationship where the agent's reward decreases as the task difficulty increases, and vice versa. To accomplish this, we established an evaluation system that quantifies the difficulty level. In the fire evacuation simulation scenario, since the exit is located on the first floor, the evacuation routes of the agents would ultimately pass through the first floor. Therefore, when there is a fire on the first floor, it can block the exit or force evacuees to go up and then down again, resulting in the highest difficulty coefficient. As the floor number increases, the difficulty decreases.

We denoted the current game difficulty as h, and the number of temporary fires generated by spreading is represented by C. The coefficient F_{ij} corresponds to the ith fire source on the jth floor. This coefficient is determined by the floor number j, specifically reflecting the height of the floor. As shown in Table I, the lower the floor, the smaller the coefficient, and the greater its reciprocal, leading to higher overall difficulty. This is because lower floors are the escape routes that those on higher floors must pass through. In other words, a fire on the first floor not only hinders the evacuation of people on the first floor but also affects the evacuation of all floors above it.

For example, assuming there are eight fire sources, with two on the first floor, three on the second floor, and three on the third floor, the difficulty should be calculated as $h = ((2/1.5) + (3/5.5) + (3/9.5)) \times C$, where C represents the total number of temporary fires.

Consequently, we can establish a formula for calculating the difficulty by using the fire source locations as difficulty coefficients and multiplying them by the temporary fires generated due to spreading

$$h = \sum_{i=1}^{8} \left((F_{ij})^{-1} \right) \times C, \quad j \in \{1, 2, 3\}. \tag{11}$$

Our design simplifies the calculation while maintaining sufficient rationality, and at the same time provides upper and lower bounds for the cumulative coefficient, allowing us to maintain a certain level of predictability over the overall situation. Finally,

$$\text{Reward} \begin{cases} -200, & \text{Invalid generation} \\ -200, & \text{Human death} \\ \begin{cases} +5 \times \text{health}, & 0 < \text{health} \leq 20 \\ +1.25 \times (100 - \text{health}), & 20 < \text{health} \leq 100 \end{cases}, & \text{Human evacuation} \end{cases} \tag{10}$$

based on our expectations and incorporating the aforementioned difficulty formula, the constructed evaluation function E is as follows:

$$E(R,h) = \left(\frac{b^{(\frac{R}{\alpha})}}{c} + \frac{h^2}{k\alpha^2} \right) / \beta. \tag{12}$$

In the provided formula, R represents the cumulative reward in the current game. The scale factor α controls the relative importance of the two parts of the formula. The base coefficient b and the control coefficient c determine the reward components of the formula. The proportion coefficient k is calculated based on these parameters. The overall evacuation efficiency, represented as E, is also an optimization target for evaluation. The coefficient β scales the overall evacuation efficiency to provide more intuitive results, known as the normalization coefficient. This formula is then used to estimate the theoretical best case. Regarding the values of the related coefficients, the part of the evaluation function concerning Reward is first isolated and set as y

$$y = \frac{b^{(\frac{R}{\alpha})}}{c}. \tag{13}$$

Setting it as a curve similar to an exponential function serves two purposes: first, to handle cases of negative rewards; and second, to utilize the properties of exponential functions.

Based on existing research, at this stage, people generally use the metric of maximizing the reward R_w (reward worst) under the worst case scenario to evaluate the training performance and robustness of RL agents with input adversarial perturbations. R_w refers to the reward under the worst possible sequence of adversarial attacks. Oikarinen et al. [63] furthered this research and proposed an alternative evaluation method called greedy worst case reward (GWC), which approximates the expected R_w and can be efficiently calculated with a predictable linear complexity. Drawing on the ideas of Oikarinen et al. [63] and applying them to our scenario, we estimated the worst case reward R_w. On the other hand, since the reward formula in the RL scenario is artificially designed, the reward in the theoretical best-case scenario is also estimable, resulting in the theoretical best-case reward R_b (reward best)

$$R_b = \sum_{i=1}^{n} \left(\sum_{j=1}^{3} \left(hp - \frac{\text{Avg}\left[\text{Min}\left(\text{len}_j\right)/\text{speed}\right]}{\text{framesp} \times \text{rate}} \right) + \text{FixReward} \right). \tag{14}$$

When RL successfully acquires the correct strategy [62], the training reward curve typically exhibits an optimistic upward trend. However, as training progresses, the rate of improvement in the reward gradually diminishes, and by the final stages, it almost ceases to rise. We contend that even a modest increase in the reward curve toward the end of training can signify a substantial breakthrough for the agent. Therefore, once the agent's reward value surpasses a specific threshold, we aim to amplify the significance of this minor improvement in our overall evaluation formula, E.

Utilizing the characteristics of the exponential-like function y, when R is below a certain value, the entire function y becomes quite small, indicating extremely poor performance of the agent group. In this scenario, the overall weight of the evaluation function E leans toward the difficulty-related part, meaning that difficulty bonuses outweigh behavior bonuses, and a base score is given just for completing the game. We called this scenario key point 1, where we consider the agent's strategy incapable of influencing the environment, with $R = R_w$. Conversely, when R exceeds a certain threshold, the entire function y explodes exponentially, affecting the balance of the entire evaluation function. We wish to set another key point 2 at the start of this exponential explosion, where R should be close to R_b. We aim to indirectly give more weight to the improvements achieved by the agent training in the later stages by using the effect of the exponential explosion, calculating the needed b and c through the two set key points.

Based on data observed during experiments, determine the appropriate value for the scaling factor α to achieve good practical effects in data processing and other aspects, aiding in enhancing the experiment's precision and reliability. With the given coefficient values, we can obtain a formula that disregards β; in this scenario, we set $\beta = e'$ as a baseline for measuring other conditions. In other words, the evaluation formula, where only the coefficient k is unknown, acts like a ruler—only needing to be calibrated with the agent's average performance, thereby estimating efficiency.

Finally, through multiple experiments, selecting data with higher rewards and lower difficulty for comparison against data with lower rewards and higher difficulty to find close performance outcomes, and incorporating their reward values R and difficulty values h into the formula, derives the proportion coefficient k, thus determining the weight distribution of the two parts of the evaluation function.

Importantly, our developed evaluation system template can be extended to similar scenarios. Specifically, in ARL environments where difficulty can be estimated, a similar method can be employed to construct an evaluation function for a systematic assessment of the overall training process.

IV. Training

In this study, we established a training framework involving two types of agents: a solver agent and an adversarial generator. The primary objective of the solver agent is to complete the evacuation task as swiftly as possible while ensuring human safety. In contrast, the adversarial generator aims to create challenging environments that increase the task's difficulty. This setup is designed to enhance the solver agent's performance under difficult conditions by introducing reasonable challenges, thereby improving robustness. The generator model is trained using the PPO algorithm, guided by a mixed-game reward function described in (10). On the other hand, the solver model is trained using the MA-POCA with centralized critic and decentralized execution. During training, each agent transmits its observations, actions, and rewards to a central controller

TABLE II
MODELS CONFIGURATION PARAMETERS

Configuration Target	**Evacuation Robot Agent Group**			**Fire Source Agent**		
Trainer type	**POCA**			**PPO**		
Hyperparameters	Batch size	1024		Batch size	1024	
	Buffer size	20 480		Buffer size	10 240	
	Learning rate	0.0001		Learning rate	0.0003	
	Learning rate schedule	Linear		Learning rate schedule	Linear	
	Beta	0.005		Beta	0.005	
	Beta schedule	Constant		Beta schedule	Constant	
	Epsilon	0.2		Epsilon	0.2	
	Epsilon schedule	Linear		Epsilon schedule	Linear	
	Lambd	0.95		Lambd	0.95	
	Num epoch	3		Num epoch	3	
Network settings	Vis encode type	Simple		Vis encode type	Simple	
	Normalize	True		Normalize	False	
	Hidden units	256		Hidden units	256	
	Num layers	3		Num layers	3	
Reward signals	Extrinsic	Gamma	0.90	Extrinsic	Gamma	0.99
		Strength	1.0		Strength	1.0
Keep checkpoints	100			80		
Checkpoint interval	1000			500		
Time horizon	64			64		
Summary freq	1000			500		

responsible for updating the policy and value networks. In the execution phase, agents independently use their policy networks for decision-making, significantly improving execution efficiency. Pseudocode for MA-POCA is provided in Appendix C. Throughout each training cycle, the generator and solver are iteratively trained, with the solver's network being updated multiple times for every update of the generator's network. Our ARL framework embodies a complex game-theoretic interaction between agents. As highlighted by Zhang et al. [43], such interactions can be categorized into cooperative, competitive, and mixed games. This study focuses on mixed games, where agents share common primary goals but also have conflicting secondary interests. Within this context, we defined a specific threshold: the generator agent receives maximum rewards when at least 20% of the human life value is retained during evacuation. This threshold was determined through iterative experiments and theoretical considerations, aiming to balance the practical needs of emergency evacuation with the educational purposes of the experimental environment. By structuring the problem in this manner, we ensured a balance between challenge and feasibility. The evacuation task is designed to be sufficiently challenging while maintaining feasibility in adversarial environments. In our framework, the solver takes action when approaching the destination or encountering flames, while the generator modifies the environment to create fire sources. Since the number of fire sources is fixed, the generator only needs to perform a limited number of actions per scenario, whereas the solver must take actions more frequently in each episode. Additionally, the solver needs to observe and learn from a broader range of variables than the generator. Consequently, the solver is allocated a larger buffer size and a lower learning rate to reduce training oscillations. The ARL framework and virtual environment are implemented using Unity3D engine version 2021.3.3f1c1. The training process is conducted on a server with 32GB RAM, an Intel i9-12900k CPU, and an Nvidia RTX 3080Ti GPU. The software environment is configured as follows:

1) PyTorch version: 1.8.2;
2) CUDA version: 11.7;
3) ML-Agents version: 0.29.0;
4) Python version: 3.8.13.

We fine-tuned the hyperparameters for optimal training performance. Specifically, we slightly reduced the initial learning rate to mitigate large fluctuations in the training curve. To achieve more accurate gradient estimates, we increased the batch size within a controlled range and expanded the buffer size accordingly. The neural network architecture also underwent careful design adjustments. Given the complexity of the observed vector correlations, we opted for sufficiently large fully connected layers. After extensive research and discussion, we used three hidden layers, each with 256 units, to decode the observed values. We applied normalization to the vector observation inputs, forming residual connections for improved inference. Moreover, the extrinsic reward signals from the environment were adjusted for the MA-POCA algorithm, setting the discount factor γ to 0.9, indicating how far into the future the agent considers rewards. The specific configuration files used for training are provided in the tables. Table II details the configurations for both the evacuation robot agent group and the fire source agent. It offers a comparative overview of the parameters set for each agent type, highlighting differences in buffer size, learning rate, and network settings tailored to their respective operational requirements.

This meticulous tuning of key training hyperparameters aims to achieve more stable and effective ARL in complex fire evacuation scenarios. During training, both the generator and solver are penalized whenever a human dies. The primary rewards for both agents are derived from the remaining human life value upon successful evacuation. However, while a higher life value results in a greater reward for the solver, the generator seeks to create a sufficiently challenging environment where the human life value remains around 20% to maximize its reward.

V. Experiments

In this section, we elaborated on the results of our experiments. For the training process, we employed the two methods proposed earlier to train both RL models and ARL models for a sufficient number of steps to discern the differences. For testing, we conducted systematic and comprehensive empirical evaluations of our methods in multiagent environments under five graded levels of difficulty and compared their performance with the advanced multiagent algorithm MA-POCA. Our testing experiments were divided into two major parts, designed to demonstrate the effectiveness of our constructed evaluation method and to compare the performance of our methods.

A. Reward Function Comparison

Pretests were conducted to evaluate the performance of two proposed methods, *ARLR-B* and *ARLR-O*, which were considered as candidate algorithms for our ARL method. Training scenarios of a pure RL algorithm without adversarial methods, namely reinforcement learning robot, basic (RLR-B) and reinforcement learning robot, optimized (RLR-O), were also provided for comparison.

Due to the inherent randomness of the RL process, performance fluctuations can occur across different training runs. We conducted five training sessions for each method to calculate the average results, and we used error bands to represent the oscillation range of these runs. The error bands in our results were calculated based on the standard error of the mean, providing a statistically valid measure of the distribution of our data points around the mean. It is important to note that all five runs were conducted under identical conditions—on the same machine and within the same simulation environment—to ensure consistency. The difference between ARLR-O and ARLR-B lies solely in the reward function, whereas the difference between ARLR and RLR is due to the inclusion of adversarial training components in ARLR.

As shown in Fig. 4, despite these fluctuations, our results demonstrate that the optimized algorithm ARLR-O achieved higher reward values at a faster rate, with the highest rewards reaching over 10 000 (on average). Meanwhile, the optimized algorithm RLR-O and the baseline algorithm RLR-B achieved nearly identical maximum rewards of approximately 6000. Although the new reward function design offers more ways to obtain rewards compared to the old one, measures were taken to balance the change in rewards. This indicates that the improvement in the ARL algorithms can be attributed to a more rational design of the new reward function rather than merely an increase in rewards. The clear contrast between these curves confirms the value of optimization. Consequently, ARLR-O was selected for further testing and evaluation as the actual ARL model [adversarial reinforcement learning robot model (ARLR)].

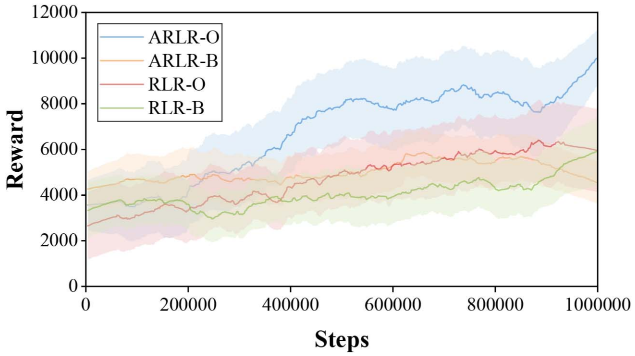


Fig. 4. By comparing the performance of two algorithms using different training methods, ARLR-O represents the ARLR algorithm under the optimized method, while ARLR-B represents the ARLR algorithm under the baseline method. Similarly, RLR-O and RLR-B are for the RLR algorithm under optimized and baseline methods, respectively. The distinction between ARLR and RLR lies solely in whether the adversarial framework developed in this study is utilized.

B. Experiments and Results

For the formal experiments, we systematically and comprehensively evaluated our methods through empirical testing in multiagent environments with different difficulty levels (L1–L5). Each level featured fixed fire generation methods and locations, with the difficulty coefficient increasing sequentially. Subsequently, we compared the performance of our methods with the multiagent algorithm MA-POCA. The experiment was divided into two main parts. The first experiment (experiment 1) aimed to demonstrate the effectiveness of the ARL evaluation method constructed in the previous section. We subsequently conducted the second experiment (experiment 2) as the primary evaluation of our proposed ARLR method and the baseline approaches.

1) Baselines and Metrics for Testing: Three baselines were utilized to compare and validate our *ARLR*. The first baseline represents the scenario where no robots exist, and humans must explore and evacuate on their own (*Baseline*). The second baseline involves using greedy algorithms to control the robots and accomplish the task (*Greedy*). The Greedy robots are designed to be able to attract followers, as they actively seek nearby human followers, taking advantage of humans' inherent ability to autonomously find leaders. The third baseline corresponds to a model trained with MA-POCA and random fire source generation, without employing adversarial training [reinforcement learning robot (RLR)]. It is important to note that the differences between these methods are designed precisely to isolate the effects of specific features.

a) ARLR Versus RLR: The key difference lies in the use of adversarial training, which allows us to evaluate the contribution of this aspect to the overall performance of the evacuation task. By comparing ARLR with RLR, we aim to understand how adversarial elements improve the robustness and adaptability of the agents in dynamic environments.

b) RLR Versus Greedy: This comparison focuses on the utilization of MARL, distinguishing the effects of collaborative agent behavior. The goal is to assess how MARL improves the coordination among multiple agents and its impact on overall evacuation efficiency compared to simpler, noncooperative strategies.

c) Greedy Versus Baseline: The primary distinction is the deployment of robotics. This comparison highlights the impact of automation on evacuation efficiency by evaluating how the presence of robots, even when controlled by a simple greedy algorithm, influences the speed and success rate of human evacuations compared to scenarios with no robotic assistance.

The test metrics are centered around the evacuation status of humans. We have identified six metrics based on the actual situation: *smoke deaths* (number of humans who died due to prolonged exposure to toxic fumes, with health decreasing by 0.01 per frame until it reaches zero), *fire deaths* (number of humans who died from direct contact with flames, resulting in rapid health depletion to zero), *survival rate* (number of humans successfully evacuated/total number of humans), *evacuation time* (average time taken for successful evacuation), *remaining health* (average health value of successfully evacuated humans), and *global health* (average health value of all humans). Based on realistic considerations, we believe that global health and survival rate are the most important indicators. This is because the survival rate sufficiently reflects the casualty situation of humans under general conditions. The metrics of smoke deaths and fire deaths are primarily used to analyze the causes of deaths in the current situation. Meanwhile, global health represents the average remaining health value of all individuals, better reflecting the survival status and injury level of humans.

2) Result and Discussion:

a) Experiment 1: This experiment aims to demonstrate the effectiveness of the ARL evaluation method [see (12)]. We conducted one thousand test iterations under five predefined difficulty conditions, aggregating the resulting data. Our objective was twofold: to obtain results where fire difficulty increases while keeping E constant, and to gather results with increasing E at fixed difficulty levels. This approach allows us to establish a clear relationship between algorithm performance and the value of E.

Due to the inherent randomness of fire difficulties, it is not possible to find perfect data that precisely meet all the requirements (e.g., when $h = 1000$ and $E = 0.5$). Additionally, it is not feasible to artificially manipulate the combination of h and E, as it goes against the original intention. Therefore, we selected the data points closest to our desired requirements on both sides of the difficulty axis, with fixed difficulties, and calculated the average metrics related to E under graded difficulties. Fig. 5 presents the results obtained from our experiments. Due to limited space, only the two most important metrics (survival rate and global health) are displayed.

Our observations reveal that as the difficulty axis (HA) increases from 2.5K to 4.5K, there is a general decrease in survival rates, accompanied by a notable decline in global health. This suggests that with increasing task difficulty, both the survival rates and the health outcomes of successfully evacuated individuals worsen. However, it is important to highlight that even with higher difficulty levels, cases where the evaluation metric (E) is relatively high (e.g., $E \geq 1.2$) still maintain a robust survival rate. For instance, at the highest difficulty level, HA4.5K, the survival rate remains approximately 82.

When $E \geq 1.4$, survival rates remain consistently high (approximately 85%–96%) across all difficulty levels. In contrast, when the evaluation metric drops to $E \geq 1.0$, there is a significant decline in survival rates, especially under higher difficulty conditions. For example, at HA4.5K, the survival rate falls to about 58.8%. Similarly, global health also remains high (ranging from 57.39 to 66.72) when $E \geq 1.4$. However, when E drops to $E \geq 1.0$, global health decreases significantly, reaching its lowest point of 43.11 under the most challenging condition (HA4.5K).

In terms of distribution, combinations of high-evaluation metrics and low-difficulty levels yield the highest survival rates. For example, the combination of $E \geq 1.4$ and HA2.5K achieves a survival rate of approximately 96%, the highest observed in the dataset. Conversely, low-evaluation metrics combined with high-difficulty levels (e.g., $E \geq 1.0$ and HA4.5K) result in a sharp decrease in survival rates, marking the lowest region in terms of both survival rate and global health.

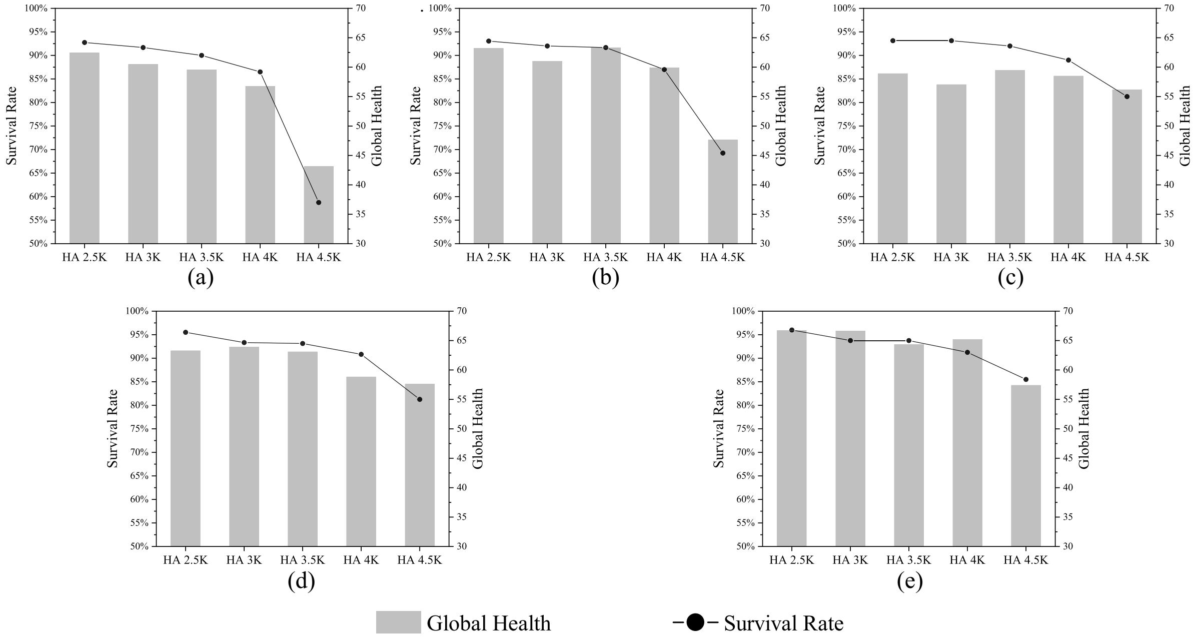
It is important to note that our evaluation formula is strongly correlated with both rewards and difficulty levels but has a weaker correlation with actual evacuation performance metrics. This can lead to discrepancies, such as those seen with the combination of $E \geq 1.2$ and HA2.5K. However, these discrepancies are within an acceptable range.

In summary, analysis of the five charts indicates that the trends in survival rate and global health are generally aligned, with the difficulty axis (HA) and the evaluation metric (E) being critical factors influencing the model's performance. The higher the evaluation metric (E), the better the Survival Rate and Global Health, indicating that the evaluation function (E) is generally effective in assessing the outcomes of ARL and exhibits consistent trends across various difficulty levels and E values.

b) Experiment 2: Experiment 2 aimed to compare the performance of our proposed method with the baselines. For each method, we conducted a minimum of 200 independent tests under various difficulty conditions and calculated the average value and standard error of the mean for each data metric. It is important to note that, in contrast to the difficulty indicators used in experiment 1, experiment 2 employed fixed fire source and spread locations across all difficulty levels. This approach ensured that the upper and lower limits of difficulties caused by flame spread remained consistent throughout the experiment.

The experimental results are depicted in Fig. 6. In this study, we employed one-way ANOVA (analysis of variance) to evaluate the effects of four different evacuation strategies (ARLR, RLR, greedy, and baseline) on evacuation efficiency. We conducted five independent ANOVA tests, each corresponding to a specific difficulty level. The variables included all metrics except the smoke deaths and fire deaths indicators. These two indicators were excluded as they are solely used to observe the causes of human fatalities.

First, the large sample size and numerous groups led to excessively significant p-values, prompting us to use random sampling to select 40 datasets for each ANOVA test. The final results demonstrated statistically significant effects of the



(a) (b) (c) (d) (e)

Fig. 5. Experiment results. Average data were obtained from five difficulty axes and different E ranges. E represents the evaluation value obtained by our evaluation algorithm, HA represents the difficulty axis and represents the data selection on both sides of the difficulty axis. The survival rate on the left axis represents the proportion of humans who survived the experiment. The global health on the right axis represents the average health of all individuals, including those who have died. (a) $E \geq 1.0$. (b) $E \geq 1.1$. (c) $E \geq 1.2$. (d) $E \geq 1.3$. (e) $E \geq 1.4$.

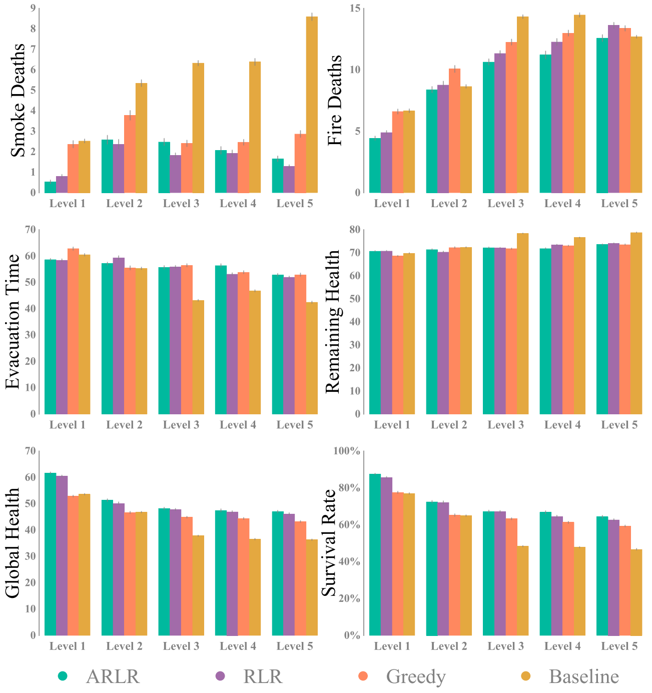


Fig. 6. Bar chart visually summarizes the key metrics obtained from experiment 2 across the different difficulty levels with the four different methods (ARLR, RLR, greedy, and baseline). The six metrics indicate the performance of each method.

evacuation method on successful evacuation numbers across all difficulty levels.

Specifically, for difficulty level 1 (survival rate: $F = 11.56$, $p < .001$; global health: $F = 23.79, p < .001$; remaining health: $F = 23.58, p < .001$; evacuation time: $F = 11.46, p < .001$), level 2 (survival rate: $F = 14.98, p < .001$; global health: $F = 20.28, p < .001$; remaining health: $F = 3.95, p < .01$; evacuation time: $F = 3.16, p < .05$), level 3 (survival rate: $F = 23.55, p < .001$; global health: $F = 24.67, p < .001$; remaining health: $F = 6.41, p < .001$; evacuation time: $F = 7.61$, $p < .001$), level 4 (survival rate: $F = 25.24, p < .001$; global health: $F = 42.74, p < .001$; remaining health: $F = 5.83$, $p < .001$; evacuation time: $F = 5.29, p < .01$), and level 5 (survival rate: $F = 36.71, p < .001$; global health: $F = 35.48, p < .001$; remaining health: $F = 4.45, p < .01$; evacuation time: $F = 9.34, p < .001$), it is evident that the effects of the strategies differ significantly across all difficulty levels, indicating that the choice of evacuation strategy has a strong impact on evacuation efficiency.

In the baseline scenario, which involved only humans, increasing environmental difficulty led to a higher incidence of individuals getting lost due to the absence of prior knowledge of the building layout and lack of robotic guidance. This trend is reflected in the smoke deaths metric, where, in the worst case, nearly nine casualties resulted from disorientation. Conversely, when robots guided the evacuation, the number of casualties due to disorientation was reduced, underscoring the importance and benefits of human–robot interaction and collaboration. For the fire deaths metric, casualties from fire spread were significantly higher than those from disorientation. The greedy and RLR methods appeared less effective in avoiding fire hazards, while the ARLR method performed best in this regard.

Regarding evacuation time, the random generation of human positions in the experimental environment and the fact that only successfully escaped individuals' times are counted can lead to misleading results. In some cases, even when the human fatality rate is high, individuals spawned near the exit may escape quickly, making the evacuation time and remaining health indicators appear satisfactory. For ARLR, the evacuation time may sometimes exceed that of RLR, likely because the experience gained from training in adversarial environments leads to additional actions, increasing the time spent. However, this becomes crucial for achieving success in rapidly changing environments. It is important to note that ARLR has the shortest evacuation time at level 5 compared to all other levels. We believe this phenomenon is reasonable in practical terms, as it further reflects the robustness of our framework. At level 5 difficulty, due to the higher number of fire sources and their more lethal locations, high-difficulty fire spread may quickly block key passages, leading to evacuation failure.

We suggest that the robot's strategy should differ depending on the environment's difficulty. In a more lenient environment, it can take detours to maximize the avoidance of danger sources, while in extreme environments, the robot's strategy should prioritize faster evacuation, minimizing detours to maximize evacuation efficiency. Compared to RLR, ARLR learns to counteract the opponent's strategy in the presence of intelligent adversaries, performing better in complex scenarios. As the difficulty increases, ARLR demonstrates certain advantages over RLR in terms of overall indicators and the critical Global Health indicator, proving its superiority and consistency.

Compared to scenarios without evacuation robots, the greedy robot (greedy) can increase the survival rate by an average of 8.4% and improve the global health score by an average of 4.18. Particularly under high-difficulty conditions, the presence or absence of robot guidance has a significant impact on evacuation outcomes. The RLR and ARLR methods implemented in this article surpass the greedy robot in key metrics, with RLR showing an average improvement of 13.5% and 8.07, and ARLR showing an average improvement of 14.7% and 8.92 compared to the Baseline. On a macro level, ARLR and RLR achieve varying degrees of success across different difficulty levels, mostly surpassing the greedy algorithm robot. Using a greedy algorithm robot shows a clear improvement over scenarios without robot guidance. In terms of the two key indicators of survival rate and global health, our ARLR maintains a dominant position, indicating that our approach is crucial for improving the efficiency of fire evacuation. Overall, our ARLR method has demonstrated satisfactory performance and strong consistency.

VI. Conclusion

This study proposes a novel framework integrating MARL with ARL to address the challenges of human–robot interaction and evacuation strategy optimization in complex, dynamic fire scenarios. By constructing a highly adaptive environment simulating a multistory building and incorporating a realistic human behavior model, our approach demonstrates the potential for enhancing both the efficiency and robustness of evacuation robots. Our framework introduces several innovative methods. First, the integration of MARL with ARL allows for the development of evacuation agents that are not only capable of collaborating effectively in dynamic and unpredictable environments but also demonstrate improved generalization and adaptability through adversarial training. This hybrid approach leverages the strengths of both MARL's distributed execution and ARL's challenge-driven learning process, resulting in a system that performs well across a wide range of complex scenarios. The experimental results indicate that ARLR demonstrates significant advantages over other methods in critical aspects. ARLR exhibits enhanced robustness and adaptability in more varied and challenging environments, where traditional RL methods such as RLR may struggle. This performance distinction highlights ARLR's potential in scenarios characterized by high-environmental dynamics and unpredictability. Furthermore, it underscores the importance of our proposed evaluation metrics, which provide a more comprehensive assessment of an agent's applicability in emergency evacuation scenarios.

Despite the advances presented in this study, several limitations remain. First, the random generation of human positions in our simulation affects the accuracy of evacuation time measurements, as individuals closer to exits may skew results. Future research should develop advanced methods for measuring evacuation performance, incorporating the variability in initial human positions and their impact on efficiency. Second, trajectory analysis is limited by the unpredictability of fire spread, reducing its effectiveness as a large-scale performance metric. Additionally, while our simulation accounts for dynamic crowd density, it assumes effective robot guidance regardless of density variations, which oversimplifies real-world complexities where high-crowd density could hinder communication and movement.

Another critical aspect for future work is the validation of the framework. Real-world validation poses challenges, as replicating fire scenarios ethically and cost-effectively is difficult, limiting the direct applicability of our findings. Future research should focus on optimizing the integration of scanned building models into our framework, aiming to deploy trained evacuation robots effectively in real-world scenarios. By refining this process, we can ensure that robots are well prepared for diverse environments and able to transition seamlessly from virtual training to real-world emergencies. However, our assumption of unconditional human trust in robots oversimplifies real-world dynamics, where trust levels can vary significantly. Future research directions include exploring the impact of varying human trust in robots through dynamic trust models and improving trajectory analysis with enhanced prediction models.

References

[1] A. S. Angel and R. Jayaparvathy, "Design and implementation of an intelligent emergency evacuation system," in *Proc. Int. Conf. Comput. Power, Energy Inf. Commun. (ICCPEIC)*, 2017, pp. 013–017.

[2] X. Zhang, "Study on rapid evacuation in high-rise buildings," *Eng. Sci. Technol.*, vol. 20, no. 3, pp. 1203–1210, 2017.

[3] M. Zhou, H. Dong, B. Ning, and F.-Y. Wang, "Recent development in pedestrian and evacuation dynamics: Bibliographic analyses, collaboration patterns, and future directions," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 4, pp. 1034–1048, Dec. 2018.

[4] J. Chen et al., "An enhanced model for evacuation vulnerability assessment in urban areas," *Comput., Environ. Urban Syst.*, vol. 84, 2020, Art. no. 101540.

[5] X. Chen and F. B. Zhan, "Agent-based modelling and simulation of urban evacuation: Relative effectiveness of simultaneous and staged evacuation strategies," *J. Oper. Res. Soc.*, vol. 59, no. 1, pp. 25–33, 2008.

[6] G. Sidiropoulos, C. Kiourt, and L. Moussiades, "Crowd simulation for crisis management: The outcomes of the last decade," *Mach. Learn. Appl.*, vol. 2, 2020, Art. no. 100009.

[7] A. Wharton, "Simulation and investigation of multi-agent reinforcement learning for building evacuation scenarios," *Report, Oxford, U.K.: St Catherine's College*, 2009.

[8] A. E. Ünal, C. Gezer, B. K. Pak, and V. Ç. Güngör, "Generating emergency evacuation route directions based on crowd simulations with reinforcement learning," in *Proc. Innovations Intell. Syst. Appl. Conf. (ASYU)*, Piscataway, NJ, USA: IEEE, 2022, pp. 1–6.

[9] L. Yang, X. Wang, J. J. Zhang, M. Zhou, and F.-Y. Wang, "Pedestrian choice modeling and simulation of staged evacuation strategies in daya bay nuclear power plant," *IEEE Trans. Comput. Social Syst.*, vol. 7, no. 3, pp. 686–695, Jun. 2020.

[10] M. Zhou, H. Dong, P. A. Ioannou, and F.-Y. Wang, "Crowd evacuation with multi-modal cooperative guidance in subway stations: Computational experiments and optimization," *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 5, pp. 2536–2545, Oct. 2023.

[11] A. Tsurushima, "Efficient crowd evacuation guidance with multiple visual signage using a middle-range agent model and black-box optimization," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, 2022, pp. 2591–2598.

[12] E. Duarte, F. Rebelo, J. Teles, and M. S. Wogalter, "Behavioral compliance for dynamic versus static signs in an immersive virtual environment," *Appl. Ergonom.*, vol. 45, no. 5, pp. 1367–1375, 2014.

[13] R. Ravnik and F. Solina, "Interactive and audience adaptive digital signage using real-time computer vision," *Int. J. Adv. Robotic Syst.*, vol. 10, no. 2, p. 107, 2013.

[14] H. Y. Wong, Y. Zhang, and X. Huang, "A review of dynamic directional exit signage: Challenges & perspectives," 2022.

[15] J. Li, Y. Hu, and W. Zou, "Dynamic risk assessment of emergency evacuation in large public buildings: A case study," *Int. J. Disaster Risk Reduction*, vol. 91, 2023, Art. no. 103659.

[16] B. Tang, C. Jiang, H. He, and Y. Guo, "Human mobility modeling for robot-assisted evacuation in complex indoor environments," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 5, pp. 694–707, Oct. 2016.

[17] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *Int. J. Robot. Res.*, vol. 39, no. 7, pp. 856–892, 2020.

[18] X.-T. Truong and T. D. Ngo, "Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model," *IEEE Trans. Automat. Sci. Eng.*, vol. 14, no. 4, pp. 1743–1760, Oct. 2017.

[19] H. Shirado and N. A. Christakis, "Locally noisy autonomous agents improve global human coordination in network experiments," *Nature*, vol. 545, no. 7654, pp. 370–374, 2017.

[20] Y. Baudoin et al., "View-finder: Robotics assistance to fire-fighting services and crisis management," in *Proc. IEEE Int. Workshop Saf., Secur. Rescue Robot. (SSRR)*, Piscataway, NJ, USA: IEEE, 2009, pp. 1–6.

[21] G. Sidiropoulos, C. Kiourt, and L. Moussiades, "Crowd simulation for crisis management: The outcomes of the last decade," *Mach. Learn. Appl.*, vol. 2, 2020, Art. no. 100009.

[22] K. Tan et al., "An IVC-based nuclear emergency parallel evacuation system," *IEEE Trans. Comput. Social Syst.*, vol. 8, no. 4, pp. 844–855, Aug. 2021.

[23] Q. Sun and Y. Turkan, "A bim based simulation framework for fire evacuation planning," in *Proc. Adv. Inform. Comput. Civil Construction Eng./Proc. 35th CIB W78 Conf.: IT Des., Constr., Manage.*, Berlin, Germany: Springer, 2019, pp. 431–438.

[24] A. Johnson, S. Zheng, A. Nakano, G. Schierle, and J.-H. Choi, "Adaptive kinetic architecture and collective behavior: A dynamic analysis for emergency evacuation," *Buildings*, vol. 9, no. 2, p. 44, 2019.

[25] P. Huang, M. Chen, K. Chen, S. Ye, and L. Yu, "Study on an emergency evacuation model considering information transfer and rerouting: Taking a simplified h-shape metro station hall as an example," *Tunnelling Underground Space Technol.*, vol. 124, 2022, Art. no. 104485.

[26] P. Huang, X. Lin, C. Liu, L. Fu, and L. Yu, "A real-time automatic fire emergency evacuation route selection model based on decision-making processes of pedestrians," *Saf. Sci.*, vol. 169, 2024, Art. no. 106332.

[27] Y. Tang, X. Zhang, R. Wang, J. Xu, L. Hu, and Y. Hao, "Crowd intelligent grouping collaboration evacuation via multi-agent reinforcement learning," in *Proc. 26th Int. Conf. Comput. Supported Cooperative Work Des. (CSCWD)*, Piscataway, NJ, USA: IEEE, 2023, pp. 796–801.

[28] M. A. Wiering and M. Van Otterlo, *Reinforcement Learning* (*Adaptation, learning, and optimization, 12*), Berlin, Germany: Springer, 2012, p. 729.

[29] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, 1996.

[30] S. Matsuzaki and Y. Hasegawa, "Learning crowd-aware robot navigation from challenging environments via distributed deep reinforcement learning," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, Piscataway, NJ, USA: IEEE, 2022, pp. 4730–4736.

[31] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, "Decentralized structural-RNN for robot crowd navigation with deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Piscataway, NJ, USA: IEEE, 2021, pp. 3517–3524.

[32] J. Chen, T. Lan, and V. Aggarwal, "Option-aware adversarial inverse reinforcement learning for robotic control," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Piscataway, NJ, USA: IEEE, 2023, pp. 5902–5908.

[33] S. Butail, "Simulating the effect of a social robot on moving pedestrian crowds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Piscataway, NJ, USA: IEEE, 2015, pp. 2413–2418.

[34] Z. Liu, B. Wu, and H. Lin, "Coordinated robot-assisted human crowd evacuation," in *Proc. IEEE Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE, 2018, pp. 4481–4486.

[35] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, Piscataway, NJ, USA: IEEE, 2019, pp. 6015–6022.

[36] F. Haghpanah, B. W. Schafer, and S. Castro, "Application of bug navigation algorithms for large-scale agent-based evacuation modeling to support decision making," *Fire Saf. J.*, vol. 122, 2021, Art. no. 103322.

[37] E. Boukas, I. Kostavelis, A. Gasteratos, and G. C. Sirakoulis, "Robot guided crowd evacuation," *IEEE Trans. Automat. Sci. Eng.*, vol. 12, no. 2, pp. 739–751, Apr. 2015.

[38] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical Rev. E*, vol. 51, no. 5, p. 4282, 1995.

[39] D. Kim, J. Jeong, S. Park, J. Go, and C. Yeom, "A study on the application of optimal evacuation route through evacuation simulation system in case of fire," *J. Soc. Disaster Inf.*, vol. 16, no. 1, pp. 96–110, 2020.

[40] M. Urmanov, M. Alimanova, and A. Nurkey, "Training unity machine learning agents using reinforcement learning method," in *Proc. 15th Int. Conf. Electron., Comput. Comput. (ICECCO)*, Piscataway, NJ, USA: IEEE, 2019, pp. 1–4.

[41] Y. Wang, H. He, and X. Tan, "Truly proximal policy optimization," in *Proc. Uncertainty Artif. Intell.*, PMLR, 2020, pp. 113–122.

[42] J. C. de Jesus, V. A. Kich, A. H. Kolling, R. B. Grando, M. A. d. S. L. Cuadros, and D. F. T. Gamarra, "Soft actor-critic for navigation of mobile robots," *J. Intell. Robotic Syst.*, vol. 102, no. 2, p. 31, 2021.

[43] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," in *Handbook of Reinforcement Learning and Control*, Berlin, Germany: Springer, 2021, pp. 321–384.

[44] B. Zhang, Z. Wei, and W. Zhu, "Intelligent close air combat design based on MA-POCA algorithm," in *Proc. IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA)*, Piscataway, NJ, USA: IEEE, 2022, pp. 408–414.

[45] T. Wu et al., "GraspARL: Dynamic grasping via adversarial reinforcement learning," 2022, *arXiv:2203.02119*.

[46] P. Bontrager and J. Togelius, "Learning to generate levels from nothing," in *Proc. IEEE Conf. Games (CoG)*, Piscataway, NJ, USA: IEEE, 2021, pp. 1–8.

[47] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2017, pp. 2817–2826.

[48] W. G. Van Toll, A. F. Cook IV, and R. Geraerts, "A navigation mesh for dynamic environments," *Comput. Animation Virtual Worlds*, vol. 23, no. 6, pp. 535–546, 2012.

[49] L. Almón-Manzano, R. Pastor-Vargas, and J. M. C. Troncoso, "Deep reinforcement learning in agents' training: Unity ML-agents," in *Proc. Int. Work-Conf. Interplay Between Natural Artif. Comput*., Berlin, Germany: Springer, 2022, pp. 391–400.

[50] S. Marsar, "Survivability profiling: How long can victims survive in a fire?" *Fire Eng.*, vol. 163, no. 7, pp. 77–82, 2010.

[51] W. D. Smart and L. P. Kaelbling, "Effective reinforcement learning for mobile robots," in *Proc. IEEE Int. Conf. Robot. Automat. (Cat. No. 02CH37292)*, vol. 4, Piscataway, NJ, USA: IEEE, 2002, pp. 3404–3410.

[52] B. Badnava, M. Esmaeili, N. Mozayani, and P. Zarkesh-Ha, "A new potential-based reward shaping for reinforcement learning agent," in *Proc. IEEE 13th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Piscataway, NJ, USA: IEEE, 2023, pp. 01–06.

[53] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Piscataway, NJ, USA: IEEE, 2018, pp. 6292–6299.

[54] K. Macek, I. PetroviC, and N. Peric, "A reinforcement learning approach to obstacle avoidance of mobile robots," in *Proc. 7th Int. Workshop Adv. Motion Control (Cat. No. 02TH8623)*, Piscataway, NJ, USA: IEEE, 2002, pp. 462–466.

[55] S. Padakandla, "A survey of reinforcement learning algorithms for dynamically varying environments," *ACM Comput. Surveys (CSUR)*, vol. 54, no. 6, pp. 1–25, 2021.

[56] W. Uther and M. Veloso, "Adversarial reinforcement learning," Tech. Rep., Carnegie Mellon University, 2003.

[57] Y. Wang, H. He, and X. Tan, "Truly proximal policy optimization," in *Proc. Uncertainty Artif. Intell.*, PMLR, 2020, pp. 113–122.

[58] D. Balduzzi et al., "Open-ended learning in symmetric zero-sum games," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 434–443.

[59] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, and S. Russell, "Adversarial policies: Attacking deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, 2019, *arXiv:1905.10615*.

[60] H. Zhang et al., "Robust deep reinforcement learning against adversarial perturbations on state observations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 21024–21037.

[61] W. Guo, X. Wu, S. Huang, and X. Xing, "Adversarial policy learning in two-player competitive games," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 3910–3919.

[62] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: A survey," *Artif. Intell. Rev.*, vol. 55, no. 2, pp. 895–943, 2022.

[63] T. Oikarinen, W. Zhang, A. Megretski, L. Daniel, and T.-W. Weng, "Robust deep reinforcement learning through adversarial loss," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 26156–26167, 2021.

Hantao Zhao received the Ph.D. degree from ETH Zurich, Zurich, Switzerland, in 2020.

He is currently an Associate Professor with the School of Cyber Science and Engineering, Southeast University, Nanjing, China. His research interests include human–computer interaction and agent simulation.

Zhihao Liang received the B.S. degree in computer science and technology from Henan University, Kaifeng, China, in 2022. He is currently working toward the master's degree in electronic information with the School of Cyber Science and Engineering, Southeast University, Nanjing, China.

His research interests include adversarial reinforcement learning, multiagent reinforcement learning, and human–robot interaction.

Tianxing Ma received the B.S. degree in cybersecurity from the Southeast University, Nanjing, China, in 2022. He is currently working toward the M.S. degree in cybersecurity with the Institution of Information Engineering, University of Chinese Academy of Sciences, Beijing, China.

His research interests include reinforcement learning, natural language processing, and other applications of deep learning in the field of security.

Xiaomeng Shi received the B.S. and Ph.D. degrees in transportation engineering from the School of Transportation, Southeast University, Nanjing, China, in 2013 and 2018, respectively.

Currently, he is an Assistant Professor with the School of Transportation, Southeast University. His research interests include pedestrian flow characteristics analysis, urban mobility analytics, and intelligent transportation systems.

Mubbasir Kapadia received the Ph.D. degree in computer science from the University of California, Los Angeles, Los Angeles, CA, USA, in 2011.

He is an Associate Professor with the Computer Science Department, Rutgers University, Newark, NJ, USA. His research interests include autonomous virtual humans, crowd simulation, computer-assisted architectural design, computational narrative, and serious games.

Tyler Thrash was a Postdoctoral Researcher in cognitive science with ETH Zurich, Zurich, Switzerland. His work focuses on spatial cognition and navigation, emphasizing an ecological/Gibsonian approach to explain higher level cognition through lower level perceptual processes. He also develops simulations for virtual environments and mathematical models to study human behavior. He is currently with the Department of Biology at Saint Louis University, St. Louis, MO, USA.

Christoph Hoelscher received the Ph.D. degree in psychology from the University of Freiburg, Freiburg, Germany, in 2000.

He is a Full Professor in cognitive science with ETH Zurich, Zurich, Switzerland, focusing on applied cognitive science. He also leads the "Architectural Cognition in Practice" module at the Future Cities Lab Global, Singapore-ETH Centre, Singapore, Singapore, and until recently he directed the Future Resilient Systems program. His research interests include human interaction with physical, technical, and social environments, emphasizing cognitive processes and task-oriented behavior.

Jinyuan Jia received the Ph.D. degree in computer science from The Hong Kong University of Science and Technology (HKUST), Guangzhou, China, in 2004.

He is a Professor with HKUST and the Dean of the Game School, Jilin Animation Institute, China. His research interests include computer graphics, computer-aided design (CAD), Web3D, virtual reality (VR), and digital entertainment.

Dr. Jia is a member of ACM and a senior member of the Chinese Computer Federation.

Bo Liu is currently a Full Professor with the School of Computer Science and Engineering, Southeast University, Nanjing, China. Her research interests include online social networks and big data, with a focus on spammer detection in social networks, the evolution of social communities, and social influence, particularly through multiagent technology and the application of agent technology to address social network challenges. She is a member of the China Computer Federation and has served as a visiting scholar at the University of Massachusetts and the University of Tennessee.

Jiuxin Cao is a Full Professor with the Southeast University, Nanjing, China, and the Leader of Jiangsu Provincial Key Laboratory of Computer Network Technology, Nanjing, China. He has led several national projects at China and has published over 80 articles in reputable journals and conferences, including World Wide Web Conference, World Wide Web Journal, and ACM International Conference on Web Search and Data Mining. His research interests include computational society, computer networks, social computing, and cross-modality media fusion.