scientific reports

Check for updates

OPEN

Emergent behaviors in multiagent pursuit evasion games within a bounded 2D grid world

Sihan Xu[1] & Zhaohui Dang[1,2]✉

This study investigates emergent behaviors in multi-agent pursuit-evasion games within a bounded 2D grid world, where both pursuers and evaders employ multi-agent reinforcement learning (MARL) algorithms to develop adaptive strategies. We define six fundamental pursuit actions—flank, engage, ambush, drive, chase, and intercept—which combine to form 21 types of composite actions during two-pursuer coordination. After training with MARL algorithms, pursuers achieved a 99.9% success rate in 1,000 randomized pursuit-evasion trials, demonstrating the effectiveness of the learned strategies. To systematically identify and measure emergent behaviors, we propose a K-means-based clustering methodology that analyzes the trajectory evolution of both pursuers and evaders. By treating the full set of game trajectories as statistical samples, this approach enables the detection of distinct behavioral patterns and cooperative strategies. Through analysis, we uncover emergent behaviors such as lazy pursuit, where one pursuer minimizes effort while complementing the other's actions, and serpentine movement, characterized by alternating drive and intercept actions. We identify four key cooperative pursuit strategies, statistically analyzing their occurrence frequency and corresponding trajectory characteristics: serpentine corner encirclement, stepwise corner approach, same-side edge confinement, and pincer flank attack. These findings provide significant insights into the mechanisms of behavioral emergence and the optimization of cooperative strategies in multi-agent games.

Pursuit-evasion (P-E) games, inspired by predator-prey interactions in nature[1,2], represent a classic problem in multi-agent systems. This problem has garnered significant research interest across diverse fields, including UAV formation control[3], robotic swarm coordination[4], and non-cooperative target engagement in spacecraft systems[5,6]. At the core of these games is the adversarial dynamic between multiple pursuers and evaders, where pursuers aim to capture their targets while evaders strive to evade pursuit by dynamically adapting their movements. Existing studies primarily focus on improving individual agents' strategies through game theory, optimization algorithms, or machine learning techniques, enhancing competitive performance[7]. However, these approaches typically concentrate on individual strategies and often overlook the potential benefits of cooperation between agents. As such, exploring how cooperation and emergent behaviors arise in P-E games is vital for a more comprehensive understanding of multi-agent decision-making and strategy optimization.

Emergent behavior refers to the spontaneous development of complex and organized patterns at the macroscopic level, driven by local interactions or individual rules within a system. As a key concept in artificial life research, emergent behavior helps in understanding the mechanisms of cooperation in biological systems, which are often emulated in computational models. In nature, phenomena such as flocks of birds and schools of fish serve as classic examples of self-organizing entities, where global structures and dynamic behaviors emerge from local interactions among individuals[8,9]. Inspired by these biological systems, artificial life models are developed to simulate and explore cooperative behaviors and their underlying mechanisms. Due to the challenges of directly obtaining data from natural systems, multi-agent systems (MAS) and multi-robot systems (MRS) are often employed as simulation tools[10]. While classical methods, such as[11] explored by Fisac and Sastry, aim to derive optimal solutions through formal analysis , they tend to limit the ability of game participants to explore new strategies. In contrast, intelligent approaches like multi-agent reinforcement learning (MARL) can capture the dynamic, adaptive nature of multi-agent interactions. These methods allow agents to learn and adjust their strategies in real time, based on continuous feedback from the environment and other agents. This adaptability enables the emergence of unexpected and sophisticated strategies, which classical differential game theory may fail to anticipate or model.

[1]School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China. [2]National Key Laboratory of Aerospace Flight Dynamics, Northwestern Polytechnical University, Xi'an 710072, China. ✉email: dangzhaohui@nwpu.edu.cn

Several studies have applied these models to investigate cooperative behaviors in pursuit-evasion tasks. For example, Yong et al.[12] employed the Multi-Agent ESP (Enforced Subpopulations) method and found that three pursuers could effectively cooperate using a chaser-blocker strategy along horizontal, vertical, or diagonal trajectories. Nitschke et al.[13] compared three artificial evolutionary approaches to designing cooperative behaviors in simulated robots, revealing an emergent role-switching strategy in predator-prey systems. In this strategy, predators dynamically switch between intercepting and striking roles, coordinating their actions to limit prey mobility. Similarly, Nishimura et al.[2] identified several collective strategies, such as rotational clustering and linear formations, within predator-prey systems. Additionally, Lee et al.[14] explored the interplay between emergent cooperative behaviors and individual actions, showing that cooperation generally improves performance, especially for agents directly involved in pursuit. In a related study, Masuko et al.[15] introduced "lazy pursuers" in pursuit-evasion games, revealing that a division-of-labor strategy significantly increased capture efficiency by enabling pincer attack patterns formed through the coordinated efforts of diligent and lazy pursuers.

Emergent behavior is a broader concept that encompasses various forms of behavior, including cooperative behavior. While cooperative behaviors often emerge from interactions among agents, they represent just one subset of the broader category of emergent phenomena. In multi-agent systems, emergent behaviors may be collaborative or non-collaborative, depending on the agents' interactions and their strategies. Traditionally, identifying emergent behavior in artificial systems involved comparing the outcomes of pursuers and evaders across various strategies, relying on human observation and qualitative analysis to describe these phenomena. However, with the advent of machine learning in recent years, the focus has shifted towards more quantitative approaches for analyzing these behaviors. For example, the Neural Information Squeezer framework enables the automatic extraction of coarse-grained strategies and macro-level dynamics[16], helping identify causal emergent phenomena from time-series data. Furthermore, Sturdivant et al.[17] proposed a conceptual model for emergence and downward causation, establishing the necessary and sufficient conditions for a phenomenon to be classified as emergent in complex systems. This model has been successfully applied to the study of symbolic emergence in robotics.

Emergent behavior has also been extensively studied in other fields, providing a strong foundation for the study of similar phenomena in multi-agent systems. From a sociological perspective, interactions between individuals and their environments give rise to emergent phenomena such as social segregation, consensus formation, behavioral contagion, epidemic outbreaks, and traffic congestion, illustrating how complex collective behaviors emerge from local rules[18,19]. In contrast, the study of emergent behavior in pursuit-evasion games remains underdeveloped, with several foundational gaps and insufficient research methods. First, the concept of emergent behavior in pursuit-evasion games remains ambiguous, and the definitions of emergent phenomena are yet to be fully articulated. Moreover, existing research methods, often based on qualitative analyses and human observation, prove insufficient as the complexity of behaviors increases across diverse strategies. If these emergent behaviors–whether cooperative or not–can be thoroughly identified and validated, they hold significant potential for applications in various domains, including telecommunications[20], aerospace and space exploration, and multi-robot systems[21,22].

To address these limitations, this paper proposes a novel analytical framework specifically designed to investigate the emergence of cooperative behaviors in pursuit-evasion games. The primary innovations of this work are as follows:

(1) Systematic cooperative action set construction: This study introduces a systematically constructed cooperative action set, which provides a structured approach for statistical analysis of emergent cooperative behavior patterns. The action set is designed to facilitate a more refined understanding of how cooperative behaviors emerge and evolve under various conditions, enabling more robust analysis of these behaviors within multi-agent systems.
(2) Clustering-based behavior recognition method: A novel clustering-based method is proposed for the effective classification and analysis of distinct types of cooperative behaviors. This approach leverages advanced clustering techniques to group similar action sequences, thereby identifying key behavioral patterns. By analyzing the action composition and unique characteristics of emergent cooperative behaviors, the study offers new insights into the underlying mechanisms driving cooperation in pursuit-evasion games. Additionally, this method provides a more rigorous and scalable solution for handling complex multi-agent interactions compared to traditional qualitative analysis.
(3) Identification of four emergent cooperative pursuit strategies and analysis of behavioral characteristics behind them: This study identifies and characterizes four distinct types of emergent cooperative pursuit strategies: serpentine corner encirclement, stepwise corner approach, same-side edge confinement, and pincer flank attack. Among these, a particularly novel finding is the emergence of lazy pursuit behavior, where one pursuer minimizes its efforts while complementing the actions of the other. This behavior highlights the importance of cooperative game theory, as it underscores the efficiency of teamwork even when individual contributions are uneven. The identification of lazy pursuit behavior contributes to a deeper understanding of how asymmetric roles within cooperative strategies can lead to more efficient pursuit dynamics, offering valuable insights into the design of multi-agent cooperative systems.

The remaining part of this paper is organized as follows: "Planar grid pursuit-evasion game model" introduces the methodology for constructing the planar grid-based pursuit-evasion game model and outlines the method for generating pursuit-evasion strategies using multi-agent reinforcement learning (MARL). "Methodology for identifying emergent behavior" details the research methodology, focusing on two key aspects: statistical analysis of cooperative pursuit actions and behavioral clustering. "Emergent behaviors and characteristics" provides a

simulation analysis, offering statistical insights into the emergent cooperative pursuit behaviors of intelligent pursuers in planar grid strategies, along with an exploration of the underlying behavioral mechanisms. Finally, the paper concludes with a summary of the key findings in "Conclusion".

Planar grid pursuit-evasion game model

Model abstraction and generalizability

In 2014, the Moser couple identified grid cells in the entorhinal cortex, a discovery that earned them the Nobel Prize in Physiology or Medicine. They revealed how the brain utilizes a "spatial coordinate system" for precise positioning and navigation[23]. Building on this breakthrough, scientists at the RIKEN Institute in Japan later uncovered predictive grid cells–neurons capable of forecasting an individual's future spatial location by accounting for both distance and time, thus enabling forward-looking planning[24]. This mechanism of spatial navigation in the brain inspired the use of grid-based models in various fields, including pursuit-evasion games. It draws inspiration not only from biological evidence, but also from established modeling practices in cognitive science and robotics. Specifically, studies have shown that both animals and humans perform spatial navigation and decision-making based on internalized "landmark-based" or "grid-like" representations–especially within the entorhinal cortex, where grid cells are found to support discretized spatial mapping. These findings support the plausibility of modeling pursuit-evasion behaviors on a discrete grid, particularly in scenarios that simulate cognitive-level planning or strategic reasoning. Just as the brain relies on spatial grids for navigation and decision-making, pursuit-evasion games can be effectively modeled within a similar grid-like environment, where the movements and strategies of pursuers and evaders unfold in a two-dimensional space. Accordingly, we adopt a grid-based model in this study to simulate pursuit-evasion dynamics and explore the emergent behaviors arising from agent interactions.

Although this abstraction may appear simplified, it is both intentional and well supported by prior literature. In real-world applications, many scenarios indeed involve continuous and dynamic motion–such as aerial drones or autonomous vehicles navigating open space. However, there also exist numerous practical cases where pursuit-evasion behaviors take place in naturally discrete or discretized environments, such as grid-based warehouse logistics, patrolling in urban environments with road constraints, and turn-based decision models in multi-agent video games. Even for inherently continuous problems, it is common practice to apply grid-based approximations for computational tractability and policy learning, as seen in robotics and reinforcement learning research. This abstraction is particularly useful for our primary objective: to analyze the emergence of multi-agent strategies from the ground up. The simplified grid setting enables clear identification of action-level and strategy-level emergent patterns, which would otherwise be significantly harder to observe and interpret in continuous or high-dimensional settings. Moreover, the methodology and analytical framework developed in this study can serve as a foundation for future extensions to more complex, dynamic, and even continuous environments.

Kinematic model

The pursuit-evasion game on a two-dimensional plane involves two pursuers and one evader, all of whom make decisions and move simultaneously. This game is typically described using five elements: Player, State, Action, Strategy, and Objective.

a. Players are the entities that execute actions within the pursuit-evasion game. They primarily consist of pursuers and evaders, each possessing independent decision-making capabilities and strategies aimed at achieving their respective objectives.
b. State refers to the configuration of the system at a given moment, including the positions of all agents on the grid. In multi-agent systems, a state is commonly represented as a vector or matrix containing the locations of all pursuers and the evader. The set of all possible states in the grid world is defined as the State Space (S).
c. Actions are the set of possible operations an agent can choose from at each time step. In a two-dimensional grid world, common actions include moving up, moving down, moving left, moving right, and staying in place. The set of actions can be expanded or restricted based on the model's specific requirements. The set of all available movements in the grid world is defined as the Action Space (M).
d. Strategies are the rules or methods agents use to select actions, typically based on the current state and historical information. Strategies may be deterministic (where each state corresponds to a specific action) or stochastic (where actions are chosen with certain probabilities). In more complex models, strategies can be optimized using learning algorithms such as reinforcement learning.
e. Objectives define the ultimate goals of the agents within the game. For pursuers, the objective is typically to capture the evaders as quickly as possible. For evaders, the goal is to evade capture for as long as possible. These clear objectives guide agents in selecting appropriate strategies and behaviors.

As shown in Fig. 1, the grid is bounded on all sides. Due to these environmental constraints, the game operates in discrete time and space. At each time step, both pursuers and the evader make decisions based on their current positions and move to their respective locations in the following step. The game assumes a setting of complete information, meaning all agents are aware of each other's positions at all times. It is important to note that this simplified planar point-mass model does not account for additional factors, such as fuel consumption. Based on these assumptions, the motion models for both the pursuers and the evader are formulated as follows:

$$\begin{aligned} &\boldsymbol{X}_i(t+1) = \boldsymbol{X}_i(t) + \boldsymbol{u}_i(t) (i \in \{P_1, P_2, E\}, t \in \mathbb{N}) \\ &s.t. \begin{cases} 0 \leq x_i \leq x_{\max} \\ 0 \leq y_i \leq y_{\max} \end{cases} \end{aligned} \tag{1}$$

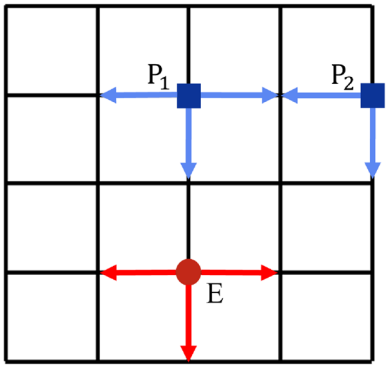

Fig. 1. The diagram of the pursuit-evasion game on a planar grid scenario.

where $\boldsymbol{X}_i(t) = [x_i(t), y_i(t)]$ represent the position of game participant i at time step t, x_{max} and y_{max} are the boundaries of the grid. The number of decisions made during the game is consistent with the total number of rounds, N. $\boldsymbol{u}_P(t)$ and $\boldsymbol{u}_E(t)$ denote the control variables of the pursuers and the evader at time step t, respectively. These control variables can be selected from the action space $\boldsymbol{M}_P$ and $\boldsymbol{M}_E$, with options for no movement or movement in one of the four directions: up, down, left, or right. The magnitude of the control variable remains constant during each movement. $u_P^{\max}$ and $u_E^{\max}$ denote the maximum control capacities of the pursuers and the evader, respectively, indicating the maximum number of grid cells they can traverse in a single step. These control variables are subject to the following constraints shown in Eqs. (2)~(3):

$$\underset{P\in\{P_1,P_2\}}{\boldsymbol{u}_P(t)} \in \boldsymbol{M}_P = \{[0, u_P^{\max}], [0, -u_P^{\max}], [-u_P^{\max}, 0], [u_P^{\max}, 0], [0, 0]\}, (u_P^{\max} \in \mathbb{N}^+) \tag{2}$$

$$\boldsymbol{u}_E(t) \in \boldsymbol{M}_E = \{[0, u_E^{\max}], [0, -u_E^{\max}], [-u_E^{\max}, 0], [0, u_E^{\max}], [0, 0]\}, (u_E^{\max} \in \mathbb{N}^+) \tag{3}$$

In the pursuit-evasion game, the objective of the pursuers is to capture the evader as quickly as possible, whereas the evader seeks to maximize the distance from the pursuers and delay capture for as long as possible. Therefore, the game objectives of both sides can be effectively represented in terms of terminal time:

$$\begin{aligned} t_f &= \min\left\{\min_{t_1\in\boldsymbol{S}_1}\{t_1\}, \min_{t_2\in\boldsymbol{S}_2}\{t_2\}, t_N\right\} \\ s.t. &\begin{cases} \boldsymbol{S}_1 = \{t \,|r_1(t) \le r_s\} \\ \boldsymbol{S}_2 = \{t \,|r_2(t) \le r_s\} \end{cases} \end{aligned} \tag{4}$$

where t_f denote the terminal time of the game, defined as the shortest time required for the pursuers to capture the evader. t_N represents the maximum duration for which the game can persist, corresponding to the total number of game rounds, N. Since both the pursuers and the evader can only move along the grid lines to adjacent vertices and are prohibited from diagonal movement, r_s is defined as the grid-line distance between the pursuers and the evader at the moment of successful capture. The distances r_1 and r_2 are the grid-line distances between pursuers P_1, P_2 and the evader respectively, calculated as follows:

$$r_1(t) = \|\boldsymbol{X}_{P_1}(t) - \boldsymbol{X}_E(t)\|_1 \tag{5}$$

$$r_2(t) = \|\boldsymbol{X}_{P_2}(t) - \boldsymbol{X}_E(t)\|_1 \tag{6}$$

To this end, the objective of the pursuers in the game is to determine the optimal control variables $\boldsymbol{u}_{P_1}^*, \boldsymbol{u}_{P_2}^*$ that minimizes the terminal time t_f, while the objective of the evader is to identify the optimal control variable $\boldsymbol{u}_E^*$ that maximizes t_f:

$$\{\boldsymbol{u}_{P_1}^*, \boldsymbol{u}_{P_2}^*\} \in S_P(\boldsymbol{u}_{P_1}, \boldsymbol{u}_{P_2}) = \arg\min_{\boldsymbol{u}_{P_1}, \boldsymbol{u}_{P_2}} t_f(\boldsymbol{X}_{P_1}(t_0), \boldsymbol{X}_{P_2}(t_0), \boldsymbol{X}_E(t_0), \boldsymbol{u}_{P_1}, \boldsymbol{u}_{P_2}, \boldsymbol{u}_E^*) \tag{7}$$

$$\{\boldsymbol{u}_E^*\} \in S_E(\boldsymbol{u}_E) = \arg\max_{\boldsymbol{u}_E} t_f(\boldsymbol{X}_{P_1}(t_0), \boldsymbol{X}_{P_2}(t_0), \boldsymbol{X}_E(t_0), \boldsymbol{u}_{P_1}^*, \boldsymbol{u}_{P_2}^*, \boldsymbol{u}_E) \tag{8}$$

Here, $\boldsymbol{X}_{P_1}(t_0)$, $\boldsymbol{X}_{P_2}(t_0)$, and $\boldsymbol{X}_E(t_0)$ denote the initial positions of the two pursuers P_1, P_2, and the evader E, respectively. The optimal strategies $\boldsymbol{u}_{P_1}^*, \boldsymbol{u}_{P_2}^*$ belong to the solution set S_P, which is obtained by minimizing the capture time t_f given a fixed evader strategy $\boldsymbol{u}_E^*$. Conversely, the evader's optimal strategy $\boldsymbol{u}_E^*$ belongs to the solution set S_E, which is derived by maximizing t_f given the pursuers' strategies $\boldsymbol{u}_{P_1}^*, \boldsymbol{u}_{P_2}^*$. It is worth noting that both S_P and S_E represent sets of admissible solutions rather than unique strategies. That is, multiple distinct strategies may satisfy the optimization criteria, reflecting the possibility of non-unique optimal solutions.

Intelligent pursuit and evasion strategies

The key to studying emergent behavior lies in ensuring that the pursuit and evasion strategies evolve naturally through self-learning and interactions, without external guidance. To achieve this, we adopt a Multi-Agent Reinforcement Learning (MARL) approach to simulate the fundamental dynamics of pursuit-evasion interactions observed in nature. Among MARL frameworks, the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm[25], which combines centralized training with decentralized execution, is particularly well-suited for this scenario. Notably, in designing the reward function for training intelligent pursuit-evasion strategies, distance is treated purely as an outcome-based reward term and is not included in the guiding rewards. Accordingly, considering the game objectives of both pursuers and evaders, the reward function is formulated as the sum of process rewards and outcome rewards, as shown in Eqs. (9) to (11) respectively:

$$R^{P_1}(t) = R_T^{P_1}(t) + R_D^{P_1}(t) \tag{9}$$

$$R^{P_2}(t) = R_T^{P_2}(t) + R_D^{P_2}(t) \tag{10}$$

$$R^{E}(t) = R_T^{E}(t) + R_D^{E}(t) \tag{11}$$

The process rewards during the game are based solely on time, with negative rewards assigned to the pursuers and positive rewards to the evader until capture occurs. The constant ρ represents the magnitude of the time-based reward $R_T^{P_1}(t)$,$R_T^{P_2}(t)$ and $R_T^{E}(t)$. The process rewards functions for pursuers and evader are defined in Eqs. (12) and (13), respectively:

$$R_T^{P_1}(t) = R_T^{P_2}(t) = \begin{cases} -\rho, & t < t_f \\ 0, & t = t_f \end{cases} \tag{12}$$

$$R_T^{E}(t) = \begin{cases} +\rho, & t < t_f \\ 0, & t = t_f \end{cases} \tag{13}$$

The terminal reward is based on the outcome of the pursuit, evaluated using a distance-based criterion. If the pursuers successfully capture the evader, they each receive a positive reward $R_D^{P_1}(t)$ and $R_D^{P_2}(t)$ respectively; conversely, if the evader evades capture, a positive reward $R_D^{E}(t)$ is awarded to the evader. The terminal reward functions for pursuers and evader are formulated in Eqs. (14) and (15), respectively:

$$R_D^{P_1}(t) = R_D^{P_2}(t) = \begin{cases} +D_L, & (r_1(t) \le r_s) \cup (r_2(t) \le r_s) \cap (t = t_f) \\ -D_L, & (r_1(t) > r_s) \cap (r_2(t) > r_s) \cap (t = t_f) \\ 0, & t < t_f \end{cases} \tag{14}$$

$$R_D^{E}(t) = \begin{cases} -D_L, & (r_1(t) \le r_s) \cup (r_2(t) \le r_s) \cap (t = t_f) \\ +D_L, & (r_1(t) > r_s) \cap (r_2(t) > r_s) \cap (t = t_f) \\ 0, & t < t_f \end{cases} \tag{15}$$

Here, D_L is a constant representing the magnitude of the terminal reward. For the pursuers, longer pursuit times result in larger negative rewards. Upon successfully capturing the evader, the pursuers receive a substantial positive terminal reward, marking the end of the game. In contrast, the evader aims to maximize the time before capture and avoid being caught. The evader earns a positive time-based reward for delaying capture but incurs a significant negative terminal reward if captured. This design encourages the evader to maintain distance from the pursuers during training.

Next, we briefly introduce the basic principles of the MADDPG algorithm[26]. MADDPG extends the Deep Deterministic Policy Gradient (DDPG) algorithm to a multi-agent reinforcement learning framework by incorporating centralized training with decentralized execution. In this setting, each agent learns a deterministic policy while leveraging a centralized critic network that has access to global state and action information, facilitating coordinated decision-making in competitive environments. Building on this foundation, the learning-based framework adopted in our study enables agents to discover high-performing strategies through interaction and feedback. Importantly, both sides–pursuers and evaders–are optimized simultaneously, allowing each agent to adapt in response to the evolving strategies of others. Specifically, our approach seeks to approximate a mutual best-response equilibrium, where each agent learns to act optimally in response to the evolving strategies of others.

Each agent i maintains a policy function $\boldsymbol{\pi}_{\boldsymbol{\theta}_i}$ parameterized by $\boldsymbol{\theta}_i$, which deterministically maps its state to a movement: $\boldsymbol{u}_i = \boldsymbol{\pi}_{\boldsymbol{\theta}_i}(\boldsymbol{O}_i)$. Here, $\boldsymbol{O}_i$ is the environmental state observed by the agent, which consists of the agent's own position and the relative position differences between itself and other agents, expressed as $\boldsymbol{O}_i = [\boldsymbol{X}_i, \boldsymbol{X}_j - \boldsymbol{X}_i, \boldsymbol{X}_k - \boldsymbol{X}_i], (i, j, k \in \{P_1, P_2, E\}, i \neq j \neq k)$. The corresponding critic function evaluates the expected return by considering the global state and the actions of all agents: $\boldsymbol{Q}_{\boldsymbol{\phi}_i}(\boldsymbol{O}_{P_1}, \boldsymbol{O}_{P_2}, \boldsymbol{O}_E, \boldsymbol{u}_{P_1}, \boldsymbol{u}_{P_2}, \boldsymbol{u}_E)$, where $\boldsymbol{\phi}_i$ is the parameters of the value estimation network for agent i. By training the critic with full state-action information while maintaining decentralized policies, MADDPG enables agents to learn competitive and cooperative strategies in pursuit-evasion games.

The optimization function of the policy estimation network is provided as $J(\boldsymbol{\theta}_i) = \mathbb{E}(R_{\boldsymbol{\pi}_i})$. The policy is improved by applying the deterministic policy gradient:

$$\nabla_{\boldsymbol{\theta}_i} J(\boldsymbol{\theta}_i) = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}_i} \boldsymbol{\pi}_i(\boldsymbol{u}_i \mid \boldsymbol{O}_i) \nabla_{\boldsymbol{u}_i} \boldsymbol{Q}_{\boldsymbol{\phi}_i}(\boldsymbol{O}_{P_1}, \boldsymbol{O}_{P_2}, \boldsymbol{O}_E, \boldsymbol{u}_{P_1}, \boldsymbol{u}_{P_2}, \boldsymbol{u}_E)\right] \tag{16}$$

During the training process, experience replay and target networks are essential components. The experience replay mechanism stores experience samples, which consist of the current joint state, joint action, environmental reward, and the subsequent joint state, encapsulating them into a single sample for further learning.

The update formula for the value estimation network is subsequently presented as follows:

$$y_i = R_i + \gamma Q'_{\phi_i}\left(\boldsymbol{O'}_{P_1}, \boldsymbol{O'}_{P_2}, \boldsymbol{O'}_E, \boldsymbol{u'}_{P_1}, \boldsymbol{u'}_{P_2}, \boldsymbol{u'}_E\right)\Big|_{\boldsymbol{u'}_j=\boldsymbol{\pi'}_j(\boldsymbol{O'}_j)} \tag{17}$$

$$\nabla_{\phi_i} J\left(\phi_i\right) = \mathbb{E}\left(\left(\boldsymbol{Q}_{\phi_i}\left(\boldsymbol{O}_{P_1}, \boldsymbol{O}_{P_2}, \boldsymbol{O}_E, \boldsymbol{u}_{P_1}, \boldsymbol{u}_{P_2}, \boldsymbol{u}_E\right) - y_i\right)^2\right) \tag{18}$$

where $\boldsymbol{Q}'_{\phi'_i}$ represents the target network for the value estimation network of agent i, while $\boldsymbol{\pi}'_{\theta'_i}$ denotes the target network for the policy network of agent i.

The calculation of the value gradient involves two steps. First, compute the label y as shown in Eq. 17. The value y is determined by the immediate reward R_i, the discount factor γ and the estimated value of the next joint state-action at the subsequent time step $\boldsymbol{Q}'_{\phi'_i}\left(\boldsymbol{O'}_{P_1}, \boldsymbol{O'}_{P_2}, \boldsymbol{O'}_E, \boldsymbol{u'}_{P_1}, \boldsymbol{u'}_{P_2}, \boldsymbol{u'}_E\right)$. At the next time step, the movement selection is provided by the policy auxiliary network based on the state at that time, while the state-action at the next time step is given by the value estimation auxiliary network. The second step, as shown in Eq. 18, involves computing the mean squared error between the label y and the estimated value from the value estimation network. This error serves as the optimization criterion for the entire value estimation network. The parameters ϕ_i of the value estimation network are updated through gradient descent to minimize this optimization criterion, completing the learning process. The auxiliary training networks here can enhance the convergence of the algorithm. The parameters of the auxiliary training networks are not updated through direct learning, but via soft updates. Let the parameters of the policy estimation auxiliary network and the value estimation auxiliary network be denoted as $\boldsymbol{\theta}'_i$ and $\boldsymbol{\phi}'_i$, respectively. The soft update process is governed by the soft update coefficient τ, as formulated in the following equation:

$$\begin{cases} \boldsymbol{\theta'}_i \leftarrow \tau\boldsymbol{\theta}_i + (1-\tau)\boldsymbol{\theta'}_i \\ \boldsymbol{\phi'}_i \leftarrow \tau\boldsymbol{\phi}_i + (1-\tau)\boldsymbol{\phi'}_i \end{cases} \tag{19}$$

The specific algorithmic process can be represented in Algorithm 1.

Initialize the state space S and action space M_P,M_E for the pursuers and evaders.
Initialize the parameters of the actor network $\boldsymbol{\pi}_{\theta_i}(\boldsymbol{O}_i)$ and the critic network $\boldsymbol{Q}_{\phi i}(\boldsymbol{O}_{P_1}, \boldsymbol{O}_{P_2}, \boldsymbol{O}_E, \boldsymbol{u}_{P_1}, \boldsymbol{u}_{P_2}, \boldsymbol{u}_E)$.
Initialize the target network $\boldsymbol{\pi}'_{\theta'_i}(\boldsymbol{O}_i)$ and $\boldsymbol{Q}'_{\phi' i}(\boldsymbol{O}_{P_1}, \boldsymbol{O}_{P_2}, \boldsymbol{O}_E, \boldsymbol{u}_{P_1}, \boldsymbol{u}_{P_2}, \boldsymbol{u}_E)$.
for $episode = 1$ to $maxepisodes$ **do**
  Randomly initialize the initial states $\boldsymbol{X}_i(t_0)$ of agent i for the pursuers and the evader on the grid points.
  for $t = 1$ to N **do**
    for $i \in \{P_1, P_2, E\}$ **do**
      Obtain the observation $\boldsymbol{O}_i$
      Calculate the control input: $\boldsymbol{u_i} = \boldsymbol{\pi}_{\theta_i}(\boldsymbol{O}_i)$, $(\boldsymbol{u}_i \in M_i = \{\text{up}, \text{down}, \text{left}, \text{right}, \text{stay}\})$
      State transition: $\boldsymbol{X}'_i = [x', y'] = \boldsymbol{X}_i + \boldsymbol{u}_i$
      if $x'(y') < 0$ **then** // boundary constraint
        $x'(y') = 0$
      else if $x'(y') > x_{max}(y_{max})$ **then**
        $x'(y') = x_{max}(y_{max})$
      end if
      Calculate the immediate reward: $R^i(t) = R^i_T(t) + R^i_D(t)$
      if $r_1 \le d_s$ or $r_2 \le d_s$ **then**
        break
      end if
      Store the experience $(\boldsymbol{O}_i, \boldsymbol{u}_i, R_i, \boldsymbol{O}'_i)$ in the experience replay buffer $\mathscr{D}$.
      Randomly sample a batch of data $(\boldsymbol{O}_i(t), \boldsymbol{u}_i, R^i, \boldsymbol{O}'_i)$ from $\mathscr{D}$.
      $y_i = R_i + \gamma Q'_{\phi i}\left(\boldsymbol{O'}_{P_1}, \boldsymbol{O'}_{P_2}, \boldsymbol{O'}_E, \boldsymbol{u'}_{P_1}, \boldsymbol{u'}_{P_2}, \boldsymbol{u'}_E\right)\big|_{\boldsymbol{u'}_j = \boldsymbol{\pi'}_j(\boldsymbol{O'}_j)}$
      Update Critic network by minimizing the loss:
$$\phi_i = \arg\min \mathscr{L}(\phi_i) = \arg\min \frac{1}{S}\sum_j \left(y_i - \boldsymbol{Q}_{\phi i}(\boldsymbol{O}_{P_1}, \boldsymbol{O}_{P_2}, \boldsymbol{O}_E, \boldsymbol{u}_{P_1}, \boldsymbol{u}_{P_2}, \boldsymbol{u}_E)\right)^2$$
      Update Actor network with sampled policy gradient:
$$\theta_i = \arg\max \nabla_{\theta_i} J = \arg\max \mathbb{E}[\nabla_{\theta i} \boldsymbol{\pi}_{\theta i}(\boldsymbol{O}_i) \nabla_{ui} \boldsymbol{Q}_{\phi i}(\boldsymbol{O}_{P_1}, \boldsymbol{O}_{P_2}, \boldsymbol{O}_E, \boldsymbol{u}_{P_1}, \boldsymbol{u}_{P_2}, \boldsymbol{u}_E)]$$
      Update target network parameters:
$$\begin{cases} \theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i \\ \phi'_i \leftarrow \tau\phi_i + (1-\tau)\phi'_i \end{cases}$$
    end for
  end for
end for

Algorithm 1. MADDPG algorithm for the pursuit-evasion game strategy in the grid world

Methodology for identifying emergent behavior

Concepts, terminology, and principles

The concept of emergence dates back to the late 19th century in the field of philosophy. British philosopher George Henry Lewes first introduced the term in his 1877 work *Problems of Life and Mind*[27], where he used it to describe phenomena within a system that cannot be fully explained by merely analyzing its individual components. Instead, such phenomena arise from the complex interactions among these components, giving rise to novel properties that are not present at the individual level but emerge at the system level.

The concept of emergent behavior was later developed to describe the collective behaviors that arise from interactions among individuals within complex systems. This idea is closely linked to the broader philosophical and scientific framework of emergence, highlighting the dynamic evolution of systems as a whole. In multi-agent pursuit-evasion games, emergent behavior exhibits distinct characteristics, often manifesting as a dynamic balance between cooperation and competition. Pursuers typically need to collaborate to successfully capture the evader; however, such cooperation may require individual agents to sacrifice their own interests for the benefit of the group. Conversely, the evader often relies on strategic planning to optimize its escape path and avoid capture. In these games, agents make decisions based on locally available information. Through the repeated execution of simple local actions governed by basic rules, spontaneous and complex global behaviors emerge. The definitions of actions and behaviors are as follows:

Definition 1: **Action**. An action refers to the fundamental operation performed by an individual at a specific moment, typically representing a direct response to the environment. In pursuit-evasion games, an action is defined as a movement from the current grid point to an adjacent grid point, encompassing both the direction and magnitude of the movement. At the j_{th} decision point t_j in a game consisting of N rounds, the individual action of an agent can be denoted as:

$$\boldsymbol{a}_i(t_j) \in \boldsymbol{A}_s(i \in \{P_1, P_2, E\}) \tag{20}$$

Definition 2: **Behavior**. Behavior refers to an organized sequence of actions that typically represents a complex, long-term, and goal-directed activity guided by specific objectives or tasks. As a higher-level response, behavior involves the coordination and integration of multiple actions to achieve broader goals. Examples include patterns such as parallel approach or serpentine movement.

Behavior as a function: A behavior b of an agent can be defined as a function that maps the history of interactions up to the current time step to a probability distribution over possible actions. Formally,

$$\boldsymbol{b} : \boldsymbol{H}_t \to \Delta(\boldsymbol{A}) \tag{21}$$

where $\Delta(\boldsymbol{A})$ denotes the set of all probability distributions over the action set $\boldsymbol{A}$, and $\boldsymbol{H}_t = \{\boldsymbol{X}_0, \boldsymbol{a}_0, \boldsymbol{X}_1, \boldsymbol{a}_1, ..., \boldsymbol{X}_{t-1}, \boldsymbol{a}_{t-1}\}$ represents the history of states and actions up to time t.

Behavior as a sequence of actions: Alternatively, particularly in scenarios with finite time horizons, behavior can be represented as a sequence of actions. In this case, a behavior b over a time horizon is a sequence of action functions:

$$\boldsymbol{b} = (\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_M) \tag{22}$$

where each a_t maps the interaction history up to time t to a specific action at time t, i.e.,

$$\boldsymbol{a}_t = \boldsymbol{b}_t(\boldsymbol{H}_t) \tag{23}$$

This sequential definition effectively captures the evolution of an agent's actions over time, as they adapt in response to past interactions.

Behavior with specific characteristics:

To incorporate specific features or constraints, subsets of behaviors can be defined based on certain properties.

(1) **Memoryless behavior**: a memoryless behavior depends solely on the current state, without consideration of the full interaction history.

$$\boldsymbol{b}(S) = \Delta(\boldsymbol{A}) \tag{24}$$

(2) **Periodic behavior**: a periodic behavior repeats a fixed sequence of actions at regular intervals, specifically every k time steps.

$$\boldsymbol{b}_t = \boldsymbol{b}_{t \bmod k} \tag{25}$$

Definition 3: **Strategy**. A strategy refers to a planned and coordinated sequence of behaviors that guides an agent's decisions over time, typically designed to achieve specific goals within the pursuit-evasion game. It is a higher-level response to the environment, wherein individual behaviors (composed of multiple actions) are combined in a systematic way to maximize the agent's performance or increase the likelihood of success. A strategy involves the selection and timing of different behaviors, taking into account both the current state of the environment and the expected reactions of the opponent. For example, a strategy might involve alternating between a "chase" behavior to close distance and a "flank" behavior to gain a positional advantage. The strategy evolves dynamically as the game progresses, based on ongoing assessments and interactions.

Classification of coordinated pursuit actions

In this section, we define single pursuit actions and coordinated pursuit actions to establish a clear theoretical framework for analyzing the complex and dynamic interactions between pursuers and the evader in the pursuit-evasion game. These definitions will serve as the foundation for the subsequent analysis of pursuit strategies and the emergent coordinated behaviors exhibited by the pursuers.

Single pursuit actions

Before examining coordinated pursuit actions, it is essential to first understand individual pursuit actions. A single pursuit action represents a direct interaction between one pursuer and the evader, and serves as the fundamental unit in a multi-to-one pursuit-evasion game. These actions encompass a variety of strategies, such as driving, chasing, intercepting, flanking, engaging, and ambushing. The action set can be defined as $\boldsymbol{A}_s = \{\boldsymbol{a}_{s_1}, \boldsymbol{a}_{s_2}, \boldsymbol{a}_{s_3}, \boldsymbol{a}_{s_4}, \boldsymbol{a}_{s_5}, \boldsymbol{a}_{s_6}\}$. Each action corresponds to specific tactical goals and conditions. For example, the driving action aims to compel the evader to alter their escape route, while the chasing action focuses on reducing the distance between the pursuer and the evader. These definitions offer a framework for pursuers to choose the most suitable strategy in response to the given circumstances.

Based on the relative position and velocity of the pursuer and evader, individual pursuit action patterns in a pursuit-evasion (P–E) game are defined as shown in Fig. 2. In this model, the evader is positioned at the origin of the coordinate system, with its velocity oriented along the y-axis. The classification of pursuit actions is determined by specific angle constraints, as detailed in Eq. 26.

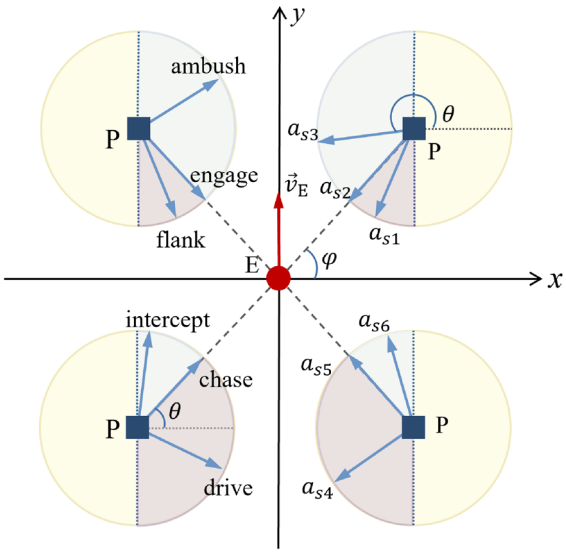

Fig. 2. Diagram of individual pursuit actions.

$$\cos\theta\cdot\cos(\varphi+\pi)\geq 0\begin{cases}0<\varphi<\pi\begin{cases}-1\leq\sin\theta<\sin(\varphi+\pi), & \text{flank}\\ \sin\theta=\sin(\varphi+\pi), & \text{engage}\\ \sin(\varphi+\pi)<\sin\theta\leq 1, & \text{ambush}\end{cases}\\ \pi\leq\varphi\leq 2\pi\begin{cases}-1\leq\sin\theta<\sin(\varphi+\pi), & \text{drive}\\ \sin\theta=\sin(\varphi+\pi), & \text{chase}\\ \sin(\varphi+\pi)<\sin\theta\leq 1, & \text{intercept}\end{cases}\end{cases}\tag{26}$$

$$\cos\theta\cdot\cos(\varphi+\pi)<0:\text{Invalid pursuit}$$

Here, φ represents the angle between the position vector of the pursuer relative to the evader and the positive direction of the x-axis, while θ represents the angle between the pursuer's velocity and the positive direction of the x-axis. Both φ and θ can be computed using Eqs. 27–30. First, the position angle of the pursuer, the velocity angle of the pursuer, and the velocity angle of the evader are calculated as follows:

$$\varphi_P=\begin{cases}\arccos\frac{(\boldsymbol{X}_P-\boldsymbol{X}_E)\cdot\boldsymbol{e}_x}{\|\boldsymbol{X}_P-\boldsymbol{X}_E\|}, & (\boldsymbol{X}_P-\boldsymbol{X}_E)\cdot\boldsymbol{e}_y\geq 0\\ 2\pi-\arccos\frac{(\boldsymbol{X}_P-\boldsymbol{X}_E)\cdot\boldsymbol{e}_x}{\|\boldsymbol{X}_P-\boldsymbol{X}_E\|}, & (\boldsymbol{X}_P-\boldsymbol{X}_E)\cdot\boldsymbol{e}_y<0\end{cases}\tag{27}$$

$$\theta_P=\begin{cases}\arccos\frac{\boldsymbol{V}_P\cdot\boldsymbol{e}_x}{\|\boldsymbol{V}_P\|}, & \boldsymbol{V}_P\cdot\boldsymbol{e}_y\geq 0\\ 2\pi-\arccos\frac{\boldsymbol{V}_P\cdot\boldsymbol{e}_x}{\|\boldsymbol{V}_P\|}, & \boldsymbol{V}_P\cdot\boldsymbol{e}_y<0\end{cases}\tag{28}$$

$$\theta_E=\begin{cases}\arccos\frac{\boldsymbol{V}_E\cdot\boldsymbol{e}_x}{\|\boldsymbol{V}_E\|}, & \boldsymbol{V}_E\cdot\boldsymbol{e}_y\geq 0\\ 2\pi-\arccos\frac{\boldsymbol{V}_E\cdot\boldsymbol{e}_x}{\|\boldsymbol{V}_E\|}, & \boldsymbol{V}_E\cdot\boldsymbol{e}_y<0\end{cases}\tag{29}$$

$$\begin{cases}\varphi=\varphi_P-\left(\theta_E-\frac{\pi}{2}\right)\\ \theta=\theta_P-\left(\theta_E-\frac{\pi}{2}\right)\end{cases}\tag{30}$$

Figure 3 illustrates the specific schematics for each type of pursuit maneuver. When the pursuer is positioned behind the evader, three types of actions can be identified: Drive, Chase, and Intercept. Conversely, when the pursuer is positioned ahead of the evader, three types of actions can be identified: Flank, Engage, and Ambush, as shown in Table 1. The definitions and constraints for each pursuit action are as follows:

Flank($\boldsymbol{a}_{s_1}$): Move to the side and behind the target to launch an attack. The pursuer approaches the evader from the front of the evader's velocity direction, with the pursuer's velocity vector directed towards the rear of the evader.

Engage($\boldsymbol{a}_{s_2}$): Move towards the target to initiate an attack. The pursuer approaches the evader from the front of the evader's velocity direction, with the pursuer's velocity vector consistently pointing towards the evader.

Ambush($\boldsymbol{a}_{s_3}$): Lie in wait along a potential route the evader might take, launching a sudden attack when the evader approaches. The pursuer approaches the evader from the front of the evader's velocity direction, with the pursuer's velocity vector directed towards the evader's front.

Drive($\boldsymbol{a}_{s_4}$): Force the target to retreat or move out of a specific area. The pursuer approaches the evader from behind the evader's velocity direction, with the pursuer's velocity vector directed towards the rear of the evader.

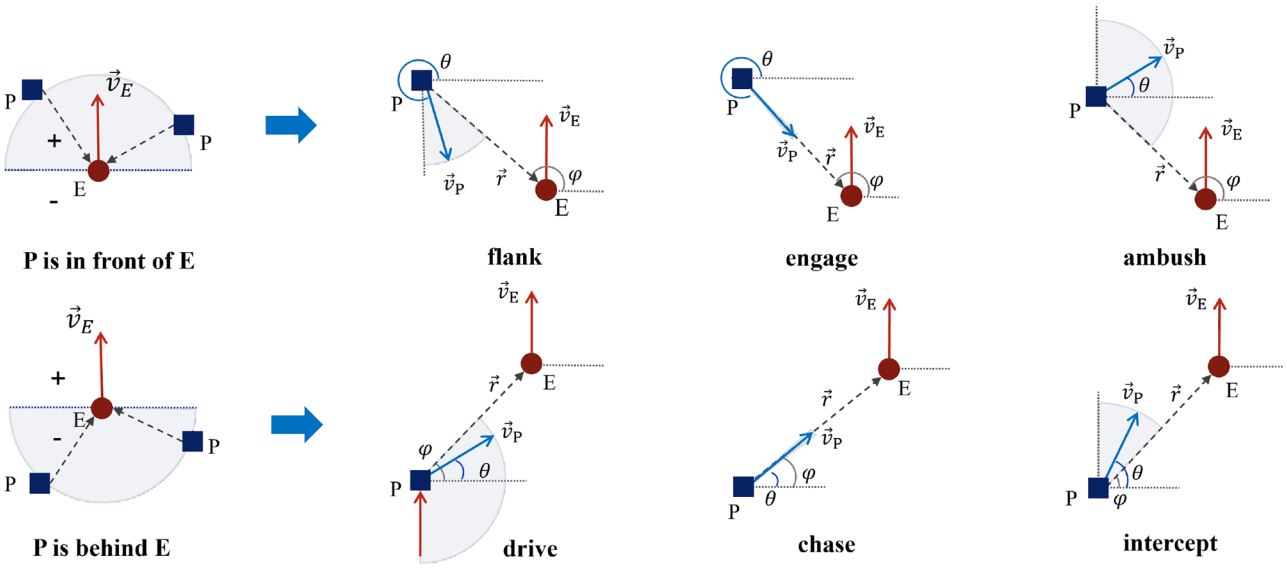

Fig. 3. The individual pursuit actions.

Number	Action	Sight direction	Velocity direction
a_{s1}	Flank	front	behind
a_{s2}	Engage	front	point to
a_{s3}	Ambush	front	front
a_{s4}	Drive	behind	behind
a_{s5}	Chase	behind	point to
a_{s6}	Intercept	front	front

Table 1. Set of individual pursuit actions.

Chase(a_{s_5}): Follow and close in on the target quickly. The pursuer approaches the evader from behind the evader's velocity direction, with the pursuer's velocity vector consistently pointing towards the evader within a certain tolerance range.

Intercept(a_{s_6}): Aim to intercept or block the target from reaching a predetermined location along its path. The pursuer approaches the evader from behind the evader's velocity direction, with the pursuer's velocity vector directed towards the front of the evader.

Coordinated pursuit actions of two pursuers

In a two-on-one coordinated pursuit, effective collaboration between the pursuers is critical. For instance, one pursuer may use a driving action to push the evader into the interception path of the other pursuer. Alternatively, both pursuers could execute a pincer movement, approaching the evader from two directions simultaneously to limit its escape routes. Such coordination requires not only clear communication and mutual understanding between the pursuers, but also the ability to anticipate the evader's actions and quickly adapt strategies.

By combining individual pursuit actions, we can define various coordinated pursuit strategies, including chase, evict, entrap, hunt, block, outflank, flank, engage, snipe, pincer attack, intercept, interfere, confront, surround, drive, encircle, contain, besiege, expel, ambush, and envelop–a total of 21 distinct actions. These strategies enhance both tactical diversity and pursuit efficiency, improving success rates.

As shown in Fig. 4, the roles of the two pursuers are interchangeable, meaning swapping their roles does not affect the coordinated maneuver. Additionally, situations where the two pursuers are positioned either on the same side or opposite sides of the evader are considered the same action, therefore $\boldsymbol{A}_c = \{\boldsymbol{a}_{c_1}, \boldsymbol{a}_{c_2}, \cdots, \boldsymbol{a}_{c_{21}}\}$. These action definitions lay the groundwork for studying game behaviors in the following section.

Identification of cooperative behaviors

Analyzing individual pursuit actions can provide insights into the game state at specific moments, such as evaluating the level of coordination in different cooperative pursuit actions or counting the frequency of each action throughout the game. However, focusing solely on individual actions offers a limited perspective, failing to capture their impact on the game's evolution and outcome. This approach overlooks the dynamic progression of strategies and the development of overall cooperative behavior. To address this limitation, attention should shift from isolated moments to the entire game process. This involves examining the trajectory evolution of both pursuers and evaders, a concept we refer to as "behavior". By using the full set of game trajectories as statistical

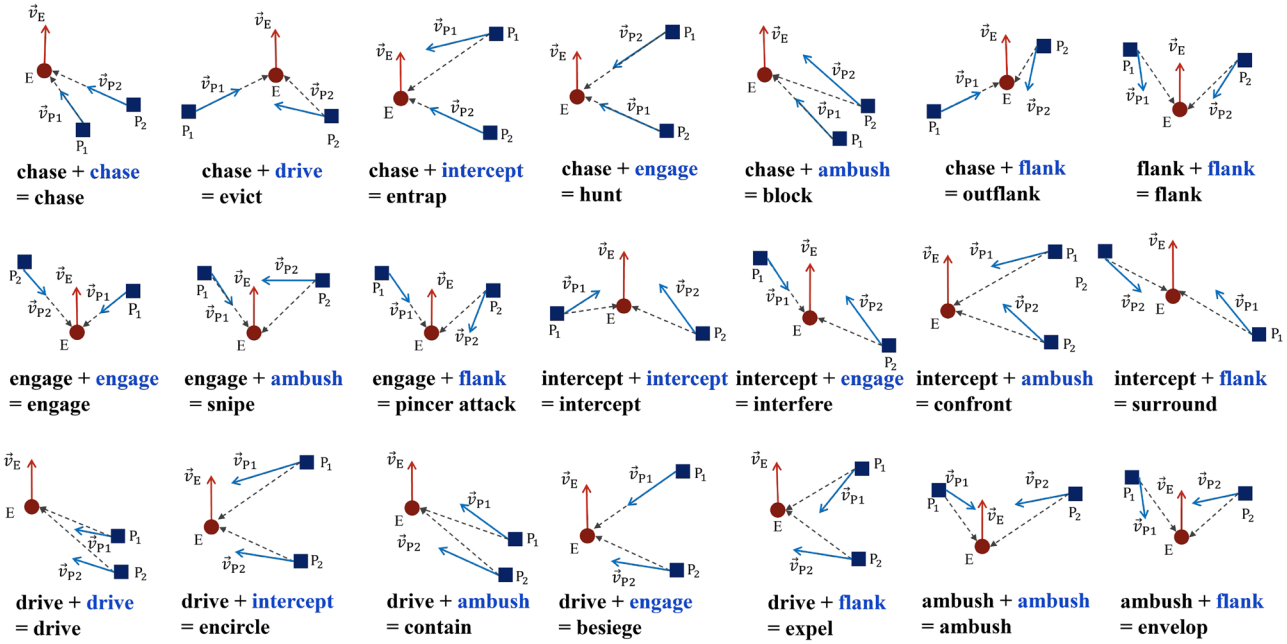

Fig. 4. The cooperative pursuit actions.

samples, methods like cluster analysis can be applied to identify distinct behavioral patterns and cooperative strategies.

The K-means algorithm is commonly used for tasks such as image segmentation, classification, and color quantization. Clustering analysis partitions data points into distinct groups, where points within the same cluster exhibit high similarity, while those in different clusters show significant dissimilarity. Since each behavior consists of a sequence of actions, we treat the action sequences across multiple pursuit-evasion trajectories as sample data points $\boldsymbol{x}_i$ for clustering. These samples, obtained through training, include the action sequences of pursuers P_1, P_2 and E: $\boldsymbol{x}_i = [\boldsymbol{a}_{P_1}, \boldsymbol{a}_{P_2}, \boldsymbol{a}_E]$. The K-means algorithm iteratively optimizes cluster assignments by allocating each data point to the nearest cluster. The centroid of each cluster is computed as the mean of all points within the cluster, representing a typical action sequence. This process is repeated until convergence. In pursuit-evasion games, this approach facilitates the identification of distinct cooperative pursuit behaviors based on representative action sequences.

The main steps of the K-means-based behaviors recognition algorithm are as follows:

Step1: Determine the number of behavior clusters. Before starting the clustering process, the number of behavior clusters K must be predefined. This can be done using the elbow method. By plotting the Sum of Squared Errors (SSE) against different values of K, the optimal K is identified at the "elbow" point, where the rate of decrease in SSE sharply slows. Given K clusters, $\boldsymbol{C}_j$ represents the j_{th} cluster, with the centroid $\boldsymbol{\mu}_j$ and action sequences $\boldsymbol{x}_i$ within the cluster. The SSE is calculated as follows:

$$SSE = \sum_{j=1}^{k} \sum_{\boldsymbol{x}_i \in \boldsymbol{C}_j} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\|_2 \tag{31}$$

As K increases, SSE typically decreases because more clusters lead to tighter grouping. However, beyond a certain point, the reduction in SSE becomes marginal, forming the "elbow" point, which is considered the optimal number of behavior clusters.

Step2: Initialization of centroids. Randomly select K data points from the action sequences of multiple pursuit-evasion scenarios as the initial cluster centroids to optimize the clustering outcome.

Step3: Assigning data points to the nearest cluster centroid. For each action sequence data point, calculate its distance to each centroid and assign it to the closest one, forming K clusters. This step groups similar action sequences together based on their proximity to the cluster centroids.

$$\boldsymbol{C}_k = \{\boldsymbol{x}_i : \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|_2 \leq \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|_2, \ \forall j = 1, 2, \ldots, K\} \tag{32}$$

Step4: Update cluster centers. For each cluster, calculate the mean of all data points within it and update the cluster center to this mean. The updated cluster centers are closer to a behavior composed of a specific action sequence.

$$\boldsymbol{\mu}_k = \frac{1}{|C_k|} \sum_{\boldsymbol{x}_i \in \boldsymbol{C}_k} \boldsymbol{x}_i \tag{33}$$

Step5: Iterate until convergence. Continue alternating between steps 3 and 4 until a termination condition is met. This condition is based on reaching a maximum number of iterations or minimal changes in cluster centers. This iterative process ensures that the clusters stabilize and accurately represent distinct types of action sequences.

Step6: Refine clusters by removing outliers. Remove the peripheral data points that are too far from the center to ensure the accuracy of classification. Due to the complexity of the trajectory patterns, removing data points outside the boundaries helps ensure clear differentiation between the clustered behavior patterns

Finally, the results are visualized using the t-SNE algorithm. This algorithm reduces the dimensionality of the data while preserving the similarity between data points, allowing us to distinguish different behavior categories in a 2D plot. The centroid of the clustered data represents an action sequence, which may encompass multiple behaviors. Data points within each cluster exhibit similar action sequences and behavioral characteristics.

Emergent behavior analysis process

The process for analyzing emergent behavior is summarized in Fig. 5. First, using a reinforcement learning algorithm, pursuit and evasion strategy networks are trained in a free exploration mode, guided only by time. These networks are then tested to generate numerous two-on-one pursuit-evasion game trajectories. These trajectories serve as the data foundation for clustering analysis, capturing the state of both pursuers and evaders at each decision point.

To ensure sufficient behavioral diversity for analysis, the underlying networks must be robustly trained. In our implementation, the model typically converges with approximately 100,000 training episodes. The network architecture includes an Actor Network and a Critic Network, each containing two hidden layers with 128 neurons. The hyperparameters of the network during training are shown in Table 2.

Next, cooperative pursuit actions are identified based on the relative position and velocity of the pursuers and evader, classifying actions such as “chasing”, “ambushing”, “intercepting”, “surrounding”, and “driving”. The occurrence frequency of each action is calculated across all trajectories to evaluate which actions lead to better outcomes.

Finally, single-step cooperative pursuit actions are combined into action sequences and subjected to clustering analysis. This identifies distinct cooperative pursuit strategies and behavioral patterns. Based on these results, categories of cooperative pursuit behavior are defined, named, and evaluated in terms of their effectiveness in specific contexts.

Emergent behaviors and characteristics

In this section, we employ the intelligent pursuit-evasion game strategy for training, then statistically analyze the cooperative pursuit actions and perform clustering algorithm to explore the emergent cooperative behaviors

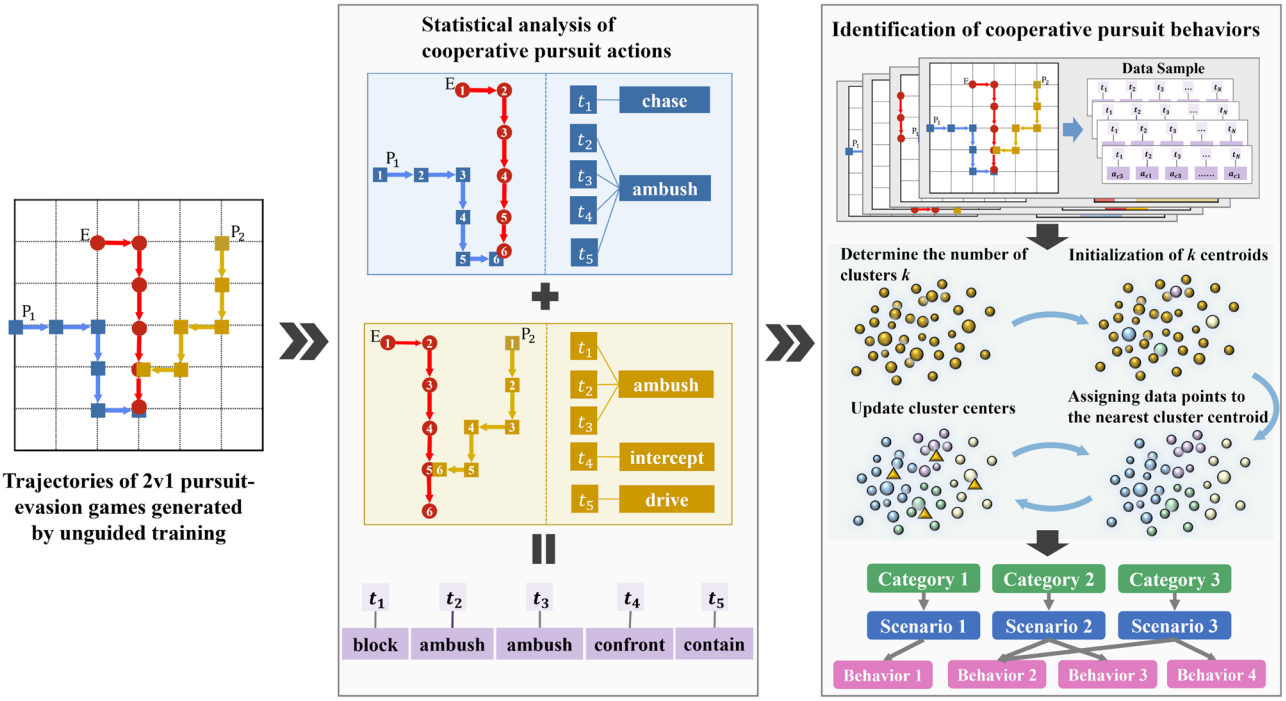

Fig. 5. Emergence behavior analysis method for pursuit-evasion games.

Name	Meaning	Value
γ	Reward discount factor	0.95
τ	Target network update rate	0.01
$Max\text{-}Episode$	Maximum number of training episodes	2×10^5
$Max\text{-}len$	Maximum decision steps per episode	50
Δt	Time interval between decisions	1s
α	Distance guidance reward coefficient	0.001
α_Q	Learning rate of critic network	0.001
α_μ	Learning rate of actor network	0.001

Table 2. Parameters of the MADDPG training system in Algorithm 1.

a_{P_2} \ a_{P_1}	drive	chase	intercept	ambush	engage	flank
drive	1963	347	3249	713	2	330
chase	289	342	504	222	6	89
intercept	2546	500	7894	591	0	217
ambush	678	153	570	460	0	180
engage	12	1	16	1	1	5
flank	264	60	227	208	0	174

Fig. 6. Statistical table of cooperative pursuit actions.

during the pursuit-evasion game. To better reflect the cooperative effect, the pursuers and the evader are assigned equal maneuvering capabilities, with the control magnitude $u_P^{\max} = u_E^{\max} = 1$. In each training session, the total number of game rounds N is set to 50. The parameters in the reward functions are denoted as $\rho = 1$ and $D_L^i = 100$. The initial position of the evader is randomly chosen within a 2×2 grid at the center of the larger grid, while the pursuers are randomly distributed across the entire grid scenario.

Specifically, we highlight that emergence in our setting occurs at two levels:

(1) **Action-level emergence**: This refers to the frequent recurrence of specific individual actions in the agents' behavior. These are statistically identified from large volumes of simulation data, reflecting regularities that arise without explicit programming or supervision.
(2) **Strategy-level emergence**: Beyond isolated actions, we observe the spontaneous formation of coordinated behavioral sequences–higher-level strategies–that emerge from repeated interactions. These are identified through unsupervised clustering of action trajectories. Importantly, these strategies were not pre-defined or hard-coded but developed dynamically through local decisions in the game environment.

Commonly used cooperative pursuit actions

After training with the MADDPG algorithm, we conduct 1000 test simulations and analyze the outcomes of the pursuit-evasion games. Among these simulations, the pursuers achieved a success rate of 99.9%. Of the successful pursuits, 675 resulted in the evader being intercepted near the boundary, while 100 instances occurred in the corners formed by two boundaries. This suggests that the pursuers intentionally drive the evader towards the boundaries and corners, effectively trapping them against the walls. The actions of the two pursuers at each moment during successful pursuits are organized and presented in tabular form, as shown in Fig. 6.

The rows and columns of the statistical table represent the individual pursuit actions a_{P_1} and a_{P_2} taken by the two pursuers, respectively. The numerical values in the table reflect how often the two pursuers jointly performed the actions represented by the respective row and column, i.e., the frequency of specific cooperative pursuit actions. Since the two pursuers are indistinguishable in terms of order, the table is symmetric about the diagonal running from the top left to the bottom right. The analysis reveals that, in successful pursuit scenarios, the most frequently used individual actions by the pursuers are "driving" and "intercepting". Moreover, the cooperative actions most frequently observed include "cooperative herding", "cooperative intercepting", and "cooperative blocking", highlighting their superior collaborative effectiveness. Additionally, scenarios where one pursuer executes an ambush while the other performs driving or intercepting are also relatively common, the rear pursuer focuses on herding, while the front pursuer sets up an ambush.

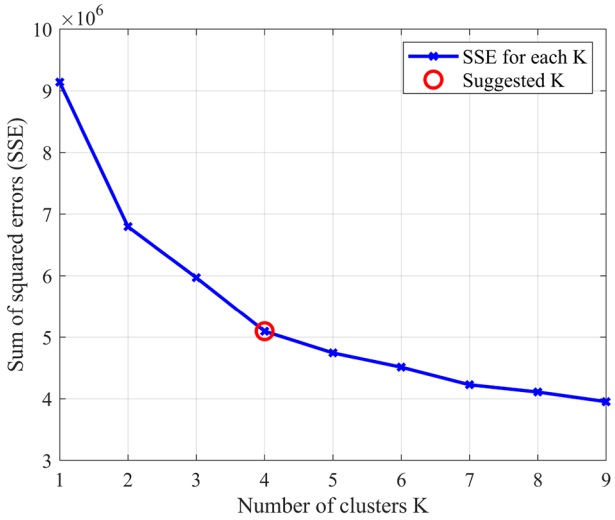

Fig. 7. Optimal *K* selection.

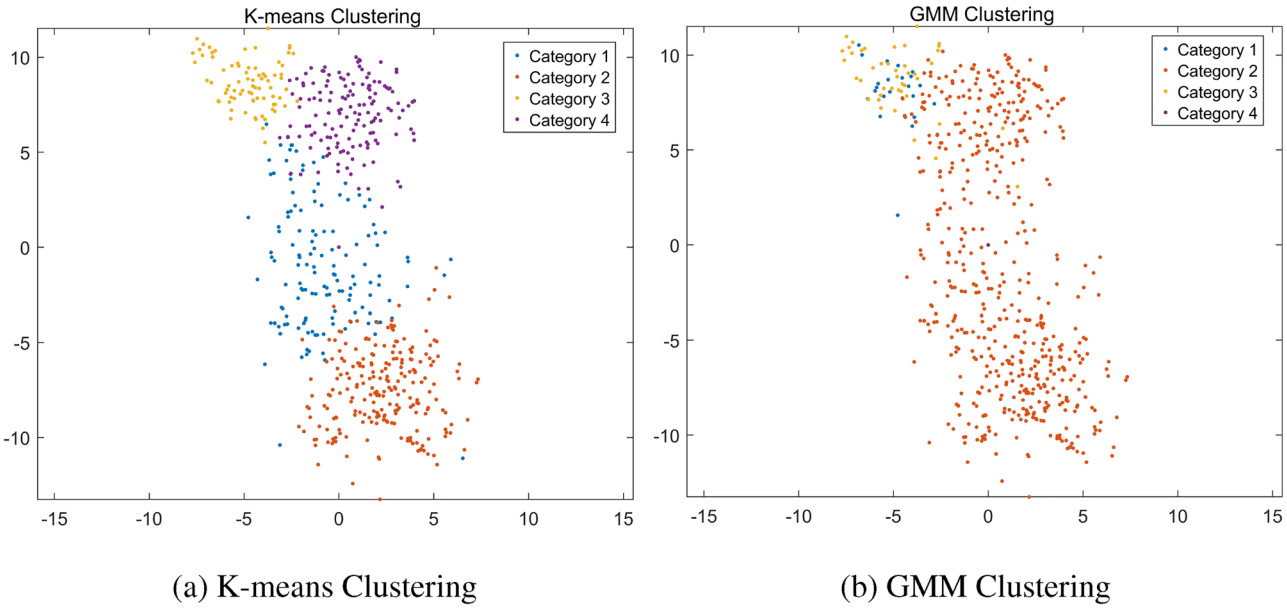

Fig. 8. Visualization of the category-wise distribution of clustering results.

Classified cooperative pursuit strategies

Following the emergent behavior research methodology outlined in the former section, simulations were performed to cluster and visualize cooperative pursuit actions. To determine the optimal number of clusters, we used the elbow method, plotting the Sum of Squared Errors (SSE) against varying values of *K*. As shown in Fig. 7, the SSE decrease slows significantly when the number of clusters *K* reaches 4, forming a clear "elbow" point. Therefore, the number of clusters is determined to be 4.

Once the number of clusters was identified, clustering and visualization analysis of the pursuit behaviors was conducted. As illustrated in Fig. 8, data points of the same color represent a single emergent behavior category, while the degree of separation between the points indicates the differences between behavior types. Points of the same color are grouped closely together, and the boundaries between categories are relatively distinct, suggesting a high-quality clustering result.

We have conducted a comparative experiment between K-means and Gaussian Mixture Model (GMM) clustering methods, which serve as classical and widely used unsupervised learning baselines. The results indicate that K-means clustering achieves more stable and interpretable clusters compared to GMM. K-means shows better cluster compactness and separation, which aligns well with the underlying spatial grouping of agents or features in our task. GMM, while more flexible due to its probabilistic nature, tends to produce overlapping clusters or blurred boundaries in our scenario, possibly due to the high dimensionality or sparsity in the data.

From a task-adaptive perspective, K-means is more effective in segmenting agent behaviors or states relevant to downstream decision-making. By plotting the game trajectories corresponding to the data points of each color, four distinct strategies with similar behaviors were identified, as shown in Fig. 9.

Next, we will select representative cases from each strategy and provide a detailed explanation of the corresponding behaviors and their constituent action sequences.

Strategy 1: serpentine corner encirclement

In this strategy, the pursuers employ serpentine movements to drive the evader into a corner. Their trajectories exhibit a high degree of similarity to the evader's path, reflecting significant overlap. This indicates that the pursuers primarily achieve encirclement by driving the evader from behind. As shown in Figure 10, the strategy is divided into three distinct phases:

Phase 1: The two pursuers move in parallel to position themselves behind the evader, gradually closing in from both sides.

Phase 2: The pursuers intensify their pressure through serpentine maneuvers, driving the evader into a corner.

Phase 3: Pursuer P_2 focuses on restricting the evader's lateral movement but does not fully engage in the chase. This behavior may manifest as the pursuer moving away from the evader or remaining stationary.

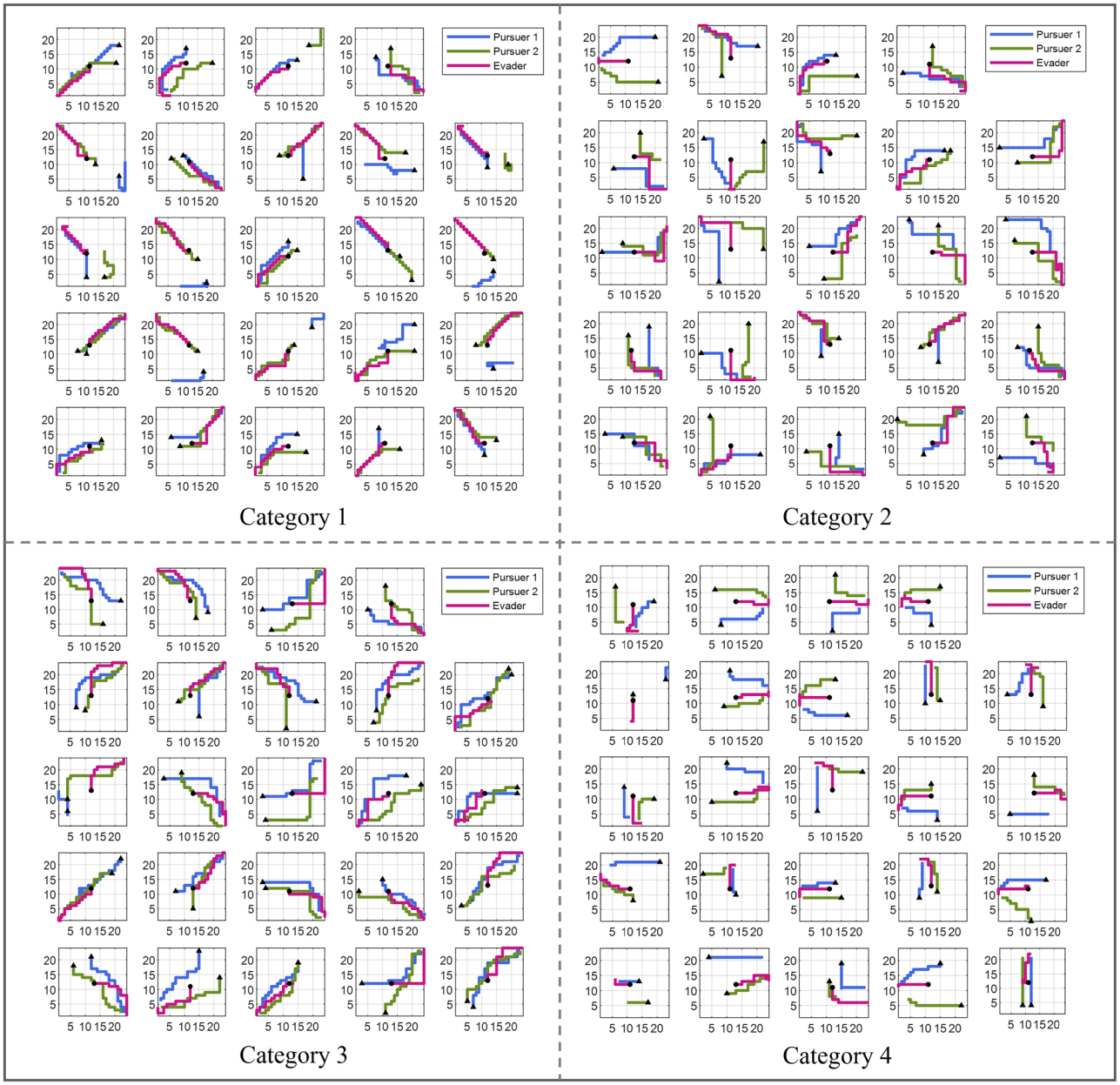

Fig. 9. Cooperative pursuit strategies classified by K-means-based behaviors recoginition algorithm (partial display).

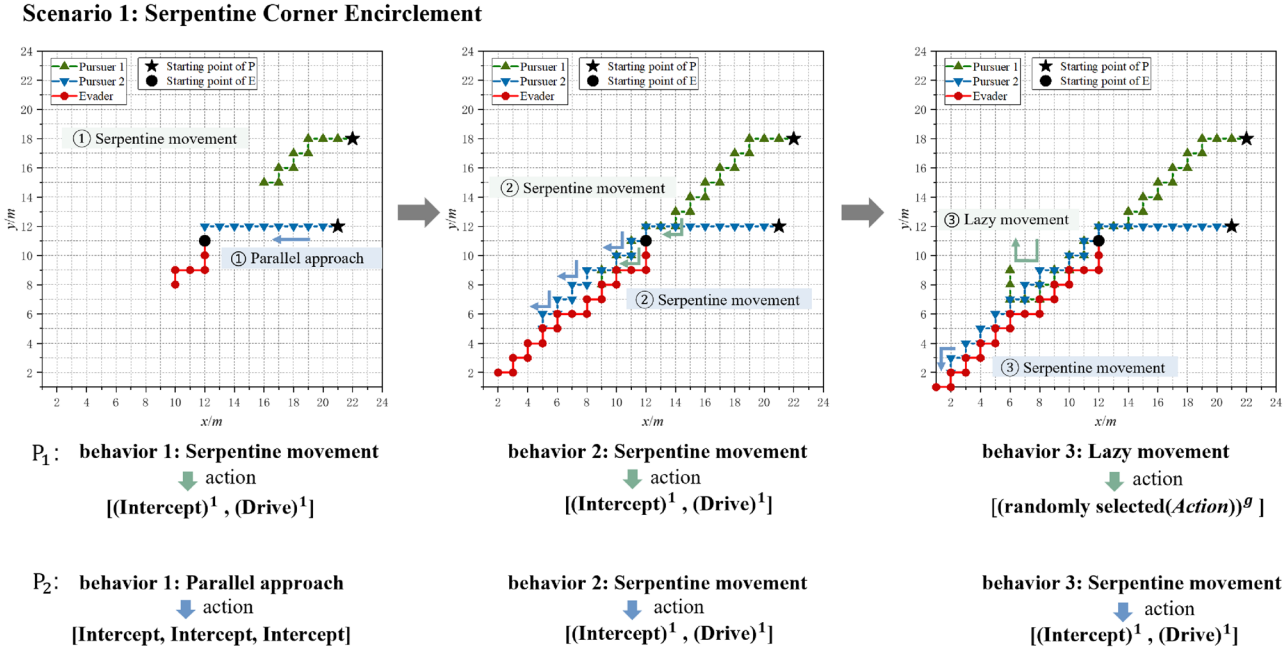

Fig. 10. Behaviors in strategy 1. The trajectories in the figure are presented in chronological order. The serpentine movement behavior consists of alternating drive and intercept actions, with the superscript "1" outside the action parentheses indicating that each identical action does not occur consecutively. The parallel approach behavior is formed by repeated consecutive intercept actions, which can create the effect of pursuing in the direction of the pursuer's velocity. The lazy movement behavior is composed of g irregular random actions, represented by randomly selected actions at each decision point.

A notable phenomenon in this strategy is the presence of "lazy pursuers", particularly when the pursuers and the evader start in the same or adjacent quadrants, with a significant disparity between one pursuer and the evader. These conditions increase the likelihood of lazy pursuit behavior.

Strategy 2: stepwise corner approach

As illustrated in Fig. 11, the pursuers employ a phased, cooperative stepwise movement to gradually approach and encircle the evader. In Phase 1, the two pursuers reduce the distance to the evader by using linear forward movements and staggered stepwise advances from behind. In Phase 2, one pursuer moves to the front of the evader to restrict its escape direction, while the other applies lateral pressure to confine its maneuvering space. Through sparse, stepwise movements, the pursuers progressively corner the evader, further limiting its mobility.

This strategy typically begins with the two pursuers positioned either adjacent to or opposite each other, and it ultimately culminates in an encirclement that drives the evader toward a boundary or into a confined area.

Strategy 3: same-side boundary suppression

Figure 12 illustrates a strategy in which the two pursuers collaborate from the same side of the evader, applying coordinated pressure that drives the evader along the boundary until capture. One pursuer consistently intercepts from behind, closely following the evader's trajectory, while the other strategically positions itself ahead to block potential escape routes, further restricting the evader's mobility.

This strategy typically begins with the pursuers and the evader positioned on the same side, often in adjacent or the same quadrants, separated by a considerable distance. The pursuers first drive the evader to the boundary, then force it to move along the boundary toward a corner, where it is eventually captured.

Strategy 4: two-sided pincer movement

The pursuers employ a phased collaborative movement strategy to gradually encircle the evader from both sides. In the first phase, the two pursuers maintain parallel linear movement toward the evader, with one pursuer driving from the flank to compress the evader's mobility, while the other intercepts along its escape trajectory. In the second phase, the pursuers adopt a sparse, staggered movement pattern, advancing from both sides to further constrain the evader's range of motion, as shown in Fig. 13. This behavior typically begins with the pursuers positioned in adjacent quadrants on the same side, ultimately forcing the evader into either a boundary or a confined interior region.

Table 3 provides a comprehensive summary of each behavior observed in the trajectory and its corresponding action composition. Table 4 summarizes the outcomes and tendencies of lethargic behavior across different strategies under varying initial conditions. Strategy 1 and strategy 3 excel in corner captures, with strategy 3 achieving the highest success rate in corner captures under non-lethargic conditions, while also exhibiting the lowest frequency of lethargic behavior, demonstrating a high level of proactivity and efficiency. In contrast, strategy 2 maintains a balanced performance between corner and boundary captures, with a moderate frequency of lethargic behavior. strategy 4, however, focuses exclusively on boundary captures, achieving the highest capture efficiency under non-lethargic conditions but failing to achieve corner captures.

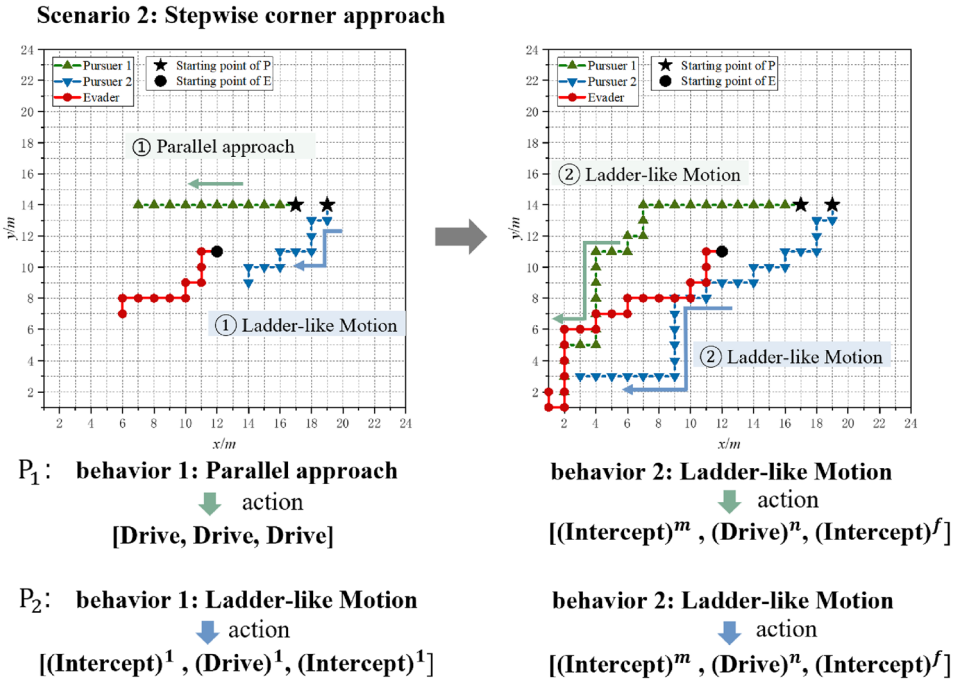

Fig. 11. Behaviors in strategy 2. The ladder-like motion behavior of P_2 comes in various forms. The superscript 1 in the action sequence below the left figure indicates that each action is used once. The letters m, n, and f below the right figure represent the respective repetition counts for the intercept, drive, and intercept actions. They are not fixed values and vary depending on the application context.

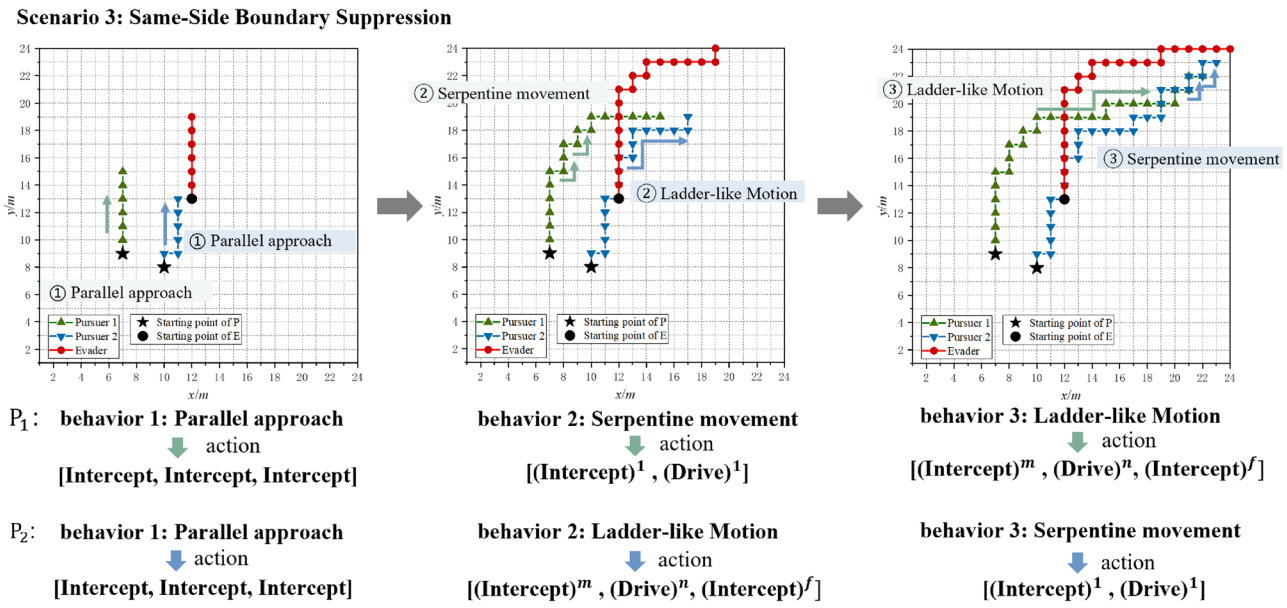

Fig. 12. Behaviors in strategy 3.

Overall, the strategies exhibit significant differences in capture modes and lethargic behavior rates. The effectiveness of each strategy depends heavily on the target strategy and the behavioral characteristics of the pursuers. These findings align well with the trajectory patterns presented in the previous sections.

Characteristics of emergent behaviors

The lazy behavior of the pursuer

Lazy pursuers may either move in a specific direction or linger near a particular location, often exhibiting behavior akin to random wandering, as shown in Fig. 15a. In this study, laziness is defined as follows: if a pursuer

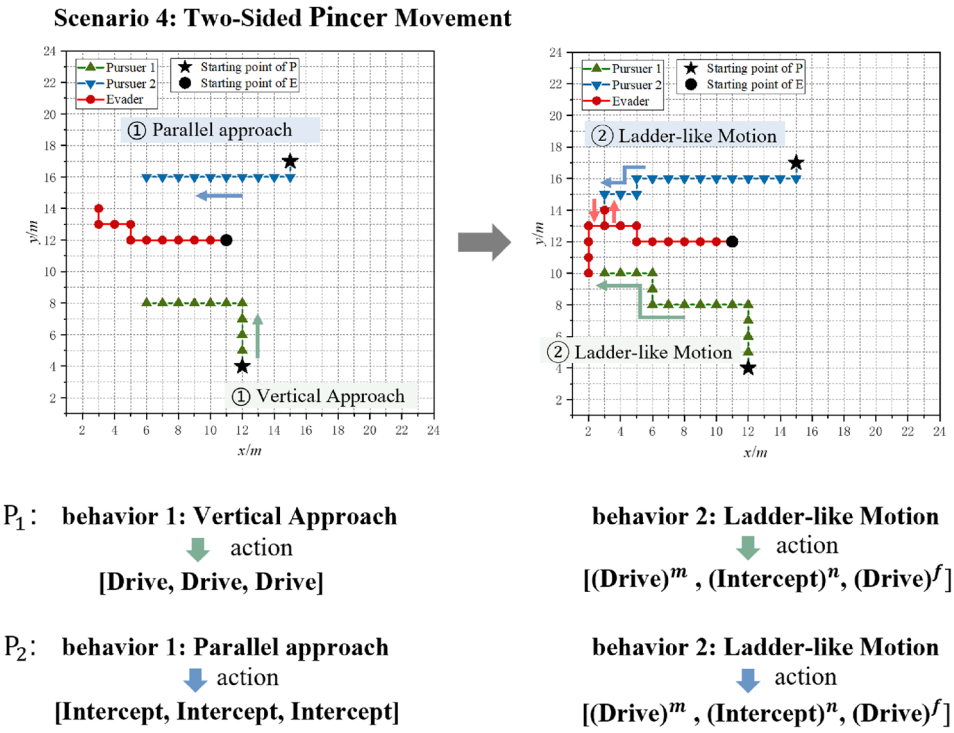

Fig. 13. Behaviors in strategy 4.

Behavior	Action composition
Parallel approach	[Intercept,Intercept,Intercept]
Vertical approach	[Drive,Drive,Drive]
Serpentine movement	$[(\text{Intercept})^1,(\text{Drive})^1]$
Ladder-like motion	$[(\text{Intercept})^m,(\text{Drive})^n],(\text{Intercept})^f$
Lazy movement	$[(\text{Randomly selected(Action)})^g)]$

Table 3. Set of pursuit behaviors.

Characteristics	Strategy 1	Strategy 2	Strategy 3	Strategy 4
Corner capture without laziness	27	57	71	0
Corner capture with laziness	65	13	7	0
Boundary capture without laziness	9	27	7	156
Boundary capture with laziness	12	1	7	4
Cases of the lazy individual	82	15	8	20
Initial position distribution of pursuers	Same,adjacent	Adjacent, opposite	Same, adjacent	Same

Table 4. The final outcomes and situational characteristics of each strategy.

repeatedly moves away from the evader or remains inactive more than five times during the game, it is classified as exhibiting lazy behavior.

Since the reward function in our setup is shared between both pursuers, where success by either is considered a victory, the emergence of lazy behavior is not only plausible but also rational. In collaborative settings, individuals may reduce their efforts when they perceive that their contributions are neither individually recognized nor rewarded. This phenomenon is known as the social loafing effect.

The presence of lazy pursuers does not necessarily compromise the overall success of the pursuit. In the context of cooperative behavior in game theory, each individual may incur certain costs for the benefit of the group, meaning that not all pursuers need to exert maximum effort for the team strategy to remain effective. Indeed, lazy pursuers can play a vital role in team coordination. While they may not actively close the distance

to the evader, their positioning can complement their counterpart's efforts by forming a pincer-like maneuver, effectively restricting the evader's escape routes. This behavior aligns closely with the findings reported in[15].

To further investigate whether this lazy behavior stems from cooperative strategies within the group, we modified the original reward function by replacing the shared cooperative reward with individually computed rewards for each pursuer. This adjustment allows us to isolate the impact of reward structure on agent behavior and examine whether individual incentives can mitigate social loafing:

$$R_D^{P_j}(t) === \begin{cases} +D_L, & (r_j(t) \le r_s) \cap (t = t_f) \\ -D_L, & (r_j(t) > r_s) \cap (t = t_f) \\ 0, & t < t_f \end{cases} \tag{34}$$

Under this modification, each pursuer focuses solely on its own interest rather than the collective benefit. The experiments showed that the lazy behavior disappeared, and both pursuers actively engaged in chasing the evader. To illustrate this more clearly, two figures in Fig. 14 present results under identical initial positions: Fig. 14a shows the outcome trained with a cooperative reward, while Fig. 14b shows the outcome with individual rewards. As can be seen, lazy behavior does not emerge in the right figure. With individual rewards, each pursuer is rewarded based on its own performance, encouraging both to actively participate. Without shared rewards, lazy behavior naturally disappears.

Boundary or corner-based encirclement behavior

In Fig. 15b, the pursuers drive the evader toward the boundary until capture. Once the evader reaches the boundary, it typically moves back and forth along it, unable to escape. If the evader attempts to flee toward the center, it accelerates its capture by the pursuers. This behavior mirrors hunting strategies observed in nature, such as those employed by dolphins. Dolphins use sonar and physical movements to cooperate during hunts, often forming a circle to drive schools of fish toward the surface, where they capture them sequentially. Additionally, dolphins have been known to use tools, such as sand or seaweed from the ocean floor, to trap their prey, enhancing their capture success rate.

The evader's sense of helplessness

When the evader is cornered at the boundary and faces no escape route, they often enter a state of confusion and helplessness, as shown in Fig. 15c. In this situation, the evader's movements become erratic as they attempt to wander aimlessly within the confined space, trying to mislead the pursuers with irregular motions to delay capture. However, due to the spatial constraints and the pursuers' approach, the evader becomes trapped in a chaotic and ineffective cycle. As time passes, the pursuers gradually close in, and the evader's options dwindle, eventually leading to capture.

Multi-scale of behaviors

The action sequences observed in pursuit-evasion strategys exhibit multi-scale characteristics, where actions of varying durations and complexities contribute to different behavioral outcomes. Shorter action sequences, typically involving rapid, immediate responses, are associated with behaviors that focus on quick adjustments to situational changes. For instance, serpentine movements rely on the gradual application of pressure over time, with actions alternating between phases to progressively reduce the evader's maneuvering space.

In contrast, longer action sequences often involve more strategic and coordinated efforts that unfold over an extended period. These actions indicate that the pursuer is capable of prediction and planning. Behaviors such as parallel approaching or ladder-like motion involve quick adaptation to the evader's movements, maintaining a

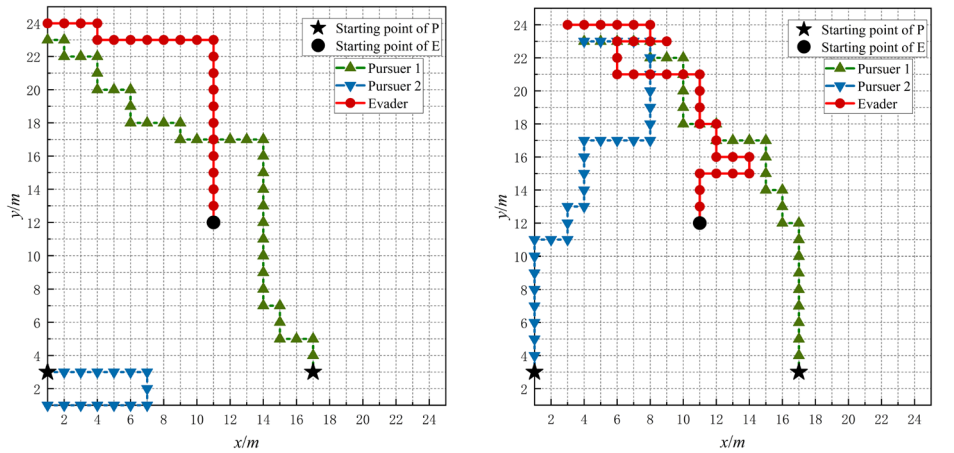

Fig. 14. The results obtained with and without cooperative rewards. (**a**) Training with cooperative reward. (**b**) Training with individual reward.

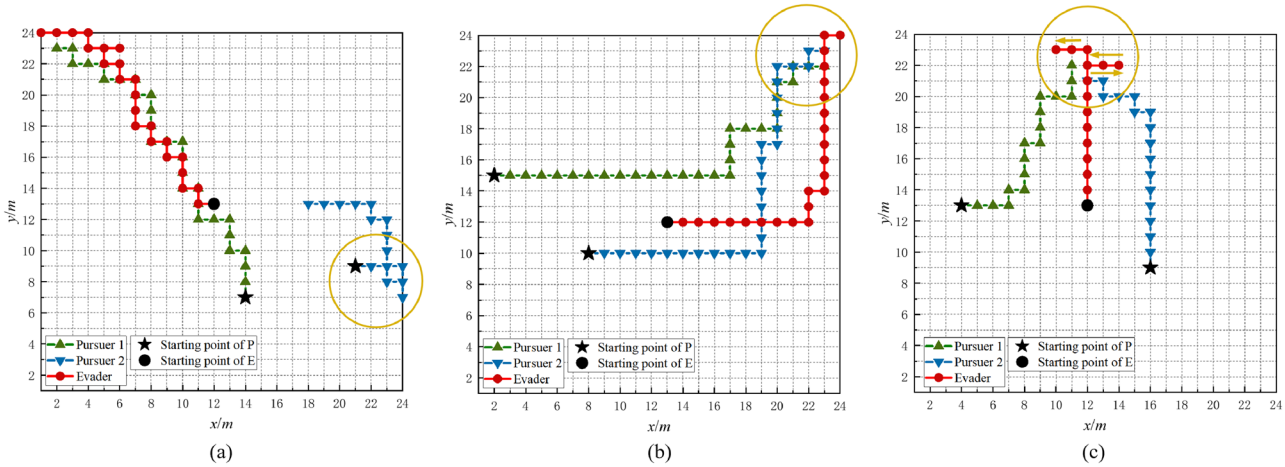

Fig. 15. Characteristics of the emergent behaviors. (**a**) pursuer P_2 exhibited lazy behavior and did not exert maximum effort in pursuit. (**b**) The two pursuers jointly cornered the evader, restricting its movement. (**c**) Upon reaching the boundary, the evader found no viable escape route and thus remained in place, wandering within a confined area.

continuous, adaptive strategy. These patterns reflect a long-term, tactical approach, with the focus on maintaining close proximity or obstructing escape routes.

The multi-scale nature of these action sequences is essential for understanding the dynamics of emergent behaviors. The combination of short, reactive actions and longer, more deliberate sequences results in complex, emergent patterns. While the examples we provided are finite, the range of behavioral patterns composed of action sequences at different scales is diverse. By considering both the immediate and prolonged effects of various action sequences, we gain a better understanding of how behavior evolves in pursuit-evasion strategys.

Scenarios with variable grid sizes, boundaries, and obstacles

While the grid-based setting imposes constraints on agent motion—particularly by limiting diagonal movements and inducing step-like trajectories—it also offers significant advantages. In particular, the discrete representation facilitates intuitive interpretation of multi-agent coordination behaviors, such as flanking, interception, and encirclement strategies. Although the resultant trajectories appear staircase-like due to discretization, these artifacts do not obscure the qualitative patterns of pursuit and evasion observed across different scenarios.

To systematically examine the influence of boundary conditions and environment size on pursuit-evasion dynamics, we analyze four representative configurations, as shown in Fig. 16. In the first scenario (Fig. 16a), where the environment extends infinitely in all directions and both pursuers and evader possess equal mobility, successful capture becomes challenging if the pursuers' initial positioning is suboptimal. Nonetheless, the evader's trajectory reveals a consistent trend of being gradually driven toward the periphery, suggesting implicit coordination among the pursuers. In contrast, Fig. 16b demonstrates that when the pursuers' maximum speed is increased, coordinated strategies alone become sufficient to achieve capture, even in the absence of environmental boundaries.

In Fig. 16c,d, we consider scenarios with no fixed boundaries but a finite number of maneuvering steps. Under these conditions, although the environment is conceptually unbounded, the trajectories remain constrained by the allowed movement duration. When mobility is symmetric (Fig. 16c), the evader can evade capture more easily, reflecting the limitations faced by pursuers without either boundary support or speed advantage. However, as shown in Fig. 16d, when the pursuers' mobility is doubled relative to the evader, successful capture is achieved through coordinated motion, even without relying on boundary effects.

These comparisons collectively highlight two key insights: (1) Boundaries can significantly assist the pursuers by constraining the evader's escape space. (2) Coordination among agents remains critical for successful capture, regardless of whether boundaries are present.

The grid-based setting imposes constraints on agent motion–most notably by preventing diagonal movements and resulting in step-like trajectories. However, despite these discretization-induced artifacts, the qualitative patterns of pursuit and evasion remain clearly observable. In particular, coordinated behaviors such as flanking, interception, and encirclement continue to emerge robustly across different scenarios.

Furthermore, to assess the adaptability of the learned strategies under more challenging conditions, we conducted further experiments in environments containing obstacles. Specifically, two rectangular obstacles of sizes 3×3 and 3×4 were placed in the upper-right corner and along the left side of the grid, respectively, both rendered in gray. In this setting, the agents were required to navigate around the obstacles to complete the pursuit task.

Despite the increased environmental complexity, the trained pursuers continued to perform effectively, achieving a high success rate of approximately 96.8%. This result demonstrates that the learned policies are robust and generalize well to environments with moderate structural constraints. As illustrated in Fig. 17, the two pursuers exhibit anticipatory behavior by adapting their paths to circumvent the obstacles, thereby maintaining

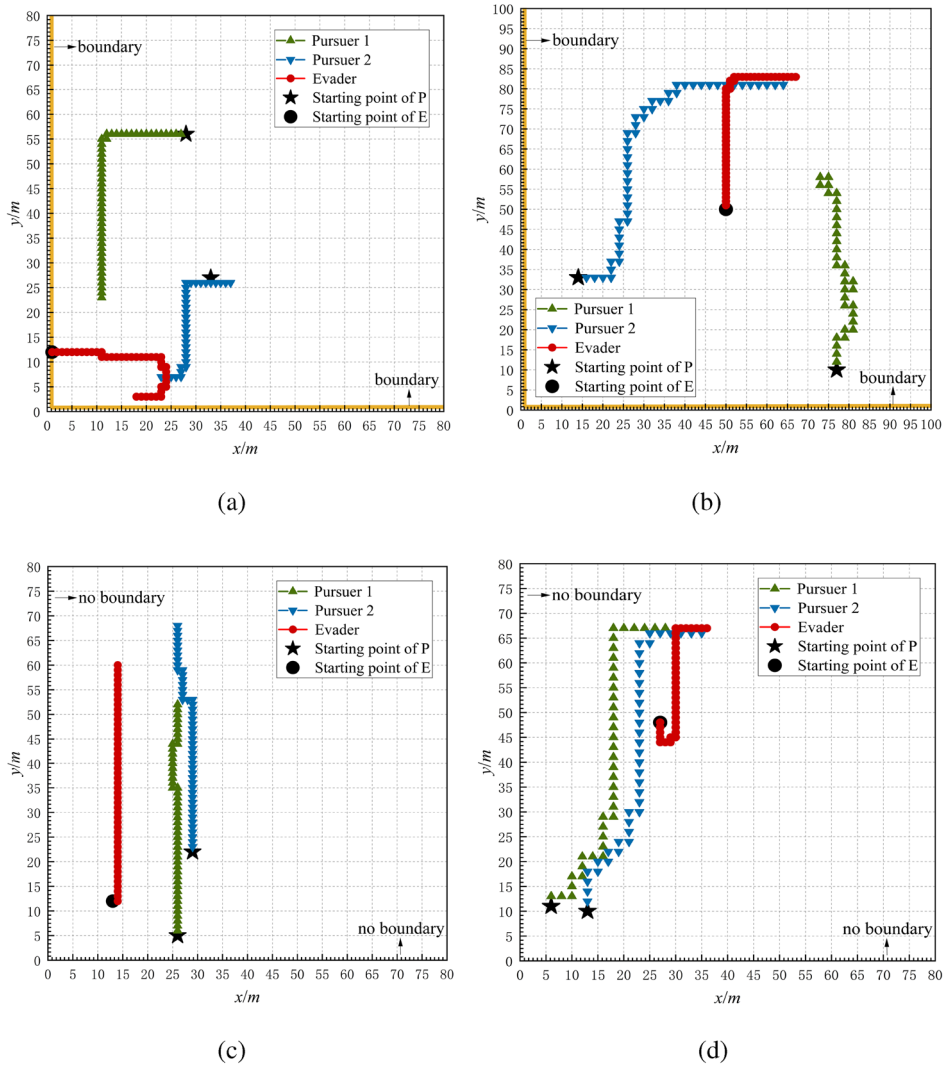

Fig. 16. Trajectory comparison under different boundary conditions and mobility scenarios. (**a**) Semi-boundary scenario (left, bottom) with $u_P^{\max}/u_E^{\max} = 1$. (**b**) Semi-boundary scenario (left, bottom) with $u_P^{\max}/u_E^{\max} = 2$. (**c**) Boundary-free scenario with $u_P^{\max}/u_E^{\max} = 1$. (**d**) Boundary-free scenario with $u_P^{\max}/u_E^{\max} = 2$.

pursuit of the evader. Meanwhile, the evader tends to flee along paths free of obstructions. This experiment and its corresponding visualization were included to highlight that while the presence of obstacles alters the trajectories, it does not significantly impede the emergence of coordinated and effective pursuit behavior.

Conclusion

This study presents a framework for classifying and defining basic behaviors, alongside a methodology for analyzing emergent behaviors in pursuit-evasion games. By employing the clustering-based methodology introduced in this paper, we group similar action sequences and observe how these patterns evolve into more complex behaviors under various conditions. The four typical strategys identified through cluster analysis include serpentine corner encirclement, stepwise corner approach, same-side edge confinement, and pincer flank attack. In each of these strategys, the pursuers collaborate and adjust their movements to progressively limit the evader's mobility, ultimately guiding them into a confined space for capture. These strategies involve sequential movements, intensified pressure, and coordinated positioning to effectively constrain the evader's escape routes, demonstrating different patterns of tactical coordination and spatial control.

Through the analysis of representative cases across these strategies, we identify distinct behavioral patterns and their constituent action sequences. For instance, the serpentine movement results from the alternating execution of drive and intercept actions, while the ladder-like motion involves drive and intercept actions at different scales. The vertical and parallel approaches rely on the continuous application of the same actions, whereas lazy movement is characterized by random action selection.

A key observation is the emergence of lazy pursuit behavior, where one pursuer reduces their effort while complementing the actions of the other. This phenomenon aligns with cooperative game theory principles,

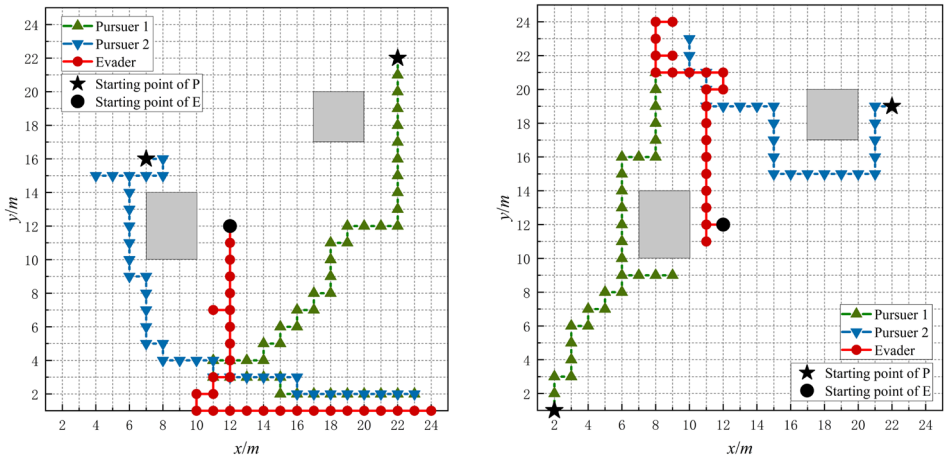

Fig. 17. Game trajectories in the obstacle-containing environment.

highlighting the efficiency of team strategies even when individual contributions vary. Furthermore, parallels are drawn between boundary-based encirclement behaviors and natural predatory tactics, such as dolphins' cooperative hunting techniques. These comparisons emphasize the role of spatial constraints and coordinated actions in shaping emergent behaviors.

These findings deepen our understanding of cooperative strategies and behavioral emergence, offering valuable insights for applications in robotics, game theory, and multi-agent systems. However, for practical implementation in engineering, further research is required, particularly in relation to more general velocity models and dynamic systems. Future work will focus on applying the proposed strategies to more complex, dynamic scenarios such as robotics, autonomous vehicles, and video games, where challenges like real-time decision-making and sensor integration come into play. Additionally, extending the proposed framework to three-dimensional pursuit-evasion scenarios is an important and challenging direction. Vertical motion, occlusions, and spatial constraints would introduce greater complexity to both agent behaviors and the analysis of emergent patterns. However, it is worth noting that even in simplified 2D grid environments, the emergence of strategic behaviors remains underexplored. To the best of our knowledge, our work is among the first to investigate the mechanisms of strategy emergence in multi-agent pursuit-evasion games from a learning-based, unsupervised perspective. This foundational step is crucial before scaling up to higher-dimensional or more realistic environments.

While the current paper focuses on the 2D case for clarity and tractability, we believe the methods and findings reported here are readily extensible. Future studies may incorporate deep pattern mining, inverse reinforcement learning, or hierarchical policy learning to support the analysis of emergent behaviors in larger-scale and 3D environments.

Data availability

The datasets and code used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 14 February 2025; Accepted: 5 August 2025
Published online: 11 August 2025

References

1. McCauley, E., Wilson, W. G. & de Roos, A. M. Dynamics of age-structured and spatially structured predator-prey interactions: Individual-based models and population-level formulations. *Am. Nat.* **142**, 412–442 (1993).
2. Nishimura, S. I. & Ikegami, T. Emergence of collective strategies in a prey-predator game model. *Artif. Life* **3**, 243–260 (1997).
3. De Souza, C. et al. Decentralized multi-agent pursuit using deep reinforcement learning. *IEEE Robot. Autom. Lett.* **6**, 4552–4559 (2021).
4. Fang, X., Wang, C., Xie, L. & Chen, J. Cooperative pursuit with multi-pursuer and one faster free-moving evader. *IEEE Trans. Cybern.* **52**, 1405–1414 (2020).
5. Yifeng, L., Liang, X. & Zhaohui, D. Nash-equilibrium strategies of orbital target-attacker-defender game with a non-maneuvering target. *Chin. J. Aeronaut.* **37**, 365–379 (2024).
6. Xie, W., Zhao, L. & Dang, Z. Game tree search-based impulsive orbital pursuit-evasion game with limited actions. *Sp. Sci. Technol.* **4**, 0087 (2024).
7. Zhao, L., Zhang, Y. & Dang, Z. Prd-maddpg: An efficient learning-based algorithm for orbital pursuit-evasion game with impulsive maneuvers. *Adv. Sp. Res.* **72**, 211–230 (2023).
8. Caraco, T., Martindale, S. & Pulliam, H. R. Avian flocking in the presence of a predator. *Nature* **285**, 400–401 (1980).
9. Ryer, C. H. & Olla, B. L. Information transfer and the facilitation and inhibition of feeding in a schooling fish. *Environ. Biol. Fish.* **30**, 317–323 (1991).

10. Mu, Z., Pan, J., Zhou, Z., Yu, J. & Cao, L. A survey of the pursuit-evasion problem in swarm intelligence. *Front. Inf. Technol. Electron. Eng.* **24**, 1093–1116 (2023).
11. Fisac, J. F. & Sastry, S. S. The pursuit-evasion-defense differential game in dynamic constrained environments. In *2015 54th IEEE Conference on Decision and Control (CDC)*. 4549–4556 (IEEE, 2015).
12. Yong, C. H. & Miikkulainen, R. *Cooperative Coevolution of Multi-Agent Systems* (University of Texas at Austin, 2001).
13. Nitschke, G. Co-evolution of cooperation in a pursuit evasion game. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*. Vol. 2. 2037–2042 (2003).
14. Lee, N., Chae, S., Baek, S. K. & Jeong, H.-C. Social dilemma in foraging behavior and evolution of cooperation by learning. *Sci. Rep.* **13**, 21948 (2023).
15. Masuko, M., Hiraoka, T., Ito, N. & Shimada, T. The effect of laziness in group chase and escape. *J. Phys. Soc. Jpn.* **86**, 085002 (2017).
16. Zhang, J. & Liu, K. Neural information squeezer for causal emergence. *Entropy* **25**, 26 (2022).
17. Sturdivant, R. L. & Chong, E. K. The necessary and sufficient conditions for emergence in systems applied to symbol emergence in robots. *IEEE Trans. Cognit. Dev. Syst.* **10**, 1035–1042 (2017).
18. Baronchelli, A. The emergence of consensus: a primer. *R. Soc. Open Sci.* **5**, 172189 (2018).
19. Helbing, D. Traffic and related self-driven many-particle systems. *Rev. Mod. Phys.* **73**, 1067 (2001).
20. Dorigo, M. The any system optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B* **26**, 1–13 (1996).
21. Levy, R. & Rosenschein, J. S. A game theoretic approach to distributed artificial intelligence and the pursuit problem. *ACM SIGOIS Bull.* **13**, 11 (1992).
22. Ji, M. et al. Cooperative pursuit with multiple pursuers based on deep minimax q-learning. *Aerosp. Sci. Technol.* **146**, 108919 (2024).
23. Moser, E. I. et al. Grid cells and cortical representation. *Nat. Rev. Neurosci.* **15**, 466–481 (2014).
24. Ouchi, A. & Fujisawa, S. Predictive grid coding in the medial entorhinal cortex. *Science* **385**, 776–784 (2024).
25. Zhao, L., Sun, Q., Xu, S. & Dang, Z. A learning-based algorithm for turn-based orbital pursuit-evasion problem with reaction-time delay. *Eng. Appl. Artif. Intell.* **145**, 110231 (2025).
26. Lowe, R. et al. Multi-agent actor-critic for mixed cooperative-competitive environments. *Adv. Neural Inf. Process. Syst.* **30** (2017).
27. Lewes, G. H. *Problems of Life and Mind* Vol. 2 (Trübner & Company, 1877).

Author contributions

Z. Dang proposed the concept and principle. S. Xu performed the research, analyzed data, prepared all figures (including Fig. 5) and wrote the initial draft of the paper. Both authors discussed the results and revised the manuscript.

Funding

This research was supported by National Key R&D Program of China: Gravitational Wave Detection Project (No. 2024YFC2207900, No. 2021YFC2202601, No. 2021YFC2202603) and National Natural Science Foundation of China (No. 12172288 and No.12472046).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Z.D.

Reprints and permissions information is available at www.nature.com/reprints.

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