

RESEARCH ARTICLE

Multiagent evacuation framework for a virtual fire emergency scenario based on generative adversarial imitation learning

Wen Zhou^{ORCID} | Wenying Jiang^{ORCID} | Biao Jie | Weixin Bian

School of Computer and Information,
Anhui Normal University, Wuhu, China

Correspondence

Wen Zhou, School of Computer and
Information, Anhui Normal University,
Wuhu, Anhui, China.
Email: w.zhou@ahnu.edu.cn

Funding information

National Natural Science Foundation of
China, Grant/Award Numbers: 61902003,
61976006; Doctoral Scientific Research
Foundation of Anhui Normal University

Abstract

One of the most common solutions for the prevention of fire accidents is to conduct extensive fire evacuation drills in crowded places. However, there are multiple salient advantages to using virtual reality technology to simulate emergency solutions, for instance, saving costs and greatly decreasing uncertain risks or accidents. Therefore, in this article, a multiagent evacuation framework for complex virtual fire scenarios is proposed and effectively used to simulate a multiagent evacuation procedure to approximate the goal of fire drills in a less costly manner. Specifically, the concept of a multihierarchy agent group model is proposed; that is, the evacuation of multiple agents is separated into leader-follower and freedom modes. Additionally, several complex actions of individual humans in actual fire drills are fully considered, and a multiagent schema is presented to characterize the associated real effects. In addition, generative adversarial imitation learning is adopted to obtain the evacuation path of the leader-agent by training numerous learning epochs. Finally, extensive experiments are conducted to validate the feasibility of our proposed method. The results show that the proposed method is superior to other methods and that it realistically and reasonably shows the procedure of multiagent evacuation in complex fire emergency scenarios.

KEYWORDS

fire evacuation drills, generative adversarial imitation learning, leader-follower, multiagent evacuation, virtual reality

1 | INTRODUCTION

In many crowded places, such as university dormitories, shopping malls, and supermarkets, fire drills are commonly used to prevent fire accidents. However, these drills often face a series of problems, such as formality, the high cost of fire simulation, and potential unexpected accidents leading to casualties, that greatly limit the scale of fire drills and disrupt the actual objective of the exercise.

In recent years, the development of virtual reality technology has provided a new effective solution for fire prevention plans via virtual fire drills, which have multiple advantages, including low cost, small accident risk, simple scheme switching, and so on. Therefore, in virtual scenes, the modeling and simulation of evacuation crowds have gradually

become an active research area in which the effective means of evacuation behavior in fire emergencies can be reproduced and forms the basis for the study of crowd behavior. In practice, the evaluation of evacuation ability, the formulation of emergency plans and emergency decision making have great significance in safety management. To a certain extent, crowd evacuation behavior can be realistically simulated to viably reduce the risk of fire to evacuees.¹ There have been several hundreds of studies on crowd evacuation. In particular, multiple evacuation models have been presented to simulate evacuation behavior in various emergency conditions, such as social force models, cellular automata models, fluid dynamics, and agent-based models. These models have been successfully widely used in many crowd evacuation simulations. Although the majority of these models are from the perspective of individuals, they plan the path for each individual. More importantly, the number of calculations that these models utilize is usually very large; thus, the simulation of large-scale crowd evacuation is more difficult.² In addition, in an actual large-scale complex environment, when emergency accidents occur, people are prone to chaos, which leads to crowd congestion and results in huge losses, even casualties. Therefore, we attempt to build an emergency evacuation model to conduct drills or exercises in a large-scale complex virtual scenario, which decreases the chaos confronted in an actual emergency and helps people become familiar with the evacuation schema of a disaster scenario accumulate the necessary experiences and account for most key human behavior factors while ensuring the computability of the approach; this approach is of great significance to ensuring safety.³

Therefore, this article proposes a multiagent evacuation framework to escape fires occurring in large-scale scenes, which is used to simulate the process of crowd escape from disaster scenes. The main contributions of this article are as follows.

1. Two kinds of multiagent evacuation models are proposed, that is, leader-follower and freedom modes. Thus, there are two different agent escape processes, which avoids excessive emphasis on the relationship between agents, namely, agents absolutely trust or absolutely distrust other agents; in this way, the behavior of virtual agents approaches that of actual evacuees.
2. Two different kinds of agent behavior models are proposed. In fact, for an evacuee, there are many different evacuation behaviors, such as runs, walks, and sprints. Therefore, in this article, two kinds of agent behaviors are used, namely, runs and walks, to better simulate evacuees in disaster emergency scenarios.
3. Generative adversarial imitation learning (GAIL) is adopted to obtain the evacuation path. Specifically, when trained in long epochs and unceasingly interacting with virtual scenarios, the agent can intelligently achieve obstacle avoidance, reduce the online computing time, and effectively reach the destination safely.

2 | RELATED WORKS

2.1 | Leader-follower model

In the process of evacuation, not all individuals know the exact evacuation route. Therefore, their behaviors are not only based on their own ideas but are also greatly influenced by the people around them. Clearly, their own evaluations and behaviors are often easily influenced by group behaviors. Most people exhibit a prominent herd mentality in an emergency; few people choose to act alone, and most evacuees prefer to follow others in a submissive manner. In an emergency event, numerous familiar individuals always gather tightly together and follow a trustworthy individual to move, which is termed social force.

Many studies^{4–8} have shown that the efficiency of individual evacuation under guidance is higher, that is, the leader-follower mode. In fact, the leader-follower mode represents reality well. Therefore, it is particularly important to consider group behavior during a crowd evacuation simulation.

Pelechano and Badler⁹ developed a two-level model to simulate leader behavior in maze evacuation scenarios. The optimal ratio of trained leaders to evacuees and the influence of escape information transmission on evacuation were studied; however, it took a large amount of time to compute the optimal ratio of leaders in crowds. Qiu and Hu¹⁰ proposed a new well-defined group structure modeling framework, and an agent-based simulation system was designed to simulate crowd behavior with a group structure. However, they considered only whole group structures in crowd simulations and did not focus on any special individual evacuations, that is, lonely evacuation. In References 11 and 12, the social power model was proposed to explore the influence of leader number and position on evacuation dynamics in a limited visible range. Moreover, another study¹³ demonstrated the necessity of leaders and the influence of the

number and spatial location of leaders during an evacuation. Clearly, these studies sufficiently show that the leader-follower model greatly influences the efficacy of the evacuation process. Zhang et al.¹⁴ proposed an improved two-level social force model to simulate and reproduce the group aggregation process. Certainly, their most important purpose was to better simulate the process of forming the leader-follower model in crowds. Additionally, in Reference 15, an improved social force model in which people are grouped according to social relations, was proposed. In this process, group members are attracted by leaders to ensure that each individual follows the leader. For some complex scenes, such as reciprocal velocity obstacles in scenarios, Juniastuti et al.¹⁶ divided the role of crowds into two parts, namely, leaders and followers. In the evacuation process, group behavior based on the leader-follower model was successfully completed. This showed that the leader-follower model is suitable for evacuation simulation in complex scenarios. Subsequently, Li et al.¹⁷ presented an evacuation process tracking model. Specifically, the dynamic Douglas-Peucker algorithm was used to extract global key nodes from dynamic partial routing. Moreover, this study considered a primary school as an example to simulate the evacuation process of students following the teacher. The efficacy of the leader-follower model was found to be robust for several complex emergency scenes. In addition, in rail transit station evacuation scenarios, Zhou et al.¹⁸ proposed a hybrid bilevel model to optimize the number and initial positions of leaders in the evacuation process and the paths of leaders in the evacuation process. However, these methods utilized only a single leader-follower model to complete the crowd evacuation simulation in different scenes. In fact, there existed some lonely evacuees; thus, to maintain the verisimilitude of simulation, the crowd diversity should be fully considered. In the present article, we propose two different models to simulate the evacuation process, that is, a leader-follower model and a free model, to attempt to maintain the diversity and reality of crowd evacuations.

2.2 | Several different evacuation model methods

To date, due to the lack of data from real evacuation, many phenomena and laws in the interaction between evacuees and their environment can only be represented by modeling.¹⁹ To explore the behavior characteristics and movement rules of crowd evacuation during an emergency, various models have been proposed, such as agent-based models,²⁰ cellular automata models,²¹ flow-based models,²² fluid dynamics models,²³ gas dynamics models,²⁴ particle system models,²⁵ and social force models.²⁶

Recently, agent-based model technology has been widely adopted to study crowd evacuation in various situations. In practice, this usually requires a higher computation cost than cellular automata and particle systems. However, this approach allows and encourages each individual to have the ability to act independently. As a result, it becomes easier to model different individuals in more diverse scenes.²⁷ Specifically, Wagner and Agrawal²⁸ proposed an agent-based crowd evacuation simulation system to simulate crowd evacuation in the case of fire disasters. In addition, they studied the evacuation performance in a variety of disaster scenarios. In this simulation system, each individual is an independent individual that remains in an independent state rather than forming a group. The advantage of this method is that it is suitable for various scenes. Furthermore, Shimada et al.²⁹ developed a simulation system to optimize the sign system design of large-scale public facilities by using an agent-based model to simulate pedestrian movement under the influence of signs. Fu et al.³⁰ obtained better results by integrating multiagent and cellular automation models. In Reference 31, a multiagent simulation collision avoidance system in a complex environment was proposed, and its application to crowd evacuation behavior was presented. Additionally, a computation framework to simulate human and social behaviors for egress analysis was presented by Pan et al.;³² the MASSEgress system was proposed to conduct simulation based on K-Means clustering analysis; however, the number of agents is less than 100, and it mainly aimed to simulate indoor scenarios, such as train stations, buildings, and so forth. Zheng et al.³³ systematically summarized crowd evacuation models based on seven methodological approaches, and the advantages and disadvantages of these approaches were discussed. Chu and Law³⁴ proposed a multiagent-based simulation framework that incorporates human behaviors; this method also aimed to simulate indoor scenarios; it only considers the effect of social behaviors on the agent egress evacuation, but does not harness the individual agent scenario decision.

Multiagent models have gradually become one of the most widely used models. Compared with other model methods, these models have several salient advantages: first, they can better handle the interactions between people in the real world; second, they allowing modeling for each individual agent.

In addition, with reinforcement learning methods rapidly developing and milestone achievements being obtained in several important areas, such as Go, this approach is gradually being used to solve the emergency evacuation problem.

For example, Zheng and Liu³⁵ proposed the agent path plan method based on the deep deterministic policy gradient (DDPG), which can effectively obtain an optimized path to evacuate by lasting learning in continuous observation space. However, this approach is not suitable for multiagent scenarios. Wang et al.³⁶ proposed an improved method by integrating DDPG learning methods³⁷ and a social force model to complete path plans. However, this method was found to be unsuitable for large-scale scenarios; above all, it continuously consumes memory and always takes a very large time to compute.

Recently, there have been multiple remarkable breakthroughs in generative models based on deep learning. Above all, a new generative model called generative adversarial networks (GANs) addressed the inherent difficulties of deep generative models associated with intractable probabilistic computations in training. GANs employed an adversarial discriminator to distinguish whether a sample is from real data or from synthetic data generated by the indicated generator. Specifically, the competition between the indicated generator and the powerful discriminator was designed as a minmax game. Moreover, GAIL, which was proposed by Ho and Ermon,³⁸ utilized a combination of the inverse reinforcement learning (IRL) idea that learns the experts' underlying reward function and the schema of the generative adversarial framework. Indeed, GAIL has been adopted in multiple literatures, such as, Song et al.,³⁹ Choi et al.,⁴⁰ and Chi et al.⁴¹

As above mentioned, in this present article, we adopt the GAIL method to gain a suitable evacuation path; furthermore, the leader-follower model and free model are utilized to simulate the evacuation process of multiple agents.

3 | PROPOSED FRAMEWORK

In this section, as Figure 1 shows, an overview of the proposed framework is presented. Our proposed framework is composed of three parts; that is, virtual disaster scenario generation, and the multiagents form a related

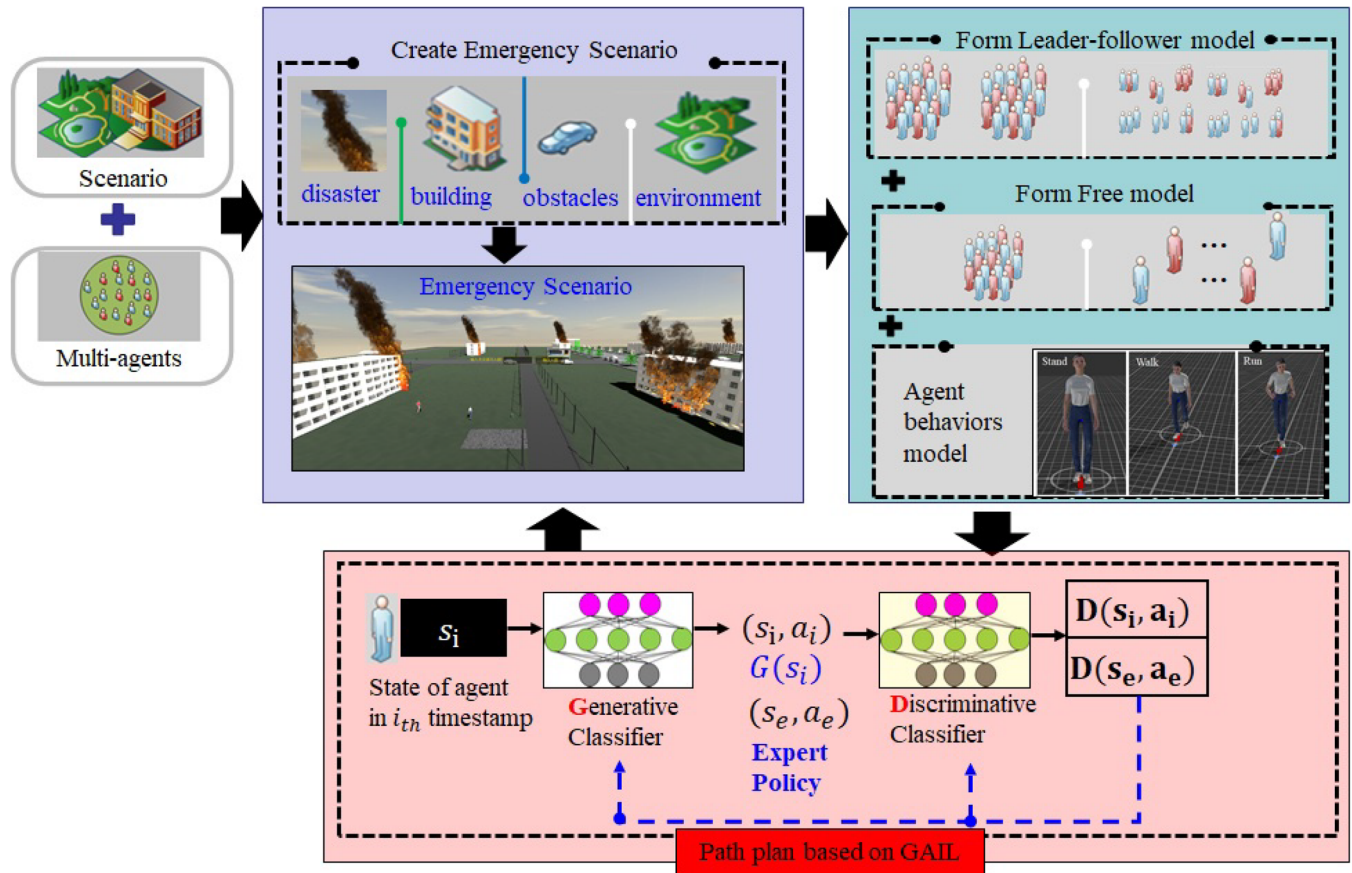


FIGURE 1 Overview of the proposed multiagent evacuation. The framework consists of three parts, that is, the create emergency scenario. A related model is generated from multiple agents, and the GAIL approach is used to train the related evacuation path

structure. Specifically, the agent grouping approach is considered to obtain a leader-follower model; moreover, to maintain reality and diversity, some lonely agents exist, that is, agents based on a free model. In addition, two different agent behavior modes are considered to better simulate the evacuation process of evacuees, specifically, the run agent and walk agent. In addition, by using sufficiently long training epochs, we can easily ensure that leader agents and lonely agents find the most suitable path to escape based on GAIL methods. In addition to leader agents and lonely agents, for many follower agents, because they follow the leader agent only, they do not need to gain evacuation paths themselves (in essence, most of the existing agents belong to the follower-agent type). Thus, the whole calculation of the evacuation is not very large; moreover, this approach aims to maintain the reality of virtual evacuation scenes.

In the following section, the details of the proposed framework are described.

4 | FRAMEWORK DESCRIPTION

4.1 | Agent grouping approach

As mentioned above, the leader-follower model can effectively simulate the process of actual scenes. In this section, we propose a belief value agent grouping approach. In addition, we assume that the trustworthiness and knowledge of an agent are closely related to the agent's distance from the egress.

Definition 1. The global trust value of Γ_i of agent i th a_i : The trustworthiness of agent a_i in the current scene is determined by the distance D_i to the escape target (exit) and the knowledge K_i . The equation of D_i is shown in Equation (1).

$$D_i = \frac{\min_G \{d(a_i, G)\} - \min_{0 \leq j \leq N} \{\min_G \{d(a_j, G)\}\}}{\max_{0 \leq k \leq N} \{\min_G \{d(a_k, G)\}\} - \min_{0 \leq j \leq N} \{\min_G \{d(a_j, G)\}\}}. \quad (1)$$

In this equation, the term G is the goal position set of the evacuation, that is, the exit (in most cases, there are probably multiple exits). Moreover, the function $d(a_i, G)$ represents the Euclidean distance set between the position of agent a_i and the set G .

The knowledge K_i of the i th agent a_i is shown in Equation (2).

$$K_i = \frac{1}{2\pi\sigma_x\sigma_z} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(z-\mu_z)^2}{2\sigma_z^2}}. \quad (2)$$

Clearly, for multiagents $A = \{a_0, \dots, a_{N-1}\}$, the terms $\sigma_x, \sigma_z, \mu_x, \mu_z$ are represented as the variance and mean of the initial position in the X - and Z -axis directions, respectively. $K_i \in [0, 1]$; therefore, the equation of the global trust value Γ_i of the i th agent a_i is denoted in Equation (3):

$$\Gamma_i = \alpha D_i + (1 - \alpha) K_i. \quad (3)$$

The number of leader agents depends on the number of agents in the scenario. In general, the greater the number of agents, the more leader agents are likely to be created to simulate a realistic scenario. Intuitively, this is likely to be consistent with realistic drill scenarios. However, when every leader agent seizes its own evacuation path, too many leader agents will probably also lead to more congestion. In such cases, the agent evacuation may often be greatly delayed; as a result, it takes multiple agents more time to complete the evacuation task. Indeed, with the increasing of the agents, the number of leader agents probably becomes constant. Hence, to appropriately simulate realistic occasions, in this article, we do not consider of the situation in which there are too many agents (i.e., $N > 1000$).

Then, the number of leader agents n in the scene is computed as shown in Equation (4).

$$n = \begin{cases} \theta \times \frac{N}{u} & \text{if } N \leq 10 \times u, \\ \varphi \times \frac{N}{v} + \tau & \text{if } 10 \times u < N \leq 10 \times v, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Here, the term N is the number of agents in the scenario. Theoretically, θ and φ present the ratio of leader agents in the respective scene; in addition, the term τ is understood to represent the minimum number of leader agents. Through the relative experiment, when the number of agents $N \leq 1000$, $\tau = 3$, $\theta = 0.5$, $\varphi = 0.5$, $u = 10$, $v = 10$, the best effect is achieved. Additionally, the various agents are uniformly generated in the initialization phase.

For agents A ($|A| < 1000$), $\forall a_i$, and its trust value Γ_i , we assume that the term L represents n leader agents.

Consequently, the remainder of agent $r_j \in R = A - L$ and its trust value t_{jk} to leader agent $l_k \in L$ (local trust value) can be denoted by Equation (5):

$$t_{jk} = \beta \times \frac{1}{d(r_j, l_k)} + (1 - \beta) \times H, \quad (5)$$

where the variable H is a random value, that is, $H \in [0, 1]$; specifically, this disturbance factor prevents the local trust values from being too concentrated, which often leads to overfitting.

In addition, to a certain extent, the agent r_j grouping depends on the local trust degree and threshold value ς , which is mainly used to maintain the diversity of agents; that is, in reality, some existing agents form a free mode to escape without a leader agent. This is shown in Equation (6).

$$\Theta_j = \begin{cases} \operatorname{argmax}(\{t_{jk} > \varsigma | t_{jk}\}) & \text{if } \exists t_{jk} > \varsigma, \\ -1 & \text{otherwise.} \end{cases} \quad (6)$$

Finally, for each group, the term Θ_j uses the k-means algorithm to update the relationship between agent r_j and the leader l_k , that is, it is harnessed to decide whether or not the agent r_j followed l_k . Furthermore, the agent grouping algorithm is shown as follows in Algorithm 1.

4.2 | Agent behavior selection

To simulate the escape process of multiple agents more realistically, it is very important to control the behavior of the agent. As Figure 2a shows, the behavior set of agent $\Phi = \{i \in [0, 2] | w_i\}$ indicates that the behaviors of the agents consist of the standing, walking and running states. Clearly, the velocity V (metric units is m/s) of the agents corresponds to the behavior set.

In an actual fire drill, in the process of crowd escape and evacuation, some participants in the drill have different postures, such as running and walking (as shown in Figure 2b). Maintaining the multiple behaviors of agents increases the fidelity and authenticity of virtual reality. Therefore, this article proposes a multiagent behavior selection method.

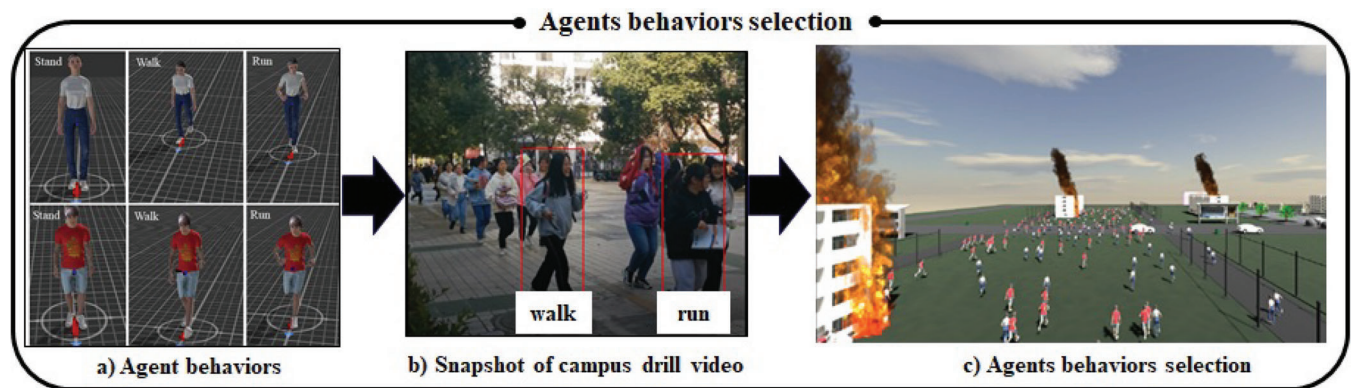


FIGURE 2 Overview of agent behavior selection. (a) There are three different behaviors. (b) In an actual campus drill, from a snapshot of a video, there are two different evacuation behaviors. (c) Agent behavior selection is used in the virtual scenario

Algorithm 1. Agent grouping algorithm

Input: Leader agent set L ; remaining agent set R
Output: L , follower agent set F ; free agent set T

```

1:  $F \leftarrow \emptyset, T \leftarrow \emptyset$ 
2: for  $l_i \in L$ 
3:    $F_i \leftarrow \emptyset$ 
4:   for  $r_j$  in  $R$ 
5:     According to Equation (6),  $\Theta_j$  is computed.
6:     if  $\Theta_j == i$ 
7:        $F_i \leftarrow r_j$ .
8:     else
9:        $T \leftarrow r_j$ 
10:    end if
11:  end for
12:   $F \leftarrow F_i$ 
13: end for
14: for  $l_k \in L$ 
15:   Compute the average distance  $\Delta_k$  between  $l_k$  and  $F_k$ .
16:   for  $a_i \in F_k$ 
17:     Compute the average distance  $\Delta_{ik}$  between  $F_k$  and  $a_i$ .
18:     Compute the distance  $\Delta_{i0}$  between  $a_i$  and  $l_k$ 
19:     Compute  $\Delta_i = \frac{\Delta_{ik} + \Delta_{i0}}{2}$ 
20:     if  $\Delta_i < \Delta_k$ 
21:        $a_i \longleftrightarrow l_k$ 
22:     end if
23:   end for
24: end for

```

Definition 2. Density of agents: the number of agents in the fixed neighborhood δ^+ of a collision point. Then, the steps of the agent behavior selection method are listed as follows:

- Step 1: Initialize the behavior state of N agents, \forall agent a_i , while its behavior state $\Phi(a_i) = \text{standby}$; then, $V(a_i) = 0$.
Step 2: The behavior state of multiple agents is preprocessed, and a random number $\rho[0, \theta]$ is generated randomly. If ρ equals 0, $\Phi(a_i) = \text{walking}$. Otherwise, $\Phi(a_i) = \text{run}$. In this article, the parameter $\theta = 9$.
Step 3: Preprocessing the behavior of leader agent l_i . To maintain the leader agent's leadership effect, the behavior of the leader agent l_i has to be set so $\Phi(l_i) = \text{run}$.
Step 4: When \exists agent a_i ($a_i \notin L$), $\text{density}(a_i) \geq \epsilon$, and $\Phi(l_i) = \text{run}$, then $\Phi(l_i) = \text{walk}$.

Actually, for agent a_i , its velocity V_i at timestamp t th satisfies Equation (7), as follows:

$$V_i = \begin{cases} 0 & \text{if } \Phi(a_i) = \text{standby}, \\ v_0 + \Psi & \text{if } \Phi(a_i) = \text{walk}, \\ v'_0 + \Psi' & \text{otherwise,} \end{cases} \quad (7)$$

where the variables v'_0 and v_0 represent the initial speed in the run state and walking state, respectively. In this article, we set the velocities $v'_0 = 1.0$ m/s, $v_0 = 0.9$ m/s, respectively. Additionally, the terms Ψ' and Ψ represent the accelerated speed of the current agent in its own state. Furthermore, the accelerated speed is usually limited by two factors, namely, the amount of knowledge of the current agent (see Equation 2) and the density of the agents (see Definition 2). In addition, the accelerated speed Ψ_i of agent a_i is shown in Equation (8).

$$\Psi_i = \max\{0, \mu \times K_i - f(\Delta(a_i) - \epsilon)\}. \quad (8)$$

Here, the term $\epsilon = 5$ refers to the experiment value used in this article. Additionally, the function Δ denotes the density of agent a_i within a unit area, that is, how many agents are in a unit area. Obviously, the term ϵ represents the common number of agents in a unit circle centered on agent a_i , including the agents in four different directions and the agent a_i . If the behavior state of agent a_i is in the walking state, that is, $\Phi(a_i) = \text{walking}$, then $\mu = 1.5$; otherwise, $\mu = 2.0$. In addition, the function f is an activation function, which is denoted as Equation (9):

$$f(x) = \min\{\max(0, x), 1\}. \quad (9)$$

To ensure that the agent maintains a certain acceleration, for agent $\forall a_i$, Ψ' is greater than 0, and the value is set as shown in Equation (10).

$$\Psi'_i = \max(0.1, \Psi_i). \quad (10)$$

4.3 | Agent path planning based on GAIL

The target of GAIL is to obtain rewards from the existing expert policy π_E and then train the GAN to obtain an approximate policy. Therefore, it is very import that expert policy π_E is obtained; unfortunately, currently, no open expert data are available. For this reason, we collect the expert policy process based on fire drills in campus videos, and then the result is saved in csv format files.

Specifically, the generator network (G) is utilized to obtain an approximate expert policy, that is, the foremost role of the generator is to create realistic synthetic trajectories; subsequently, the discriminative classifier network (D) is trained to obtain suitable parameters based on the mini-batch stochastic gradient descent (SGD) optimizer. More specially, the discriminator (D) attempts to solve the classification problem by distinguishing real expert trajectories from generated trajectories.

Indeed, the generator network (G) consists of a policy function network and value estimator network, and the discriminator (D) is comprised of a discriminator network; moreover, for the discriminator (D), a primary target is to output the reward value $R(s_t, a_t)$ of the synthetic trajectory (s_t, a_t) , which serves to bring the training value network to convergence through a multiple iteration roll-out. In other words, the generator (G) works as a reinforcement learning agent to produce the policy function and value estimator, whereas the discriminator works as an IRL representative to obtain a related reward of the synthetic trajectory of state s_t and action a_t . Furthermore, by applying the policy steps, the related cost function is adopted based on mini-batch SGD to obtain a series of approximate policies. An overview of GAIL is intuitively summarized in Figure 3.

In reinforcement learning, a value function $Q_\pi(s_t, a_t)$ is often used to figure out the expected return of the actions a_t at the current state s_t . In particular, the expected return, or the estimated value, which is created by the value estimator (i.e., the value network), is used as a coefficient when updating the policy generator. Moreover, the value network is trained to minimize the value objective loss function J_{value} , which is defined as Equation (11):

$$\begin{aligned} J_{\text{value}} &= E\{[Q_\pi(s_t, a_t) - (R(s_t, a_t) + \gamma E(R(s_{t+1}, a_{t+1})) + \gamma^2 E(R(s_{t+2}, a_{t+2})) + \dots)]^2\} \\ &= E\{[Q_\pi(s_t, a_t) - (R(s_t, a_t) + \gamma \sum \pi(a_{t+1}|s_{t+1}))Q(s_{t+1}, a_{t+1})]^2\}, \end{aligned} \quad (11)$$

where the term γ is the actual discounted value. Additionally, the term $R(s_t, a_t)$ is the reward value from the output of the discriminator network (D). Meanwhile, by taking a series of policy steps θ , the objective of the policy update is to maximize the expected cumulative reward function; hence, the loss function J_{policy} of the policy network maximized the policy objective to enhance the policy generator at every iteration. Therefore, the policy network is increasingly updated to maximize the loss function J_{policy} with the related gradient as Equation (12) shows:

$$\begin{aligned} \nabla_\theta J_{\text{policy}}(\theta) &= \hat{E}_{r_i} [\nabla_\theta \log \pi_\theta(a|s)Q(s, a)] - \lambda \nabla_\theta H(\pi_\theta), \\ \text{Subjected to : } Q(\bar{s}, \bar{a}) &= \hat{E}_{r_i} [\log(D_{w+1}(s, a))|s_0 = \bar{s}, a_0 = \bar{a}], \end{aligned} \quad (12)$$

where the term r_i is the reward in i th timestep, and θ is the parameters of the policy network. Additionally, among the below-constrained equation, the term w is the parameters of the discriminator (G), and these terms $s, a, \tilde{s}, \tilde{a}$ represent the state of the action of the current timestep and the state of the action of the next timestep, respectively.

Indeed, as the generator is improved to produce more realistic trajectories, the discriminator's ability to classify the generated trajectories from the real trajectories is also significantly improved through iterative parameter updates and fine-tuning. In essence, this competition between the two neural networks (G and D) is the fundamental concept of the generative adversarial learning framework. In particular, for every update step of the discriminator, the samples from the real trajectory dataset (i.e., expert data) are labeled as 0, by comparison, the samples from the generator are labeled as 1. Therefore, for both expert and generated trajectories, we put the sequence observation and the action taken at the last observation as an input, process the sequence of observation into a belief state by a multilayer perceptron (MLP) embedding layer, and calculate the probability ($D_w(s, a)$) that the given sequence of observations and the action are from the generator (G).

Clearly, for the discriminator D and the expert policy π_E , the parameters of the discriminator are increasingly updated to minimize the binary cross-entropy loss. The w -parameterized discriminator is updated to minimize the discriminator loss function J_{discr} , with the following gradient, as Equation (13) shows:

$$\nabla_w J_{discr} = \hat{E}_{r_i} [\nabla_w \log(D_w(s, a))] + \hat{E}_{r_E} [\nabla_w \log(1 - D_w(s, a))]. \quad (13)$$

On the other hand, the discriminator (D) generally provides the training signal to the generator (G) through the reward function ($R(s_t, a_t)$), as shown in Figure 3. In comparison, the generator (G) is always trained to achieve the target of maximizing the binary cross-entropy loss of the discriminator (D). In fact, the parameters of the generator (G) are closely related to the first term of Equation (13), that is, the primary objective of the generator (G) is to maximize the first term of Equation (13). As a result, the reward function can be easily represented in Equation (14), as follows:

$$R(s_t, a_t) = -\log(D_w(s, a)). \quad (14)$$

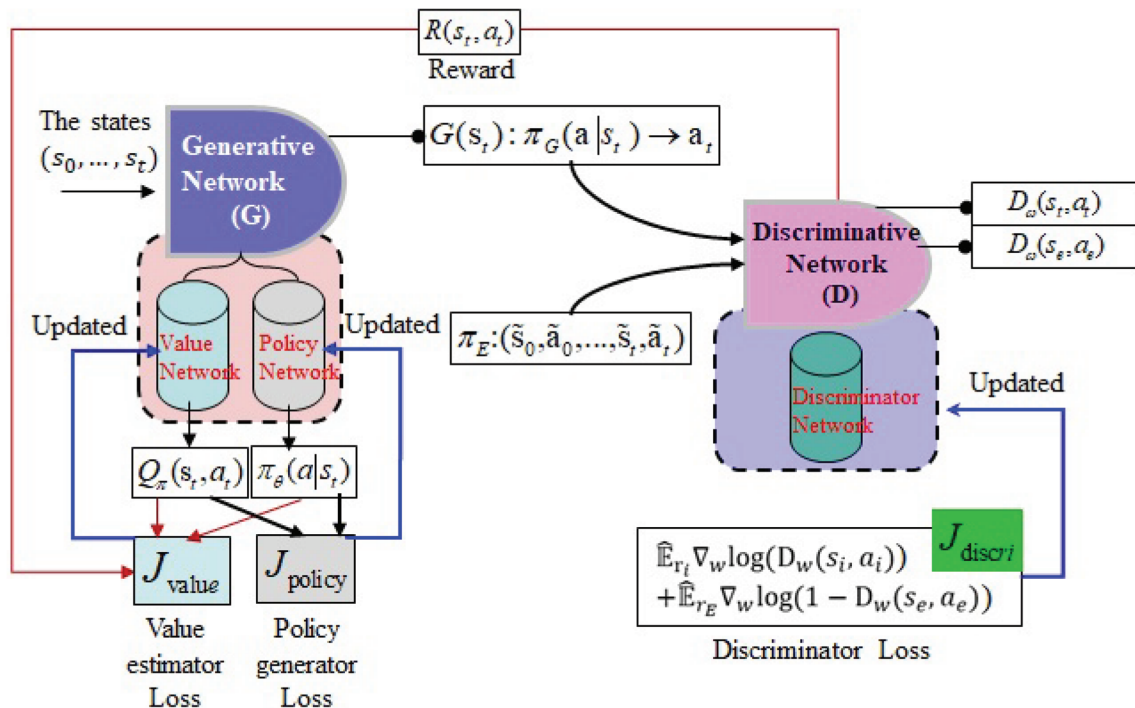


FIGURE 3 Overview of the framework of generative adversarial imitation learning. Specifically, the target of the training process is to make each network (including the value network, policy network, and discriminator network) reaches convergence. In this case, the output policy based on such a convergent policy network can obtain maximization of the value function

5 | EXPERIMENTS

5.1 | Setting the experiment environment

To validate our proposed framework, we set the related experimental environment, including a virtual fire scenario, 3D model actors (agents), several obstacles (buildings, cars, trees), and two evacuation exits (two safety exits). Specifically, we assume the scenario to be a campus where fires occur in dormitories. In addition, the 2D plane of the virtual campus is shown in Figure 4. Furthermore, the configuration of the computer is shown in Table 1.

Additionally, in this article, for the value estimator network, policy network, and discriminator network, the same network structure is adopted, that is, multilayer perceptron (MLP) with two hidden layers. Additionally, the primary parameters of GAIL are shown in Table 2.

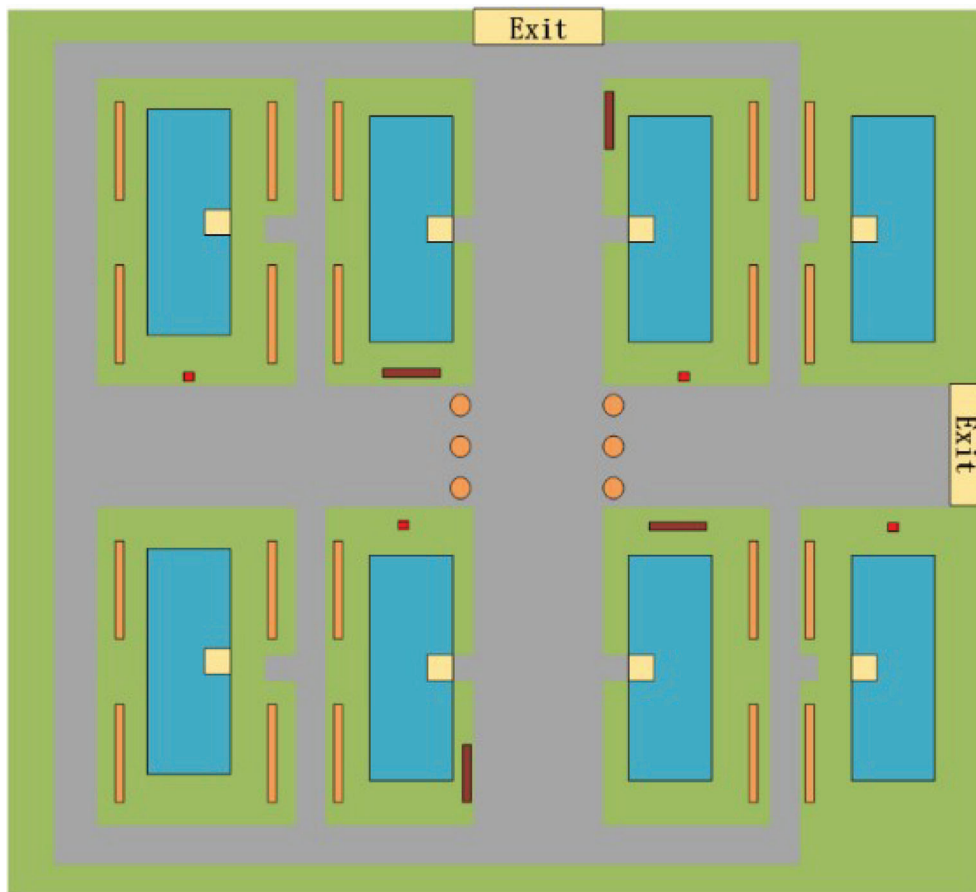


FIGURE 4 2D plane of the virtual environment. The orange and red bars are the obstacles, and the blue blocks denote the dormitories struck by fire accidents. In addition to these, there are two safety exits

TABLE 1 Main configuration of the computer

Operation system	Windows 10
Processor	Intel Pentium Dual Core I7
Graphics card	NVIDIA GeForce GTX 1060
Memory	32 GB
ML framework	Tensorflow 2.4
Physic engine	Unity 3D

TABLE 2 Main configuration of the presented GAIL network

Config	Value
Training epochs	100,000
Size of expert batch	2000
Learning rate	1e−4
Network	Two-layer perceptron [256, 256]
Optimizer	Adam

**FIGURE 5** Set the number of agents for the evacuation environment in the beginning timestep

5.2 | Results and analysis

According to the above experimental environment, we have completed the related experiments, including a multiagent obstacle avoidance experiment, a leader-follower model, an agent path planning based on the GAIL network (hereinafter referred to as the L-F model), and an agent behavior choice test. In addition, as an important criterion, the evacuation time of existing agents is used for quantitative analysis to verify the feasibility of our proposed method.

In the beginning, while the number of agents is inputted, a given number of agents are effectively generated to complete the creation of the virtual scenario. Figure 5 shows the beginning evacuation scenario that awaits the inputting of the number of agents.

Furthermore, Figure 6 shows the automatic obstacle avoidance process of multiple agents based on the GAIL training model, which can greatly save online calculation time in comparison to a traditional plan algorithm, such as the A* method. Moreover, the planning path of a leader agent is shown, as are various follower agent movement trajectories in L-F mode. In addition, in Figure 7, the paths of multiple leader agents are shown to more intuitively present the evacuation trajectory of agents in L-F mode and to verify the feasibility and robustness of the agent behavior selection approach, that is, that this behavior is reasonable and that walking and running agents exist. In fact, in the actual scene of a fire drill, there are several different crowd behaviors, namely, walking evacuees and running evacuees. To maintain verisimilitude, it is necessary to keep two different evacuation behaviors in the scenario.

The left part of Figure 8 shows the escape process of the agent in the complete walking state (ideal state), whereas the right part of Figure 8 shows the escape process of the agent in the mixed state (closer to reality). To quantitatively analyze the evacuation process of the existing agents, we use the evacuation time as the evaluation criterion. In the real scene, there is less escape time without considering casualties, which means that the exercise plan is better. Nevertheless, because the initial position of the agent is randomly generated in our experiment, there are some errors in the experiment,

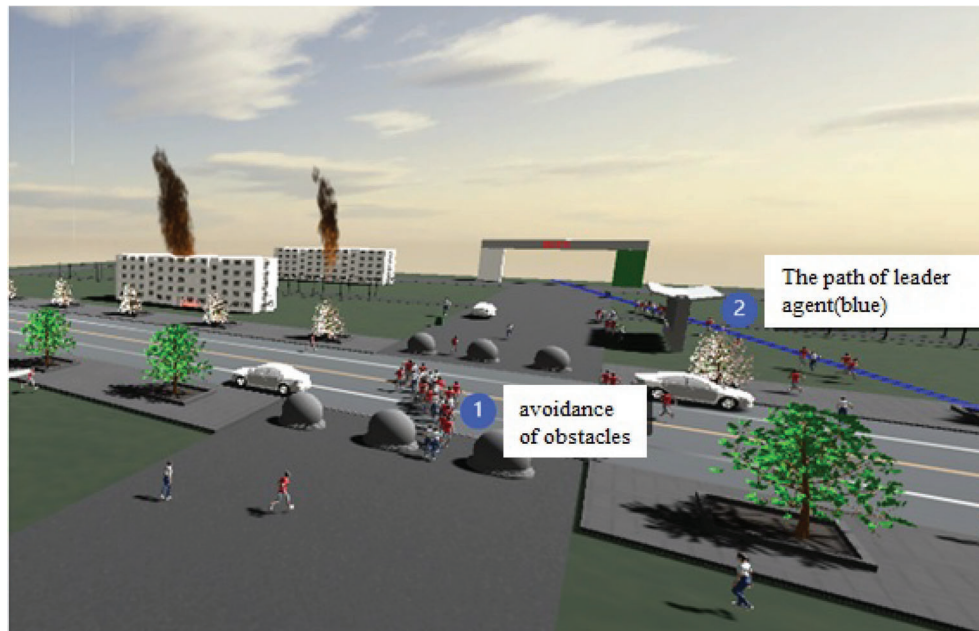


FIGURE 6 Smartly avoid obstacles, and show a leader agent's path (blue line)

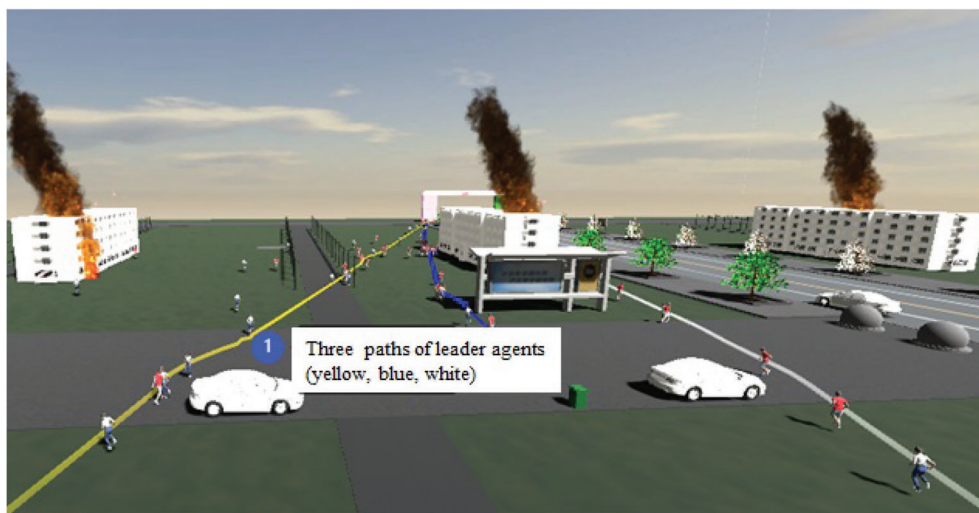


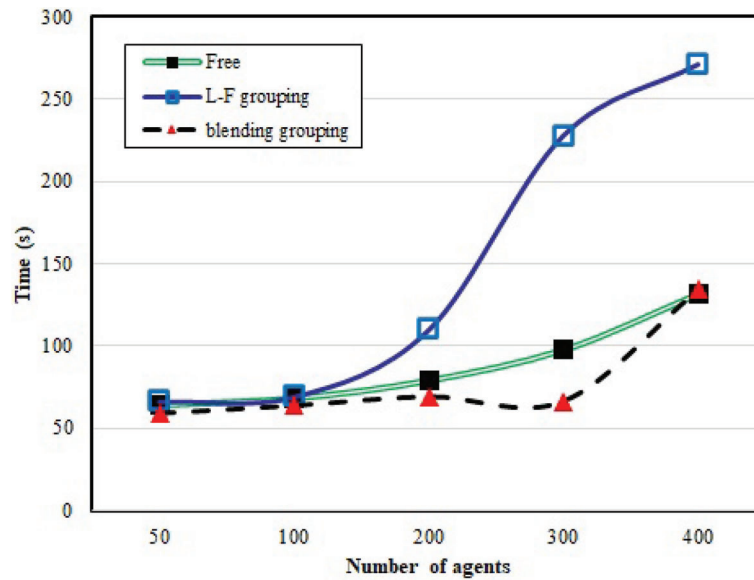
FIGURE 7 Several leader agent evacuation paths based on the L-F grouping schema



FIGURE 8 Comparison between agents with walking behavior and blending behaviors (running and walking) in the scenario

TABLE 3 Evacuation time comparison between different agent grouping schema (unit is seconds)

Number of agents	Free mode	L-F mode	Blended mode
50	64.01	66.5	59.17
100	68.16	69.35	63.89
200	79.05	109.88	69.36
300	97.81	227.61	66.54
400	131.94	270.71	134.13

**FIGURE 9** Evacuation time comparison between different agent grouping schema**TABLE 4** Evacuation time comparison between different behavior schema (unit is seconds)

Number of agents	Running behavior	Walking behavior	Blending ($\theta = 1$)	Blending ($\theta = 4$)	Blending ($\theta = 9$)
50	57.68	70.14	66.91	74.1	59.17
100	58.97	78.72	78.37	71.13	63.89

but this does not affect the overall results. Hence, the experimental results do not consider the error caused by the random position of the agent.

Table 3 shows the evacuation time of 50–400 agents under different grouping strategies. In the blended mode, there are certain advantages in evacuation time; the results are in line with the real scene.

Moreover, Figure 9 shows the comparison of the evacuation time of the above different agent groups. The blended mode is probably the best scheme under the evacuation time indicator. In Step 2 of Section 4.2, the running and walking behavior selections of agents are heavily determined by the parameter θ . Specifically, the larger θ is, the fewer walking agents exist. In our experiment, we take $\theta = 1$, $\theta = 4$, and $\theta = 9$ to represent 50%, 20%, and 10% of walking agents, respectively (to maintain the existence of the L-F mode, for the leader agents, we assume that their behavior consists of running to maintain the leader position). In addition, we compare the differences in the evacuation time between running agents and walking agents (except for leader agents). The results are shown in Table 4 and Figure 10. Figure 10 shows that when $\theta = 9$, the evacuation time is very close to that of the state when all agents maintain the running behavior mode. Objectively, such a case is associated with both a lower attempted evacuation time and a priority on the relative authenticity of

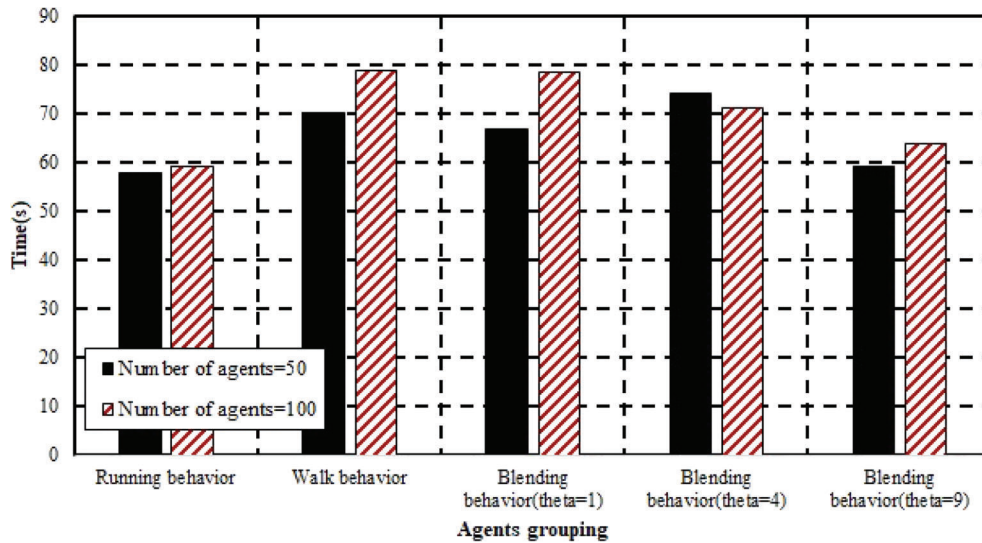


FIGURE 10 Evacuation time comparison between different action schema when the numbers of agents are 50 and 100

the scenario. Subsequently, we conduct further experiments on agent grouping. In the previous experiments, we showed that the best results are based on the blended grouping. However, how to effectively divide the L-F mode and free mode is the object of our following experimental evaluation. In Equation (6), we utilize the threshold value ζ to control the ratio of the two modes. As Table 5 shows, we set the evacuation time in the range of ζ from 0.1 to 1. The smaller ζ is, the more agents in the L-F mode are in the blended mode. Interestingly, when $\zeta = 1$, the blended mode is close to the completed free mode, whereas when $\zeta = 0$, the blended mode is close to the completed L-F mode.

Figure 11 shows the evacuation time of 200 agents under different threshold values ζ . When $\zeta = 0.4$, the blended mode is likely to approximate the best result because the evacuation time is smallest. Indeed, these agents in the L-F

TABLE 5 Evacuation time comparison between 10 different values of threshold ζ when the number is 200 (unit is seconds)

0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
118.13	92.9	82.27	78.94	84.27	85.19	91.83	81.72	82.7	82.46

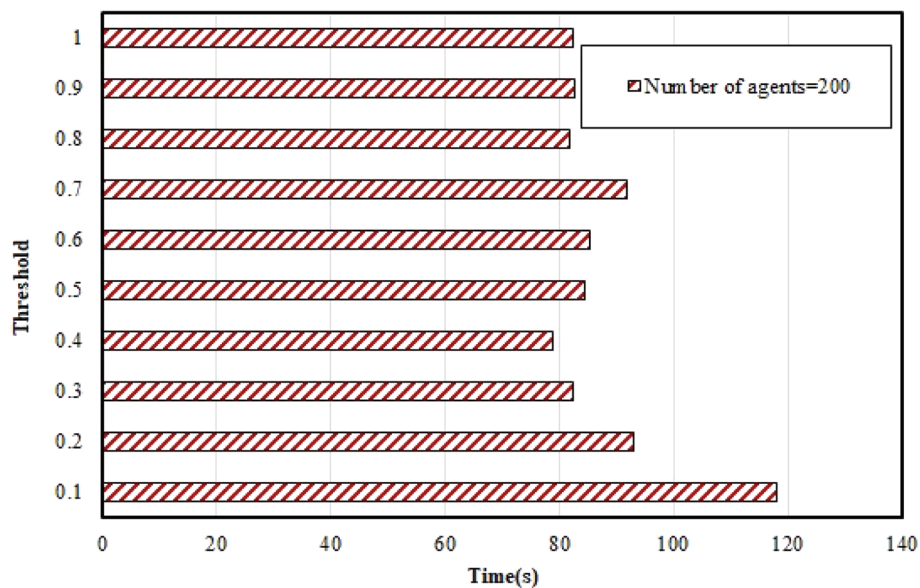


FIGURE 11 Evacuation time comparison between ten different values of threshold ζ when the number is 200

mode are too concentrated, so congestion may occur; almost certainly, the evacuation time is affected. Nonetheless, too many agents in free mode means that more calculation workload and inference of the planning path based on the GAIL training model are needed, which also delays the evacuation time to some extent. Therefore, $\varsigma = 0.4$ produces the best results.

Moreover, Figure 12 shows the result of GAIL training. There are three indicators, that is, entropy, loss, and cumulative reward based on the environment. The training epoch of GAIL is ten million to make the agent in any place find the best path and avoid related obstacles.

To validate the superiority of path plan schema based on GAIL with traditional methods, such as the A* method, time consumption is utilized as an indicator, and the result is shown in Figure 13. Figure 13 intuitively shows that regarding the comparison result, it is not difficult to draw a deterministic conclusion that the GAIL method obtains a superior result to the traditional A* method; while the population size of agents is less than 100, the average computation time using the A* method is relatively shorter. The reason can be summarized as follows: agent congestion plays a role in the evacuation process. Specifically, when the number of agents is few, agent congestion does not easily occur or its scale is small; in this case, as a distance optimization A* method, there may be a certain advantage. Nevertheless, when the population of agents becomes large enough, the agent congestion situation has to be considered; in this manner, the average computation time is certainly affected by such congestion. Figure 14 shows the agent evacuation process of the virtual scene, which can be seen in the main different stages of the emergency evacuation exercise simulation process. In Figure 14a, evacuation begins. The top left corner shows that the total number of agents in the scene and the number

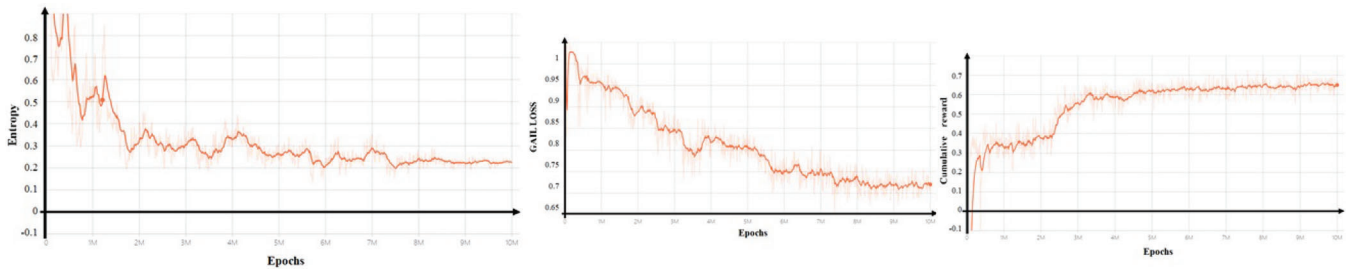


FIGURE 12 For the three indicators (entropy, loss, cumulative reward), the GAIL training results are shown

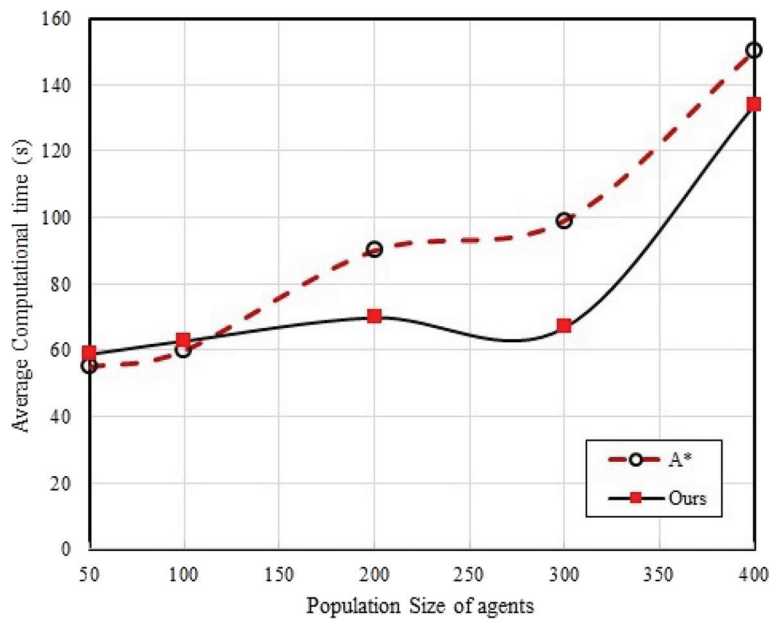


FIGURE 13 Comparison between the A* method and our presented GAIL approach regarding the average computational indicator while the emergency evacuation task is conducted

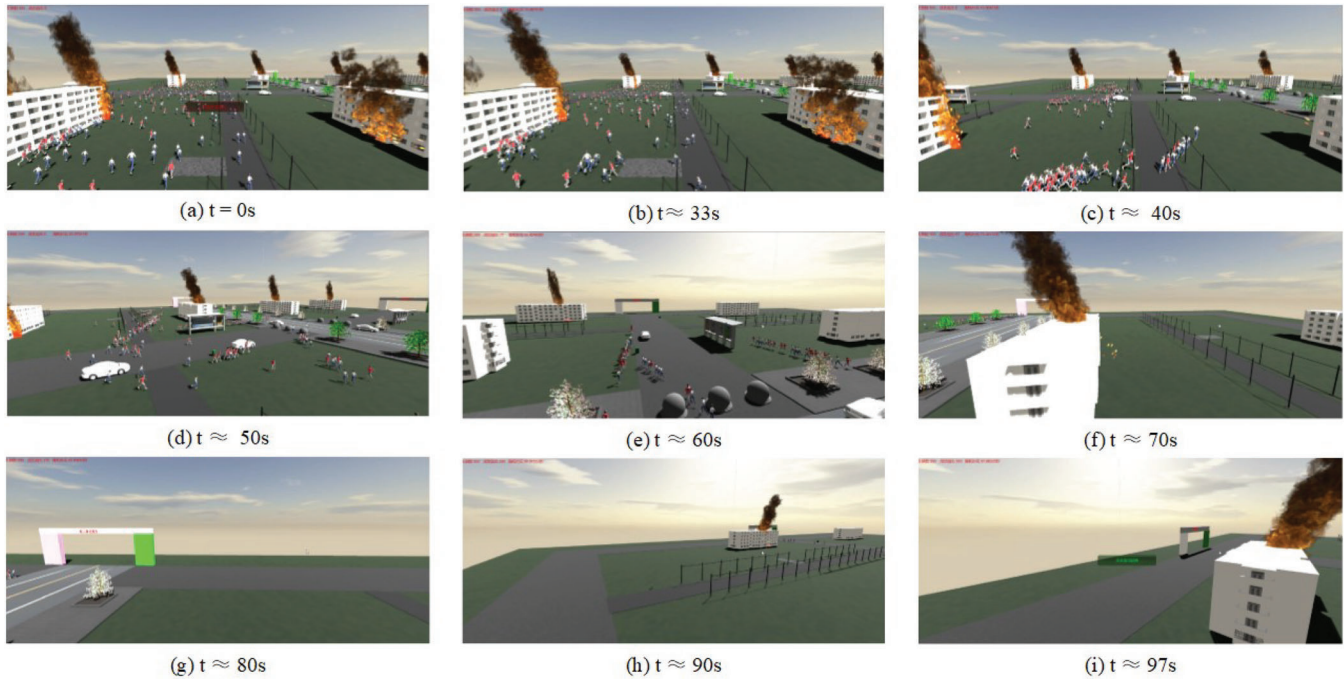


FIGURE 14 Three-hundred-agent evacuation simulation process, shown from (a) to (i)

of successful evacuations are 300 and 0, respectively. In Figure 14b, most of the agents are “aware” of the fire, and the agents who are far from the exit try to follow the leader agent in front of the exit position. The agents find that there is a time delay from the fire to the reaction to evacuate; in fact, the scene is relatively large, and the existing agents are located far from the exit. In Figure 14c,d, some agents who find their leader agents in the escape process ultimately form the L-F model agent. Here, we set up only two exits. Due to the different directions of the exits, agents mainly evacuate from these two exits. Subsequently, different evacuation queues naturally form. In Figure 14e, the agents in the same group are close to each other and have the same goal. They show a tendency to act together to avoid obstacles. The leader agent is at the forefront of the team and leads many follower agents to evacuate. In Figure 14g,h, the agents who do not finish evacuation increasingly move toward the two exits. Finally, the number of successful escapes in the scene is 230 and 288. In Figure 14i, all existing agents in the scene have successfully evacuated, and the fire drill is over.

5.3 | Limitation

Although we can obtain good results based on the proposed framework, there still exist dozens of works and bottlenecks that need to be addressed. Specifically, they are as follows. First, the expert data are lacking diversity, which leads to a lack of enough fidelity of the simulation. To solve this problem, in the future, more expert data should be built to enhance the diversity. Second, the grouping process of the agents is static; there exist some special occasions that are not considered, such as follower agent defection so as to reselect the group. Third, more emergency scenarios should be tested in our proposed framework, such as the city flood scenario, earthquake scenario, violence escape scene, and so on.

Additionally, there is much extended work to perform. One such area of research, for example, is based on a deep learning method to automatically build disaster scenarios instead of manual establishment. In this way, the generality of our proposed method has further enhanced and applied more scenes to create a virtual visualization decision platform.

6 | CONCLUSIONS

In this article, a multiagent evacuation framework based on blended grouping and blended behavior patterns is proposed. Importantly, the GAIL method is trained to obtain a related model that is utilized for path planning inference.

To some extent, this method greatly reduces the online computing time and improves the evacuation efficiency. Moreover, by building a 3D simulation environment to simulate the virtual fire scenario, the simulation results and extensive experiments are given to validate the feasibility of our proposed framework.

However, this article considers only the evacuation time and does not consider the casualties of the agents. Moreover, the obstacles in the scene are static, and our framework is very effective. In follow-up work, we plan to attempt more complex scenarios, including dynamic obstacles, and to fully consider how to avoid agent casualties. In addition, to further improve the reality of virtual evacuation, in the future, we plan to create a large-scale real emergency video scene dataset and use more learning methods to study the behaviors of agents in emergency scenarios.

ACKNOWLEDGMENTS

The authors appreciate the comments and suggestions of all the anonymous reviewers, whose comments helped us to significantly improve this article. This work is supported in part by the National Natural Science Foundation of China (Grant Nos. 61902003, 61976006) and the Doctoral Scientific Research Foundation of Anhui Normal University.

ORCID

Wen Zhou  <https://orcid.org/0000-0002-1266-1864>

Wenyang Jiang  <https://orcid.org/0000-0001-7647-1202>

REFERENCES

1. Li Y, Chen M, Dou Z, Zheng X, Cheng Y, Mebarki A. A review of cellular automata models for crowd evacuation. *Phys A Stat Mech Appl*. 2019;526(1):120–32.
2. Liu H, Liu B, Zhang H, Li L, Qin X, Zhang G. Crowd evacuation simulation approach based on navigation knowledge and two-layer control mechanism. *Inf Sci*. 2018;436–437(1):247–67.
3. Ma K, Zhang P, Mao Z. Study on large-scale crowd evacuation method in cultural museum using mutation prediction RFID. *Pers Ubiquitous Comput*. 2019;24(5):111–23.
4. Tian Z, Zhang G, Hu C, Lu D, Liu H. Knowledge and emotion dual-driven method for crowd evacuation. *Knowl Based Syst*. 2020;208(1):106–20.
5. Li Z, Huang H, Li N, Zan ML, Law K. An agent-based simulator for indoor crowd evacuation considering fire impacts. *Autom Constr*. 2020;103–25.
6. Bakar A, Adam K, Majid MA, Allegra M. A simulation model for crowd evacuation of fire emergency scenario. *Proceedings of the 2017 8th International Conference on Information Technology ICIT*; 2017. p. 61–368; Amman, Jordan.
7. Liu S, Liu J, Wei W. Simulation of crowd evacuation behaviour in outdoor public places a model based on shanghai stampede. *Int J Simul Model*. 2019;18(1):86–99.
8. Xie W, Lee E, Li T, Shi M, Cao R, Zhang Y. A study of group effects in pedestrian crowd evacuation: experiments, modelling and simulation. *Saf Sci*. 2020;133(1):105–29.
9. Pelechano N, Badler N. Modeling crowd and trained leader behavior during building evacuation. *IEEE Comput Graph Appl*. 2006;26(6):80–6.
10. Qiu F, Hu X. Modeling group structures in pedestrian crowd simulation. *Simul Model Pract Theory*. 2006;18(2):190–205.
11. Hou L, Liu J, Pan X, Wang BH. A social force evacuation model with the leadership effect. *Phys A Stat Mech Appl*. 2014;400(1):93–9.
12. Ma Y, Yuen K, Lee E. Effective leadership for crowd evacuation. *Phys A Stat Mech Appl*. 2016;450(1):333–41.
13. Yang X, Dong H, Yao X, Sun X, Wang Q, Zhou M. Necessity of guides in pedestrian emergency evacuation. *Phys A Stat Mech Appl*. 2016;442(1):397–408.
14. Zhang H, Liu H, Qin X, Liu B. Modified two-layer social force model for emergency earthquake evacuation. *Phys A Stat Mech Appl*. 2018;492(1):1107–19.
15. Li Y, Liu H, Liu G, Li L, Moore P, Hu B. A grouping method based on grid density and relationship for crowd evacuation simulation. *Phys A Stat Mech Appl*. 2017;473(1):319–36.
16. Juniastuti S, Fachri M, Nugroho S, Li L, Moore P, Hu B. A grouping method based on grid density and relationship for crowd evacuation simulation. *Phys A Stat Mech Appl*. 2017;473(1):319–36.
17. Li W, Li Y, Yu P, Gong J, Shen S. The trace model: a model for simulation of the tracing process during evacuations in complex route environments. *Int J Feder Eur Simul Soc*. 2016;60(1):108–21.
18. Zhou M, Dong H, Zhao Y, Ioannou PA, Wang FY. Optimization of crowd evacuation with leaders in urban rail transit stations. *IEEE Trans Intell Transp Syst*. 2019;20(12):4476–87.
19. Bakar A, Majid A, Adam K. Simulation and modelling the human crowd evacuation. *Mater Sci Eng*. 2019;551(12):12–38.
20. Farhan J. An agent-based multimodal simulation model for capacity planning of a cross-border transit facility. *Transp Res C Emerg Technol*. 2015;60(12):189–210.

21. Miyagawa D, Ichinose G. Cellular automaton model with turning behavior in crowd evacuation. *Phys A Stat Mech Appl*. 2020;549(1):124–37.
22. Luo X, Yuan Y, Li Z, Zhu M, Xu Y, Chang L, et al. FBVA: a flow-based visual analytics approach for citywide crowd mobility. *IEEE Trans Comput Soc Syst*. 2019;6(2):277–88.
23. Cao S, Zhang X, Cao W, Liu C, Li Y, Li P, et al. Approach for detecting crowd panic behavior based on fluid kinematic features and entropy. *Proceedings of the International Workshop on Advanced Computational Intelligence & Intelligent Informatics*; Fukuoka, Japan; 2011.
24. John D. The role of social identity processes in mass emergency behavior: an integrative review. *Eur Rev Soc Psychol*. 2018;29(1):38–81.
25. Li G, Zhang J, Yang J. Ocean simulation based on particle system. *IOP Conf Ser Earth Environ Sci*. 2020;440(5):52–76.
26. Han Y, Liu H. Modified social force model based on information transmission toward crowd evacuation simulation. *Phys A Stat Mech Appl*. 2017;469(1):499–509.
27. Kahn K. An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo. *Phys Today*. 2015;68(8):55–5.
28. Wagner N, Agrawal V. An agent-based simulation system for concert venue crowd evacuation modeling in the presence of a fire disaster. *Expert Syst Appl*. 2014;41(6):2807–15.
29. Shimada E, Yamane S, Ohori K, Yamada H, Takahashi S. Agent-based simulation for evaluating signage system in large public facility focusing on information message and location arrangement. *9th JSAI International Symposium on Artificial Intelligence*; 2017: 67–82; Yokohama, Japan.
30. Fu Y, Liang J, Liu Q, Hu XQ. Crowd simulation for evacuation behaviors based on multi-agent system and cellular automaton. *Proceedings of the 2014 International Conference on Virtual Reality and Visualization*; Shenyang, China; 2014.
31. Chennoufi M, Bendella F, Bouzid M. Multi-agent simulation collision avoidance of complex system: application to evacuation crowd behavior. *Int J Ambient Comput Intell*. 2018;9(1):43–59.
32. Pan X, Han C, Law H, Latombe JC. A computational framework to simulate human and social behaviors for egress analysis. *Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering*; 2006. p. 1206–15; Montreal, Canada.
33. Zheng X, Zhong T, Liu M. Modeling crowd evacuation of a building based on seven methodological approaches. *Build Environ*. 2019;44(3):437–45.
34. Chu M, Law K. Computational framework incorporating human behaviors for egress simulations. *J Comput Civil Eng*. 2013;27(6):699–707.
35. Zheng S, Liu H. Improved multi-agent deep deterministic policy gradient for path planning-based crowd simulation. *IEEE Access*. 2019;7(1):47755–147770.
36. Wang Q, Liu H, Gao K, Zhang L. Improved multi-agent reinforcement learning for path planning based crowd simulation. *IEEE Access*. 2019;7(99):73841–55.
37. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, et al. Continuous control with deep reinforcement learning; 2015. arXiv preprint arXiv:1509.02971.
38. Ho J, Ermon S. Generative adversarial imitation learning. *Adv Neural Inf Process Syst*. 2016;29:4572–80.
39. Song J, Ren H, Sadigh D, Ermon S. Multi-agent generative adversarial imitation learning; 2018. arXiv preprint arXiv:1807.09936.
40. Choi S, Kim J, Yeo H. TrajGAIL: generating urban vehicle trajectories using generative adversarial imitation learning. *Transp Res C Emerg Technol*. 2021;128:91–103.
41. Chi W, Dagnino G, Kwok Y, Nguyen A, Kundrat D, Abdelaziz ME, et al. Collaborative robot-assisted endovascular catheterization with generative adversarial imitation learning. *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*; 2020. p. 2414–20; Paris, France.

AUTHOR BIOGRAPHIES



Wen Zhou received his Ph.D. degree from the School of Software Engineering, Tongji University, China, in 2018. Since 2018, he has been affiliated with the School of Computer and Information, Anhui Normal University, Wuhu, China, where he is currently a lecturer; he is also an IEEE Member and a Member of the Chinese Computer Federation (CCF). His research interests include WebVR visualization, virtual reality, sketch-based retrieval, and machine learning.



Wenying Jiang is a master's degree candidate at Anhui Normal University, China. She received her B.Sc. degree in digital media technology from Huaibei Normal University in 2019 and is now pursuing a master's degree in computer science and technology at Anhui Normal University. Her research interests include machine learning and 3D visualization.



Biao Jie received a Ph.D. degree in computer science from Nanjing University of Aeronautics and Astronautics, China, in 2015. He joined the School of Computer and Information, Anhui Normal University, in 2006 and is a professor at present. His research interests include machine learning and medical image analysis.



Weixin Bian received a Ph.D. degree in computer science from China University of Mining and Technology, Xuzhou, China, in 2018. He joined the School of Computer and Information, Anhui Normal University, in 2005 and is an associate professor at present. His research interests include machine learning and digital image analysis.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Zhou W, Jiang W, Jie B, Bian W. Multiagent evacuation framework for a virtual fire emergency scenario based on generative adversarial imitation learning. *Comput Anim Virtual Worlds*. 2021;e2035. <https://doi.org/10.1002/cav.2035>