



Adaptive formation control and transformation in virtual crowds via deep reinforcement learning

Libo Sun¹ · Yongchun Qiu¹ · Wenhui Qin¹

Received: 7 July 2025 / Accepted: 20 August 2025 / Published online: 4 September 2025
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2025

Abstract

In the field of computer graphics, crowd simulation is crucial for constructing realistic virtual environments. This paper introduces a novel approach to adaptive formation control and transformation in virtual crowds using deep reinforcement learning. Our method enables virtual agents to autonomously generate, maintain, and adaptively transform formations relying solely on local perception and interaction. The proposed framework comprises three key components: dynamic formation configuration based on alignment, a formation control algorithm embedded with formation constraints, and an adaptive formation transformation mechanism. Simulation results demonstrate that our method achieves formation errors around 5% in formation tasks, surpassing existing approaches in both convergence speed and formation control precision. It further exhibits rapid, stable, flexible, and highly scalable formation control capabilities, offering a novel paradigm for creating realistic and environmentally adaptive crowd animations. Code and trained policies for this paper are at <https://github.com/SYDDX/AFCTVC-DRL>.

Keywords Crowd formation · Deep reinforcement learning · Formation control · Formation transformation

1 Introduction

With advances in computer graphics and artificial intelligence, the learning and control of crowd behavior has become an important research area. Realistic crowd simulation enhances the visual fidelity of virtual environments and offers a reliable platform for crowd-related research. As a result, it is widely used in real-world applications such as computer games [1], animated films [2, 3], and crowd performance design [4], and it plays an important role in public safety [5], cooperative autonomous driving [6], and UAV formation demonstrations [7].

Crowd simulation methods are generally categorized into macroscopic [8, 9] and microscopic [10–12] models. Macroscopic models focus on approximating the overall movement

of crowds, often representing individuals as particles in a fluid flow governed by changes in density and velocity over time, while ignoring interactions between individuals. In contrast, microscopic models emphasize local interactions among individuals, preserving agent-level characteristics and personality. Although currently applied mainly to smaller crowds for real-time performance, microscopic models enable more realistic autonomous behaviors and social dynamics within the crowd.

Crowd formation is a relatively new yet important direction in crowd simulation research. Formation control typically involves guiding a crowd under strict formation constraints, with individuals navigating intelligently toward their goals. During movement, the crowd may need to transition from one formation to another. Maintaining order and coherence throughout the process is essential. Current methods for crowd formation mainly focus on formation generation, maintenance, and transformation. Formation generation and maintenance emphasize the ability to quickly form and stably preserve formations under environmental disturbances, while formation transformation describes the matching and path planning between two crowd formations. Formation generation typically relies on geometric constraints [7] and consensus theory [13]. Subsequently, formation maintenance

✉ Libo Sun
sunlibo@seu.edu.cn

✉ Wenhui Qin
qinwenhui@seu.edu.cn

Yongchun Qiu
220233663@seu.edu.cn

¹ School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China

and transformation are implemented through behavior models [14], collision avoidance models [15], and other similar methods. With the development of deep reinforcement learning theory, this approach has been increasingly applied to crowd formation control. Although existing methods have made significant progress in crowd formation control and transformation, they often rely on manual intervention or expert knowledge during formation generation and adjustment. As the crowd size increases, these approaches become more complex and labor-intensive, limiting their scalability and adaptability in dynamic environments.

This paper proposes a multi-agent deep reinforcement learning strategy for crowd formation control and transformation in complex, obstacle-rich environments, enabling agents to autonomously generate, maintain, and adaptively adjust orderly formations without external guidance. The proposed approach achieves rapid formation generation, stable formation control, and flexible transformation in complex environments. The main contributions are as follows:

- (1) Dynamic formation configuration based on alignment: Each agent estimates the formation center using local perception and interactions, aligning the current spatial configuration of the group with a predefined formation pattern. A distributed CBBA algorithm then assigns formation positions based on relative locations, which helps minimize unnecessary movement from suboptimal position assignments and promotes smooth, coordinated motion across the crowd.
- (2) Formation control algorithm based on MAPOCA: This approach embeds formation constraints directly into each agent's local observations, enabling agents to avoid collisions and maintain consistent, orderly formations without relying on global information or complex communication. It supports adaptation to formations of different scales, shapes, and densities, resulting in coordinated and well-structured group movements that enhance the overall uniformity of crowd animations.
- (3) Adaptive formation transformation mechanism: Agents generate a globally consistent scaling factor through local interactions to guide formation transformation. When the scaling requirement exceeds a threshold, the group performs uniform scaling; otherwise, the system selects the optimal transformation strategy based on an improved Hausdorff distance, balancing obstacle avoidance and formation constraints to ensure smooth and natural transitions.

2 Related work

To achieve ideal formation control, researchers have proposed various methods, including leader–follower method

[16–18] and behavior-based method [19, 20]. Leader–follower approach explains collective behavior by assigning leaders to determine movement trajectories, with followers maintaining formation. Behavior-based methods simulate complex crowd dynamics by modeling individual actions as a set of simple rules. These methods assume that each agent makes decisions based on perceived environmental cues and internal behavioral strategies. Common behaviors include obstacle avoidance, goal seeking, cohesion, and velocity matching.

Reinforcement learning (RL) serves as a computational method for solving Markov Decision Processes (MDPs), which offer a rigorous mathematical framework for sequential decision-making. Deep reinforcement learning (DRL) [21, 22] leverages neural networks to learn near-optimal policies through interaction without relying on reference data. Early research efforts [23–26] predominantly employed DRL in robot formation control, establishing its efficacy in coordinating multi-robot systems. In the domain of DRL-based crowd simulation, Lee et al. [27] developed a deep reinforcement learning framework with a predefined reward function for crowd navigation in diverse complex environments. Charalambous et al. [28] introduced a guided reinforcement learning approach that incorporates reference crowd data, effectively capturing both overt and nuanced human behaviors. Kwiatkowski et al. [29] investigated reward function design for crowd simulation using reinforcement learning. Zhang et al. [30] proposed a crowd evacuation path planning strategy, utilizing LC-DDPG for leader-based global planning and an enhanced social force model to assist followers in obstacle avoidance and leader tracking. Xu et al. [31] employed knowledge distillation to derive reward functions from expert policies trained via behavior cloning, facilitating interaction in fully decentralized multi-agent settings. Nevertheless, these studies primarily emphasize collision avoidance rather than structured crowd formation. Wang et al. [32] introduced a rule-based action generator (RAG) for actor-critic methods, enabling user-directed crowd control through leader–agent interactions. Sun et al. [33] devised a hierarchical formation control framework for crowd simulation, wherein the lower level employs a composite group reward function to train agents for cooperative behavior, while the upper level implements a formation-switching mechanism to manage compression and splitting dynamics.

In formation control, agents must maintain predefined formations to ensure coordinated task execution. However, environmental obstacles and dynamic mission requirements often demand real-time formation adjustments. To address this challenge, Zhao et al. [34] developed a stress matrix-based leader–follower strategy that enables agents to track essential affine transformations including translation, rotation, and scaling. Extending this work, Li et al. [35] proposed a dynamic hierarchy-reorganization method that reassigns

leadership roles for seamless formation switching. Li et al. [36] also created a distributed adaptive formation strategy that automatically adjusts formation scale based on local agent density. Recent advances include Pan et al.'s [37] innovative use of Hausdorff distance for formation similarity measurement combined with MADDPG-based optimal path planning. Xu et al. [38] achieved fluid formation transitions by integrating balanced allocation with social force models. Gu et al. [39] established a dual-layer control framework that simultaneously optimizes formation contour alignment and internal distribution. Most recently, Yan et al. [40] introduced a spatiotemporal transformer-enhanced GAN to learn style-specific crowd transition patterns.

Despite significant progress in geometric transformation, leadership reorganization, and scale adaptation, most existing methods do not adequately address complex environmental constraints. In computer graphics applications, realistic scenarios frequently involve dynamic obstacles and spatial restrictions, requiring high adaptability and smooth transitions in formation control. Current approaches often struggle to balance the need to maintain formation consistency with the challenge of environmental responsiveness, limiting their effectiveness in crowd simulations and other virtual applications.

3 Preliminaries

3.1 Markov decision processes (MDPs)

Distributed multi-agent formation control is a fully cooperative problem formulated as a Markov game [41], an extension of the Markov decision process to multiple agents within the multi-agent reinforcement learning framework. The Markov game involving N agents can be defined by $\{N, \mathcal{S}, \mathcal{O}, \mathcal{A}, P, r, \gamma\}$, where $\mathcal{S}$ denotes the state space of the environment, and $\mathcal{O} := \mathcal{O}_1 \times \dots \times \mathcal{O}_N$ represents the joint observation space of all agents. The environmental state $\mathcal{S}$ includes the observations of all agents as well as the total number of active agents. $\mathcal{A} := \mathcal{A}_1 \times \dots \times \mathcal{A}_N$ stands for the joint action space of all agents, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the state transition probability function, and $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the shared reward function. $r(s, \mathbf{a})$ is the reward acquired by all agents when the joint action $\mathbf{a}$ is taken in the environmental state s , and γ is the discount factor.

3.2 Problem statement

This paper studies a crowd formation consisting of N virtual agents, denoted by the set $\mathcal{N}$, and environmental obstacles denoted by the set $\mathcal{M}$. The position vectors of the agents are represented by the vector $[p_1, \dots, p_N]$, with each agent having a radius r and a communication radius R . Agents

within distance R of agent i are considered its neighbors, forming the neighbor set $\mathcal{N}_i$. Agents exchange information bidirectionally within this limited communication range. The communication topology is modeled as a weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. An edge $(i, j) \in \mathcal{E}$ indicates a bidirectional communication link between agents i and j , implying mutual neighborhood.

In a predefined topology, each agent's formation constraints are determined by its relative positions to neighboring agents. The formation objectives and constraints become fixed once position assignment is completed, and the desired shape is achieved only when these constraints are satisfied. Formation control regulates agent velocities to comply with these constraints, enabling formation generation, maintenance, navigation, obstacle avoidance, scaling, and reconfiguration. Specifically, formation generation is the process by which agents transition from an initial disordered state to a desired spatial configuration through distributed cooperation; formation maintenance preserves predefined spatial relationships during motion despite disturbances; formation scaling adjusts formation size while preserving topology to adapt to environmental constraints; and formation reconfiguration restructures the formation when scaling alone cannot ensure passage through constrained spaces.

4 Methodology

4.1 Overview of the proposed framework

This paper aims to improve the autonomy and flexibility of crowd formations by coordinating individual motion based on local environmental perception, enabling formation generation, maintenance, and adaptive shape transformation. For this purpose, we propose a distributed formation control framework composed of three key components, as shown in Fig. 1. The dynamic formation configuration based on alignment assigns suitable final positions and associated constraints to each agent. Subsequently, a distributed control algorithm based on MAPOCA adjusts agents' linear and angular velocities according to local observations, ensuring formation constraints are satisfied and guiding agents toward their assigned positions. Simultaneously, the algorithm produces scaling factors that act as transformation commands. During simulation, the adaptive transformation mechanism autonomously chooses between scaling and reconfiguration modes based on these factors, enabling flexible formation adaptation in response to environmental constraints. The distributed system architecture improves scalability by requiring only minimal information exchange among agents, such as the estimated formation center and scaling factors, thereby reducing the communication burden.

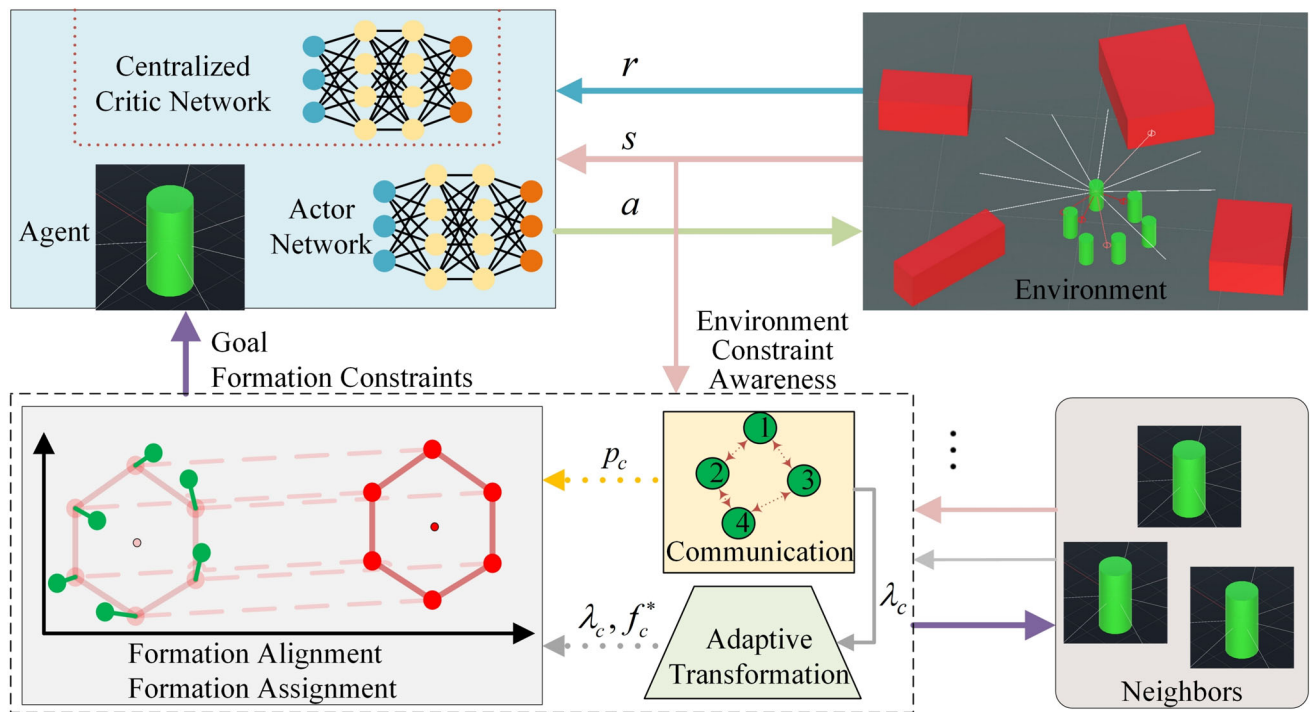


Fig. 1 Illustration of the framework of the distributed formation control algorithm. Formation alignment and target allocation assign final goals and formation constraints to agents. The control policy which is trained using the MAPOCA algorithm, guides the agents to adjust their

actions and interact with the environment. At the same time, the formation adaptively transforms based on environmental constraints and scaling factors

4.2 Dynamic formation configuration approach

The dynamic formation configuration approach adaptively assigns each agent a final target position along with the corresponding formation constraints. This method consists of two stages: formation alignment and formation configuration, as shown in Fig. 2.

Formation alignment dynamically computes a reference center via local communication to reflect agent distribution, guiding goal assignment and formation generation while minimizing adjustments and interference. Each agent initializes its estimate with its position and updates it by exchanging information with neighbors, eventually reaching consensus. At discrete step k , agent i 's estimate is denoted as $p_i^c(k)$, and is updated as follows:

$$p_i^c(k+1) = w_{ii} p_i^c(k) + \sum_{j \in \mathcal{N}_i} w_{ij} (p_j^c(k) - p_i^c(k))$$

$$w_{ij} = \frac{\tilde{w}_{ij}}{\sum_{k \in \mathcal{N}_i \cup \{i\}} \tilde{w}_{ik}} \quad (1)$$

$$\tilde{w}_{ij} = \begin{cases} \frac{1}{d_{ij} + \epsilon_1}, & j \in \mathcal{N}_i \\ 0, & j \notin \mathcal{N}_i \end{cases}, \tilde{w}_{ii} = \frac{1}{\epsilon_1}$$

where d_{ij} denotes the distance between agent i and agent j , and ϵ_1 is a small positive constant.

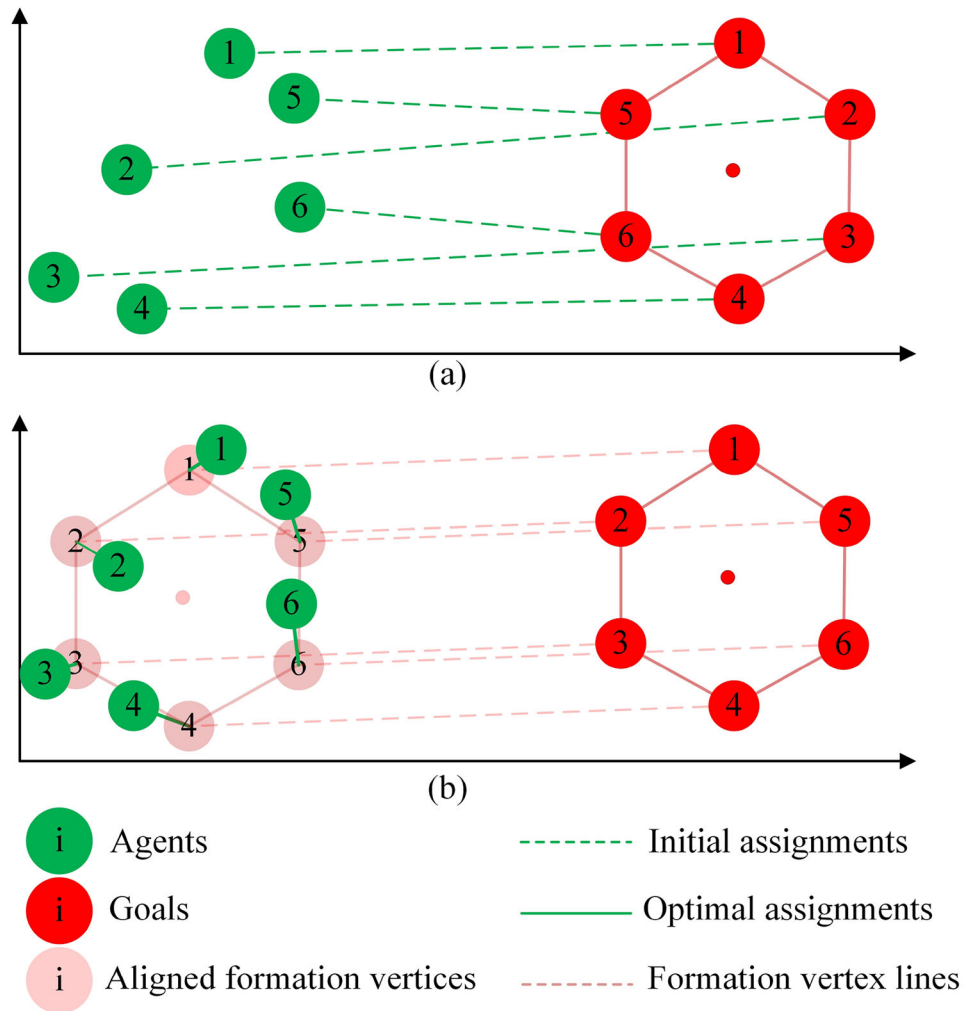
These steps are repeated until agents' estimated formation centers satisfies the convergence condition:

$$\|p_i^c(k) - p_j^c(k)\| < \delta, i \in \mathcal{N}, j \in \mathcal{N}_i \quad (2)$$

where δ denotes the permissible estimation error. Upon satisfying the convergence condition, all agents achieve a consensus estimate of the formation center p_c . The predefined desired formation topology f is then aligned with p_c , generating a set of vertex position vectors $\{p_1^f, \dots, p_N^f\}$. A vertex j is defined as a neighbor of vertex i if their separation distance is less than the perception radius R , with the neighbor set denoted as $\mathcal{N}_i^f$. The relative position vector p_{ij} represents vertex j 's position relative to vertex i , and the collection $\{p_{ij}\}_{j \in \mathcal{N}_i^f}$ constitutes the formation constraint for vertex i . Furthermore, alignment of the desired formation topology with the global target position p_g produces corresponding sub-goals $\{p_1^g, \dots, p_N^g\}$, which maintain a one-to-one correspondence with the aligned formation vertices.

The objective of formation assignment is to determine the optimal matching between agents and the vertices of the desired formation. In this work, we adopt a distributed

Fig. 2 Illustration of formation alignment and configuration. **a** Without alignment, directly assigning formation goals may lead to large transition distances for some agents to their goals, as well as disordered assignments that may cause deadlock. **b** Without changing agent positions, applying formation alignment results in optimal goal assignments with smaller and more uniform transition distances, thereby accelerating the formation process



Consensus-Based Bundle Auction (CBBA) algorithm [42], where agents actively bid on formation vertices based on the lowest associated cost, thereby acquiring the corresponding sub-goals and formation constraints. To achieve optimal assignment, we design a cost function driven by spatial similarity, which jointly considers the consistency in distance and bearing between agents and formation vertices relative to the formation center. The cost function is defined as follows:

$$J(i, \sigma(i)) = w_\theta |\theta_i^c(t) - \theta_{\sigma(i)}^c| + w_d |d_i^c(t) - d_{\sigma(i)}^c| \quad (3)$$

where σ denotes the assignment map from agents to formation vertices, and $\sigma(i)$ indicates the formation vertex assigned to agent i . $\theta_i^c(t)$ and $d_i^c(t)$ represent the orientation angle and distance of agent i relative to the formation center p_c at time t , while $\theta_{\sigma(i)}^c$ and $d_{\sigma(i)}^c$ denote the orientation angle and distance of vertex $\sigma(i)$ relative to p_c . The cost function quantifies the spatial similarity between agents and formation vertices. A lower cost indicates higher similarity, meaning that the agent's topological position within the formation closely

matches that of the corresponding vertex in the desired formation and only minimal adjustments in speed and direction are needed to satisfy the assigned formation constraints.

The overall objective of the assignment can be formulated as:

$$\sigma^* = \min_{\sigma} \sum_{i \in \mathcal{N}} J(i, \sigma(i)) \quad (4)$$

Upon determining the optimal assignment, each agent autonomously configures its corresponding sub-goal and formation constraints based on the assigned formation vertex. Specifically, the terminal goal position for agent i is given by $p_{\sigma^*(i)}^g$, with the associated formation constraints defined as the set $\{p_{\sigma^*(i)\sigma^*(j)}^f\}_{j \in \mathcal{N}_i}$. The primary objective of formation control is to continuously regulate the agent's linear and angular velocities such that the desired relative positions with respect to neighboring agents are strictly maintained during navigation. This control problem can be formally expressed

as:

$$\lim_{t \rightarrow \infty} \|p_{ij}(t) - p_{\sigma^*(i)\sigma^*(j)}^f\| = 0, \forall i \in \mathcal{N}, j \in \mathcal{N}_i \quad (5)$$

4.3 Formation control algorithm based on MAPOCA

4.3.1 MAPOCA algorithm

This paper employs the Multi-Agent POsthumous Credit Assignment (MAPOCA) algorithm [43] to train core control capabilities in multi-agent systems. Traditional methods often rely on predefined rules or fixed models, limiting adaptability to complex dynamic environments and flexibility in coordination. Although reinforcement learning provides a means to learn adaptive and cooperative strategies, many multi-agent RL algorithms lack robustness and adaptability when handling variable-sized inputs. In contrast, MAPOCA leverages a residual self-attention mechanism to efficiently process variable-sized inputs, significantly enhancing robustness and adaptability. By integrating counterfactual advantage estimation and a centralized training with decentralized execution (CTDE) framework, it improves coordination among agents, reduces policy conflicts, and accelerates stable convergence. Consequently, MAPOCA outperforms traditional methods and other standard multi-agent RL architectures on standard multi-agent coordination tasks.

Assuming there are N active agents at time t , and the local observation of agent i is denoted as o_i^t , the MAPOCA algorithm independently encodes each local observation into $g_i(o_i^t)$. The encoded observations $g_i(o_i^t)_{1 \leq i \leq N}$ of all active agents are then passed through a Residual Self-Attention (RSA) module to the Critic network. The centralized state-value function is represented as $V_\phi(RSA(g_i(o_i^t)_{1 \leq i \leq N}))$. The training loss function of the Critic network is given by:

$$J(\phi) = (V_\phi(RSA(g_i(o_i^t)_{1 \leq i \leq N})) - y^{(\lambda)})^2 \quad (6)$$

$$y^{(\lambda)} = (1 - \lambda) \sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)} \quad (7)$$

$$G_t^{(n)} = \sum_{l=1}^n \gamma^{l-1} r_{t+l} + \gamma^n V_\phi(RSA(g_i(o_i^{t+n})_{1 \leq i \leq N})) \quad (8)$$

where ϕ denotes the parameters of the value network, r_{t+l} is the sum of the team and individual rewards received by the agent at time $t + l$, and γ is the discount factor.

The MAPOCA algorithm employs a baseline network to estimate the counterfactual baseline expectation for agent i . It also uses the RSA module to encode both the agent's observation and observation-action pairs. The baseline function is

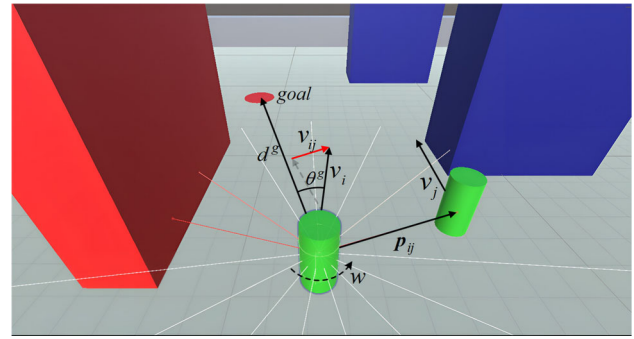


Fig. 3 Local observation of the agent

denoted as $RSA(g_i(o_i^t), f_j(o_j^t, a_j^t)_{1 \leq j \leq N, j \neq i})$, and the loss function of the baseline network is defined as:

$$J(\psi) = (Q_\psi(RSA(g_i(o_i^t), f_j(o_j^t, a_j^t)_{1 \leq j \leq N, j \neq i})) - y^{(\lambda)})^2 \quad (9)$$

The advantage function of agent i is defined as:

$$Adv_i = y^{(\lambda)} - Q_\psi(RSA(g_i(o_i^t), f_j(o_j^t, a_j^t)_{1 \leq j \leq N, j \neq i})) \quad (10)$$

The update of the actor network is given by:

$$\nabla_{\theta_i} J(\theta_i) = \nabla_{\theta_i} \log \pi_i(a_i | o_i) Adv_i \quad (11)$$

where θ_i denote the actor network parameters of agent i .

4.3.2 Observation space

Each agent is equipped with a ray sensor for detecting surrounding obstacles and other agents, with a sensing radius R , equal to the communication radius. The agent's observation space consists of three parts, including self-state, formation state and environmental state detected by the ray sensor. The agent's self-state $\mathbf{o}_{self} = \{v, w, \theta^g, d^g\}$ includes the agent's translational velocity, angular velocity, the angle difference between its movement direction and the goal direction, and the distance to the goal. The formation state $\mathbf{o}_{form} = \{\{v_{ij}(t), p_{ij}(t) - p_{\sigma^*(i)\sigma^*(j)}^f\}_{j \in \mathcal{N}_i, \|p_{ij}\| \leq R}\}$ includes the relative velocity between the agent and its neighboring agents, as well as the deviation between the actual relative position and the desired formation constraint. The environmental state detected by the ray sensor $\mathbf{o}_{sensor}$ is formed by concatenating all ray outputs, where the k -th ray returns a triplet $\mathbf{o}_{sensor}^k = \{h_k, d_k, t_k\}$. h_k denotes the ray's hit status, d_k corresponds to the normalized distance, and t_k represents the encoding of hit object's label (Fig. 3).

4.3.3 Action space

The action space $a = \{v, w\}$ of each agent includes forward translational velocity and rotational angular velocity. Specifically, the translational velocity $v \in [0, v_{\max}]$, the rotational angular velocity $w \in [-w_{\max}, w_{\max}]$, both are continuous variables.

4.3.4 Reward function

One of the most important tasks in reinforcement learning is the design of the reward function. In this work, a composite reward function is designed to address the complex tasks of obstacle avoidance, navigation, formation generation, and maintenance during formation control. The overall reward is defined as:

$$r_i = r_i^{\text{navi}} + r_i^{\text{avoid}} + r_i^{\text{form}} + r_i^{\text{time}} \quad (12)$$

where r_i^{navi} is the navigation reward, r_i^{avoid} is the collision avoidance reward, r_i^{form} is the formation reward, and r_i^{time} is the time penalty.

The navigation reward is designed to reflect the efficiency of an agent moving toward its destination. It comprises distance and orientation components, and is defined as:

$$r_i^{\text{navi}} = w_1(d_i^g(t) - d_i^g(t-1)) + w_2(\theta_i^g(t) - \theta_i^g(t-1)) \quad (13)$$

where $d_i^g(t)$ is the distance between agent i and the final goal p_g at time step t , and $\theta_i^g(t)$ is the angular difference between the agent's movement direction and the target orientation. w_1 and w_2 are weights.

The collision avoidance reward r_i^{avoid} is designed to quantify the agent's ability to avoid collisions with other agents and obstacles. The penalty increases as the agent gets closer to other agents and obstacles. It is defined as:

$$r_i^{\text{avoid}} = w_3 \sum_{j \in \mathcal{N}, j \neq i} e^{|R-d_{ij}|} \cdot \mathbb{1}(d_{ij} \leq R) + w_4 \sum_{m \in \mathcal{M}} e^{|R-d_{im}|} \cdot \mathbb{1}(d_{im} \leq R) \quad (14)$$

where d_{ij} is the distance between agent i and agent j at time step t , and d_{im} is the distance between agent i and obstacle m . w_3 and w_4 are negative weights.

The formation reward is designed based on the difference between the actual relative positions among agents and their assigned formation constraints. The formation reward

for each agent is defined as:

$$r_i^{\text{form}} = w_5 \sum_{j \in \mathcal{N}_i} \|p_{ij}(t) - p_{\sigma^*(i)\sigma^*(j)}^f\| + w_6 \sum_{j \in \mathcal{N}_i} \|v_{ij}(t)\| \quad (15)$$

where v_{ij} denotes the relative velocity vector between agent i and its neighbor j at time t , and w_5 and w_6 are weights.

Furthermore, each agent receives a small constant time penalty $r_i^{\text{time}} = c$ at each time step, encouraging active exploration, shorter trajectories, and faster movement.

4.4 Adaptive formation transformation approach

In this section, we introduce the environment constraint awareness vector and a scaling factor, and propose an adaptive formation transformation mechanism. Within a distributed decision-making framework, agents collaboratively adjust the formation based on local perception and communication, coordinating obstacle avoidance with formation integrity by optimizing inter-agent spacing or reconfiguring the topology.

4.4.1 Environmental constraint awareness vector

To assess the severity of environmental constraints on the agents, we propose an environmental constraint awareness vector derived from ray sensor data, as illustrated in Fig. 4. In the agent's local coordinate system, its velocity direction defines a central axis, dividing the sensing field into two awareness sectors $\mathcal{A}_l, \mathcal{A}_r$, with angular ranges $\theta_l \in [-\frac{5\pi}{6}, -\frac{\pi}{6}]$ and $\theta_r \in [\frac{\pi}{6}, \frac{5\pi}{6}]$. Their corresponding angle bisector unit vectors are denoted as $\{u_l, u_r\}$. The detection data of rays within these two sectors effectively reflect the degree of environmental constraints imposed on the agent. Each sector is further subdivided into three subregions: a shrinking region, a keeping region, and an expanding region. The normalized distances of all rays within each sector are extracted and averaged to obtain the mean normalized distance d_q , where $q \in \{l, r\}$. The environmental constraint awareness vector is defined as follows:

$$D = u_{q*} d_{q*} \quad (16)$$

where $q* = \arg \min_{q \in \{l, r\}} (d_q)$.

4.4.2 Adaptive formation transformation mechanism

We introduce the scaling factor $\{\lambda_i\}_{i \in \mathcal{N}}$ to represent each agent's tendency toward expansion, contraction, or reconfiguration. Each agent adjusts its own scaling factor based on

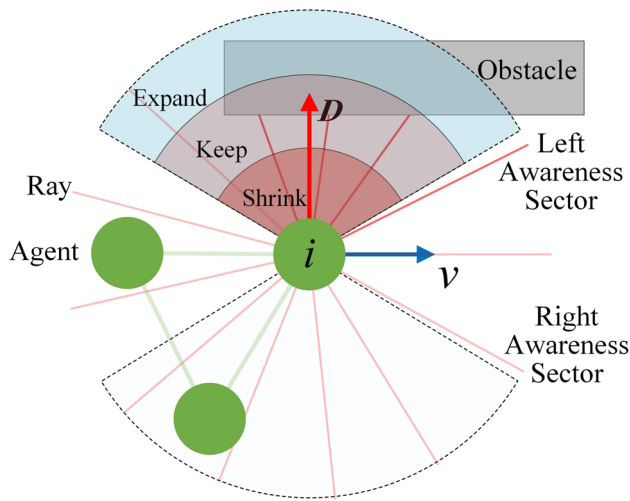


Fig. 4 Environmental constraint awareness vector

environmental constraints, as follows:

$$\lambda_i = \begin{cases} \lambda_-, & \text{if } \mathbf{D} \in \mathcal{R}_s \\ \lambda_c^{\text{old}}, & \text{if } \mathbf{D} \in \mathcal{R}_k \\ \lambda_+, & \text{if } \mathbf{D} \in \mathcal{R}_e \end{cases} \quad (17)$$

$$\lambda_- = \begin{cases} \max\{\lambda \in \Lambda \mid \lambda < \lambda_c^{\text{old}}\}, & \text{if } \exists \lambda \in \Lambda : \lambda < \lambda_c^{\text{old}} \\ \lambda_c^{\text{old}}, & \text{otherwise} \end{cases}$$

$$\lambda_+ = \begin{cases} \min\{\lambda \in \Lambda \mid \lambda > \lambda_c^{\text{old}}\}, & \text{if } \exists \lambda \in \Lambda : \lambda > \lambda_c^{\text{old}} \\ \lambda_c^{\text{old}}, & \text{otherwise} \end{cases}$$

where $\mathcal{R}_s$, $\mathcal{R}_k$ and $\mathcal{R}_e$ represent the shrinking, keeping, and expanding regions, respectively. λ_c^{old} denotes the previous global scaling factor of the formation, initially set to 1. $\Lambda = \{\lambda_{\max}, \lambda_{\text{mid}}, \lambda_{\min}\}$ denote the set of scaling factors.

During the information exchange process, each agent updates its scaling factor by selecting the minimum value from the set comprising its own signal and those received from neighboring agents. Assuming the communication graph remains connected, the scaling factors of all agents will converge to the minimum value among the initial individual signals. This value is then defined as the updated global scaling factor of the formation λ_c , as follows:

$$\lambda_c^{\text{new}} = \lim_{k \rightarrow \infty} \lambda_i(k), \quad \forall i \in \mathcal{N} \quad (18)$$

$$\lambda_i(k) = \min_{j \in \mathcal{N}_i \cup \{i\}} \lambda_j(k-1)$$

where $\lambda_i(0) = \lambda_i$.

Formation Scaling. When the factor $\lambda_c \in \{\lambda_{\text{mid}}, \lambda_{\max}\}$, the formation undergoes a uniform scaling transformation. In this case, both the formation topology and the correspondence between each agent and its assigned goal remain unchanged. However, the formation constraints must be

scaled accordingly based on the scaling factor:

$$\mathbf{p}_{\text{new}, \sigma^*(i)\sigma^*(j)}^f = \lambda_c \mathbf{p}_{\text{old}, \sigma^*(i)\sigma^*(j)}^f, \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{N}_i \quad (19)$$

Formation Reconfiguration. Formation reconfiguration is achieved by dynamically selecting the optimal formation for agents using the Hausdorff distance, reflecting the adaptability of the formation configuration. When the factor $\lambda_c = \lambda_{\min}$, a modified Hausdorff distance is adopted to evaluate the similarity between the local topology and the other desired formation topology, guiding dynamic formation adjustment. The standard Hausdorff distance focuses on the maximum deviation between two point sets, which makes it sensitive to outliers. In contrast, the average Hausdorff distance reduces the influence of outliers by averaging the maximum distances. Its calculation formula is given by:

$$H_{\text{avg}}(X, Y) = \frac{1}{2} \left[\frac{1}{|X|} \sum_{x \in X} \max_{y \in Y} d(x, y) + \frac{1}{|Y|} \sum_{y \in Y} \max_{x \in X} d(y, x) \right] \quad (20)$$

where $d(\cdot, \cdot)$ represents the Euclidean distance function. Let $\mathcal{P}_i = \{\mathbf{p}_{ij}\}_{j \in \mathcal{N}_i}$ denote the set of actual relative positions between agent i and its neighbors. We compute the average Hausdorff distance between the actual relative position set $\mathcal{P}_i$ of agent i and the formation constraints defined by each vertex in all candidate formation templates. The formation template corresponding to the vertex with the minimum average distance is then selected as the target formation for agent i . Its calculation formula is given by:

$$f_i^* = \arg \min_{f \in \mathcal{F}} \left(\min_{j \in \mathcal{N}^f} H_{\text{avg}}(\mathcal{P}_i, \mathcal{P}_j^f) \right) \quad (21)$$

where $\mathcal{F}$ is the set of formation templates, and $\mathcal{P}_j^f$ represents the formation constraint set corresponding to vertex j in a given template f . To ensure formation consistency, each agent updates its optimal formation at each time step by selecting the most frequent choice among itself and its neighbors. The consensus optimal formation f' is obtained after several iterations, as shown in Eq. (22):

$$f' = \lim_{k \rightarrow \infty} f_i^*(k), \quad \forall i \in \mathcal{N} \quad (22)$$

$$f_i^*(k) = \arg \max_{f \in \{f_j^*(k-1)\}_{j \in \mathcal{N}_i \cup \{i\}}} N(f)$$

where $f_i^*(0) = f_i^*$, and $N(\cdot)$ is a counting function that denotes the number of times an element appears in the set. Upon reaching consensus, the Dynamic Formation Configuration Approach proposed in Sect. 4.2 executes formation

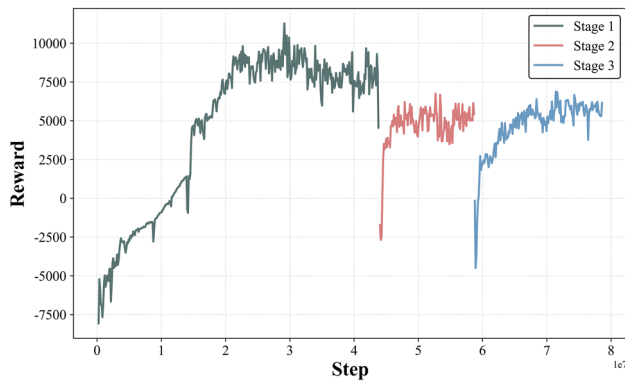


Fig. 5 Reward function learning curve

reconfiguration, updating the formation constraints as follows:

$$p_{\text{new}, \sigma^*(i)\sigma^*(j)}^{f'} = DFC(f'), \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{N}_i \quad (23)$$

where $DFC(\cdot)$ denotes the Dynamic Formation Configuration Approach.

5 Experiments and results

5.1 Training configuration

Simulation experiments were conducted on a laptop running the Windows 11 operating system, equipped with a 13th Gen Intel(R) Core(TM) i5-13500H processor, 16 GB RAM, and an NVIDIA GeForce RTX 4070 GPU. Simulation environments were constructed using Unity 2023.2.17f1 and ML-Agents release version 21. The deep learning framework employed was PyTorch 2.3.1.

In complex virtual environments, direct training of the network leads to unstable or non-convergent behaviors in crowd agents. To mitigate this challenge and facilitate more stable learning progression, we employ a curriculum learning strategy [44], dividing the training process into three progressively difficult stages. Stage 1 focuses on basic navigation in scenes containing only boundary walls, allowing agents to learn fundamental movement capabilities. Stage 2 introduces formation generation and maintenance tasks, guiding agents to form and preserve coherent group structures. Stage 3 incorporates diverse static obstacles into the environment to enhance agents' collision avoidance capabilities while maintaining formation. Transitions between stages are triggered once the average reward stabilizes, ensuring steady skill acquisition. The structure of the reward function is shown in Fig. 5. Detailed agent-related parameters are summarized in Table 1, while the model's training settings are provided in Table 2.

Table 1 Agent-related parameters

Parameters	Value
r	0.5 m
R	8 m
$v_{\max}$	0.7 m/s
$w_{\max}$	0.75π rad/s
Number of rays	13
$\lambda_{\max}$	1
λ_{mid}	0.7
$\lambda_{\min}$	0.5

Table 2 Training parameters of MAPOCA

Parameters	Value
Batch size	1024
Buffer size	10240
λ	0.95
β	0.01
Learning rate	0.0003

To evaluate the effectiveness of the proposed method, we define two key performance metrics: the formation error and the convergence time, calculated as follows:

$$e_f = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} \left\{ \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \frac{\|p_{ij}(t) - p_{\sigma^*(i)\sigma^*(j)}^f\|}{\|p_{\sigma^*(i)\sigma^*(j)}^f\|} \right\} \quad (24)$$

$$t_f = \min\{t \mid |e_f - e_f^s| \leq \epsilon_2\} \quad (25)$$

where $e_f^s = \frac{1}{K} \sum_{i=t_{\text{end}}-K+1}^{t_{\text{end}}} e_f(t)$, $e_f(t)$ denotes the time series data for the formation error e_f , t_{end} denotes the last time index of the error data sequence and ϵ_2 is a small constant. The formation error quantifies the ability of the agents to establish and maintain the desired formation pattern during movement by measuring the average deviation between the agents' actual relative positions with respect to their neighbors and the prescribed formation constraints. Meanwhile, the convergence time reflects the rate at which the formation error converges to a steady state.

5.2 Simulation results

5.2.1 Evaluation of formation alignment performance

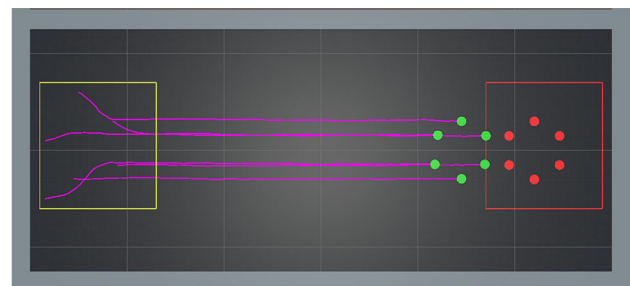
We evaluated the formation alignment strategy from Sect. 4.2 through comparative experiments in circular formation tasks. Under random initial distributions, agents were assigned to formation positions either with or without our alignment method, using Euclidean distance as the cost function to ensure fair comparison. The results in Fig. 6 display agent positions as colored circles connected by lines to their assigned target positions. Fig. 6a demonstrates that with-

out alignment, agents starting on the left side frequently receive distant target assignments on the right side, resulting in excessive path crossings and unnatural movement patterns that both slow formation assembly and compromise visual realism. In contrast, Fig. 6b shows that when applying alignment, agents obtain target assignments that better match their initial spatial distribution, leading to significantly reduced trajectory adjustments and consequently smoother, more natural-looking formation transitions.

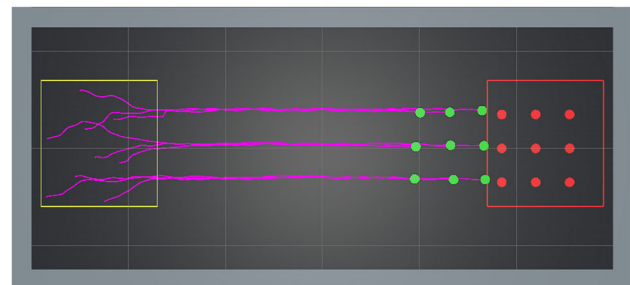
5.2.2 Simulation results in obstacle-free scenarios

We conducted experiments to evaluate the generalization capability of our proposed method under varying shapes and agent numbers in an obstacle-free environment. The tests included a 6-agent circular formation, a 9-agent square formation, and a 10-agent triangular formation. As shown in Fig. 7, the agents quickly formed the desired formations without collisions and maintained them while moving smoothly toward the final goal. Building on the capabilities of formation generation, maintenance, and navigation, we simulated two-way motion between two 6-agent groups, as shown in Fig. 8. Initially, the green and blue groups occupied separate start areas, with each group's target located in the other's initial region. Each group first formed a circular formation and then navigated toward its target. As the formations converged near the scene center, their paths adjusted to avoid collisions. After successful collision avoidance, both groups proceeded to their respective targets.

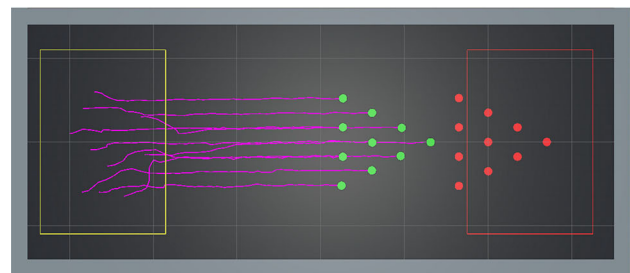
We further conducted comparative experiments to evaluate the formation control performance of the proposed strategy using formation error and convergence time as metrics. Specifically, we compared the ORCA method enhanced with consistency theory (ORCA-C), the method by Sun et al. (CCA), and our proposed approach in triangular formation tasks involving six and ten virtual agents. After normalizing



(a)

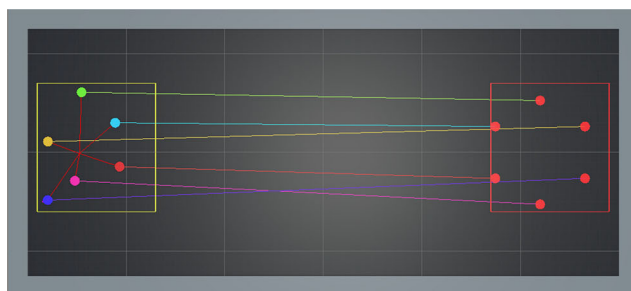


(b)

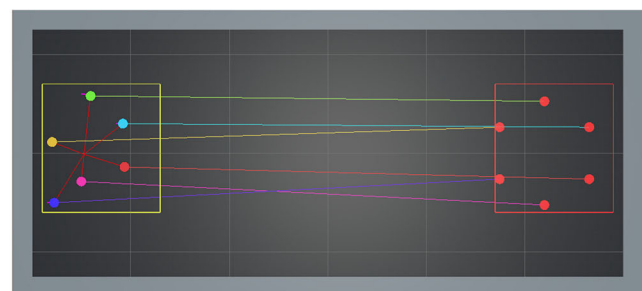


(c)

Fig. 7 Formation simulation results under varying agent numbers and desired shapes. **a** 6 agents quickly form and maintain a stable circular structure; **b** 9 agents quickly form and maintain a stable square structure; **c** 10 agents quickly form and maintain a stable triangular structure



(a)



(b)

Fig. 6 Comparison of target assignment under random initial position distribution: **a** Without alignment: agents are assigned in a crosswise manner, causing some agents to require larger movement distances to form the formation, which increases the time required for formation. **b**

With alignment: agents are assigned to more appropriate targets based on their spatial distribution, resulting in more balanced movement distances

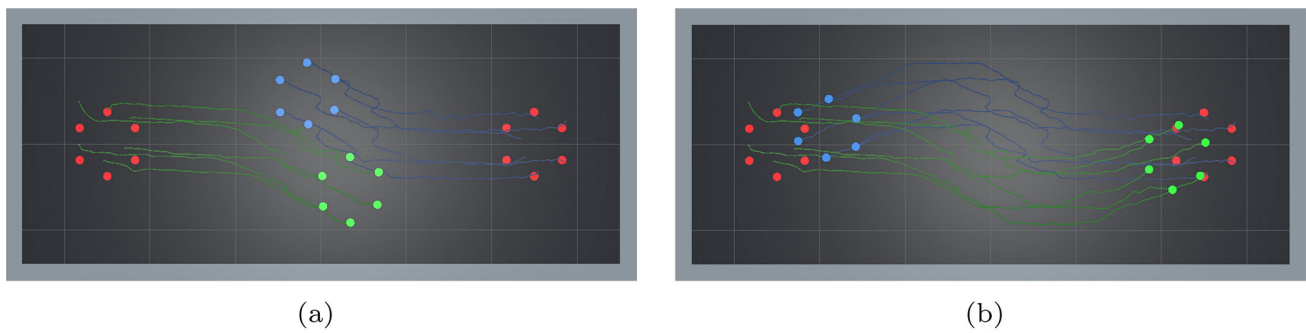


Fig. 8 Simulation results of two circular formations moving toward each other. **a** The blue and green formations quickly form and begin to avoid each other as they approach. **b** After the two formations avoid each other, they proceed toward their respective targets

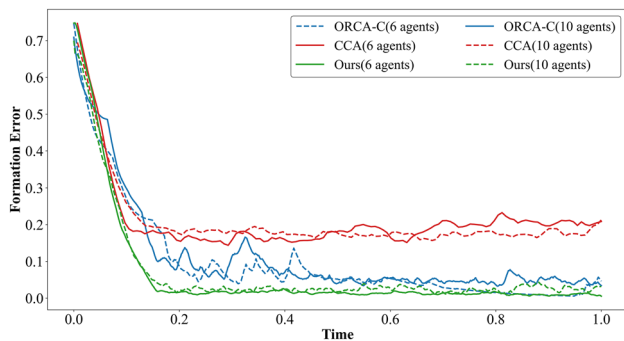


Fig. 9 Comparison of formation error among the Our method, ORCA-C and CCA

the motion time for each method, we analyzed the evolution of the formation error e_f throughout the process. As shown in Fig. 9, the convergence times for CCA, ORCA-C, and our method were approximately 0.11, 0.22, and 0.16 of the total duration, respectively. Although our method exhibited a longer convergence time than CCA, it achieved a lower formation error compared to both CCA and ORCA-C. After convergence, CCA maintained an error of approximately 20%, ORCA-C around 6%, while our method further reduced the error to below 5%, demonstrating superior control precision. These results highlight the advantages of the proposed

Table 3 Average frame rate with varying numbers of agents

Number of agents	FPS
6	221.1
10	159.6
15	88.9

method in both convergence speed and formation control accuracy.

To quantitatively assess the real-time performance of the proposed formation control framework, we measured the frame rate of triangular formation control under different agent numbers, summarized in Table 3. Although the frame rate exhibits a decreasing trend with increasing agent numbers, the system consistently sustains real-time performance, maintaining frame rates above 80 FPS even with 15 agents.

5.2.3 Simulation results in multi-obstacle scenarios

We evaluated the obstacle avoidance capability and formation stability of the proposed algorithm in three obstacle-rich scenarios. Model parameters were initialized using those trained in the obstacle-free environment. In the first scenario, two semi-cylindrical obstacles were placed at the top and bottom of the environment. Agents started from the left and

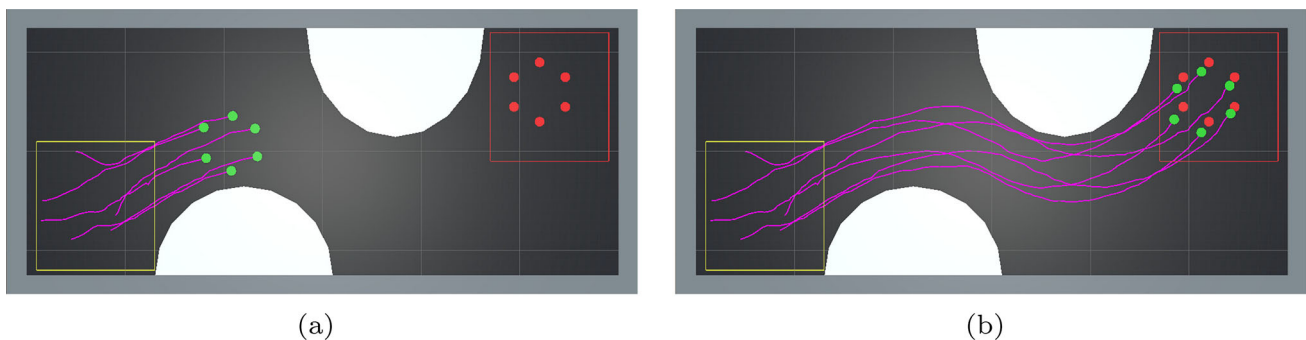


Fig. 10 Simulation results of circular formation in a rectangular scenario with semi-cylindrical obstacles. **a** 6 agents quickly form a circular formation. **b** The circular formation follows an “S”-shaped path to avoid obstacles and reach the final target

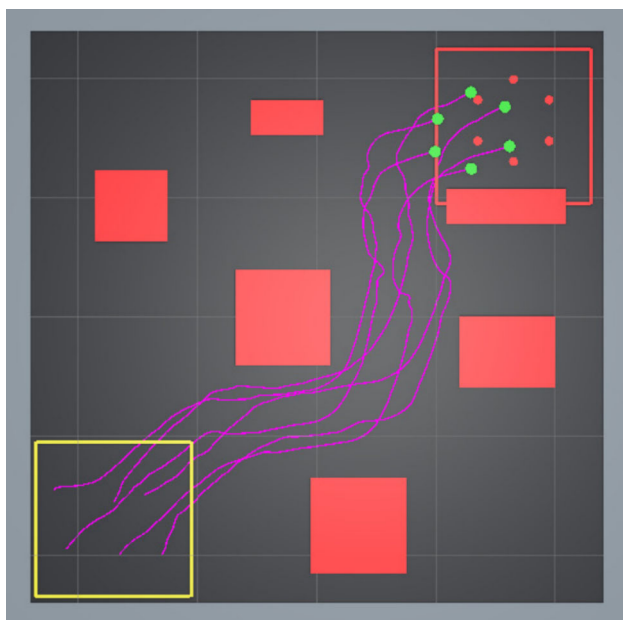


Fig. 11 Simulation results of circular formation in a square scenario with rectangular obstacles. The circular formation quickly forms, avoids the red obstacles, maintains the formation stably, and reaches the final target

aimed to reach targets on the right. As shown in Fig. 10, the agents quickly formed a circular formation and followed an S-shaped trajectory to bypass the obstacles while maintaining formation stability. In the second scenario, the agents similarly formed a stable circular formation and successfully

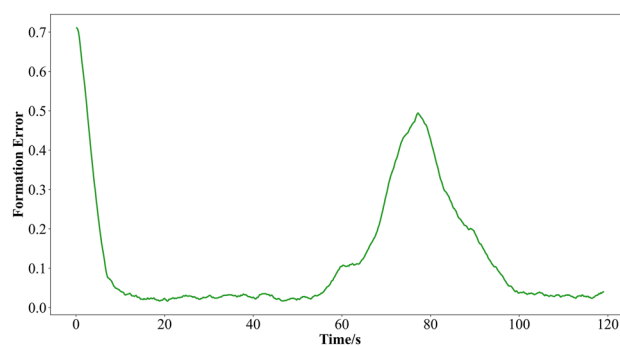
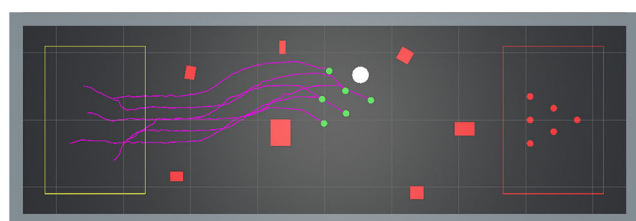


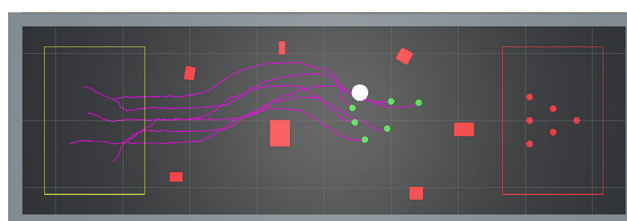
Fig. 13 The trend of formation error changes in a scenario with dynamic and static obstacles

avoided obstacles while moving toward the target in the top-right corner, as illustrated in Fig. 11.

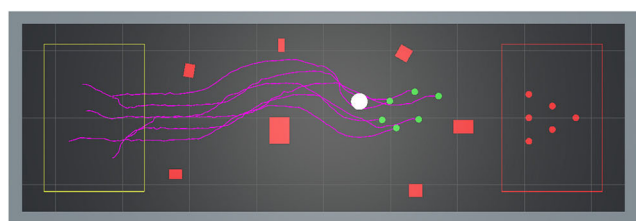
In the third scenario, multiple static and dynamic obstacles were placed, with the dynamic obstacles moving vertically within a limited range at random speeds. As shown in Fig. 12a, the agents quickly formed a triangular formation and avoided the static obstacles. When encountering a dynamic obstacle moving downward, the formation was compressed, causing deformation as shown in Fig. 12b. After passing the obstacle, the formation exhibited significant distortion, as seen in Fig. 12c. The formation then rapidly recovered to its triangular shape and continued moving toward the goal, as depicted in Fig. 12d. Figure 13 shows the formation error trend throughout this process.



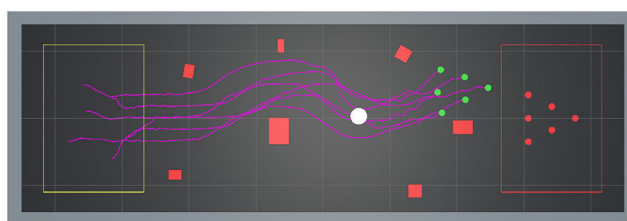
(a)



(b)



(c)



(d)

Fig. 12 Simulation results of triangular formation in a rectangular scenario with dynamic and static obstacles: **a** 6 agents form a triangular formation and approach a dynamic obstacle; **b** the formation starts to

deform; **c** severe deformation occurs as the formation moves away from the dynamic obstacle; **d** the formation quickly recovers and continues toward the goal

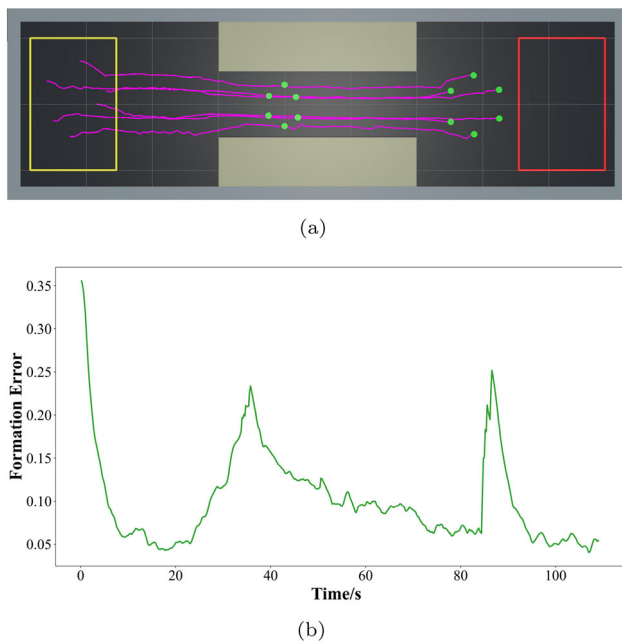


Fig. 14 Simulation results of formation shrinking: **a** The formation shrinks after entering the passage and restores its original shape after exiting. **b** The trend of formation error changes

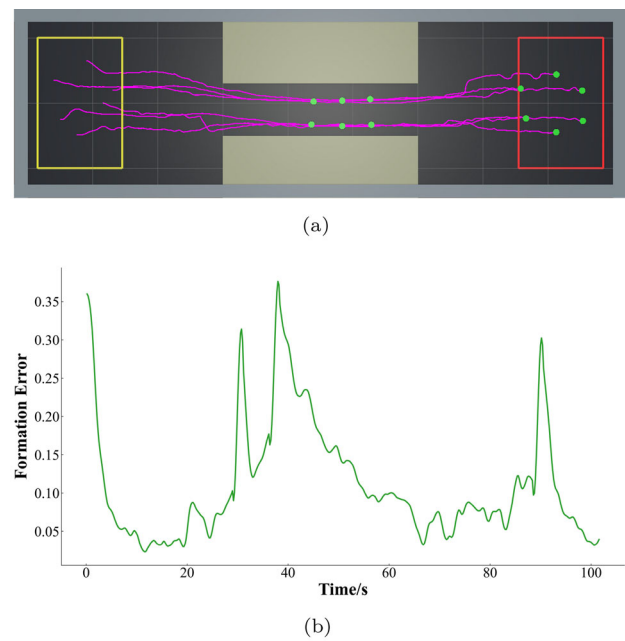


Fig. 15 Simulation results of formation shrinking followed by immediate restructuring: **a** The formation quickly shrinks upon entering the passage and promptly restructures into a rectangular shape, then restores to its original form after exiting. **b** The trend of formation error changes

5.2.4 Simulation results in narrow passage scenarios

We evaluate the proposed adaptive transformation mechanism in several narrow-passage scenarios. In Fig. 14a, when passing a moderately narrow passage, agents initiate shrink signals, and the formation uniformly scales down to 70% of its original size. It maintains this size during traversal and restores upon exit. As shown in Fig. 14b, formation error increases at 25 s due to deformation, but quickly converges after scaling and reallocation at 37 s and again after expansion at 88 s.

In the narrower passage scenario shown in Fig. 15a, sustained spatial constraints on boundary agents lead to continuous shrink requests, driving a decentralized reconfiguration into a rectangular formation that restores upon exit. The formation error peaks at approximately 25 s and then declines after coordinated shrinkage and reallocation, converging before the passage is completed, as illustrated in Fig. 15b. Figure 16a depicts traversal through two successive passages of different widths. The formation first uniformly scales to 70 percent in the wider section and subsequently adapts to a rectangular layout in the narrower section. A marked reallocation occurs near 30 s, and recovery toward the original circular geometry begins near 90 s, as shown in Fig. 16b. In Fig. 17a, positional heterogeneity causes the formation to split into two triangular subgroups. Reallocation and dual-triangle reconfiguration occur within approximately 35 to 42 s, as presented in Fig. 17b. These

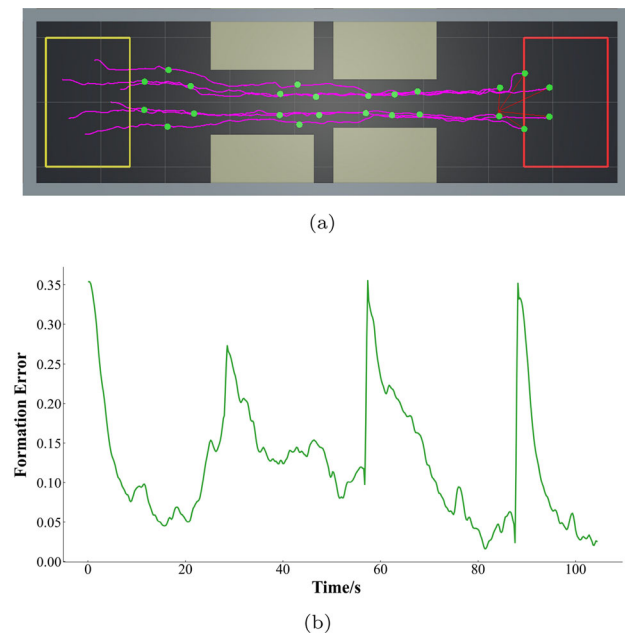


Fig. 16 Simulation results of the formation shrinking and restructuring: **a** The formation shrinks after entering the first passage, then restructures upon entering the second passage, and restores to its original shape after exiting; **b** the trend of formation error

observations confirm the mechanism's capability for distributed structural adaptation under spatial constraints.

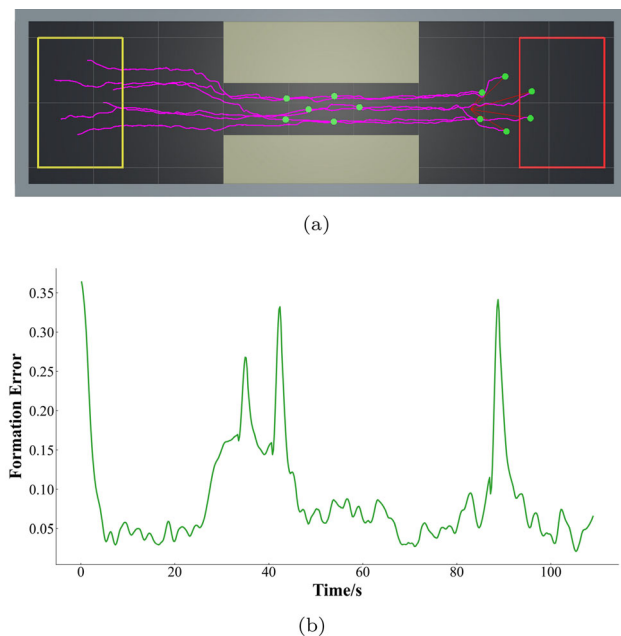


Fig. 17 Simulation results of formation shrinking followed by immediate restructuring: **a** The formation first shrinks upon entering the passage, then quickly restructures into two triangular sub-formations, and restores to its original shape after exiting. **b** The trend of formation error over time

6 Conclusion

This paper presents a MAPOCA-based distributed formation control algorithm tailored for virtual crowd simulation, integrating target allocation, navigation, formation generation, maintenance, and adaptive shape transformation. To minimize unnecessary agent displacement and ensure natural motion continuity, a formation alignment strategy is introduced, considering the initial spatial distribution of individuals during target assignment. A carefully designed reward function, along with a curriculum learning scheme, facilitates efficient training and enables agents to perform complex collective behaviors. In addition, an adaptive transformation mechanism is developed to dynamically switch between transformation modes based on environmental constraints and the current formation state, facilitating smooth and responsive shape adjustments. Experimental results demonstrate that the proposed model achieves low formation error and exhibits strong generalization across both open and cluttered virtual environments. Future work will focus on scaling the method to larger crowds and exploring richer and more diverse formation transformation behaviors for realistic crowd animation.

Acknowledgements This work was supported by the Key R&D Program of Jiangsu Province under Grant BE2023010-3, Jiangsu modern agricultural industry single technology research and development

project under Grant CX(23)3120 and National Key Laboratory of Ship Structural Safety.

Author Contributions LS and YQ conceived and designed the experiments; LS and YQ performed the experiments, analyzed results and wrote the paper, WQ given some advice, reviewed and edited the paper.

Data Availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

- Henry, J., Shum, H.P.H., Komura, T.: Environment-aware real-time crowd control. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. SCA '12, pp. 193–200. Eurographics Association, Goslar, DEU (2012)
- Gustafson, S., Arumugam, H., Kanyuk, P., Lorenzen, M.: Mure: fast agent based crowd simulation for VFX and animation. In: ACM SIGGRAPH 2016 Talks. SIGGRAPH '16. Association for Computing Machinery, New York, NY, USA (2016). <https://doi.org/10.1145/2897839.2927396>
- Kanyuk, P., Park, L.J.W., Weihrich, E.: Headstrong, hairy, and heavily clothed: animating crowds of Scotsmen. In: ACM SIGGRAPH 2012 Talks. SIGGRAPH '12. Association for Computing Machinery, New York, NY, USA (2015). <https://doi.org/10.1145/2343045.2771822>
- Ren, J., Jin, X., Deng, Z.: Crowd formation generation and control. In: Handbook of Human Motion, pp. 1–12 (2017)
- Feng, T., Yu, L.-F., Yeung, S.-K., Yin, K., Zhou, K.: Crowd-driven mid-scale layout design. ACM Trans. Graph. **35**(4), 132–1 (2016)
- Cai, M., Xu, Q., Chen, C., Wang, J., Li, K., Wang, J., Zhu, Q.: Formation control for connected and automated vehicles on multi-lane roads: relative motion planning and conflict resolution. IET Intell. Transp. Syst. **17**(1), 211–226 (2023)
- Zheng, X., Zong, C., Cheng, J., Xu, J., Xin, S., Tu, C., Chen, S., Wang, W.: Visually smooth multi-UAV formation transformation. Graph. Models **116**, 101111 (2021)
- Hughes, R.L.: The flow of human crowds. Annu. Rev. Fluid Mech. **35**(1), 169–182 (2003)
- Narain, R., Golas, A., Curtis, S., Lin, M.C.: Aggregate dynamics for dense crowd simulation. ACM Trans. Graph. **28**(5), 1–8 (2009). <https://doi.org/10.1145/1618452.1618468>
- Karamouzas, I., Heil, P., Van Beek, P., Overmars, M.H.: A predictive collision avoidance model for pedestrian simulation. In: Motion in Games: Second International Workshop, MIG 2009, Zeist, The Netherlands, November 21–24, 2009. Proceedings 2, pp. 41–52. Springer (2009)
- Parisi, D.R., Dorso, C.O.: Microscopic dynamics of pedestrian evacuation. Phys. A **354**, 606–618 (2005)
- Tang, K., Zhang, J., Shen, Y., Li, C., He, G.: KDPM: knowledge-driven dynamic perception model for evacuation scene simulation. Comput. Anim. Virtual Worlds **35**(3), 2279 (2024)
- Ren, W.: Consensus strategies for cooperative control of vehicle formations. IET Control Theory Appl. **1**(2), 505–512 (2007)
- Weiss, T.: Fast position-based multi-agent group dynamics. In: Proceedings of the ACM on Computer Graphics and Interactive Techniques, vol. 6, no. 1, pp. 1–15 (2023)

15. Van Den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. In: *Robotics Research: The 14th International Symposium ISRR*, pp. 3–19. Springer (2011)
16. Li, L., Lyu, C., Liu, H., Lyu, L.: Safety indication signs-based crowd division and leader selection approach for evacuation guidance. *IEEE Sens. J.* **24**(6), 8937–8948 (2024)
17. Wu, H., Wang, C., He, G.: Dynamic leader role modeling for self-organizing crowd evacuation simulation. *Comput. Anim. Virtual Worlds* **34**(1), 2115 (2023)
18. Xu, J., Xu, D., Wu, J., Shi, X.: Modeling the collective behavior of pedestrians with the spontaneous loose leader-follower structure in public spaces. *Comput. Aided Civ. Infrastruct. Eng.* **40**, 1956–1974 (2025)
19. Mao, T., Wang, J., Meng, R., Yan, Q., Liu, S., Wang, Z.: A fusion crowd simulation method: integrating data with dynamics, personality with common. *Comput. Graph. Forum* **41**, 131–142 (2022). (Wiley Online Library)
20. Liu, P., Chao, Q., Huang, H., Wang, Q., Zhao, Z., Peng, Q., Yip, M.K., Liu, E.S., Jin, X.: Velocity-based dynamic crowd simulation by data-driven optimization. *Vis. Comput.* **38**(9), 3499–3512 (2022)
21. Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction. *IEEE Trans. Neural Netw.* **9**(5), 1054–1054 (1998). <https://doi.org/10.1109/TNN.1998.712192>
22. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017)
23. Bai, C., Yan, P., Pan, W., Guo, J.: Learning-based multi-robot formation control with obstacle avoidance. *IEEE Trans. Intell. Transp. Syst.* **23**(8), 11811–11822 (2021)
24. Sui, Z., Pu, Z., Yi, J., Wu, S.: Formation control with collision avoidance through deep reinforcement learning using model-guided demonstration. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(6), 2358–2372 (2020)
25. Chang, L., Shan, L., Zhang, W., Dai, Y.: Hierarchical multi-robot navigation and formation in unknown environments via deep reinforcement learning and distributed optimization. *Robot. Comput. Integr. Manuf.* **83**, 102570 (2023)
26. Wei, X., Wang, H., Tang, Y.: Deep hierarchical reinforcement learning based formation planning for multiple unmanned surface vehicles with experimental results. *Ocean Eng.* **286**, 115577 (2023)
27. Lee, J., Won, J., Lee, J.: Crowd simulation by deep reinforcement learning. In: *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*, pp. 1–7 (2018)
28. Charalambous, P., Pettre, J., Vassiliades, V., Chrysanthou, Y., Pelechano, N.: Greil-crowds: crowd simulation with deep reinforcement learning and examples. *ACM Trans. Graph. (TOG)* **42**(4), 1–15 (2023)
29. Kwiatkowski, A., Kalogeiton, V., Pettr , J., Cani, M.-P.: Reward function design for crowd simulation via reinforcement learning. In: *Proceedings of the 16th ACM SIGGRAPH Conference on Motion, Interaction and Games*, pp. 1–7 (2023)
30. Zhang, Y., Yang, B., Zhu, J.: A double-layer crowd evacuation simulation method based on deep reinforcement learning. *Comput. Anim. Virtual Worlds* **35**(3), 2280 (2024)
31. Xu, P., Karamouzas, I.: Human-inspired multi-agent navigation using knowledge distillation. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8105–8112. IEEE (2021)
32. Wang, Z.-S., Song, C.G., Lee, J., Kim, J.-H., Kim, S.-J.: Controllable swarm animation using deep reinforcement learning with a rule-based action generator. *IEEE Access* **10**, 48472–48485 (2022)
33. Sun, L., Yan, J., Qiu, Y., Qin, W.: The crowd cooperation approach for formation maintenance and collision avoidance using multi-agent deep reinforcement learning. *Vis. Comput.* **41**, 1–15 (2024)
34. Zhao, S.: Affine formation maneuver control of multiagent systems. *IEEE Trans. Autom. Control* **63**(12), 4140–4155 (2018)
35. Li, Y., Dong, W.: Flexible affine formation control based on dynamic hierarchical reorganization. *IEEE Robot. Autom. Lett.* **9**(12), 11290–11297 (2024). <https://doi.org/10.1109/LRA.2024.3490407>
36. Li, X., Zhou, R., Sun, G.: Adaptive shape formation against swarm-scale variants in robot swarms. *IEEE Robot. Autom. Lett.* **9**(8), 6776–6783 (2024). <https://doi.org/10.1109/LRA.2024.3414210>
37. Pan, C., Yan, Y., Zhang, Z., Shen, Y.: Flexible formation control using Hausdorff distance: a multi-agent reinforcement learning approach. In: *2022 30th European Signal Processing Conference (EUSIPCO)*, pp. 972–976. IEEE (2022)
38. Xu, M., Wu, Y., Ye, Y.: Smooth and efficient crowd transformation. In: *Proceedings of the 20th ACM International Conference on Multimedia*, pp. 1189–1192 (2012)
39. Gu, Q., Deng, Z.: Generating freestyle group formations in agent-based crowd simulations. *IEEE Comput. Graph. Appl.* **33**(1), 20–31 (2011)
40. Yan, D., Huang, K., Zhang, L., Ding, G.Y.: Stylized crowd formation transformation through spatiotemporal adversarial learning. *Adv. Intell. Syst.* **6**(3), 2300563 (2024)
41. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: *Machine Learning Proceedings 1994*, pp. 157–163. Elsevier (1994)
42. Choi, H.-L., Brunet, L., How, J.P.: Consensus-based decentralized auctions for robust task allocation. *IEEE Trans. Robot.* **25**(4), 912–926 (2009)
43. Cohen, A., Teng, E., Berges, V.-P., Dong, R.-P., Henry, H., Mattar, M., Zook, A., Ganguly, S.: On the use and misuse of absorbing states in multi-agent reinforcement learning. *arXiv preprint arXiv:2111.05992* (2021)
44. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48 (2009)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.