Universiteit Leiden

Master Computer Science

[In Trust We Survive: Emergent Cooperation without Explicit Rewards in Multi-Agent Systems]

Name:	[Qianpu Chen]
Student ID:	[S3864553]
Date:	[04/3/2025]
Specialisation:	[Data Science]
1st supervisor:	[Mike Preuss]
2nd supervisor:	[Giulio Barbero]

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

In Multi-Agent Reinforcement Learning (MARL), fostering cooperation without explicit rewards remains a significant challenge. Many conventional approaches rely on carefully designed reward structures to incentivize collaborative behavior. However, real-world scenarios often lack predefined cooperative incentives, necessitating alternative mechanisms for emergent cooperation. To explore this problem, we propose a novel Tower Environment, inspired by the movie The Platform, where agents must make strategic decisions regarding resource consumption in a hierarchical setting with scarce resources. Unlike traditional MARL settings, our environment does not provide explicit cooperation rewards, making it an ideal testbed for studying emergent altruism and self-regulation. To tackle this challenge, we introduce the Imitation Evolutionary Game Strategy (IE-GS), which integrates trust-based decision-making, memory mechanisms, and adaptive exploration strategies to promote cooperation in multi-agent systems. Experimental results demonstrate that IE-GS significantly improves cooperation rates and fairness indices, outperforming baseline reinforcement learning algorithms. To further evaluate the adaptability and robustness of IE-GS, we extended our experiments to the Iterated Prisoner's Dilemma (IPD), a well-established game-theoretic framework for studying cooperation and defection dynamics. Our findings indicate that IE-GS not only maintains cooperative tendencies but also exhibits resilience against exploitative strategies in repeated interactions. This suggests that trust and adaptive learning mechanisms can serve as fundamental drivers of cooperation across diverse multi-agent settings.

Contents

1 Introduction **4**

2 Related Work **6**
2.1 Multi-Agent Systems . . . 6
2.2 Cooperation in Multi-Agent Systems . . . 7
2.3 Centralized Training with Decentralized Execution (CTDE) . . . 7
2.4 Communication in Multi-Agent Systems . . . 8
2.5 Trust Mechanisms in Muiti-Agent System . . . 10

3 Methodology **11**
3.1 Environment Design . . . 11
3.1.1 Reward and Penalty Mechanism . . . 12
3.2 Algorithm Design . . . 14
3.2.1 Algorithm Overview . . . 14
3.2.2 Memory Mechanism . . . 14
3.2.3 Trust Mechanism . . . 16
3.2.4 Exploration Mechanism . . . 17

4 Experiment **19**
4.1 Experiments in the Tower Environment . . . 19
4.1.1 Algorithm Comparison . . . 19
4.1.2 Impact of Initial Conditions on Learning . . . 20
4.1.3 Sensitivity to Hyperparameters . . . 20
4.1.4 Evaluation Metrics . . . 21
4.2 Experiments in the Iterated Prisoner's Dilemma . . . 21

5 Results **23**
5.1 Baseline Algorithm Performance . . . 23
5.2 Effect of Greedy Initialization . . . 25
5.3 Impact of Different Greedy Initialization Durations on Learning . . . 27
5.4 Hyperparameter Sensitivity . . . 29
5.4.1 Effect of Trust Update Learning Rate α . . . 29
5.4.2 Effect of Trust Adjustment Parameter β . . . 31
5.4.3 Effect of Exploration Strategies . . . 32
5.5 Results in the Prisoner's Dilemma . . . 33
5.5.1 Baseline Strategy Comparison: Strengths and Weaknesses of IE-GS . . . 34
5.5.2 Challenges from Delayed Strategies: Can Trust Be Exploited? . . . 37

6 Discussion **39**

7 Future Work **42**
7.1 Scaling to Larger Systems . . . 42
7.2 Heterogeneous Agent Strategies . . . 42
7.3 Hierarchical and Decentralized Trust Models . . . 43

8 Conclusion **44**

1 Introduction

The emergence of cooperation in multi-agent systems (MAS) is a fundamental and long-standing challenge in reinforcement learning (RL). Many existing approaches facilitate cooperation through explicit reward design, providing direct incentives for collaborative behaviors such as resource sharing, task completion, or achieving global objectives. However, in many real-world applications, predefined cooperative rewards may not always be feasible. Scenarios such as distributed resource management, emergency response coordination, and traffic control require agents to develop cooperative behaviors autonomously, without externally imposed incentives. Instead, cooperation must emerge as a result of individual decision-making, environmental feedback, and agent interactions[1].

To investigate cooperation in settings where explicit cooperative rewards are absent, we introduce a novel environment and algorithm inspired by the movie The Platform [2]. The film depicts an extreme resource allocation scenario in which individuals in a vertical structure receive food sequentially from top to bottom. Since individuals can freely decide how much to consume, excessive consumption by those on upper levels frequently leads to starvation for those below. This setting encapsulates key challenges in cooperative decision-making, including resource scarcity, competition, and high interdependence of agent choices. Based on this premise, we design the Tower Environment, a controlled multi-agent system where agents must regulate their food consumption as a platform moves downward through different levels. Crucially, the environment lacks any explicit cooperative rewards: agents receive rewards only for eating and penalties for starvation, with no direct incentives for preserving resources for others. Moreover, agents are periodically repositioned across different floors, introducing dynamic role reversals that require them to continuously adapt their strategies.

To explore whether cooperation can emerge in multi-agent systems without explicit cooperative rewards, we introduce algorithm, Imitation Evolutionary Game Strategy (IE-GS). Unlike traditional reinforcement learning approaches such as Q-Learning and Monte Carlo methods, which primarily optimize individual rewards, IE-GS incorporates trust-based interactions and environmental feedback to encourage cooperative behaviors. Instead of relying on predefined reward shaping, IE-GS allows agents to adjust their strategies dynamically based on observed interactions, enabling them to transition from self-interested behavior to long-term cooperation.

At the core of IE-GS is the idea that trust can serve as an implicit mechanism for cooperation. Agents evaluate the behaviors of others over time, adjusting their trust values accordingly—restrained consumption fosters trust, while selfish actions erode it. This dynamic trust mechanism influences decision-making, encouraging cooperative strategies without direct incentives. Moreover, by incorporating memory mechanisms and adaptive exploration-exploitation strategies, agents can refine their behaviors based on both immediate interactions and long-term trends in the environment.

However, several fundamental questions remain: Can cooperation truly emerge in the absence of explicit cooperative rewards? How does trust influence the stability and effectiveness of cooperation? Under what conditions does a trust-based mechanism succeed or fail? Furthermore, while trust can promote cooperation, it may also be vulnerable to exploitation. To examine the robustness of IE-GS, we extend our investigation beyond resource allocation in the Tower Environment and apply it to a broader strategic setting—the Iterated Prisoner's Dilemma (IPD). This classic framework introduces additional challenges, such as strategic deception and delayed cooperation, providing a more rigorous testbed for evaluating trust-based learning

in dynamic and adversarial environments.
To systematically investigate these challenges, we define the following key research questions:

- **RQ1: Can cooperation emerge in a multi-agent system where individual incentives actively conflict with collective welfare?** Many reinforcement learning methods rely on reward shaping to encourage cooperation. However, in our Tower Environment, individual rewards are directly tied to personal resource consumption, creating an inherent conflict between self-interest and collective welfare. This study examines whether cooperation can still emerge under such conditions, where consuming more food yields higher rewards but simultaneously reduces resources available for others, increasing the risk of systemic starvation.

- **RQ2: Can a trust-based decision-making mechanism like IE-GS mitigate selfish behavior and promote stable cooperation in a competitive multi-agent environment?** Trust is a crucial factor in human cooperation but remains underexplored in MAS. In the Tower Environment, agents face strong incentives to prioritize their own survival, leading to overconsumption and resource depletion. IE-GS introduces a trust mechanism where agents adjust their behavior based on observed interactions, increasing trust in cooperative peers and decreasing trust in selfish ones. This study examines whether such a mechanism can effectively shift agents from purely self-serving behaviors to sustainable cooperative strategies, even in the absence of explicit cooperative rewards.

- **RQ3: How do different initial conditions, such as initial trust values and early-stage greedy behavior, impact the long-term emergence of cooperation?** The initial setup of an MAS significantly influences its learning dynamics. This study evaluates how factors such as low initial trust or early selfish behaviors affect the eventual development of cooperative interactions.

- **RQ4: How do hyperparameters such as exploration rate, trust update rate, and trust adjustment parameter affect the performance of agents in terms of cooperation, fairness, and stability?** Hyperparameter choices in RL affect exploration, learning, and adaptation. We analyze how these factors influence long-term cooperative behaviors.

- **RQ5: Can a trust-based cooperation mechanism remain effective when agents face dynamic and deceptive opponent strategies?** Real-world cooperation is rarely static; adversaries may adjust their behaviors to exploit or deceive cooperative agents. By evaluating IE-GS in IPD against adaptive defectors and delayed-response strategies, we explore its resilience to long-term strategic manipulation.

This paper is structured as follows: Section 3 details the design of the Tower Environment and the IE-GS algorithm. Section 4 presents the setup for our comparative experiments. Section 5 provides empirical evaluations of agent cooperation dynamics in both the Tower Environment and the Iterated Prisoner's Dilemma. Section 6 interprets key findings, analyzes algorithmic limitations, and discusses broader implications for cooperative multi-agent learning. Section 7 explores potential extensions of our work, including scaling to larger systems, handling heterogeneous agent strategies, and developing hierarchical and decentralized trust models. Finally, Section 8 summarizes our contributions and highlights the impact of our findings on future research directions.

2 Related Work

This section reviews key research areas relevant to our study, including multi-agent systems (MAS), cooperation mechanisms, centralized training with decentralized execution (CTDE), communication strategies, and trust mechanisms in MARL.

2.1 Multi-Agent Systems

Multi-agent systems (MAS) have become a fundamental paradigm for modeling complex, decentralized environments where multiple autonomous entities, referred to as agents, interact within a shared system. Unlike traditional single-agent frameworks, MAS involves independent decision-makers that operate within an environment characterized by uncertainty, partial observability, and dynamic interactions. These agents may exhibit cooperative, competitive, or mixed-motive behaviors depending on the nature of their objectives and interactions. The study of MAS spans various domains, including robotics, economics, distributed computing, and artificial intelligence, where the ability to design and control autonomous agents has far-reaching implications[3].

At the core of multi-agent systems lies the challenge of decentralized decision-making. Each agent must process local observations, infer relevant information about the environment and other agents, and make strategic choices to optimize its own objectives. In cooperative settings, agents must align their individual goals with those of the collective, requiring mechanisms for coordination and communication. Conversely, in competitive scenarios, agents must anticipate the strategies of others while maximizing their own utility, often resulting in adversarial dynamics. The complexity of MAS further increases when agents operate under partial observability, where access to global state information is limited, making efficient coordination and long-term planning difficult[4].

The applications of MAS are diverse, ranging from autonomous vehicle fleets and robotic swarms to financial markets and distributed sensor networks. In robotic systems, multiple autonomous agents can collaborate to perform tasks such as search-and-rescue operations or warehouse logistics, requiring real-time adaptation and communication[5]. Similarly, in traffic management[6], MAS is used to optimize traffic flow through decentralized decision-making among autonomous vehicles. Financial markets also exhibit multi-agent characteristics, where algorithmic trading agents interact in highly dynamic environments, responding to market conditions and competing for optimal trade execution. In each of these cases, the underlying mechanisms of decision-making, learning, and coordination define the system's overall efficiency and performance.

While multi-agent systems provide a powerful framework for solving complex real-world problems, the primary challenge remains how to design and train agents that can effectively learn optimal behaviors in dynamic, multi-agent settings. Reinforcement learning has emerged as a key approach to addressing this challenge, where agents learn policies through trial-and-error interactions with the environment. However, standard reinforcement learning approaches, designed for single-agent settings, often fail to scale effectively to multi-agent domains due to non-stationarity, credit assignment difficulties, and the need for strategic reasoning about the actions of others[7]. These challenges underscore the necessity of developing specialized learning algorithms tailored to multi-agent settings, particularly those that facilitate emergent cooperation.

2.2 Cooperation in Multi-Agent Systems

Cooperation in multi-agent systems (MAS) refers to scenarios where multiple agents must coordinate their actions to achieve shared objectives, optimize collective performance, or ensure equitable resource distribution. Unlike single-agent settings, where an agent independently learns an optimal policy, multi-agent cooperation requires agents to account for the actions and strategies of others[8]. This introduces challenges such as coordination, credit assignment, and long-term stability. While cooperative behavior can emerge naturally in some environments, in many cases, specialized mechanisms are necessary to facilitate and sustain cooperation.

A key theoretical foundation for studying cooperation in MAS is game theory[9], which provides formal models for analyzing interactions among rational agents. In cooperative game theory, agents form coalitions to maximize joint rewards, employing mechanisms such as the Shapley value and core stability to ensure fair distribution of collective gains. In contrast, non-cooperative game theory assumes that agents act independently to maximize their own utility, with Nash equilibria representing stable outcomes where no agent has an incentive to unilaterally deviate. Many real-world MAS scenarios involve social dilemmas such as the Prisoner's Dilemma, Public Goods Game, and Tragedy of the Commons, where individual rationality often leads to collectively suboptimal outcomes unless explicit cooperation mechanisms are introduced[10].

In multi-agent reinforcement learning (MARL), fostering cooperation becomes even more complex due to the issue of non-stationarity—agents continuously adapt their strategies, altering the environment dynamically. Traditional reinforcement learning (RL) algorithms struggle in such settings because they typically assume a static environment, whereas in MARL, each agent's learning process directly influences the behavior of others. Additionally, credit assignment remains a significant challenge, as rewards are often distributed among multiple agents, making it difficult to attribute success to specific individuals. These challenges underscore the need for advanced learning strategies that promote stable and effective cooperation in multi-agent environments.

To address these challenges, various cooperation mechanisms have been proposed in the MARL literature. These mechanisms generally fall into three categories: centralized training with decentralized execution (CTDE)[11], explicit communication, and implicit cooperation via information mechanisms. Each approach offers unique advantages but also faces inherent limitations, particularly when applied to real-world, large-scale MAS. The following sections provide a detailed discussion of these mechanisms and highlight their differences from the approach used in this study.

2.3 Centralized Training with Decentralized Execution (CTDE)

One of the predominant paradigms for facilitating cooperation in multi-agent reinforcement learning (MARL) is Centralized Training with Decentralized Execution (CTDE), a framework designed to leverage centralized learning during the training phase while allowing agents to operate independently during execution. This approach is particularly effective in addressing the challenges associated with decentralized decision-making, as it enables agents to exploit global state information during training without requiring direct coordination or communication at inference time. The underlying motivation for CTDE is to optimize cooperative behaviors by utilizing a centralized training architecture that provides each agent with access to privileged information that would otherwise be unavailable in a fully decentralized setting. By learning from this global perspective, agents can develop strategies that are more aligned with collective

objectives while still maintaining autonomy when deployed in real-world environments where centralized control is often impractical[12].
Within the CTDE framework, several algorithmic approaches have been proposed to decompose the complexities of multi-agent interactions while preserving cooperation. One of the most influential methods is Multi-Agent Deep Deterministic Policy Gradient (MADDPG), which extends the Deep Deterministic Policy Gradient (DDPG) algorithm to multi-agent settings by employing a centralized critic for each agent during training[13]. The centralized critic has access to the joint state-action space of all agents, allowing it to compute more informed value estimates that incorporate the dependencies between agents' actions. However, during execution, each agent acts independently using only its local observations, ensuring scalability and decentralized decision-making. This formulation enables agents to learn cooperative behaviors that would be challenging to discover in a fully decentralized reinforcement learning setting, where the non-stationarity introduced by multiple learning agents often destabilizes training.
Another notable class of CTDE-based methods is value decomposition techniques, which aim to decompose the joint value function into individual agent-specific components to facilitate decentralized decision-making. One such approach is Value Decomposition Networks (VDN), which assumes that the joint Q-function can be expressed as a simple summation of individual Q-values, thereby enabling independent agents to optimize their policies while still contributing to a shared objective[14]. While effective in structured cooperative environments, VDN's simplistic value decomposition limits its ability to capture complex inter-agent dependencies. QMIX extends this idea by introducing a mixing network that learns a more flexible function for aggregating individual Q-values into a joint Q-function, enforcing a monotonicity constraint that ensures optimality under decentralized execution. These approaches have demonstrated strong performance in cooperative multi-agent tasks, particularly in domains such as robotic swarm control, autonomous driving, and real-time strategy games.
Despite the empirical successes of CTDE, several limitations remain. First, CTDE relies heavily on explicit reward shaping, which assumes that cooperative incentives can be predefined and embedded into the learning process. This assumption does not always hold in real-world scenarios where cooperative behavior must emerge from intrinsic interactions rather than being externally prescribed. The need for a well-designed global reward structure introduces a dependence on domain-specific knowledge and limits the generalizability of these methods to environments where cooperative incentives are not explicitly defined. Additionally, the scalability of CTDE-based approaches is a fundamental challenge, as the requirement for centralized training becomes computationally expensive when the number of agents increases. The centralized critic or value decomposition models must process exponentially growing state-action spaces, leading to bottlenecks in training efficiency and memory consumption[15]. Finally, CTDE does not necessarily generalize well to dynamic environments, particularly in open multi-agent systems where new agents may enter or exit over time. Many existing CTDE methods assume a fixed agent population with stable interaction patterns, making them less suited for dynamic settings where cooperation mechanisms must continuously adapt to evolving agent compositions and objectives.

2.4 Communication in Multi-Agent Systems

Communication plays a crucial role in facilitating cooperation within multi-agent systems (MAS), particularly in environments where agents must coordinate their actions to achieve

shared objectives[16]. The ability to exchange information allows agents to reduce uncertainty about others' intentions, synchronize strategies, and mitigate suboptimal behaviors caused by partial observability. In multi-agent reinforcement learning (MARL), communication mechanisms are often introduced to enhance cooperative behaviors by enabling agents to share relevant state or policy information, thereby improving joint decision-making[17]. However, designing effective communication protocols remains a challenging problem due to constraints such as communication bandwidth limitations, delayed or noisy transmissions, and the risk of strategic deception when agents have competing incentives.

Several approaches have been developed to integrate communication into MARL frameworks, ranging from explicit message passing to learned communication protocols. One of the most common paradigms involves differentiable communication, where agents learn to exchange continuous-valued messages through neural network-based architectures. This approach is exemplified by CommNet[18], which employs a shared recurrent neural network to aggregate messages from multiple agents and process them alongside individual agent states, enabling implicit coordination through distributed learning. Another widely studied method is DIAL (Differentiable Inter-Agent Learning)[19], which incorporates discrete communication signals into deep Q-learning, allowing agents to learn when and what to communicate by treating messages as additional network inputs. These methods enable agents to develop context-aware communication strategies, adjusting their message content based on situational demands rather than relying on fixed communication protocols.

Beyond explicit message exchange, some MARL frameworks adopt attention-based communication mechanisms to enhance the efficiency of inter-agent information sharing[20]. TarMAC (Targeted Multi-Agent Communication) introduces an attention-based message selection mechanism that enables agents to dynamically decide whom to communicate with, preventing unnecessary message exchanges and improving scalability[21]. Similarly, IC3Net (Iterated Communication for Cooperative Agents) leverages recurrent attention networks to facilitate iterative message refinement, allowing agents to converge on more effective coordination strategies over multiple communication rounds[22]. These approaches address a fundamental challenge in multi-agent communication: selective and efficient information sharing, ensuring that communication is both meaningful and computationally feasible in large-scale systems.

Despite the successes of communication-based MARL, several inherent limitations restrict its applicability to real-world multi-agent cooperation[23]. First, communication introduces additional learning complexity, as agents must simultaneously learn both task-specific policies and effective message-passing strategies. This can lead to convergence instability, particularly in environments where optimal communication protocols are nontrivial to discover. Additionally, the assumption of reliable communication may not hold in real-world settings, where factors such as bandwidth constraints, packet loss, or adversarial agents may disrupt information exchange. In fully decentralized environments, where agents operate asynchronously and independently, reliance on explicit communication can introduce vulnerabilities, making systems more susceptible to failures when communication is compromised.

Furthermore, many existing communication-based MARL methods assume that agents are inherently cooperative, which may not be a valid assumption in open multi-agent systems where individual incentives are misaligned. In mixed-motive environments, agents may strategically manipulate messages to mislead others, requiring the integration of trust mechanisms or mechanism design principles to ensure truthful communication. Some studies have attempted to address this issue through adversarial training, where agents are trained against potential deceptive communicators to enhance robustness. However, these methods remain largely ex-

perimental, and designing communication strategies that are both scalable and resilient to manipulation remains an open problem in the field.

2.5 Trust Mechanisms in Muiti-Agent System

In multi-agent reinforcement learning (MARL), trust mechanisms are pivotal for fostering cooperation, particularly in environments where agents may exhibit unreliable or adversarial behaviors. Traditional trust models[24] often rely on direct experiences, where agents assess the trustworthiness of their peers based on past interactions. For instance, decentralized trust mechanisms enable agents to independently decide which neighbors to communicate with, thereby mitigating the influence of unreliable agents and improving consensus success rates.
Another approac[25] involves the development of trust evaluation models that assess the trustworthiness of target agents within the MARL framework. These models aim to provide a quantitative measure of trust, which agents can use to make informed decisions about cooperation and coordination.
While these existing trust mechanisms have contributed to enhancing cooperation in MARL, they often depend on explicit trust evaluations and predefined communication protocols. In contrast, our proposed approach distinguishes itself by not relying on explicit trust assessments or direct communication. Instead, it leverages implicit trust signals derived from agents' observed behaviors and historical interactions. This method allows for the emergence of cooperative strategies through evolutionary game dynamics, enabling agents to adapt their behaviors based on the inferred trustworthiness of their peers. By focusing on implicit trust cues and adaptive learning, our approach offers a more flexible and scalable solution for fostering cooperation in complex, dynamic multi-agent environments.

3 Methodology

This section describes the methodology used to investigate emergent cooperation in a multi-agent environment. We first introduce the environment design, where agents interact under resource constraints within a hierarchical structure, making decisions that directly impact others' survival. Then, we introduce the algorithm design, detailing the Imitation Evolutionary Game Strategy (IE-GS), which enables agents to learn adaptive behaviors through memory mechanism, evolving trust mechanism, and exploration balancing.

3.1 Environment Design

This study draws inspiration from the movie "The Platform". To avoid the violent screen Fig1 of the movie, a simplified visual representation of the tower environment was implemented in Pygame, using hand-drawn cartoon to simulate the resource allocation dynamics. A virtual

Figure 1: Movie Screen

environment was designed, consisting of a four-story tower with one agent assigned to each floor, resulting in a total of four agents. The sole resource in the tower is food, which is carried by a movable platform starting at the top floor and descending one level at a time. The platform begins with an initial amount of 4 food units, just sufficient to meet the minimum survival requirement of all agents (i.e., each agent needs to consume 1 food unit to survive). Agents can choose to consume 0, 1, or 2 food units when the platform reaches their floor. Unconsumed food continues to the next floor, creating a direct interdependence between the decisions of agents on higher floors and the survival of those below. An agent's hunger level increases linearly over time, and if it reaches the predefined maximum hunger threshold H_{max}, the agent is considered "dead." The dynamic hunger update for each agent is governed by the following equation:

$$H_i(t+1) = \min\left(H_{\text{max}}, H_i(t) + \Delta H - \kappa A_i\right)$$

where $H_i(t)$ is the hunger level of agent i at time t, ΔH is the natural increase in hunger per round, κ is the reduction in hunger per unit of food consumed, and A_i is the agent's food consumption.

At the end of each round, the positions of all agents are randomly reassigned to different floors, introducing further uncertainty into the environment. This dynamic allocation requires

Figure 2: The tower environment in the movie

agents to develop adaptive strategies to cope with changing conditions and to explore diverse behaviors within the environment.

3.1.1 Reward and Penalty Mechanism

The reward structure in this environment is intentionally simple yet challenging, consisting of two components: consumption rewards and survival penalties.
Agents earn rewards based on the amount of food consumed when the platform reaches their floor. The reward increases with the consumption amount and is defined as:

$$R_i = A_i, \quad A_i \in \{0, 1, 2\}$$

where R_i is the reward for agent i, and A_i is the agent's food consumption.
If an agent's hunger level reaches H_{max}, the agent is considered dead and incurs a penalty of -1. The overall reward for an agent is therefore expressed as:

$$R_i = \begin{cases} A_i, & \text{if } H_i(t) < H_{\text{max}} \\ -1, & \text{if } H_i(t) \geq H_{\text{max}} \end{cases}$$

This reward structure is entirely short-term oriented, with rewards closely tied to immediate consumption actions. This design introduces several challenges. First, agents are incentivized to consume as much food as possible to maximize their immediate rewards, potentially at the expense of agents on lower floors. Second, the absence of explicit cooperation rewards makes it difficult for agents to develop collaborative strategies. Instead, the design indirectly discourages cooperation by aligning rewards with selfish behavior. Third, the dynamic and random reassignment of agent positions at the end of each round further complicates strategy development, as agents must continuously adapt to new environmental conditions. To better illustrate the dynamics of the environment, Fig 3 shows the initial state of the environment, where the food platform starts at the top of the tower, and agents are randomly assigned to

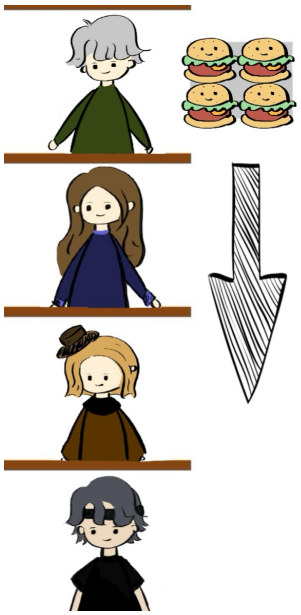


Figure 3: environment

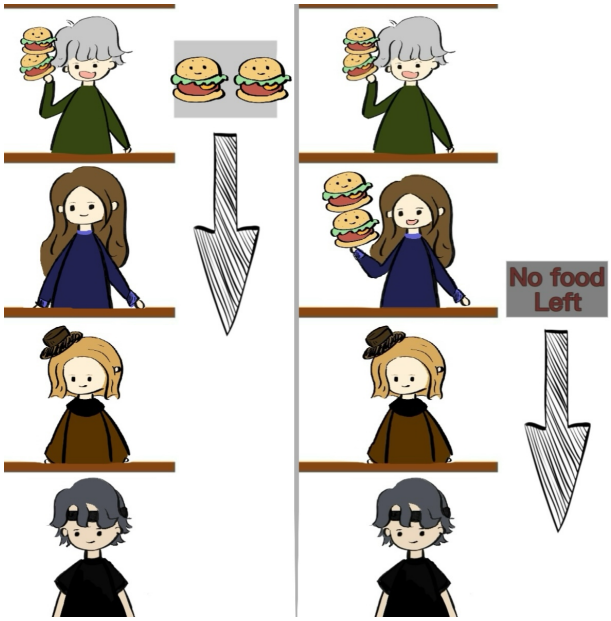


Figure 4: bad situation

floors. Fig 4 and 5 demonstrate two typical scenarios. Fig 4 shows the situation that agents on lower floors are left without resources and eventually die When agents on higher floors overconsume food. Fig 5shows the situation that all agents survive when agents learn to moderate their consumption, with each consuming exactly 1 food unit.

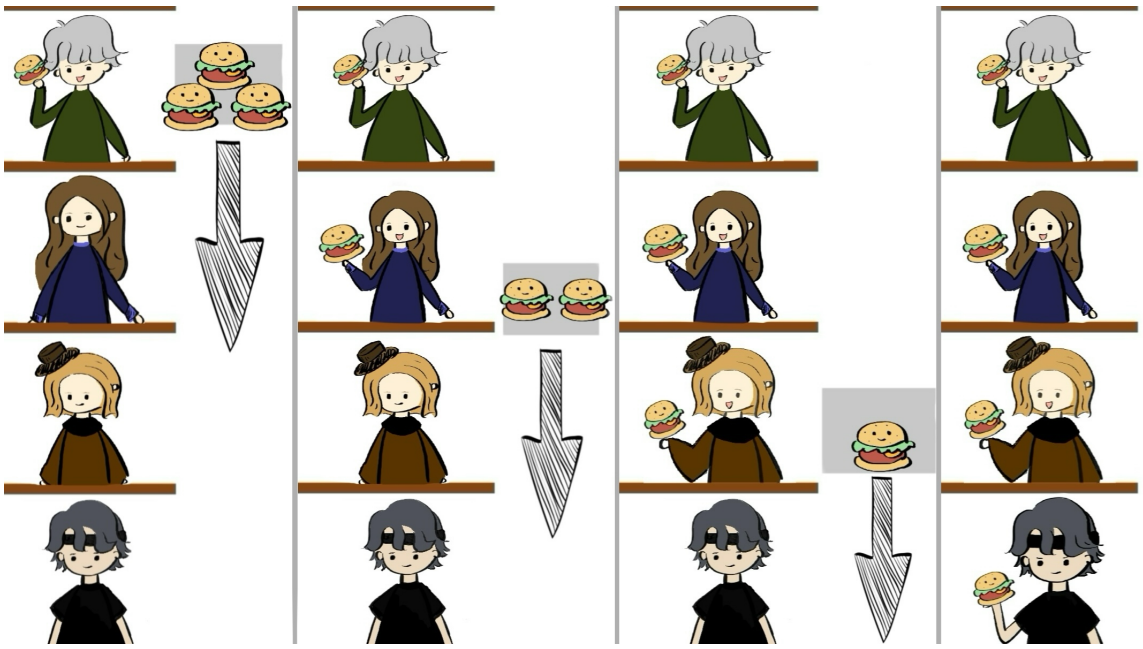


Figure 5: good situation

3.2 Algorithm Design

This section provides a detailed explanation of the various components of the proposed algorithm.

3.2.1 Algorithm Overview

The Imitation Evolutionary Game Strategy (IE-GS) is designed to address the challenges of decentralized decision-making in a multi-agent system with limited resources. This algorithm enables agents to develop emergent cooperative behaviors without explicit cooperation rewards, relying instead on adaptive decision-making driven by memory retention, evolving trust, and exploration-exploitation balancing.

Each agent in the system must decide how much food to consume while taking into account both immediate survival needs and long-term cooperation incentives. Unlike traditional reinforcement learning approaches that rely on predefined cooperative rewards, IE-GS leverages an evolving trust mechanism that emerges from agents' interactions. Trust acts as an implicit signal reflecting past behaviors and is used to modulate the agent's decision-making strategy. Exploration is dynamically adjusted based on trust stability and historical performance, ensuring that agents maintain adaptability while converging towards sustainable policies.

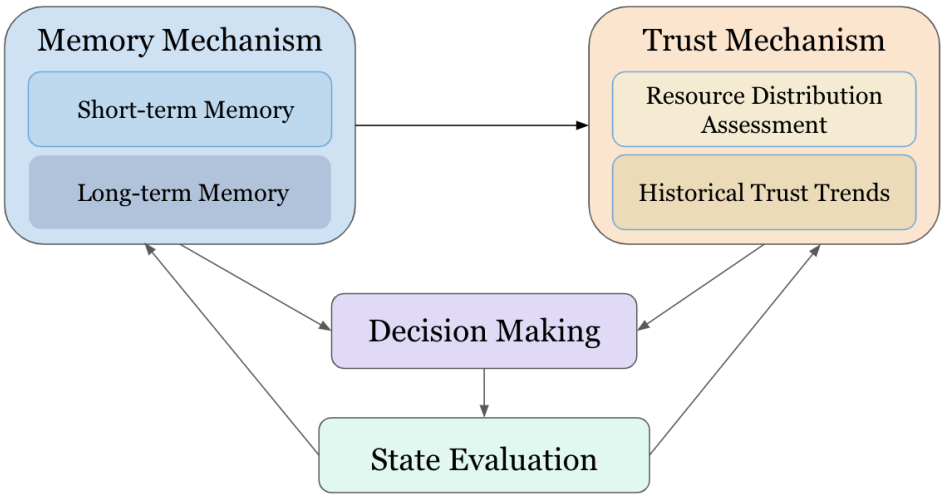


Figure 6: Algorithm Structure

The algorithm operates within an environment where agents occupy discrete hierarchical positions. The availability of resources at each level is influenced by the consumption behaviors of those above. Over time, agents learn patterns of resource distribution, adapt their strategies, and refine their decision-making based on trust dynamics and memory-driven learning. The following subsections provide a detailed description of the three fundamental mechanisms underlying IE-GS.

3.2.2 Memory Mechanism

Memory serves as the foundation of the IE-GS algorithm, allowing agents to retain and utilize past interactions to optimize their decision-making processes. It captures both short-term

Algorithm 1 Imitation Evolutionary Game Strategy (IE-GS)

Require: Environment $\mathcal{E}$, Trust update parameters α, β, γ, Exploration decay parameters $\epsilon_{\max}, \epsilon_{\min}, \lambda$

Ensure: Emergent cooperative behavior among agents

Initialize trust values $T_{ij} \leftarrow 0$ for all agents i, j
Initialize short-term memory $\mathcal{M}_s$ and long-term memory $\mathcal{M}_l$
Set exploration probability $\epsilon \leftarrow \epsilon_{\max}$
while not converged **do**
    Reset environment $\mathcal{E}$ and assign agents to random floors
    for each episode e **do**
        Observe initial state s_0
        for each time step t **do**
            for each agent i **do**
                Retrieve state $s_t^i = (H_i^t, F^t, T_{ij})$
                Select action a_i^t based on trust, hunger, and exploration:
                if $T_{ij} > 0.5$ and $H_i^t < 0.8 H_{\max}$ **then**
                    $a_i^t \leftarrow 1$ ▷ Moderate consumption
                else if $T_{ij} < -0.3$ or $H_i^t > 0.8 H_{\max}$ **then**
                    $a_i^t \leftarrow 2$ ▷ Max consumption
                else
                    $a_i^t \leftarrow$ exploration-based decision
                end if
                Execute action a_i^t and observe reward R_i^t
                Store (s_t^i, a_i^t, R_i^t) in $\mathcal{M}_s$ and $\mathcal{M}_l$
            end for
            Update environment state s_{t+1}
        end for
        for each agent i **do**
            for each preceding agent j **do**
                Compute trust update ΔT_{ij}:

$$\Delta T_{ij} = \alpha \cdot \frac{F^t}{F_{\max}} - \beta \cdot \max(0, H^t - \theta_h) + \gamma \cdot (\bar{T}_{ij}^t - T_{ij}^t)$$

                Apply trust update:

$$T_{ij}^{t+1} = \text{clip}(T_{ij}^t + \Delta T_{ij}, -1, 1)$$

                Store updated T_{ij}^{t+1} in memory
            end for
        end for
        Update exploration probability ϵ:

$$\epsilon_{e+1} = \begin{cases} \min(\epsilon_{\max}, \epsilon_e \times 1.1), & \text{if trust variance is high,} \\ \max(\epsilon_{\min}, \epsilon_e \times \lambda), & \text{otherwise.} \end{cases}$$

        Randomly reassign agent positions
    end for
end while
return Emergent cooperative behavior

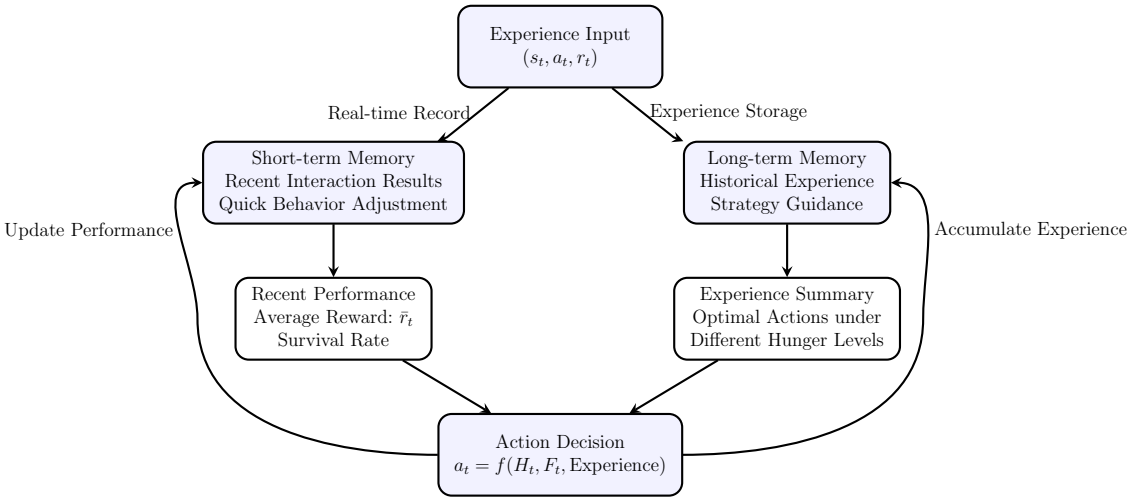


Figure 7: Memory Mechanism: Quick behavior adjustment through short-term memory while building stable strategies with long-term memory

fluctuations in behavior and long-term patterns of resource distribution, enabling agents to refine their strategies over time.

Each agent maintains two distinct memory buffers: a short-term memory (STM) and a long-term memory (LTM). STM is designed to store recent experiences within a limited temporal window, allowing agents to quickly adapt to environmental fluctuations. Formally, the STM buffer is defined as:

$$\mathcal{M}_s = \{(s_t, a_t, r_t)\}_{t=T-N_s}^{T} \tag{1}$$

where (s_t, a_t, r_t) represents the observed state, executed action, and received reward at time step t, and N_s is the fixed size of the short-term memory buffer.

In contrast, LTM accumulates information across a significantly longer time horizon, capturing historical trends in trust evolution, cooperative behaviors, and survival strategies. The LTM buffer is defined as:

$$\mathcal{M}_l = \{(s_t, a_t, r_t)\}_{t=0}^{T} \tag{2}$$

where T is the total number of episodes experienced by the agent.

Memory directly influences decision-making by reinforcing successful past actions and mitigating suboptimal behaviors. If an agent's STM indicates that a particular action consistently leads to starvation, it will adjust its short-term strategy accordingly. However, if its LTM suggests that temporary resource conservation benefits long-term survival, the agent may tolerate short-term sacrifices in favor of sustainable cooperation.

3.2.3 Trust Mechanism

Trust plays a critical role in regulating cooperation within the IE-GS framework. Each agent maintains a dynamic trust value towards its neighboring agents, which reflects past observations of resource-sharing behaviors. Trust is continuously updated based on the proportion of food received from agents on higher floors, the agent's own hunger level, and long-term behavioral trends. For an agent i at time step t, the trust value towards its neighboring agent on the preceding floor, denoted as T_{ij}^t, is computed as follows:

$$T_{ij}^{t+1} = \text{clip}\left(T_{ij}^t + \Delta T_{ij}, -1, 1\right) \tag{3}$$

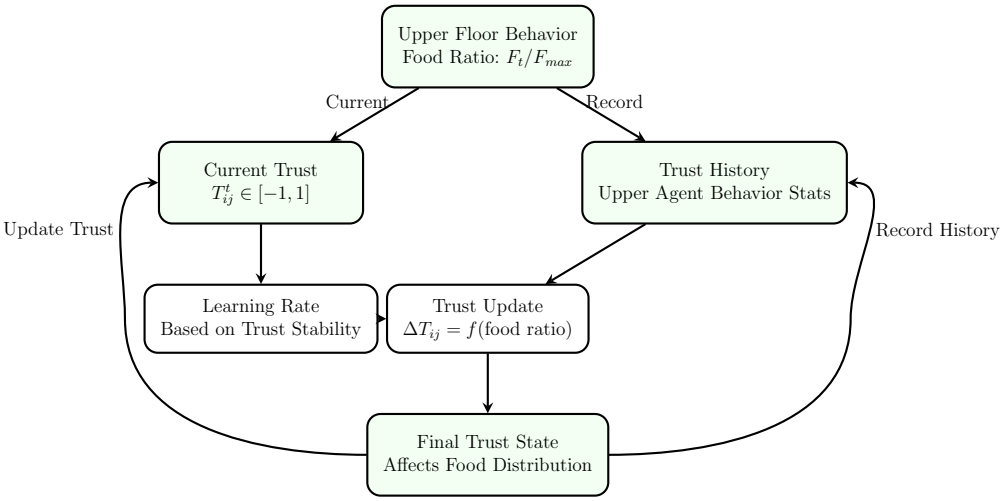


Figure 8: Trust Mechanism: Dynamically adjusts trust based on food distribution behavior to guide cooperation decisions

where ΔT_{ij} represents the trust adjustment factor.
The change in trust ΔT_{ij} is influenced by three primary factors. The first is the observed resource consumption pattern. If an agent on the preceding floor leaves a significant portion of food, trust increases proportionally to the food ratio:

$$\Delta T_{ij} = \alpha \cdot \frac{F^t}{F_{\max}} \tag{4}$$

where F^t is the remaining food observed by agent i and $F_{\max}$ represents the total initial food supply.
The second factor is the agent's own hunger level. If an agent experiences prolonged high hunger levels due to insufficient food supply, trust in the preceding agent is penalized:

$$\Delta T_{ij} = -\beta \cdot \max(0, H^t - \theta_h) \tag{5}$$

where H^t is the current hunger level and θ_h is a predefined threshold.
The third factor accounts for historical trust trends, ensuring that trust updates are not overly reactive to single interactions. The long-term trust trajectory $\bar{T}_{ij}^t$ is used to modulate updates:

$$\Delta T_{ij} = \gamma \cdot \left(\bar{T}_{ij}^t - T_{ij}^t\right) \tag{6}$$

where γ determines the weight of historical trust.
By integrating these factors, the trust mechanism allows agents to adapt their behaviors based on past interactions, fostering emergent cooperation without requiring explicit rewards for altruistic actions.

3.2.4 Exploration Mechanism

Exploration in IE-GS is dynamically adjusted to balance adaptation and stability. Unlike traditional reinforcement learning approaches where exploration follows a fixed decay schedule, IE-GS conditions exploration probability on both trust stability and recent agent performance. At each time step, the exploration probability ϵ_t is updated as follows:

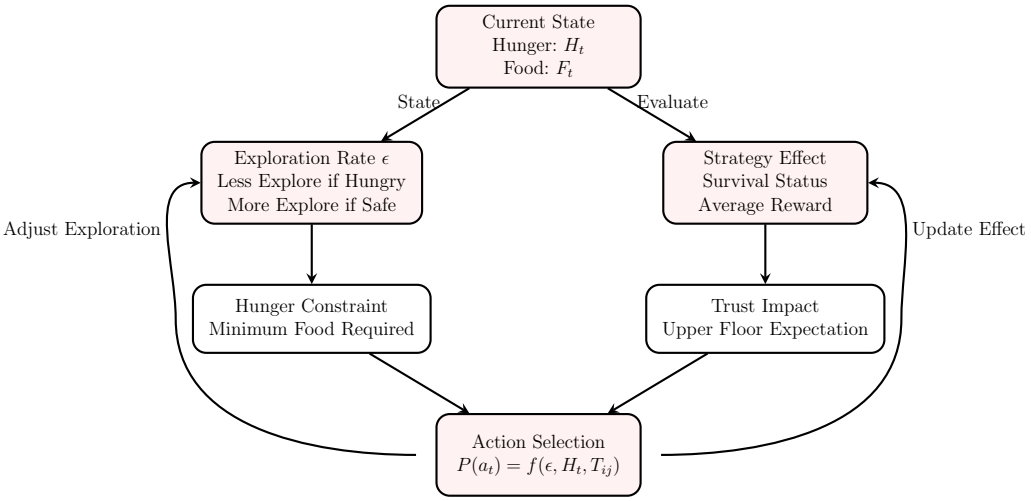


Figure 9: Exploration Mechanism: Balances between survival and exploration by adjusting exploration rate based on hunger and trust

$$\epsilon_{t+1} = \begin{cases} \min(0.9, \epsilon_t \times 1.1) & \text{if} \quad \sigma_T > \theta_T, \\ \max(\epsilon_{\min}, \epsilon_t \times 0.995) & \text{otherwise.} \end{cases} \tag{7}$$

where σ_T represents the variance in trust values over recent episodes and θ_T is a stability threshold. When trust variance is high, indicating unstable cooperation, exploration is increased to encourage alternative strategies. When trust variance is low, exploration is reduced, allowing the agent to exploit learned strategies.

Exploration is further modulated by trust-influenced action selection. The probability of selecting a specific action a_i^t is determined by both the current exploration rate and the trust value:

$$P(a_i^t|s_t) = \begin{cases} 1 - \epsilon_t & \text{if} \quad T_{ij} > 0.5 \quad \text{and} \quad H_i^t < 0.7H_{\max}, \\ \epsilon_t & \text{otherwise.} \end{cases} \tag{8}$$

This formulation ensures that highly trusted agents are more likely to exploit cooperative strategies, while agents with uncertain trust values engage in exploratory actions.

Over time, exploration naturally declines as long-term memory stabilizes trust values and cooperative behaviors emerge. However, stochastic exploration is retained at a minimal level to prevent agents from becoming trapped in locally optimal but globally suboptimal strategies.

4 Experiment

To evaluate the effectiveness of the Imitation Evolutionary Game Strategy (IE-GS) in fostering cooperation, we conducted experiments in two distinct environments: the Tower Environment and the Iterated Prisoner's Dilemma (IPD). The first environment models cooperative behavior under competitive resource constraints, while the second explores strategic decision-making in a repeated game setting. These two experiments together provide a comprehensive evaluation of how trust-based learning can drive cooperation in different multi-agent scenarios.

4.1 Experiments in the Tower Environment

The experimental environment is inspired by the movie *The Platform*, where resources (food) are distributed sequentially across different hierarchical levels. This environment is designed to test whether cooperation can emerge in a multi-agent system (MAS) without explicit cooperative rewards. The Tower Environment consists of:

- Four agents, each occupying one of four floors.
- A food platform that starts at the top floor and moves downward.
- Limited food resources, initially set to four units per round.
- Dynamic floor reassignment after each round to prevent fixed positional advantages.
- A hunger mechanism, where agents receive a penalty if they died due to hunger reaching the upper limit.

Each agent must decide how much food to consume when the platform reaches its floor. The challenge is that overconsumption by higher-floor agents leaves lower-floor agents with insufficient resources, which can lead to starvation.
Agents operate under partially observable conditions, with access only to their local state, which includes hunger level, available food, and estimated trust values for adjacent layers. Since trust is not explicitly rewarded, its formation and stability depend on how agents interpret and respond to past interactions.
The environment is implemented in Python using Pygame for visualization, and it tracks various performance metrics, including cooperation rate, success rate, and fairness index.
The primary objective of this experiment is to evaluate the effectiveness of the proposed Imitation Evolutionary Game Strategy (IE-GS) and examine its ability to foster cooperation in a competitive multi-agent environment without explicit cooperative rewards. To achieve this, we systematically design a series of experiments that investigate (i) how IE-GS compares to traditional MARL algorithms, (ii) how different initial conditions influence learning, and (iii) the sensitivity of IE-GS to hyperparameter choices.

4.1.1 Algorithm Comparison

Q-Learning is a classical value-based reinforcement learning algorithm. Agents learn action-value functions by updating a Q-table. Prioritizes maximizing individual long-term rewards, often leading to selfish behaviors.
Monte Carlo (MC) Method uses episodic sampling to update value estimates. Agents adjust

Feature	Q-Learning	Monte Carlo	IE-GS
Decision Basis	Value-based	Return-based	Trust-driven
Adaptation Speed	Fast but short-term	Slow but long-term	Dynamic
Memory Mechanism	None	Episodic	Short- and Long-term
Exploration	ϵ-greedy	Episodic updates	Adaptive

Table 1: Comparison of Q-Learning, Monte Carlo, and IE-GS in the Tower Environment.

their strategies based on returns observed at the end of episodes. Learning is less sensitive to short-term fluctuations but may still favor self-interested actions.
These baselines provide a contrast to IE-GS, which incorporates trust-based learning, memory mechanisms, and adaptive decision-making.

4.1.2 Impact of Initial Conditions on Learning

The initial conditions of an agent-based learning system can significantly influence the emergent behavior and the long-term evolution of cooperative strategies.
To investigate the impact of initial conditions on learning, we conduct a forced greedy initialization experiment. In this experiment, all agents are constrained to follow a greedy consumption strategy for a predetermined number of episodes, where they consume the maximum available resources up to a limit of 2 units per round. This setup simulates an extreme competition scenario where agents prioritize immediate self-interest, preventing them from developing cooperative behaviors in the early stages.
The primary objective of this experiment is to determine whether agents can recover cooperation after being exposed to enforced greedy behavior. Additionally, we analyze how the duration of this greedy phase influences learning outcomes by varying the enforced greediness period to 200, 800, and 2000 episodes. These settings allow us to examine whether prolonged exposure to non-cooperative behaviors makes it increasingly difficult for agents to transition toward sustainable resource-sharing strategies.

4.1.3 Sensitivity to Hyperparameters

To systematically assess the effects of different learning dynamics, several key hyperparameters are varied across experiments. One of the most critical parameters is the trust update learning rate (α), which dictates how quickly agents adjust their trust values based on observed interactions.
Another essential hyperparameter is the trust weighting factor (β), which determines the sensitivity of trust updates to an agent's hunger state
Exploration strategies are also tested, as they significantly influence how agents balance novel strategy discovery and exploitation of learned behaviors. Three exploration schedules are examined. The extreme high exploration strategy maintains a persistently high exploration rate, ensuring that agents continuously experiment with different strategies. The moderate exploration strategy gradually shifts from exploration to exploitation following a decay schedule, while the minimal exploration strategy minimizes randomness from the start, allowing agents to refine their behaviors early on.

All experiments run for 5000 training episodes, with performance metrics averaged over multiple independent runs to ensure statistical reliability. Training is conducted using a parallelized simulation framework to efficiently explore different hyperparameter configurations.

4.1.4 Evaluation Metrics

To evaluate the performance and emergent behaviors of our multi-agent system, we employ four key metrics:
The success rate quantifies the proportion of agents exhibiting moderate or minimal consumption behavior:

$$\text{Success Rate} = \frac{\text{Number of Agents with Action} \leq 1}{\text{Total Number of Agents}}$$

This metric serves as a primary indicator of collective self-regulation, measuring the agents' ability to avoid overconsumption. A high success rate suggests that agents have learned to regulate their resource consumption, either through complete abstention (Action $= 0$) or moderate consumption (Action $= 1$), which is crucial for sustainable resource management in multi-agent systems.
The cooperation level specifically measures the proportion of agents choosing optimal moderate consumption:

$$\text{Cooperation Level} = \frac{\text{Number of Agents with Action} = 1}{\text{Total Number of Agents}}$$

Unlike the success rate, this metric focuses exclusively on agents that select the socially optimal action of moderate consumption (Action $= 1$). This distinction is important as it differentiates between passive non-consumption and active cooperation. A high cooperation level indicates that agents have learned to balance individual needs with collective sustainability.
The fairness index evaluates the equality of resource distribution using a modified Gini coefficient:

$$\text{Fairness Index} = 1 - G$$

where G is the Gini coefficient calculated over the episode's cumulative consumption:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n \sum_{i=1}^{n} x_i}$$

Here, x_i represents the total resources consumed by agent i during an episode, and n is the number of agents. This metric ranges from 0 to 1, where 1 indicates perfect equality in resource distribution. The fairness index is particularly important in multi-agent systems as it reflects the system's ability to maintain equitable resource allocation.
These metrics collectively provide a comprehensive evaluation framework for analyzing both individual agent behavior and emergent system-level properties in our multi-agent resource management scenario. They capture different aspects of the system's performance: individual decision-making (Success Rate), cooperative behavior (Cooperation Level) and distributive justice (Fairness Index).

4.2 Experiments in the Iterated Prisoner's Dilemma

The Iterated Prisoner's Dilemma (IPD) is a well-established game theory model that captures the essence of decision-making in environments where agents must balance cooperation and

competition. The key challenge in IPD is that short-term incentives often conflict with long-term benefits—defecting may yield immediate rewards, but cooperation maximizes long-term gains if mutual trust is established.

IE-GS is fundamentally designed as a trust-based decision-making algorithm, making IPD an ideal environment to evaluate its effectiveness. Traditional strategies like Tit-for-Tat (TFT) rely on direct reciprocity—cooperating initially and mirroring the opponent's previous action—while IE-GS instead dynamically adjusts trust levels based on long-term behavioral trends. This difference raises key questions:

Can IE-GS outperform Tit-for-Tat by achieving higher cooperation rates and long-term rewards?

Is IE-GS resilient against purely exploitative strategies like Always Defect?

How does IE-GS respond to deceptive opponents who change their behavior over time?

To answer these questions, we conducted a two-stage evaluation. First, we compared IE-GS against baseline IPD strategies to determine its strengths and weaknesses in standard settings. Second, we introduced Delayed-Defect and Delayed-Cooperate strategies to examine whether IE-GS's trust mechanism can be exploited or recovered in non-stationary environments.

5 Results

In this section, we present the experimental results evaluating the effectiveness of our proposed Imitation Evolutionary Game Strategy (IE-GS) compared to traditional reinforcement learning and game-theoretic strategies. The Baseline Algorithm Performance subsection analyzes how IE-GS performs in a resource-limited environment and compares it against standard reinforcement learning approaches such as Q-Learning and Monte Carlo. The Effect of Greedy Initialization subsection investigates how agents recover from an enforced phase of selfish behavior, examining whether cooperative strategies can emerge despite unfavorable starting conditions. We then extend this analysis in the Impact of Different Greedy Initialization Durations on Learning, exploring whether prolonged exposure to greediness impairs the ability to regain cooperation. Next, the Hyperparameter Sensitivity subsection delves into the effects of key trust-related parameters, including the trust update learning rate and trust adjustment parameter, assessing their impact on cooperation level and Fairness index. Furthermore, the Effect of Exploration Strategies subsection examines different exploration settings to determine the optimal balance between exploration and exploitation for achieving sustainable cooperation. Finally, in the Iterated Prisoner's Dilemma subsection, we assess IE-GS's performance in a classic game-theoretic environment, comparing it against well-established strategies such as Tit-for-Tat, Always Defect, and Random, followed by an evaluation against Delayed-Defect and Delayed-Cooperate strategies to test whether trust-based cooperation mechanisms can withstand dynamic opponent behavior. Together, these experiments provide a comprehensive analysis of IE-GS's robustness, adaptability, and long-term strategic viability in multi-agent cooperative settings.

5.1 Baseline Algorithm Performance

One of the fundamental challenges in reinforcement learning is the ability to learn cooperative strategies in environments where there are no explicit rewards for cooperation. This issue becomes particularly evident in our experimental setting, where agents must learn to moderate their resource consumption without receiving a direct reward for doing so. Figure 10 illustrates the success rate of three different learning strategies—Q-Learning, Monte Carlo, and our proposed Imitation Evolutionary Game Strategy (IE-GS).

The success rate in this context is defined as the proportion of episodes in which agents successfully moderate their consumption such that food is preserved for lower-level agents, ensuring long-term sustainability. Unlike traditional reinforcement learning problems where an explicit reward function directly guides optimal policy learning, our environment requires agents to infer the benefits of cooperation from indirect signals and interactions.

As shown in Figure 10, Q-Learning and Monte Carlo both fail to achieve stable success rates and exhibit persistent fluctuations. Q-Learning (blue line) oscillates between 30% and 70% success, while Monte Carlo (green line) stabilizes at an even lower range. In contrast, the IE-GS algorithm (red line) maintains a near-constant success rate close to 90% throughout training.

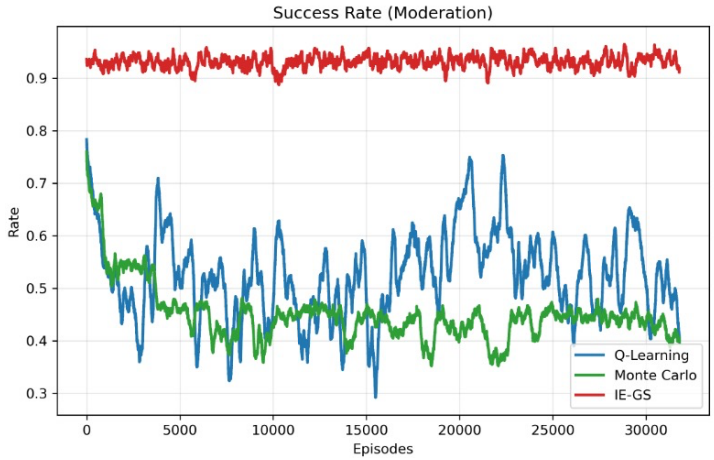


Figure 10: Comparison of success rates among Q-Learning, Monte Carlo

The failure of Q-Learning and Monte Carlo in this task can be attributed to their reliance on direct reinforcement signals. In standard reinforcement learning, agents learn by associating actions with rewards, updating their value functions accordingly. However, in our environment, agents do not receive immediate rewards for moderation. Instead, the consequences of over-consumption only manifest in later timesteps, making it difficult for traditional algorithms to propagate meaningful learning signals.

Q-Learning, which relies on bootstrapped value updates, struggles because it continuously updates its estimates based on immediate feedback, leading to a reinforcement loop where no clear moderation strategy emerges. Instead, agents remain stuck in a state of persistent exploration, oscillating between different consumption levels without converging to an optimal policy.

Monte Carlo methods, which update policies based on episode-level returns, also fail in this setting. Because successful moderation only produces indirect benefits rather than immediate reinforcement, Monte Carlo updates do not consistently reinforce cooperative behaviors. As a result, the learned policy remains unstable, and agents continue to explore inconsistently.

Unlike traditional reinforcement learning, IE-GS introduces a trust-based cooperation mechanism, adaptive exploration-exploitation balancing, and memory-enhanced learning, which collectively enable stable cooperation.

First, the trust mechanism ensures that agents do not need explicit rewards to learn cooperation. Instead of relying solely on immediate reinforcement, each agent maintains a trust value for neighboring agents. This value is updated based on observed behaviors, allowing agents to establish implicit cooperation over time. Once a high-trust equilibrium is reached, agents maintain stable moderation strategies, explaining the flat success rate in results.

Second, the adaptive exploration-exploitation strategy prevents agents from getting stuck in endless exploration. Traditional RL methods continue exploring suboptimal behaviors because they receive no direct feedback indicating which strategy is superior. In contrast, IE-GS dynamically reduces exploration as trust values stabilize, ensuring that once an agent learns a successful moderation strategy, it retains it instead of reverting to random behaviors.

Lastly, memory-enhanced learning allows IE-GS agents to retain and reinforce successful strategies. Unlike Q-Learning, which updates based only on recent experiences, or Monte Carlo, which relies on episodic returns, IE-GS incorporates both short-term and long-term memory. The short-term memory captures recent strategic shifts, while long-term memory ensures the preservation of learned policies. This prevents the instability observed in other methods and

results in the near-perfect success rate.
While the success rate of IE-GS remains consistently high, one intriguing observation is the lack of visible learning dynamics in its performance curve. Unlike Q-Learning and Monte Carlo, which exhibit fluctuations indicative of ongoing exploration and adaptation, the IE-GS success rate remains nearly constant from the early episodes onward.
There are several possible explanations for this phenomenon. One hypothesis is that IE-GS quickly discovers an effective cooperative strategy and maintains it due to its trust-based reinforcement mechanism. Once agents establish a high-trust equilibrium, there is no incentive to deviate from their learned behaviors, leading to a nearly flat performance curve. This would suggest that IE-GS is capable of achieving rapid convergence to an optimal policy under favorable conditions.
Another possible explanation is that the trust mechanism may suppress learning variability by reinforcing early interactions. If initial trust values and early behaviors lead to cooperation, the algorithm stabilizes before any significant exploration occurs. This raises an important question: what happens if the agents do not start with favorable conditions? If early interactions establish low trust values or uncooperative behaviors, would the system still be able to recover and achieve cooperation?
To further investigate this, we conduct a bad initialization experiment in coming subsection. This experiment explores scenarios where agents begin with distrust, suboptimal consumption behaviors, or erratic decision-making. By examining the adaptability of IE-GS under these conditions, we aim to determine whether the algorithm is robust enough to recover from poor initial states and still achieve long-term cooperation.

5.2 Effect of Greedy Initialization

In this experiment, all agents were subjected to a forced greedy strategy for the first 200 episodes, where they consumed food whenever available, up to a maximum of 2 units per round. This setup simulates an extreme resource competition scenario, preventing agents from developing cooperative behaviors during the early stages. The key research question is: After experiencing enforced greedy behavior, can agents recover cooperation, and can the system return to a stable state?

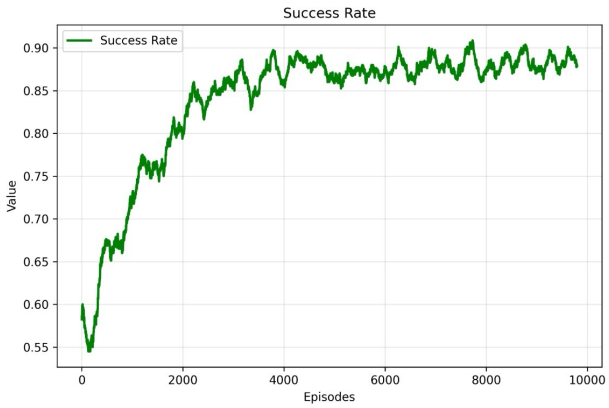
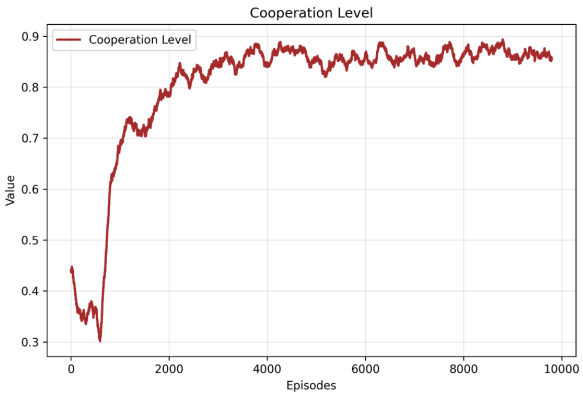


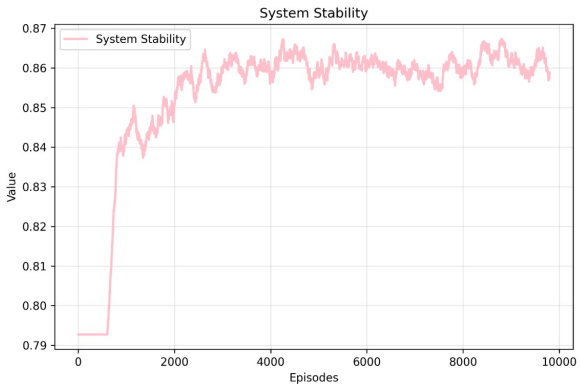
Figure 11: Success Rate Recovery After Greedy Initialization

The success rate measures whether agents adopt a moderate consumption strategy, defined as the proportion of agents choosing to consume less than 2 units of food per round. Due to the

enforced greedy strategy in the first 200 episodes, the success rate shown in Fig11 remained extremely low, as agents prioritized maximizing their immediate resource intake. However, once learning commenced, the success rate increased rapidly, reaching approximately 90% around 3000 episodes and stabilizing thereafter. This result indicates that even after an initial phase of extreme selfish behavior, agents can still learn to develop stable cooperative strategies.
Cooperation level is defined similarly to the success rate as the proportion of agents selecting moderate food consumption (1 unit per round). As shown in Fig12a, cooperation levels sharply declined and reached as low as 30% around episode 100 during the first 200 episodes, , reflecting the widespread adoption of a non-cooperative greedy strategy. However, once learning was enabled, cooperation recovered relatively quickly, surpassing 80% by episode 2000 and eventually stabilizing. This suggests that despite the initial state of extreme non-cooperation, agents were able to adjust their strategies and gradually form effective cooperative behaviors through learning.



(a) Cooperation Level Recovery from Forced Greedy Behavior



(b) System Stability Improvement After Initial Greed

Figure 12: Comparison of cooperation level recovery and system stability improvement after initial greed

System stability quantifies the variance in agent hunger levels, indicating whether resources are distributed evenly among agents. Higher stability signifies a more balanced system where hunger levels are relatively uniform. As shown in Fig12b, hunger level variance was extremely high during the first 200 greedy episodes, resulting in very low stability. However, stability increased rapidly after the learning phase began, reaching a steady state of approximately 0.86 around 3000 episodes. This trend suggests that even after an initial phase of intense resource competition, agents can gradually learn to distribute resources more equitably, stabilizing the system.

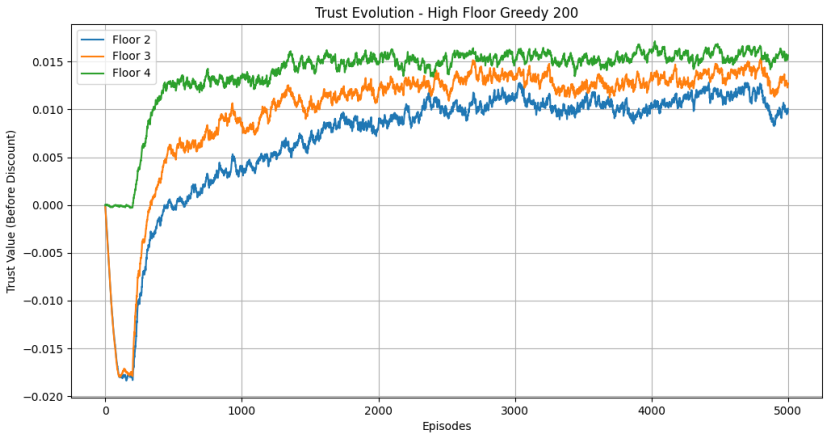


Figure 13: The Evolution of Trust in Greedy-200 Experiment

An interesting observation from Fig13 illustrates the evolution of trust between different floors and their immediate upper floors. Notably, higher floors tend to recover trust more quickly and, in some cases, are barely affected by the enforced greedy phase. In contrast, lower floors experience a more significant negative impact, with slower trust recovery and lower final trust values at convergence. This suggests that position-based resource access asymmetry influences trust dynamics, where agents with greater initial resource availability are more resilient to disruptions in cooperation.

Overall, this experiment reveals that even after an enforced phase of extreme greed, agents can progressively recover cooperation and develop stable resource-sharing mechanisms. Despite being forced into a non-cooperative state for 200 episodes, they adapted their strategies during learning, achieving high cooperation levels (above 80%) and improved fairness (lower Gini coefficients). System stability also reached equilibrium around 3000 episodes, suggesting that agents can establish sustainable resource distribution patterns despite early-stage competition. However, it is important to note that the recovery process requires several thousand episodes to reach stability, indicating that initial strategy biases can have long-lasting effects on learning. A natural follow-up question is: if the forced greediness lasts longer, will the agent still be able to recover? Does the speed of recovery decrease as the greediness time increases?

To further investigate this question, we will explore the impact of different forced greediness durations on the agent's learning ability in the next experiment to analyze whether extreme priming may lead to permanent policy deviations or whether the agent still has the ability to recover.

5.3 Impact of Different Greedy Initialization Durations on Learning

In the previous experiment, agents that underwent a short period of forced greediness (200 episodes) were still able to recover cooperation and gradually establish a fair resource allocation strategy. However, a key question remains: how does the duration of greedy initialization affect agents' ability to learn and ultimately restore cooperation? To explore this, we extend the greedy initialization period to three different settings: 200, 800, and 2000 episodes, and examine the learning process under these conditions. The experimental results in Fig 14, Fig 15a, and Fig 15b demonstrate that while all experimental groups ultimately achieved a high success rate and converged to stable cooperation, the duration of the greedy phase affected the speed

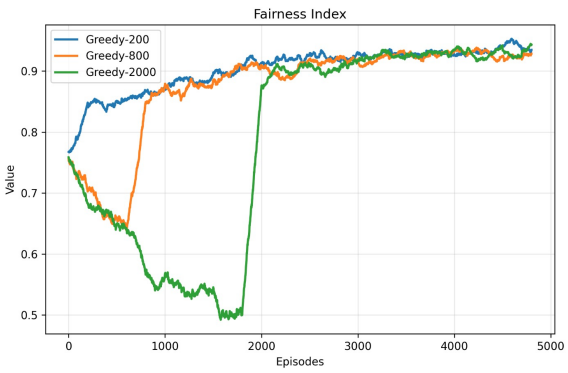
Figure 14: Impact of Different Greedy Initialization Durations-Success Rate

of recovery. In the early stages, agents with shorter greedy periods adapted faster and exhibited cooperative behavior sooner. In contrast, agents exposed to longer greedy phases experienced slower recovery but still managed to reach comparable levels of cooperation after approximately 4000 episodes. This suggests that although extended greediness temporarily reinforces self-interested strategies, it does not permanently prevent agents from learning cooperative behavior.

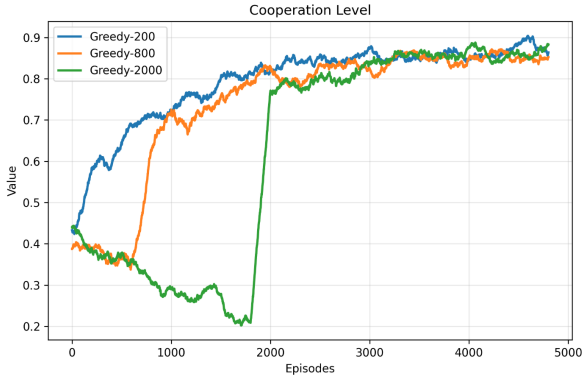
A similar trend is observed in the fairness index, which measures the balance of resource distribution among agents. This index is derived from the Gini coefficient (G), where higher values indicate greater inequality. The fairness index is defined as:

$$\text{Fairness Index} = 1 - G \tag{9}$$

where a lower fairness index implies highly unequal resource allocation.



(a) Impact of Different Greedy Initialization Durations - Fairness Index



(b) Impact of Different Greedy Initialization Durations - Cooperation Level

Figure 15: Comparison of fairness index and cooperation level under different greedy initialization durations.

In this experiment, the Greedy-200 group quickly restored fairness, reaching a fairness index above 0.9 within 1000 episodes and maintaining stability thereafter. The Greedy-800 group experienced a deeper decline but still recovered fairness by around 2000 episodes. The Greedy-2000 group exhibited a more prolonged drop in fairness and required approximately 3000 episodes to stabilize. However, after 4000 episodes, all groups reached a similar level of fairness,

suggesting that extended greedy phases only delay, rather than prevent, the development of equitable resource-sharing behaviors.

The recovery of cooperation level further supports these observations. Cooperation level is defined as the proportion of agents selecting moderate consumption (i.e., choosing action 1) in each episode, reflecting the extent to which agents coordinate their actions under resource constraints. Across different greedy initialization conditions, cooperation level followed a consistent pattern. The Greedy-200 group recovered rapidly, surpassing 0.8 within 1500 episodes and maintaining high stability. The Greedy-800 group showed a slightly delayed recovery, reaching similar levels around 2000 episodes. The Greedy-2000 group exhibited the slowest recovery, with cooperation levels remaining low (between 0.2 and 0.3) for nearly 2000 episodes before gradually improving. However, by 4000 episodes, cooperation levels across all groups became comparable, reinforcing the conclusion that while prolonged greediness slows adaptation, it does not fundamentally alter the system's ability to learn cooperation.

5.4 Hyperparameter Sensitivity

This section presents the impact of key hyperparameters on agent performance.

5.4.1 Effect of Trust Update Learning Rate α

In this experiment, as shown in Fig16 and Fig17, we analyze the impact of the trust update rate parameter α on the performance of the IE-GS algorithm in a multi-agent environment. The parameter α controls the rate at which trust values are updated; a higher α allows agents to adjust trust values more rapidly, while a lower α results in slower trust adaptation. An appropriate choice of α is crucial for balancing stability and adaptability, thereby facilitating the emergence of trust, cooperative behavior, and equitable resource distribution. To comprehensively evaluate the effect of α on agent decision-making and overall system performance, we focus on four key metrics: *Success Rate*, *Cooperation Level* and *Fairness Index*.

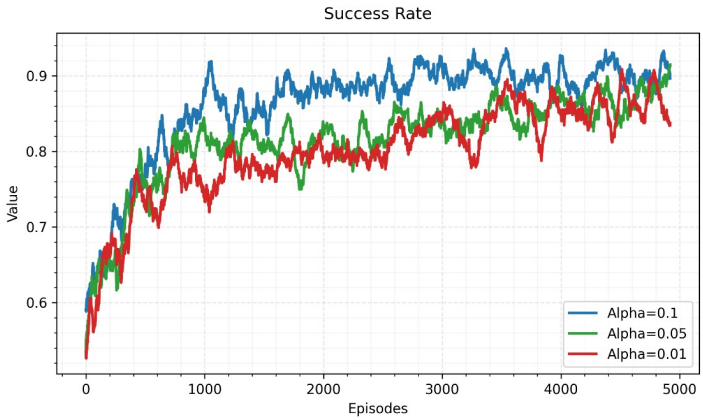
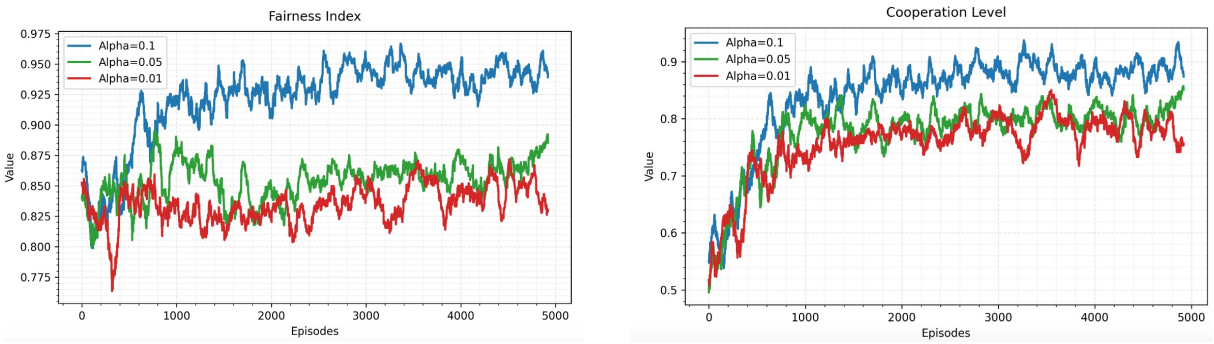


Figure 16: Success Rate under Different α Values.

The success rate measures the agents' ability to regulate resource consumption effectively, preventing excessive depletion and enhancing system sustainability. The results indicate that with a high trust update rate of $\alpha = 0.1$, the success rate rises rapidly and stabilizes above 90% after approximately 2000 episodes. This suggests that a high α enables agents to quickly

adjust their trust relationships and learn adaptive survival strategies earlier. In contrast, with a medium trust update rate of $\alpha = 0.05$, the success rate increases at a slower pace, eventually stabilizing around 80%, indicating that agents require a longer period to develop stable resource management strategies. For $\alpha = 0.01$, the success rate remains around 75% and exhibits significant fluctuations, suggesting that a low trust update rate hinders agents from quickly adapting to dynamic environmental changes, leading to persistent overconsumption behavior that negatively impacts overall survival rates.



(a) Fairness Index (b) Cooperation Level

Figure 17: Comparison of Fairness Index and Cooperation Level under Different α Values.

In terms of cooperation level, α directly affects the cooperative strategies learned by the agents. When $\alpha = 0.1$, the cooperation level stabilizes around 90%, indicating that agents not only avoid excessive consumption but also adopt moderate consumption behaviors, striking a balance between individual needs and collective sustainability. However, with $\alpha = 0.05$, the cooperation level decreases to approximately 80%, showing that while agents can still learn cooperative strategies, the slower trust update rate delays the convergence of cooperative behaviors. For $\alpha = 0.01$, the cooperation level remains below 75% and exhibits greater oscillations throughout training. This suggests that when trust updates are too slow, agents struggle to develop stable cooperative mechanisms within a limited number of training episodes, making them more susceptible to short-term self-interest decisions that hinder long-term cooperation. The fairness index results further corroborate these trends. When $\alpha = 0.1$, the fairness index reaches approximately 0.95, indicating a high degree of resource distribution equity among agents. This demonstrates that a high trust update rate facilitates the rapid formation of resource-sharing strategies, leading to more balanced resource allocation across the system. In contrast, with $\alpha = 0.05$, the fairness index stabilizes at around 0.85, suggesting that while resource distribution remains relatively fair, it is slightly less balanced compared to the $\alpha = 0.1$ case. This implies that some agents may still have an advantage in resource acquisition, causing minor inequities. When $\alpha = 0.01$, the fairness index remains low at approximately 0.82 and exhibits considerable fluctuations. This suggests that a slow trust update rate hinders the establishment of stable trust relationships, making it difficult for agents to fairly distribute resources. Consequently, some agents may accumulate significantly more resources than others, leading to a lower overall fairness index.

In summary, the choice of α has a direct impact on agent learning speed, cooperation level, and resource distribution fairness. A higher α (e.g., 0.1) enables faster trust adjustments, allowing agents to quickly adapt to environmental dynamics and learn stable cooperative strategies, ultimately improving success rates, promoting fair resource allocation, and enhancing system sustainability. In contrast, a lower α (e.g., 0.01) results in slower convergence, preventing

agents from forming stable cooperation behaviors and leading to lower success rates, reduced cooperation levels, and increased inequality in resource distribution. Additionally, in terms of convergence speed, $\alpha = 0.1$ allows the system to reach stability significantly faster, whereas lower values of α require much longer to achieve equilibrium. Therefore, under the current experimental conditions, $\alpha = 0.1$ is the optimal setting, as it facilitates trust evolution, accelerates cooperative strategy learning, and improves overall system performance.

5.4.2 Effect of Trust Adjustment Parameter β

In this experiment, as shown in Fig18 and Fig19, we analyze the impact of the trust adjustment parameter β on the performance of the IE-GS (Imitation Evolutionary Game Strategy) algorithm in a multi-agent environment. The parameter β determines the sensitivity of trust updates to an agent's hunger state, where a higher β causes agents to adjust their trust more aggressively in response to food shortages, while a lower β results in more gradual trust modifications. The choice of β directly affects how agents react to immediate resource deprivation, influencing the stability of cooperative behavior, fairness in resource distribution, and overall system success.

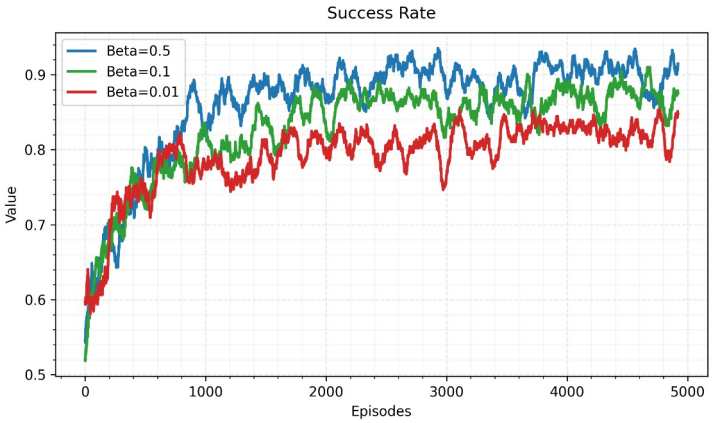
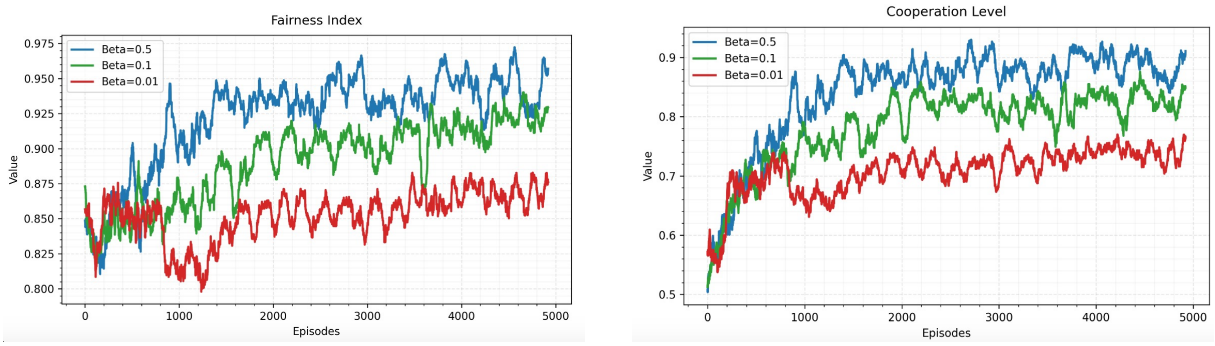


Figure 18: Success Rate for Different β Values.

Regarding the success rate, β significantly affects how quickly agents develop effective resource management strategies and their ultimate performance. When $\beta = 0.5$, the success rate increases rapidly and stabilizes above 90% after approximately 2000 episodes. This indicates that a stronger reaction to hunger penalties enables agents to adapt more quickly, discouraging exploitative behavior and accelerating the formation of cooperative strategies. When $\beta = 0.1$, the success rate also steadily improves, reaching 85%, suggesting that agents still learn effective resource management strategies, though their adaptation is slightly slower as they rely more on gradual trust adjustments. When $\beta = 0.01$, the success rate stabilizes at 80%, slightly lower than the higher β values. This suggests that while agents can still develop viable survival mechanisms, a lower hunger sensitivity results in more persistent reliance on past trust values, slowing down the adaptation process.

For cooperation level, a higher β leads to faster cooperation emergence, as agents more aggressively adjust trust values in response to food deprivation. When $\beta = 0.5$, the cooperation level ultimately reaches 90%, indicating that agents successfully learn a sustainable resource-sharing strategy and exhibit consistent moderate consumption behavior. When $\beta = 0.1$, the cooperation level stabilizes around 85%, demonstrating a strong cooperative tendency, although the



(a) Fairness Index for Different β Values (b) Cooperation Level for Different β Values

Figure 19: Comparison of Fairness Index and Cooperation Level under Different β Values.

reliance on more gradual trust updates results in slightly slower learning. When $\beta = 0.01$, the cooperation level remains around 80%, which, while lower than the higher β values, still shows that agents can develop cooperative strategies. However, since trust updates respond less aggressively to hunger penalties, cooperative behavior tends to be slightly less stable over time.

The fairness index further supports these findings by showing the impact of β on equitable resource distribution. When $\beta = 0.5$, the fairness index stabilizes at approximately 0.95, indicating that agents distribute resources more evenly, as the stronger response to hunger penalties discourages excessive resource hoarding. When $\beta = 0.1$, the fairness index reaches 0.90, suggesting that agents still develop fair resource allocation strategies, albeit with slightly lower fairness than in the $\beta = 0.5$ case, as adjustments are made more gradually. When $\beta = 0.01$, the fairness index stabilizes around 0.85, indicating that while agents still develop an equitable resource-sharing strategy, the weaker reaction to immediate hunger results in slightly more persistent inequalities in resource distribution.

In summary, the choice of β significantly influences the stability of agent learning, cooperation levels, and resource fairness. A higher β (e.g., 0.5) enables agents to adjust trust more rapidly in response to food shortages, leading to faster adaptation, higher success rates, and fairer resource distribution. Lower β values (e.g., 0.1 and 0.01) still enable cooperative strategies to emerge, but with relatively slower adaptation and slightly reduced long-term stability. In terms of convergence speed, $\beta = 0.5$ allows the system to reach stability more quickly, whereas lower β values require more time to establish a stable cooperation model. Therefore, under the current experimental settings, $\beta = 0.5$ emerges as the most effective choice, as it facilitates fast adaptation, enhances cooperation, and optimizes fairness and sustainability in the system, while $\beta = 0.1$ and $\beta = 0.01$ remain viable but exhibit relatively slower adaptation and slightly reduced long-term stability.

5.4.3 Effect of Exploration Strategies

Exploration determined how an agent balances between trying new actions and exploiting known rewarding behaviors. In this experiment, three different exploration strategies were evaluated, each defined by its initial exploration rate (ϵ), decay rate, and minimum exploration threshold.

The first strategy, Extreme High Exploration, maintains a consistently high exploration rate. It starts with an initial ϵ of 1.0 (meaning the agent initially explores completely randomly),

decays at an ultra-slow rate of 0.9999 per episode, and maintains a high minimum exploration level of 0.5. This strategy ensures that the agent continually explores throughout training, even in later stages.
The second strategy, Moderate Exploration, follows a balanced approach between exploration and exploitation. It begins with an initial ϵ of 0.9, decays at a moderate rate of 0.995, and stabilizes at a minimum exploration level of 0.1. This allows the agent to explore extensively in early training phases while gradually shifting towards more exploitative behavior.
The third strategy, Minimal Exploration, prioritizes exploitation from the beginning. It starts with a relatively low initial ϵ of 0.2, maintains a constant exploration rate (ϵ decay of 1.0), and has a very low minimum exploration threshold of 0.01. This setting ensures that the agent primarily exploits learned behaviors with minimal exploration over time.
The results, as illustrated in Figures 20 and 21, reveal key insights into the effects of these strategies. In the early training phase, both the moderate and minimal exploration strategies

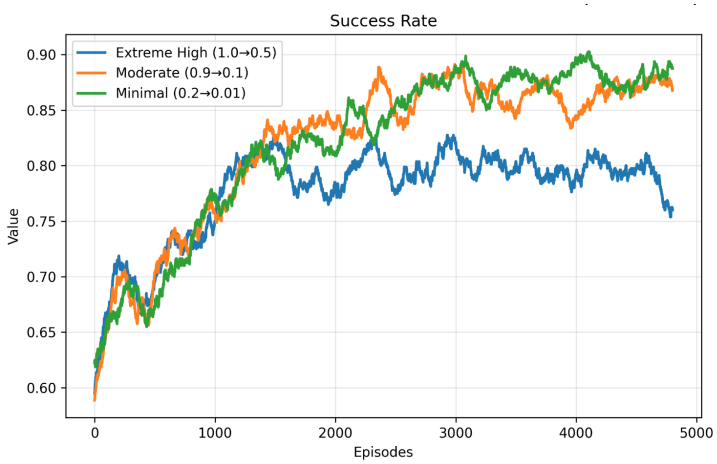
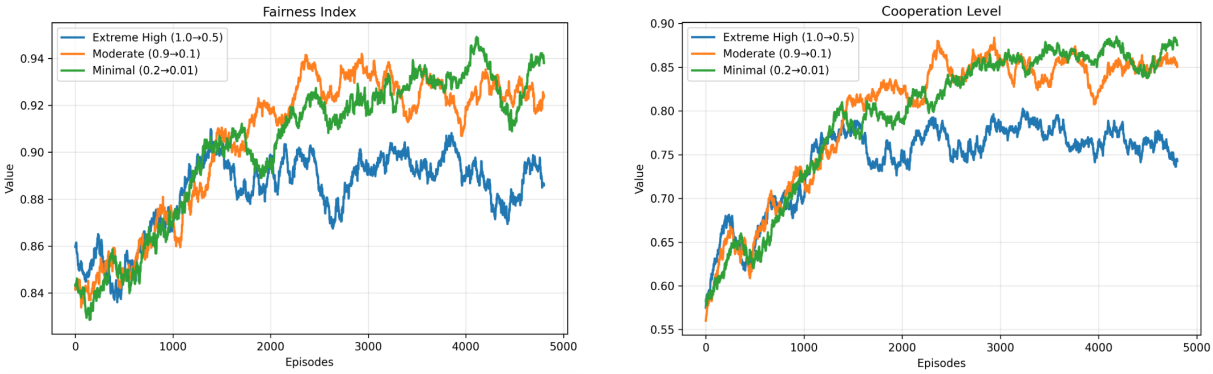


Figure 20: Exploration - Success Rate

show similar success rates. However, as training progresses, the moderate strategy temporarily outperforms the minimal exploration setting, reaching a higher peak before eventually converging to a performance level close to that of the minimal strategy. This suggests that a moderate level of early exploration provides some advantage in the mid-training phase but does not significantly alter final performance.
In contrast, the extreme high exploration strategy exhibits consistently lower performance across all metrics. The success rate, fairness index, and cooperation level of this strategy remain significantly below those of the other two strategies even after full convergence. This indicates that excessive exploration inhibits the agent's ability to solidify effective strategies, preventing it from efficiently learning and leveraging stable cooperative behavior.
Overall, these findings suggest that while exploration is necessary for initial learning, maintaining too high an exploration rate over time leads to suboptimal results. The moderate exploration strategy appears to provide a good balance, while excessive exploration hampers long-term performance.

5.5 Results in the Prisoner's Dilemma

The primary objective of this experiment is to evaluate the effectiveness of the IE-GS algorithm in the Iterated Prisoner's Dilemma (IPD) environment, particularly when competing against



(a) Exploration - Fairness Index

(b) Exploration - Cooperation Level

Figure 21: Comparison of fairness index and cooperation level under different exploration strategies.

various predefined opponent strategies. The experiment is divided into two parts: first, a comparison between IE-GS and baseline strategies, including Tit-for-Tat (TFT), Random, Always Cooperate, and Always Defect, to assess the advantages and weaknesses of IE-GS in standard game settings. Second, we introduce Delayed-Defect and Delayed-Cooperate strategies to examine whether IE-GS can maintain its performance in environments where opponents change their behavior over time.

Player A	**Player B**	**Payoff (A, B)**
Cooperate (C)	Cooperate (C)	(3, 3)
Cooperate (C)	Defect (D)	(0, 5)
Defect (D)	Cooperate (C)	(5, 0)
Defect (D)	Defect (D)	(1, 1)

Table 2: Payoff Matrix of the Iterated Prisoner's Dilemma. Each cell represents the rewards received by Player A and Player B based on their actions.

5.5.1 Baseline Strategy Comparison: Strengths and Weaknesses of IE-GS

In the baseline experiments, as shown in Table3 and Table4, IE-GS achieved the highest overall average score (2.44) and the highest win rate (66.44%), slightly outperforming Tit-for-Tat (2.44, 60.00%). This indicates that IE-GS possesses strong adaptability in most environments. However, further analysis reveals that IE-GS's advantage lies in its ability to adapt to dynamic environments, whereas it exhibits some weaknesses against fixed strategies. Table5 shows the detailed scores against each opponent.

Tit-for-Tat is a classic cooperative strategy based on a simple rule: cooperate in the first round, then mimic the opponent's previous action. Its strength lies in its stability, as it never initiates defection but immediately retaliates against defection. This makes it a strong contender in a stable environment. However, IE-GS surpasses Tit-for-Tat in several ways, demonstrating greater adaptability and strategic depth. One key advantage of IE-GS is its ability to establish long-term cooperation by dynamically adjusting its trust. Unlike Tit-for-Tat, which rigidly follows the opponent's last move, IE-GS is capable of tolerating occasional defections and making decisions based on broader behavioral trends. Against Tit-for-Tat, IE-GS maintained

Strategy	Win Rate (%)	Avg. Score
IE-GS	66.44	2.44
Random	31.24	2.27
Tit-for-Tat	40.00	2.07
Always Coop.	60.00	2.09
Always Defect	40.00	2.25

Table 3: Overall Performance Comparison Across Strategies

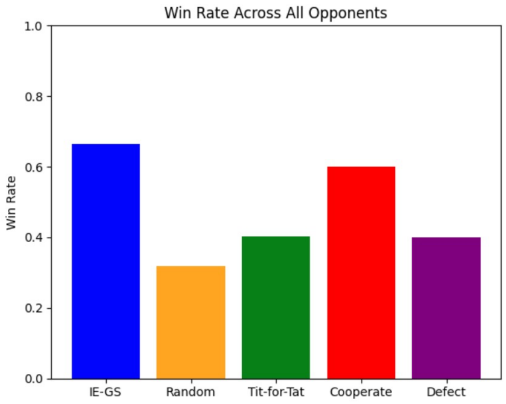


Table 4: Baseline Strategy Comparison - Win Rate

a cooperation rate of 97.58%, comparable to Tit-for-Tat's 97.60%, indicating that IE-GS successfully recognized TFT's cooperative tendency while maximizing its own payoff. Another

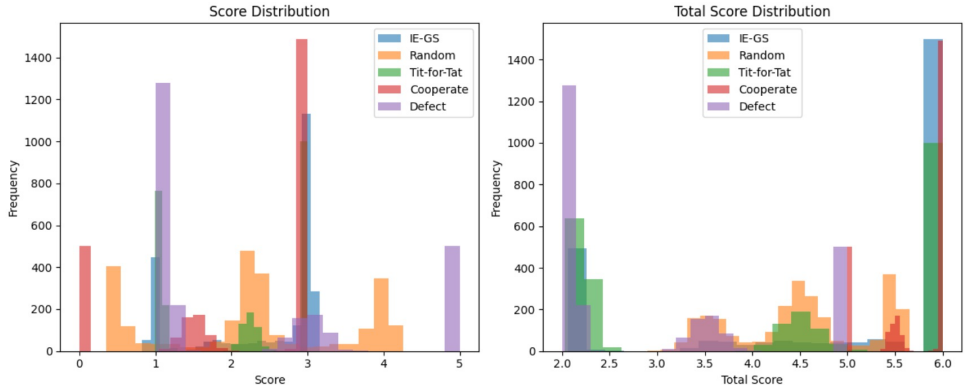


Figure 22: Score Distribution

notable advantage of IE-GS is its resilience to noise. In noisy environments where accidental defections occasionally occur, Tit-for-Tat may struggle, as it responds to every defection with immediate retaliation, potentially leading to unnecessary cycles of conflict. In contrast, IE-GS incorporates a trust mechanism that buffers against such fluctuations, allowing it to sustain cooperation even when occasional errors occur. This trust mechanism also plays a crucial role in preventing exploitation. Some opponents attempt to manipulate Tit-for-Tat by cooperating initially to gain trust before defecting at a crucial moment to maximize their own gains. Since Tit-for-Tat blindly mirrors its opponent, it is vulnerable to such deceptive strategies. IE-GS mitigates this risk by adjusting its trust gradually rather than reacting instantly, making it more robust against strategic exploitation.

While IE-GS demonstrates clear advantages in fostering cooperation and resisting manipulation, it faces challenges when dealing with highly uncooperative opponents such as Always Defect. Always Defect is the most aggressive strategy, never cooperating under any circumstances. Against this opponent, IE-GS achieved an average score of 0.96, which is slightly lower than Tit-for-Tat's 0.99. This performance gap can be attributed to IE-GS's initial trust, which leads to early losses. Since IE-GS starts with a trust value of 0.5, it attempts cooperation in the first few rounds, but because Always Defect never reciprocates, these early cooperative

attempts result in unnecessary losses. Additionally, IE-GS's trust adjustment is slower compared to Tit-for-Tat's immediate response. While Tit-for-Tat ceases cooperation immediately after encountering defection, IE-GS requires multiple rounds to lower its trust value, making it slower to adapt to opponents that consistently defect.

These results suggest that while IE-GS excels in cooperative environments, its slower adaptation to consistently uncooperative opponents remains a limitation. Future improvements could involve accelerating trust reduction when facing prolonged defection, allowing IE-GS to respond more efficiently to aggressive strategies while retaining its ability to foster cooperation in mixed environments.

Opponent	**IE-GS Avg Score**	**Cooperation Rate (%)**	**Avg Trust Value**
Against IE-GS			
IE-GS	2.98	97.55%	1.00
Random	2.35	50.12%	–
Tit-for-Tat	1.10	5.47%	–
Cooperate	2.93	100.00%	–
Defect	1.18	0.00%	–
Against Random			
IE-GS	2.27	48.23%	0.47
Random	2.26	50.30%	–
Tit-for-Tat	2.24	50.10%	–
Cooperate	1.50	100.00%	–
Defect	3.01	0.00%	–
Against Tit-for-Tat			
IE-GS	2.97	97.32%	1.00
Random	2.26	49.78%	–
Tit-for-Tat	3.00	100.00%	–
Cooperate	3.00	100.00%	–
Defect	1.04	0.00%	–
Against Cooperate			
IE-GS	3.05	97.57%	1.00
Random	4.00	50.22%	–
Tit-for-Tat	3.00	100.00%	–
Cooperate	3.00	100.00%	–
Defect	5.00	0.00%	–
Against Defect			
IE-GS	0.96	4.47%	0.00
Random	0.50	49.91%	–
Tit-for-Tat	0.99	1.00%	–
Cooperate	0.00	100.00%	–
Defect	1.00	0.00%	–

Table 5: Detailed Performance Against Each Opponent-Baseline Strategy

5.5.2 Challenges from Delayed Strategies: Can Trust Be Exploited?

While the baseline experiments demonstrate IE-GS's adaptability against standard strategies, real-world interactions often involve opponents whose behaviors change over time. To further investigate the robustness of IE-GS, we introduce two delayed-response strategies: Delayed-Defect and Delayed-Cooperate. Delayed-Defect starts by cooperating for several rounds, building trust with its opponent before suddenly switching to defection, making it an explicitly exploitative strategy designed to take advantage of trust-based algorithms. In contrast, Delayed-Cooperate begins with defection, potentially discouraging cooperation in the early stages, but later transitions into cooperative behavior, allowing us to examine whether IE-GS can recognize and recover trust after an initial period of uncooperative interactions. By testing against these delayed strategies, we aim to understand whether IE-GS can effectively detect and respond to delayed betrayal, as well as assess its ability to rebuild trust when faced with late-emerging cooperation.

In the Delayed Strategies experiments, as shown in Table6 and Fig7, IE-GS achieved the highest overall average score (2.49) and the highest win rate (66.52%),

Strategy	Win Rate (%)	Avg. Score
IE-GS	66.52	2.49
Tit-for-Tat	60.00	2.44
Delayed-Coop.	59.72	2.14
Random	31.88	2.26
Delayed-Defect	36.76	2.24

Table 6: Win Rates and Scores of Strategies

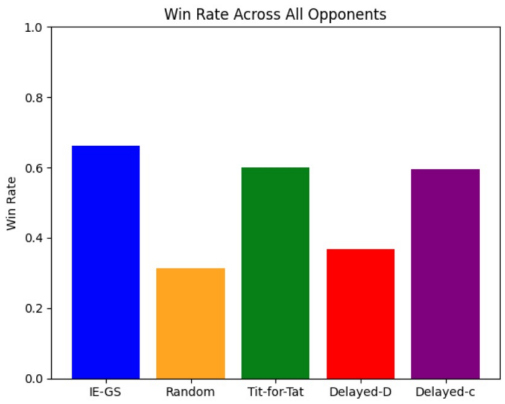


Table 7: Delayed Strategy Comparison - Win Rate

Table8 shows the detailed scores against each opponent.

Against Delayed-Defect, IE-GS achieved a score of 1.47, the highest among all tested strategies. This is a notable result because Delayed-Defect is explicitly designed to exploit trust-based agents by initially cooperating to establish trust before switching to defection. Despite this challenge, IE-GS still outperformed Tit-for-Tat (1.39) and other baselines, demonstrating its ability to mitigate long-term exploitation.

The key reason for IE-GS's superior performance against Delayed-Defect is its gradual trust decay mechanism. Unlike Tit-for-Tat, which always reacts immediately to defection, IE-GS reduces trust progressively over multiple rounds, allowing it to identify and respond to shifting behaviors. While IE-GS initially suffers losses due to its trusting nature, it ultimately stabilizes by reducing cooperation over time, preventing long-term exploitation.

That said, IE-GS's score against Delayed-Defect remains relatively low, indicating that while it limits damage, it does not fully eliminate the impact of delayed betrayal. A possible improvement would be implementing a short-term trust adjustment mechanism, where IE-GS detects sudden changes in opponent behavior and reacts faster.

Delayed-Cooperate presents a different challenge—it begins with defection, potentially damaging trust early on, but later shifts to cooperation. Against this opponent, IE-GS achieved

a high score of 2.75, outperforming Tit-for-Tat (2.61). This demonstrates IE-GS's ability to rebuild trust over time, an ability that Tit-for-Tat lacks.
When facing Delayed-Cooperate, IE-GS initially lowers trust due to early defections. This prevents it from being continuously exploited. As cooperation increases in later rounds, IE-GS raises its trust value, eventually stabilizing around an 80% cooperation rate.
This result highlights an important strength of IE-GS—it does not simply react to the previous action but rather adapts to long-term behavioral trends. This flexibility allows it to recover from early betrayals and re-establish cooperation, an advantage over more rigid strategies like Tit-for-Tat.

Opponent	**Strategy**	**Average Score**	**Cooperation Rate (%)**
IE-GS	IE-GS	2.98	97.54
	Random	2.29	50.29
	Tit-for-Tat	2.97	97.60
	Delayed-Defect	1.64	20.00
	Delayed-Cooperate	2.67	80.00
Random	IE-GS	2.25	50.44
	Random	2.26	50.06
	Tit-for-Tat	2.24	50.47
	Delayed-Defect	2.71	20.00
	Delayed-Cooperate	1.79	80.00
Tit-for-Tat	IE-GS	2.98	97.58
	Random	2.27	50.17
	Tit-for-Tat	3.00	100.00
	Delayed-Defect	1.44	20.00
	Delayed-Cooperate	2.61	80.00
Delayed-Defect	IE-GS	1.47	23.38
	Random	1.20	49.85
	Tit-for-Tat	1.39	21.00
	Delayed-Defect	1.40	20.00
	Delayed-Cooperate	1.00	80.00
Delayed-Cooperate	IE-GS	2.75	78.52
	Random	3.30	50.03
	Tit-for-Tat	2.61	80.00
	Delayed-Defect	4.00	20.00
	Delayed-Cooperate	2.60	80.00

Table 8: Detailed Performance of Each Strategy Against Delayed Strategy

6 Discussion

The results of this study provide evidence that cooperation can emerge in multi-agent reinforcement learning (MARL) environments without explicit cooperative rewards, driven purely by interaction-based trust mechanisms. Through extensive experimentation in the tower environment, we demonstrated that our proposed Imitation Evolutionary Game Strategy (IE-GS) facilitates cooperation through an evolving trust-based decision-making framework. Unlike conventional MARL algorithms such as Q-Learning and Monte Carlo, which struggle to develop stable cooperation, IE-GS enables agents to autonomously discover cooperative strategies through learned trust dynamics, even in the presence of initially self-interested behaviors. The implications of these findings extend beyond the specific environment studied here, highlighting the potential for trust-based learning as an alternative paradigm for promoting cooperation in decentralized, multi-agent systems.

A key observation from our experiments is that cooperation can emerge purely from local interactions without the need for explicit global incentives. Traditional MARL approaches often rely on carefully designed shared reward functions to align the interests of individual agents with a collective objective. However, in many real-world scenarios, defining such reward functions is impractical, as cooperation is often an emergent phenomenon rather than an explicitly prescribed goal. The findings from this study suggest that by leveraging implicit trust mechanisms, agents can develop cooperative behaviors naturally, even when their immediate incentives do not directly reward such behaviors. The ability of agents to learn trust dynamically, based on past observations and interactions, demonstrates a promising direction for future research in decentralized learning systems where global coordination is infeasible.

One of the critical factors influencing the emergence of cooperation in this study is the role of trust as an implicit coordination mechanism. Unlike reward-driven approaches, where cooperation is externally enforced through structured incentives, the trust model in IE-GS allows agents to modulate their strategies dynamically based on observed behaviors. The results indicate that trust serves as a stabilizing force in learning, reinforcing cooperative strategies while deterring exploitative behaviors. Agents that demonstrate restraint in consuming resources gain the trust of their peers, which in turn influences their own future decisions. This self-reinforcing feedback loop enables the gradual development of cooperation without requiring explicit communication or predefined cooperative objectives. The success of this mechanism suggests that trust-based learning could be a viable alternative to explicit reward shaping in environments where cooperative incentives cannot be predefined.

Another important aspect of our findings is the impact of initial conditions on long-term cooperation dynamics. The forced greedy initialization experiments demonstrate that the learning trajectory of agents is highly sensitive to early-stage interactions. Agents that begin with aggressive, self-serving behaviors take significantly longer to transition towards cooperation, as their early experiences reinforce competitive tendencies. However, the fact that cooperation was still able to emerge, even after extended periods of greedy behavior, underscores the robustness of the trust mechanism. This suggests that while initial conditions influence the rate of learning, they do not necessarily dictate the final outcome. Nevertheless, the prolonged adaptation period observed in the greedy initialization experiments raises important questions about how to accelerate the transition to cooperative behavior. One possible avenue for future research is the introduction of structured trust bootstrapping techniques that provide agents with an initial trust baseline, reducing the reliance on purely trial-and-error learning in the early stages.

Our findings indicate that hyperparameter selection plays a crucial role in determining the learning speed and final stability of cooperative strategies under the IE-GS framework. The trust update rate (α) directly influences how quickly agents adjust their trust values. A higher α enables agents to adapt rapidly to changes in trust, accelerating the convergence to cooperative strategies, while a lower α results in slower adaptation, making cooperation more difficult to establish and sustain. Similarly, the trust adjust factor (β) determines the sensitivity of trust updates to an agent's hunger state. A higher β causes agents to adjust their trust more aggressively in response to food shortages, while a lower β results in more gradual trust modifications. These observations highlight the importance of dynamic trust evolution in ensuring the success of cooperative learning. Proper hyperparameter tuning can balance learning efficiency and robustness in multi-agent systems.

Exploration strategies also significantly impact the effectiveness of the IE-GS mechanism. Our results show that excessive exploration inhibits the stabilization of cooperative strategies, as agents continually experiment with new behaviors instead of committing to established cooperation. Conversely, minimal exploration may prevent agents from adequately discovering the benefits of cooperation, trapping them in locally optimal but suboptimal selfish behaviors. The optimal strategy involves an initial phase of high exploration followed by gradual reduction, allowing agents to sufficiently explore the environment before settling into stable cooperative behavior. This finding supports the notion that balancing exploration and exploitation is a key factor in decentralized multi-agent learning. Future work could investigate adaptive exploration mechanisms that dynamically adjust exploration rates based on environmental complexity and interaction patterns to optimize learning efficiency.

Experiments in the Iterated Prisoner's Dilemma (IPD) further demonstrate the adaptability of IE-GS in complex game-theoretic settings. When facing cooperative strategies such as Tit-for-Tat and Always Cooperate, IE-GS successfully identifies cooperative trends and reinforces long-term trust relationships, resulting in a high cooperation rate. However, against deceptive strategies such as Delayed-Defect, where opponents initially cooperate before shifting to defection, IE-GS incurs early losses due to its initial high trust values. Nevertheless, over time, it learns to adjust trust downward, mitigating long-term exploitation. This suggests that dynamic trust adjustment mechanisms can help agents maintain adaptability in multi-agent systems with shifting opponent behaviors.

While IE-GS performs well in cooperative environments, its adaptation to persistently non-cooperative opponents, such as Always Defect, remains relatively slow. This is primarily due to the gradual trust decay mechanism, which initially encourages cooperation even when the opponent is continuously defecting, leading to prolonged losses. This observation suggests that in more adversarial environments, accelerating trust decay could improve response efficiency, allowing agents to recognize and counter persistent defection more effectively.

Furthermore, IE-GS exhibits strong trust recovery capabilities when facing Delayed-Cooperate strategies, where opponents initially defect before transitioning to cooperation. Unlike rigid strategies such as Tit-for-Tat, which immediately retaliate against defection and may struggle to rebuild cooperation, IE-GS adapts dynamically. It first lowers trust in response to early defections but later re-establishes cooperation as the opponent's behavior shifts. This flexibility enables it to outperform Tit-for-Tat in environments where agents' behaviors change over time.

Overall, our experimental results demonstrate that IE-GS enables stable cooperative learning in decentralized multi-agent environments without requiring explicit cooperative rewards. This finding is particularly significant for distributed intelligent systems where global incentive structures are impractical, and agents must rely on local interactions and trust mechanisms

to achieve long-term cooperation. However, our results also highlight that in the presence of persistent non-cooperation or highly deceptive strategies, IE-GS faces adaptation challenges. This suggests that future research should explore further optimizations to the trust update mechanism, enabling faster adaptation in adversarial settings. Additionally, future work could examine the generalization of IE-GS in more complex game-theoretic scenarios, such as mixed-strategy multi-agent games or heterogeneous agent systems, to assess its broader applicability.

7 Future Work

While this study has demonstrated the effectiveness of the Imitation Evolutionary Game Strategy (IE-GS) in facilitating cooperation in constrained multi-agent environments, several promising directions remain unexplored. Future research can expand upon our findings by addressing scalability to larger systems, incorporating heterogeneous agent strategies, and developing hierarchical and decentralized trust models to better reflect real-world multi-agent interactions.

7.1 Scaling to Larger Systems

One natural extension of our work is to explore the scalability of IE-GS in larger multi-agent environments. Our current experiments were conducted in a constrained setting, such as the four-level Tower Environment and small-scale Iterated Prisoner's Dilemma (IPD) interactions. However, real-world multi-agent systems often involve hundreds or even thousands of interacting agents. In such large-scale environments, emergent cooperation may be influenced by increased competition, communication constraints, and coordination complexity. Future research should investigate how IE-GS performs when scaled to a significantly larger number of agents, possibly incorporating grid-based spatial simulations, multi-tier hierarchical decision-making, and distributed reinforcement learning approaches to manage computational complexity.

A key challenge in scaling up is the increased difficulty in maintaining trust and cooperative behavior across a large population. As the number of agents grows, localized interactions and partial observability may play a more dominant role, leading to fragmented cooperation dynamics. Potential solutions include clustering methods where agents primarily interact with a subset of their peers, thereby reducing the cognitive load of tracking all other agents. This approach would allow agents to learn localized cooperation patterns, which could then propagate across the system through indirect interactions. Additionally, adaptive communication protocols may be needed to enable efficient information sharing among agents in large-scale environments without overwhelming computational resources.

7.2 Heterogeneous Agent Strategies

In our current study, all agents followed the same decision-making process governed by the IE-GS framework. However, in more complex environments, agents may employ heterogeneous strategies, including different learning mechanisms, varying levels of rationality, or distinct goals. Investigating the interplay between trust-based agents and agents using alternative strategies—such as self-interested reinforcement learning, adversarial learning, or probabilistic cooperation models—could provide deeper insights into the robustness and adaptability of IE-GS.

A particularly interesting avenue for future work is the inclusion of evolutionary dynamics in heterogeneous agent populations. By allowing agents to evolve their strategies over time, it would be possible to study whether trust-based cooperation remains a dominant behavior in the presence of strategy mutation and adaptation. Moreover, adversarial agents could be introduced to test whether trust-based agents can effectively identify and neutralize exploitative behaviors through adaptive trust decay mechanisms. This line of inquiry is particularly relevant for applications in cybersecurity, where malicious actors may attempt to manipulate trust networks to gain an unfair advantage.

Another consideration is the influence of cognitive biases and bounded rationality in heterogeneous agent systems. Unlike purely rational decision-makers, real-world agents (including humans) often exhibit biases such as loss aversion, over-trusting behaviors, or misinterpretation of intent. Future work could explore the impact of these cognitive limitations on the emergence of cooperation, testing whether IE-GS can be adapted to mimic human-like trust dynamics or counteract suboptimal decision-making caused by biases.

7.3 Hierarchical and Decentralized Trust Models

In our current implementation of IE-GS, trust is modeled as a single-layer, direct interaction mechanism, where agents update their trust values based solely on observed behaviors in one-to-one interactions. While effective in small-scale environments, this approach may not be sufficient for complex systems where trust relationships operate at multiple levels of abstraction. Future research could explore multi-level trust frameworks, where agents maintain different levels of trust toward individuals, subgroups, and the overall system.

A hierarchical trust model could introduce community-based trust structures, where agents track trust at both local and global levels. For example, in a large-scale multi-agent economy, agents may form cooperative clusters or alliances, maintaining high trust within their subgroup while still engaging in broader interactions with the global population. This structure could be inspired by federated learning or multi-agent reputation systems, where trust updates propagate hierarchically based on both direct experiences and aggregate community feedback.

Additionally, decentralized trust models may provide a more scalable and resilient approach to cooperation. Instead of relying on global trust scores, agents could engage in trust propagation via networked structures, where trust relationships are dynamically reinforced based on indirect interactions. This would enable agents to infer trustworthiness without direct interactions, reducing the reliance on costly trial-and-error learning. Techniques such as graph neural networks (GNNs) or blockchain-based trust mechanisms could be explored to maintain robust decentralized trust models in large, distributed agent populations.

Furthermore, adaptive trust thresholds could be implemented in hierarchical systems, allowing agents to contextually adjust their trust tolerance based on environmental conditions. For instance, in high-risk situations (e.g., limited resources, adversarial settings), agents might require stronger trust evidence before engaging in cooperative behavior, whereas in low-risk scenarios, they may adopt a more lenient trust policy to encourage faster cooperation formation.

8 Conclusion

This study explores the emergence of cooperation in multi-agent reinforcement learning (MARL) environments where explicit cooperative rewards are absent. By designing a structured environment inspired by resource allocation dilemmas and introducing the Imitation Evolutionary Game Strategy (IE-GS), we demonstrate that trust-based decision-making enables agents to develop cooperative behaviors solely through interaction-driven learning. Unlike traditional MARL approaches that rely on predefined shared rewards, IE-GS allows cooperation to emerge dynamically, driven by agents' evolving trust relationships and adaptive decision-making processes.

The findings of this study contribute to the broader understanding of multi-agent cooperation by showing that explicit cooperative incentives are not always necessary for achieving collective coordination. The ability of agents to develop cooperative behaviors purely through local trust interactions suggests that decentralized learning mechanisms can be effective in complex environments where global coordination is impractical. The trust mechanism introduced in this study provides a robust alternative to explicit reward shaping, enabling agents to autonomously regulate their behaviors and form sustainable cooperation strategies. By reinforcing cooperative behaviors and discouraging exploitative actions through dynamic trust updates, IE-GS demonstrates that cooperation can emerge as a natural outcome of repeated interactions, even in highly competitive settings.

Empirical results confirm that IE-GS significantly outperforms traditional MARL algorithms such as Q-Learning and Monte Carlo in terms of fostering long-term cooperation, improving resource fairness, and enhancing system stability. The success rate of cooperation in IE-GS remains consistently high, and agents exhibit improved fairness in resource distribution compared to purely self-interested baselines. Moreover, the ability of agents to recover cooperation after an extended period of greedy behavior highlights the resilience of the trust-driven learning framework. These results provide strong evidence that trust-based MARL can be an effective paradigm for multi-agent coordination, particularly in environments where designing explicit cooperative incentives is challenging or infeasible.

Despite these promising findings, challenges remain in extending this approach to more complex, real-world scenarios. Future work should focus on scaling trust-based learning to environments with dynamic agent populations, where new agents may enter or exit the system over time. Additionally, exploring decentralized and hierarchical trust models could improve scalability, enabling cooperation in large-scale, distributed systems. Further research is also needed to investigate the application of IE-GS in heterogeneous agent environments, where different agents may have varying trust-building mechanisms and learning capabilities.

Overall, this study offers initial insights into the potential of trust-driven MARL as an alternative to explicit reward-based cooperation mechanisms. While the findings suggest that cooperation can emerge naturally through implicit trust signals, further research is needed to explore the broader applicability and limitations of this approach. By continuing to refine and extend these ideas, we hope to contribute to the development of more adaptive and self-organizing multi-agent systems in complex environments.

References

[1] Gronauer, S., and Diepold, K. (2022). Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, **55**, 895–943. Available at: `https://doi.org/10.1007/s10462-021-09996-w`.

[2] Galder Gaztelu-Urrutia. *The Platform*. Film. Basque Films, 2019.

[3] Du, Y., Leibo, J. Z., Islam, U., Willis, R., and Sunehag, P. (2023). A review of cooperation in multi-agent learning. *arXiv preprint arXiv:2312.05162*.

[4] Cai, H., Su, Y., and Huang, J. (2022). Cooperative control of multi-agent systems. *Cham, Switzerland: Springer Cham*. Springer.

[5] Wang, X., Zhao, C., Huang, T., Chakrabarti, P., and Kurths, J. (2023). Cooperative learning of multi-agent systems via reinforcement learning. *IEEE Transactions on Signal and Information Processing over Networks*, **9**, 13–23.

[6] Mushtaq, A., Haq, I. U., Sarwar, M. A., Khan, A., Khalil, W., and Mughal, M. A. (2023). Multi-agent reinforcement learning for traffic flow management of autonomous vehicles. *Sensors*, **23**(5), 2373.

[7] Canese, L., Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M., and Spanò, S. (2021). Multi-agent reinforcement learning: A review of challenges and applications. *Applied Sciences*, **11**(11), 4948.

[8] Wang, J., Hong, Y., Wang, J., Xu, J., Tang, Y., Han, Q.-L., and Kurths, J. (2022). Cooperative and competitive multi-agent systems: From optimization to games. *IEEE/CAA Journal of Automatica Sinica*, **9**(5), 763–783.

[9] Barron, E. N. (2024). *Game theory: An introduction*. John Wiley & Sons.

[10] Patel, P. (2021). Modelling cooperation, competition, and equilibrium: The enduring relevance of game theory in shaping economic realities. *Social Science Chronicle*, **1**, 1–19.

[11] Amato, C. (2024). An introduction to centralized training for decentralized execution in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2409.03052*.

[12] Saifullah, M., Papakonstantinou, K. G., Andriotis, C. P., & Stoffels, S. M. (2024). Multi-agent deep reinforcement learning with centralized training and decentralized execution for transportation infrastructure management. *arXiv preprint arXiv:2401.12455*.

[13] Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2020). Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. Retrieved from `https://arxiv.org/abs/1706.02275`

[14] Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., & Graepel, T. (2017). Value-decomposition networks for cooperative multi-agent learning. Retrieved from `https://arxiv.org/abs/1706.05296`

[15] Zhou, Y., Liu, S., Qing, Y., Chen, K., Zheng, T., Huang, Y., Song, J., & Song, M. (2023). Is centralized training with decentralized execution framework centralized enough for MARL? *arXiv preprint arXiv:2305.17352*.

[16] Zhu, C., Dastani, M., & Wang, S. (2024). A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*, *38*(1), 4. Springer.

[17] Hausknecht, M. J. (2016). *Cooperation and communication in multiagent deep reinforcement learning* (Doctoral dissertation).

[18] Sukhbaatar, S., Szlam, A., & Fergus, R. (2016). Learning multiagent communication with backpropagation. *arXiv preprint arXiv:1605.07736*. Retrieved from `https://arxiv.org/abs/1605.07736`

[19] Foerster, J. N., Assael, Y. M., de Freitas, N., & Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. *arXiv preprint arXiv:1605.06676*. Retrieved from `https://arxiv.org/abs/1605.06676`

[20] Gao, Y., Li, D., Chen, X., & Zhu, J. (2023). Attention-based mechanisms for cognitive reinforcement learning. *Applied Sciences*, *13*(13), 7361. MDPI.

[21] Das, A., Gervet, T., Romoff, J., Batra, D., Parikh, D., Rabbat, M., & Pineau, J. (2019). Tarmac: Targeted multi-agent communication. In *Proceedings of the International Conference on Machine Learning* (pp. 1538–1546). PMLR.

[22] Singh, A., Jain, T., & Sukhbaatar, S. (2018). Learning when to communicate at scale in multiagent cooperative and competitive tasks. *arXiv preprint arXiv:1812.09755*. Retrieved from `https://arxiv.org/abs/1812.09755`.

[23] Khan, R., Khan, N., & Ahmad, T. (2023). Communication in multi-agent reinforcement learning: A survey. *The Nucleus, 60*(2), 174–184.

[24] H. L. Fung, V.-A. Darvariu, S. Hailes, and M. Musolesi, "Trust-based Consensus in Multi-Agent Reinforcement Learning Systems," arXiv preprint arXiv:2205.12880, 2024. Available: https://arxiv.org/abs/2205.12880.

[25] J. Haoran, Y. Yuyu, H. Qiang, Z. Pengqian, and G. Ting, "Multi-Agent Trust Evaluation Model based on Reinforcement Learning," in *Proceedings of the 2021 8th International Conference on Dependable Systems and Their Applications (DSA)*, Yinchuan, China, 2021, pp. 608-613, doi: 10.1109/DSA52907.2021.00088.