Agent-based Model for Iterated Prisoner's Dilemma

Authors:

Shriran Chandran, Bjarni Kárason, Khai Phan

Complex Social Systems: Modeling Agents, Learning, and Games HS2023
December 19, 2023

Abstract

This project introduces an agent-based model (ABM) for the Iterated Prisoner's Dilemma with a network of players across multiple rounds. The model incorporates dynamic strategies and strategy-switching mechanisms, including, two knowledge-based strategy switchers – Adaptive (employing Bayesian reasoning) and SoftMajor (decision-making based on opponent history). A credit system is implemented for resource accumulation, while a community knowledge system allows players to gather information about others without direct visibility. The ABM explores the impact of diverse strategies and adaptive behavior in a network of agents over iterated interactions in various experiments, with a focus on emergent phenomena and cooperation strategies. This project contributes to the vast literature in the Prisoner's Dilemma and its extensions, helping to understand complex decision-making in multi-agent zero-sum systems, with potential applications in various social sciences.

1 Introduction

The prisoner's dilemma is a classic game theoretical thought experiment involving two rational agents who independently choose to cooperate with or betray (defect against) their opponent or partner. It illustrates how decisions made under collective rationality may differ from those made under individual rationality.
Albert W. Tucker formalized the game by framing the rewards in terms of prison sentences and subsequently named it the *prisoner's dilemma* [7], and is as follows:

> Two members of a criminal gang are arrested and imprisoned. Each prisoner is in solitary confinement with no means of speaking to or exchanging messages with the other. The police admit they don't have enough evidence to convict the pair on the principal charge. They plan to sentence both to a year in prison on a lesser charge. Simultaneously, the police offer each prisoner a Faustian bargain. If he testifies against his partner, he will go free while the partner will get three years in prison on the main charge. Oh, yes, there is a catch ... If *both* prisoners testify against each other, both will be sentenced to two years in jail. The prisoners are given a little time to think this over, but in no case may either learn what the other has decided until he has irrevocably made his decision. Each is informed that the other prisoner is being offered the very same deal. Each prisoner is concerned only with his own welfare — with minimizing his own prison sentence. [8]

The actions selected by the prisoners and their corresponding rewards can be conveniently illustrated in a payoff matrix, as depicted in Figure 1.

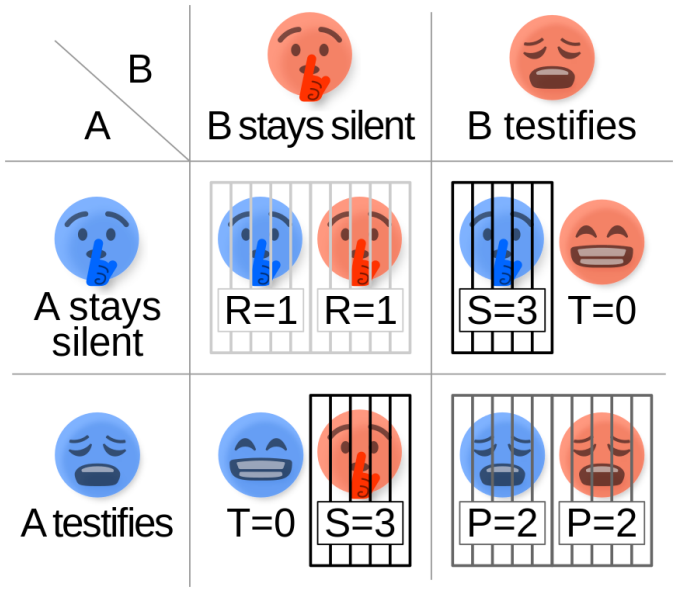


Figure 1: An example prisoner's dilemma payoff matrix (from Wikipedia)

Many prisoner's dilemma models share a similar outcome and reward structure, as described above, and can be summarized as follows:

1. If both individuals cooperate, they both receive a moderate reward.
2. If one individual cooperates while the other defects, the defector receives a high reward while the cooperator receives a low reward.

3. If both individuals defect, they both receive a low reward.

The dilemma arises from the fact that each individual must decide what to do without knowledge of the other's decision. Both individuals have an incentive to defect due to the potential for a high reward. However, if both individuals defect, they end up with a lower reward than if they had cooperated. Popular game-theory notions of equilibrium such as the *Nash equilibrium* formulated by John Nash in 1950 [6] fail to find the optimal solution, finding instead that both players should always defect.
The prisoner's dilemma serves as a valuable model for various real-world situations involving strategic behavior, wherein agents could potentially achieve greater rewards through cooperation or face adverse consequences for failing to do so. This scenario is particularly relevant in situations where agents are unable to communicate their decisions to one another.
A classic illustration of the prisoner's dilemma, demonstrating how individual self-interest can result in a suboptimal outcome for all, is the *tragedy of the commons* introduced by Garret Hardin [4]. In this context, the *commons* denotes a shared resource accessible to all members of a society. The tragedy unfolds when individuals, driven by their self-interest, deplete this shared resource. Hardin's example involves a common pasture where each herdsman tends to his cattle. The more a herdsman grazes his cattle, the less grazing space remains for the next herdsman's cattle. This cycle of overgrazing leads to the degradation of the pasture. If the herdsmen had collectively made decisions, prioritizing the well-being of the pasture over individual interests and grazing their cattle moderately, they could have averted the degradation of the shared resource.
The prisoner's dilemma has undergone extensive study [1] and finds applications in diverse fields, including economics, political theory, psychology, and biology. It has been found that simple strategies such as repeating the actions of your opponent often work surprisingly well, but adaptive strategies which consider the opponent's entire move history can unsurprisingly work better than non-adaptive strategies [5]. Recently, adaptive strategies that further consider the stability of the strategy have been proposed, arguing that players who do not frequently change their strategy appear more reliable [2]. The prisoner's dilemma offers valuable insights into decision-making, cooperation, and the inherent tension between individual and collective interests. Additionally, researchers have explored an iterative version of the dilemma, where the game is played repeatedly with the same set of players, further contributing to our understanding of the dynamics involved in repeated interactions.
In contrast to the classical version of the prisoner's dilemma, where players engage in a single game, the iterative version involves repeated gameplay. This extended format enables players to learn from each other's behaviors and adapt their strategies accordingly. Across multiple games, each player's final score is the sum of their scores accumulated over all the games. The primary objective for each player remains maximizing their score, but now with the added consideration that betraying an opponent may result in a loss of trust. The iterated prisoner's dilemma holds significance in various theories related to human cooperation and trust as it serves as a versatile model for understanding interactions requiring trust, such as corporate strategies, geopolitical issues, and more [3]. Classical strategies examined in the iterated prisoner's dilemma include: [Further content may be added here as needed.]

1. Always playing cooperation
2. Always playing defection
3. Randomly playing cooperation or defection

4. Cooperate until the opponent defects, then only play defection
5. Start by either cooperating or defecting, and then mirror the opponent
6. Mirror the opponent until he defects twice in a row, then only play defection
7. Mirror the opponent but retaliate by playing two defections for every defection the opponent plays

Both versions of the prisoner's dilemma can be extended to involve more than two players, providing a more nuanced model of trust evolution in a crowd setting. This extension requires defining player pairings and determining whether a player's strategy remains fixed throughout a round or can adapt after interacting with a single opponent. The multiplayer prisoner's dilemma is thus versatile in modeling various social, political, and economic scenarios.
For example, consider a labor market scenario where each trade union aims to negotiate wages exceeding the inflation rate — a strategy aligned with individual self-interest. However, if all trade unions collectively pursue such agreements, it could lead to a rise in the prices of goods and services, ultimately leaving everyone worse off than if they had exercised restraint.
In this report, we specifically explore a multiplayer and iterated prisoner's dilemma, investigating how the introduction of community knowledge, adaptive Bayesian players, and a credit system influences crowd dynamics.

2 Model Description

In the context of a multiplayer and iterated prisoner's dilemma, it is required to define the network connections between players, the stability of their strategies during a round, the sequence of games, and the information available for strategic adaptation.
Our rendition of the prisoner's dilemma closely mirrors the classical model but introduces innovative elements. Community knowledge allows an agent to inquire about its neighbors' past encounters with a specific opponent, gaining insights into historical behavior from other players. An adaptive Bayesian strategy uses this information to optimize the next move, maximizing rewards based on an opponent's historical behavior.
Our investigation explores the consequences of excluding consistently defecting players from subsequent games, aiming to understand its impact on the convergence and stability of the overall system.
For this project, we arrange the players on a 2-dimensional grid and consider two different connection models, the *lattice* and *extra lattice* networks, shown in Figure 2 and Figure 3 respectively.
In the lattice network, each player connects with those above, below, to the left, and to the right. In the extra lattice network, players are additionally connected to those diagonally adjacent. Every round involves players competing against all connected peers. The order of games in a round is randomized, and players can adjust their strategies between games.
An agent's behavior is influenced by two components: its strategy and its strategy switcher. The strategy dictates the agent's next move independently of an opponent's history. On the other hand, the strategy switcher determines the agent's next strategy, and it may consider an opponent's history. This allows the composition of simple strategies using the strategy switchers to create more intricate behavior. Our model incorporates classic prisoner's dilemma strategies and introduces two new strategy-switching behaviors: `SoftMajor` and `Adaptive`. The strategies we implemented include `AlwaysCooperate`, `AlwaysDefect`,

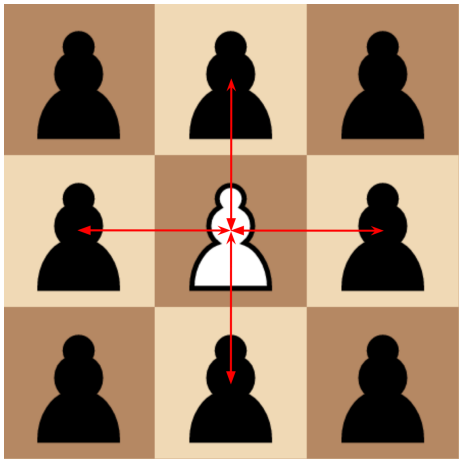


Figure 2: Connections of a single player in the lattice network

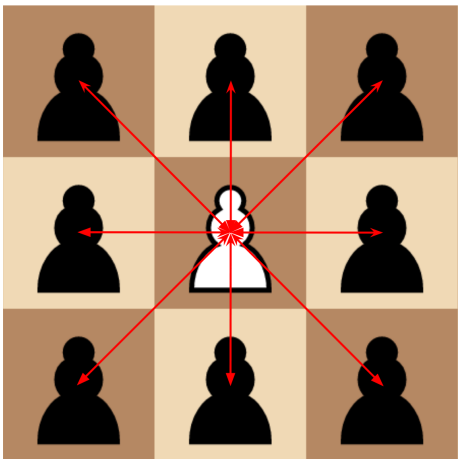


Figure 3: Connections of a single player in the extra lattice network

`ListRepeated`, `Alternating`, and `Random`. The strategy switchers we consider include `NOP`, `Random`, `TitForTat`, `CooperateUntilNDefectionsInARow`, `RetaliateWithTwoDefections`, `SoftMajor`, and `Adaptive`. The behavior of the strategies and strategy switchers is discussed in Section 3.

The players in our model have limited access to their opponent's history, being able to observe only the moves played against them. However, they can leverage community knowledge by consulting their neighbors about their past interactions with the same opponent. In this project, players store information solely about the moves played by their neighbors. When queried, neighbors provide truthful accounts of their experiences. Note that in the lattice network, there is no valuable community knowledge between an agent and opponent pair, as they have no mutual neighbors. In the extra lattice network, however, an agent and opponent pair have two common neighbors that can share their experiences, simulating a scenario where community knowledge is available. This mirrors real-life situations where parties cannot directly inquire about the trustworthiness of their counterparts but can seek insights from shared acquaintances or mutual experiences.

Additionally, we introduce a credit system where each player starts with a specific amount of credits. Players receive rewards for participating in games against opponents, with the reward determined by the prisoner's dilemma payoff matrix. Once a player's credits fall below zero, they are eliminated and no longer participate in subsequent games. Using these systems we investigate how penalizing repeat defectors by excluding them from the game impacts the overall convergence and stability of the system.

Our `Adaptive` and `SoftMajor` strategy switchers leverage community knowledge to enhance decision-making. The `Adaptive` strategy switcher employs Bayesian statistics to select a move expected to maximize the player's reward, considering the opponent's historical actions and a payoff matrix. This approach mirrors real-world decision-making, where individuals evaluate choices based on historical performance and beliefs about their counterpart's behavior. The `SoftMajor` strategy switcher is a simplified version of `Adaptive`, focusing solely on the opponent's historical cooperation rate. It cooperates if this rate exceeds a specified threshold.

We conducted experiments by playing games in both fixed and random orders. However, the fixed-order approach yielded undesired results. For instance, when populating a 2-dimensional grid with cooperating players, who turn to defection when someone defects against them, and introducing a single defecting player on the grid, we anticipated a ripple effect where defection spreads outward from the original defector. Surprisingly, this

phenomenon did not manifest consistently when games were played in a fixed order. To illustrate this, Figures 4 and 5 depict the number of rounds it takes for every player to adopt a defective strategy, based on the starting position of the defector. In Figure 4, games are played from left-to-right, top-to-bottom, whereas in Figure 5, games are played in a random order. The former demonstrates that the defector corrupts surrounding players more rapidly when placed in the top-left corner, where the defection flows downstream. Conversely, if placed in the bottom-right corner, where the defection must flow upstream, it takes considerably more rounds to corrupt every other player. In the latter case, where games are played in a random order, the spread of defection occurs equally fast in all directions. Consequently, we opt to exclusively consider playing games in a random order.

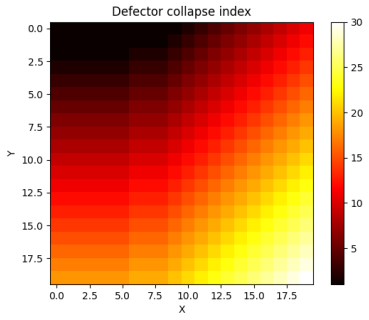


Figure 4: Defector Collapse Index with Fixed Order of Games

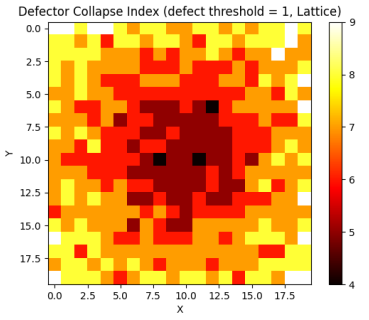


Figure 5: Defector Collapse Index with Random Order of Games

3 Implementation

3.1 Strategy

The `Strategy` class is an abstract base class (ABC) that defines the structure for player strategies. Subclasses must implement the `move` method. The `move` method gives a move following the strategy. These strategy implementations form the basis for the players' decision-making in the context of the simulation.

3.1.1 ListRepeatedStrategy

The `ListRepeatedStrategy` is a concrete strategy where a predefined list of moves `["C", "C", "D"]` is repeated. The strategy keeps track of its current move index and cycles through the list.

3.1.2 AlwaysCooperateStrategy

The `AlwaysCooperateStrategy` is a naively faithful strategy that always cooperates (chooses `"C"`).

3.1.3 AlwaysDefectStrategy

The `AlwaysDefectStrategy` is an evil strategy that always chooses to defect (chooses `"D"`).

3.1.4 AlternatingStrategy

The `AlternatingStrategy` alternates between cooperating (`"C"`) and defecting (`"D"`) based on the last move in its history.

3.1.5 RandomStrategy

The `RandomStrategy` makes a move randomly, either cooperating or defecting.

3.1.6 DeadStrategy

The `DeadStrategy` is a placeholder strategy that always returns `None`, created for "dead" players with negative credit as explained in Section 2.

3.2 StrategySwitcher

The Strategy Switcher module introduces a set of classes responsible for dynamically switching player strategies. Each strategy switcher class is a subclass of the abstract class `StrategySwitcher` Subclasses must implement the `check` method which is responsible for checking required criteria and changing the strategy of a player. These strategy switchers introduce dynamic decision-making to the player strategies.

3.2.1 NOPSwitcher

The `NOPSwitcher` (No Operation Switcher) is a strategy switcher that never triggers a strategy switch. It always returns `False` and `None`.

3.2.2 SoftMajor

The `SoftMajor` switcher assesses the opponent's cooperation rate and, based on a specified threshold, may switch the strategy to `AlwaysCooperateStrategy` or `AlwaysDefectStrategy`.

3.2.3 AdaptiveSwitcher

The `AdaptiveSwitcher`, a Bayesian strategy switcher, evaluates the opponent's cooperation rate and the payoff matrix to determine if switching to `AlwaysCooperateStrategy` or `AlwaysDefectStrategy` is more favorable.

3.2.4 RandomSwitcher

The `RandomSwitcher` selects a random strategy from the available strategy pool (excluding the `DeadStrategy`).

3.2.5 TitForTat

The `TitForTat` switcher responds to the opponent's previous move, switching between `AlwaysCooperateStrategy` and `AlwaysDefectStrategy` accordingly.

3.2.6 CooperateUntilNDefectionsInARow

The `CooperateUntilNDefectionsInARow` switcher cooperates until a certain threshold of `N` consecutive defections by the opponent is reached.

3.2.7 RetaliateWithTwoDefections

The `RetaliateWithTwoDefections` switcher retaliates with two consecutive defections for every defection by the opponent.

3.3 Player

The `Player` class models an agent in the prisoner's dilemma game. Each player is initialized with a strategy, strategy switcher, initial credit, and a list of neighbors. This class is designed to capture the dynamic nature of players in the prisoner's dilemma, allowing for to adapt strategy based on both direct interactions and observations of neighbors' behaviors.

1. **Initialization:** The player is initialized with the initial `strategy`, a `strategy_switcher`, the initial `credit`, and a list `neighbors` of neighboring players with whom interactions occur.
2. **Strategy Switching:** The player can adapt its strategy based on the conditions specified by the strategy switcher. Strategy switching occurs before each move also taking into account the opponent's behavior.
3. **Interaction with Neighbors:** Players keep track of their neighbors' move history with them and move tallies as a form of community knowledge. The `gather_opponent_tally` method collects information about an opponent's moves from both direct interactions and indirect information provided by neighbors.
4. **Credit Management:** The player's credit is updated based on the outcomes of interactions. If the player's credit falls below zero, it switches to a special `DeadStrategy` and stops participating in the game in any subsequent rounds.
5. **Playing Moves:** The `play` method executes the player's move based on its current strategy. The player's move is determined solely by the strategy.
6. **Utility Methods:** Methods such as `update_history`, `check_credit`, and `latest_move` among others provide functionalities for updating move history, checking credit status, and retrieving the latest move, providing a complete abstraction layer and interface to the player.

3.4 Network

The `Network` class serves as a representation of a social network within the context of the prisoner's dilemma game. It facilitates the creation of diverse network structures based on specified types and parameters.

1. **Initialization:** Networks are initialized with a few essential parameters. `N` and `M` denote the number of rows and columns in the network grid, respectively. `NType` defines the type of network structure, which can be Graph, Lattice, CLattice, or ExtraLattice. `density` is a parameter influencing edge generation in the network. `spread` is a parameter used specifically in generating edges for CLattice-type networks.
2. **Network Generation:** The `generate` method orchestrates the creation of the network based on the specified type and parameters. Each network type employs a distinct algorithm for edge generation, encompassing random graphs, lattices, and extended lattice structures as described earlier.
3. **Edge Retrieval:** The `getEdges` method enables the retrieval of a list of neighboring positions for a given position within the network. This functionality is crucial for initializing players in a network and facilitating interactions among network members.

4. **Printing Information:** The `printNet` method is a utility function that provides a concise summary of the network. It includes essential details such as the number of vertices and edges, and also prints an adjacency list, offering an overview of the network's structure.

3.5 Simulator

The `Simulator` class encapsulates the dynamics of the multiplayer and iterated prisoner's dilemma scenario. It creates a network based on its attributes and orchestrates the interactions between players, their strategies, and the evolving landscape of the game over multiple rounds.

1. **Attributes:** The `Simulator` class is equipped with several attributes that play pivotal roles in shaping the dynamics of the simulation. `n` and `m` are attributes that define the dimensions of the grid, determining the spatial organization of players within the simulation. The `pm` attribute holds the payoff matrix that governs the rewards and penalties associated with different outcomes in the prisoner's dilemma `population` and `cooperation_percentage` provide insights into the distribution of strategies within the population and the overall cooperation levels throughout the simulation. `status_matrix` and `strat_matrix` matrices serve as dynamic representations of the state of individual players, indicating their life status (alive or eliminated) and the strategies they employ. Finally, `network`, `grid`, and `pairs` define the network connecting players, the spatial arrangement of players on the grid, and the pairs of players engaged in the game.

2. **Initialization:** `init_network` and `init_grid` methods set the stage for the simulation. `init_network` configures the network structure based on user-defined parameters like `type`, `density`, and `spread`. Meanwhile, `init_grid` establishes the initial grid of players, forming the foundation for subsequent interactions.

3. **Game Simulation:** The `match` method orchestrates the simulation of a single game between two players. It encapsulates the logic of strategy execution, payoff calculation, and strategy adaptation based on the opponent's moves into a single interface. The `round` method extends the game to a full round, where pairs of players engage in matches. Importantly, the order of game execution is randomized, as explained before.

4. **Population Dynamics:** The `count_strategies` method provides a snapshot of the current distribution of strategies within the population. This insight is crucial for tracking the evolution of strategies over time.

5. **Dynamic Updates:** The `update_status` and `update_strat` methods dynamically reflect changes in player status (alive or eliminated) and strategy choices. These updates are vital for maintaining an accurate representation of the evolving state of the simulation.

6. **Simulation Execution:** Ultimately, the `simulate` method serves as the conductor of the entire simulation. It orchestrates multiple rounds, collects data on scores, strategies, and credits, updates the state of the attributes, and performs a comprehensive view of the evolving dynamics within the simulated environment.

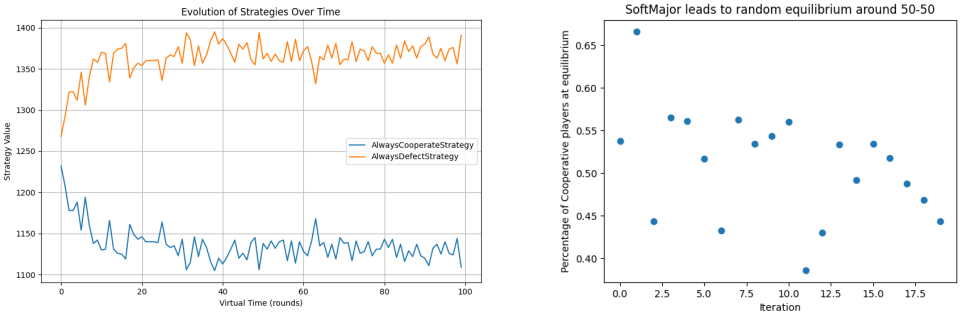
4 Experiments and Results

4.1 Experiment 1: Mimicry in a Neutral Population

This experiment serves as a straightforward illustration to delve into the concept of cooperative-defective symmetry within the framework of evolutionary game theory. By concentrating on a homogeneous majority-mimicking strategy, the experiment aims to illustrate how the population fluctuates around a balanced state of $50 - 50$ under symmetry. The simplicity of this setup makes it an ideal foundational benchmark for our simulation framework, providing a baseline for evaluating the effectiveness of our algorithm.
While the setup may seem rudimentary, it effectively captures a fundamental force driving multi-agent systems: conformity.
Specifically, the experimental setup is outlined as follows:

1. **Initial condition:** The initial population of the simulation is intentionally evenly divided between two types of strategists: cooperators and defectors. This balanced composition ensures an unbiased starting point.

2. **Adaptation Strategy:** The model is structured in a way that individuals within the simulation adapt by mimicking the strategies of the majority. This adaptation mechanism mirrors a fundamental aspect of social dynamics, wherein entities adjust their behaviors based on the prevailing strategies observed within their environment.



(a) Cooperative vs. Defective Populations: A Snapshot from an Iteration

(b) Final Percentage of Cooperative Players Across Iterations.

Figure 6: Random Fluctuations around Symmetric Cooperation-Defect Ratio.

As anticipated, in Figure 6, around 50% of the players display cooperative behavior, accompanied by minor random fluctuations in subsequent iterations. In the following experiment, with a slight modification to the setup, we intentionally break the symmetry. This alteration highlights that under the `SoftMajor` condition, the initially observed symmetry is inherently unstable.

4.2 Experiment 2: Minority-Induced Behavioral Cascade

In this experiment, we simulate a classic scenario illustrating a pathological situation where a small number of defectors have the potential to corrupt the system through pure imitative adaptation. Conversely, the scenario can be reversed by introducing a few purely cooperative players, steering the system towards cooperation. For this experiment, we choose to implement the latter scenario, although both scenarios are entirely analogous. The setup is conceptually intriguing and yields somewhat predictable outcomes. While

we anticipate the eventual corruption of the system with the introduction of a sufficient number of defectors or cooperators, the exact number required for this effect remains to be determined.
The setup and results illustrate scenarios in which a small number of influential players exert significant persuasive power in a majority that is imitative and neutral, often due to insufficient information. Examples include panic selling in financial markets, a cascade phenomenon triggered by a few pessimistic agents, and binary decisions made by the majority influenced by misinformation originating from a limited number of central sources.

1. **Initial condition:** The population is evenly divided between cooperators and defectors.
2. **Adaptation strategy:** Participants mimic the major strategies of their immediate neighbors, except for a few immutable cooperators.

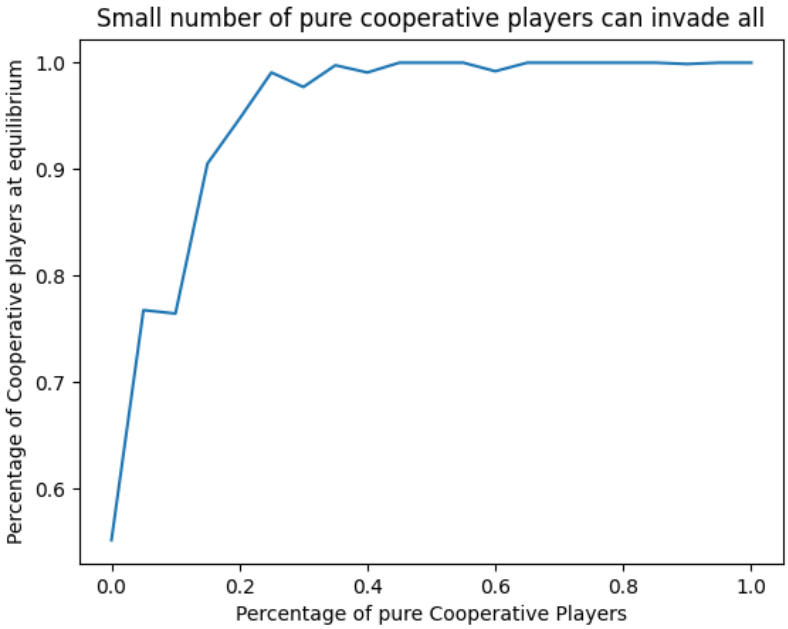


Figure 7: Equilibrium Cooperative Percentage vs. Initial Cooperative Players (`SoftMajor`).

Examining Figure 7, it becomes apparent that only a minimal number of initial cooperative players are needed to disrupt the symmetry. The significant impact of a small proportion of 'pure' (i.e., immutable) players is reminiscent of physical systems where symmetry spontaneously breaks due to instability, similar to a particle atop a hill or an inverted pendulum.

4.3 Experiment 3: Bayesian Thinking in a Neutral Population

Within the context of symmetric initial conditions, we explore scenarios where more sophisticated agents make decisions using Bayesian statistics, informed by their opponents' historical actions. Specifically, each agent calculates the expected reward $R(A)$ for each of their actions $A \in \{C, D\}$, representing cooperation and defection, respectively:

$$R(C) = R(C|D)P_{\text{op}}(D) + R(C|C)P_{\text{op}}(C), \tag{1}$$

$$R(D) = R(D|D)P_{\text{op}}(D) + R(D|C)P_{\text{op}}(C), \tag{2}$$

where $R(A|B)$ denotes the conditional reward from the payoff matrix, and $P_{\text{op}}(B)$ is the probability of the opponent's action B.
With the same initial conditions as the previous two experiments, we now delve into the impact of introducing logical reasoning, which has the potential to alter the cascade effects caused by immutable agents. In upcoming experiments, we will further explore how the authority, represented by the game designer, can favor Bayesian players through modifications to the payoff matrix. This form of reasoning before making binary decisions mirrors a crucial tactic in the buy/sell process, where players carefully weigh the risks and rewards, leveraging their knowledge of the historical performance of external companies.

1. **Initial conditions:** The population is evenly divided between cooperative and defective players.
2. **Strategy adaptation:** Strategies are adapted through Bayesian analysis of an opponent's available history. Decisions are influenced by a careful assessment of risks and rewards based on the information provided by the payoff matrix.

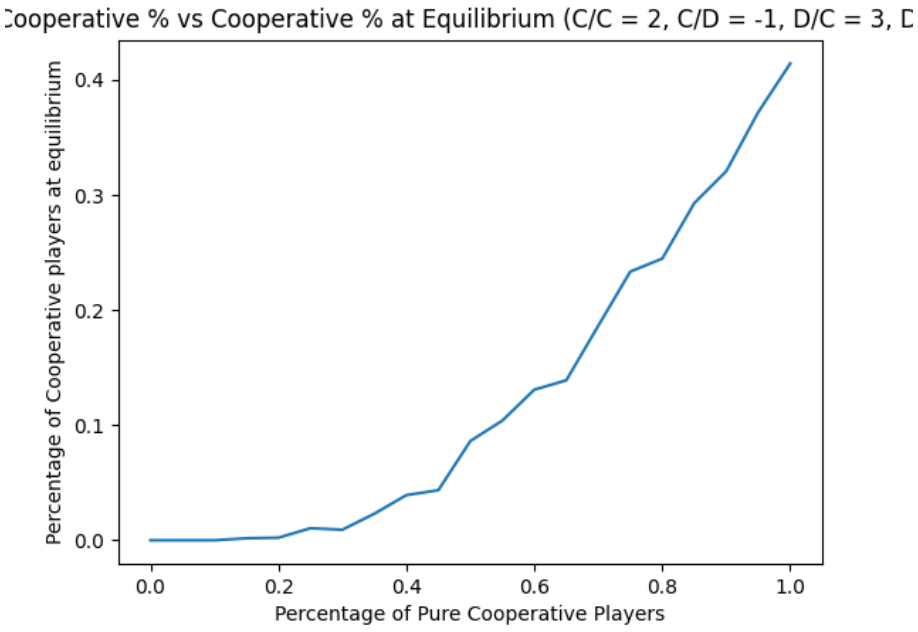


Figure 8: Equilibrium Cooperative Percentage vs. Initial Cooperative Players (Bayesian).

In contrast to the population of imitators, in this case, as shown in Figure 8, the immutable defectors exert no influence on the system. Consequently, the proportion of defectors is capped at half the player population, contingent upon the number of initial defectors that remain inflexible.

4.4 Experiment 4: Pure Defectors in a Cynical Cooperative Population

In this experiment, we explore the propagation of pure defectors within an initially predominantly cooperative environment. In contrast to the scenario discussed in Section 4.2,

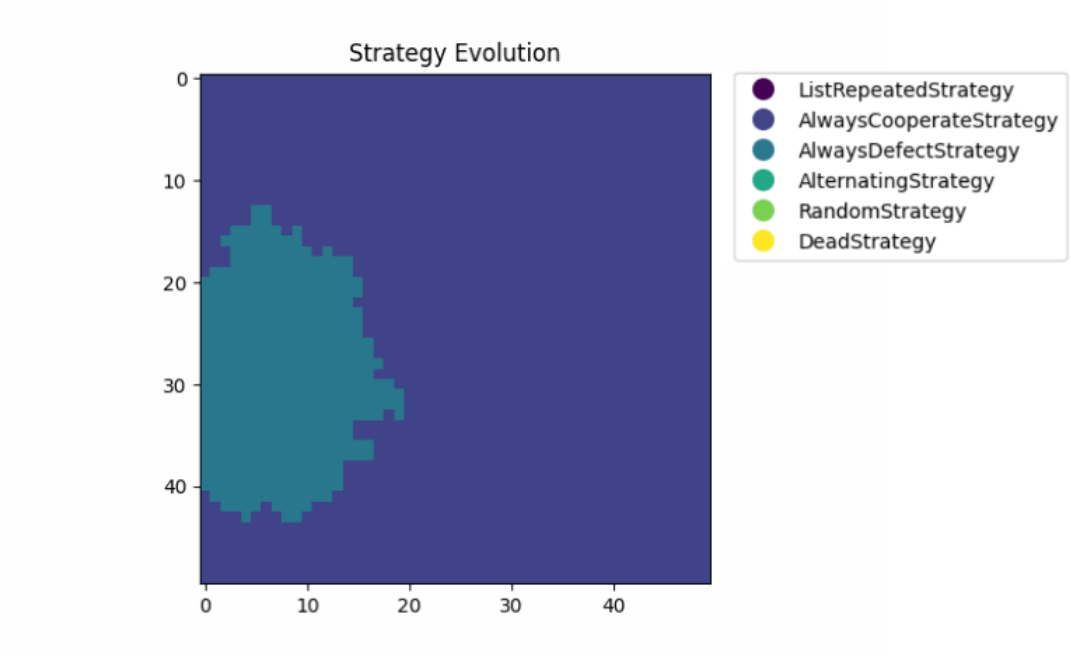


Figure 9: Defective decision spreading from a single source initial defector.

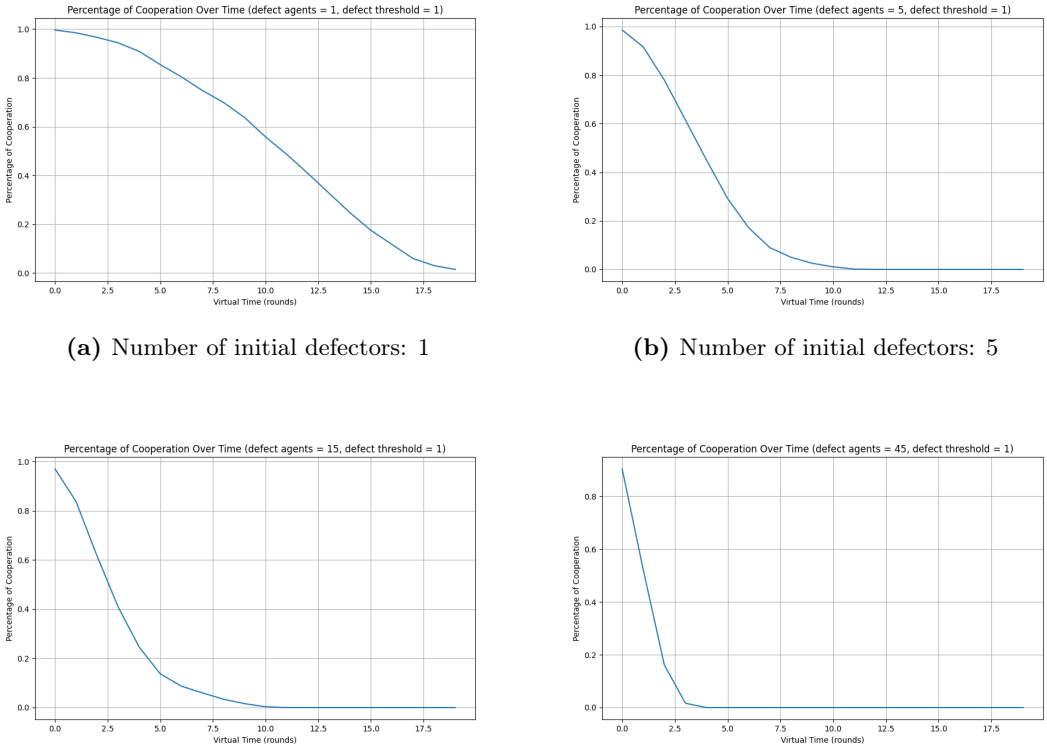
where the population started with an asymmetry favoring cooperation, this setup involves a cynically oriented adaptation strategy among players, leading to a contagion-like spread of defection. The experiment is structured as follows:

1. **Initial conditions:** The population is primarily cooperative, except for a specified small number (n) of defectors.
2. **Strategy adaptation:** Players adhere to cooperation until encountering betrayal in x consecutive instances, at which point they transition to a defective strategy.

Figure 9 depicts the proliferation of defectors stemming from a single defector acting as the source.
By keeping the threshold x constant and varying the number of initial defectors, the corresponding trend shown in Figure 10 reveals that an escalation in the initial number of defectors leads to a swifter decline in the percentage of cooperative players, resulting in a more pronounced convex curve. In contrast, when holding the number of initial defectors constant and adjusting the threshold x, the cooperative player percentage displays a more concave pattern, as illustrated in Figure 11. This finding implies an intuitive relationship: an increase in optimism, represented by a higher threshold x, effectively mitigates the propagation of defection.

4.5 Experiment 5: Single Defector in a Network - Impact Analysis

Now, we turn our attention to exploring how the initial placement of the defector in the network influences the spreading phenomenon observed in the previous experiment. Although our network lacks an intrinsic topology resembling a complex social network, the algorithm governing the chronological order in which players engage in the game, coupled with the location of players in the grid that affects their distances to other players, introduces a hierarchy of impact, resembling a form of *influence* in a network. This abstract



(a) Number of initial defectors: 1

(b) Number of initial defectors: 5

(c) Number of initial defectors: 15

(d) Number of initial defectors: 45

Figure 10: Evolution of cooperation percentage for different initial numbers of defectors.

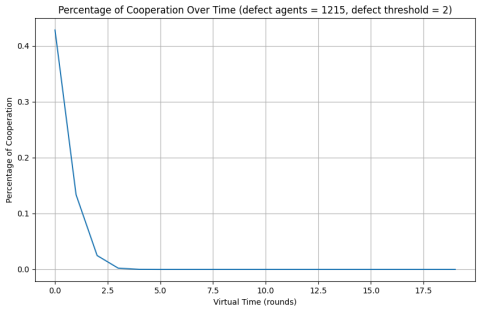
notion mirrors how individuals with specific characteristics, such as politicians, influencers, or those with privileged access to information, wield greater leverage to propagate influence in a social system. Specifically, our setup involves the following:

1. **Initial conditions:** The entire player population is cooperative, except for a single defector.
2. **Strategy adaptation:** Players maintain a cooperative stance until experiencing betrayal x times consecutively, at which point they switch to a defective strategy.
3. **Variable:** The sole defector is placed in different locations within the system.
4. **Collapse Index Measurement:** The duration (in the number of steps) it takes for the cooperative player population to diminish to below 10%.
5. **Investigation Focus:** Analyzing how the initial placement of the defector influences the rate at which the system collapses.

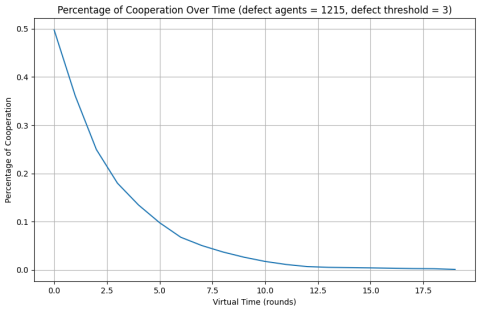
Examining Figure 12, it becomes evident that a central position in the network has a more significant impact on the collapsing index. Moreover, as anticipated, the collapse index displays rational symmetry within our network.

4.6 Experiment 6: Defector Density's Impact Analysis

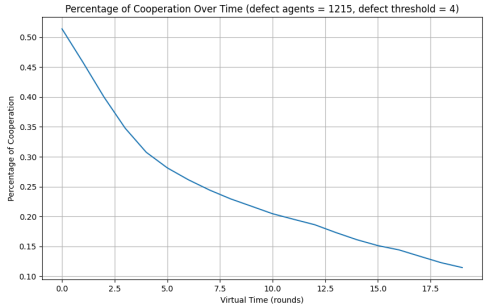
Using a setup similar to the previous two experiments, we now focus on examining the effect of the number of initial defectors, averaged over iterations where their locations are randomized. Notably, we observe that the average collapse index decreases exponentially with an increase in the number of initial defectors, as shown in fig.13.



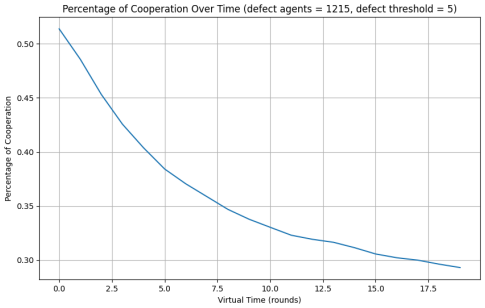
(a) Defective threshold: 2



(b) Defective threshold: 3

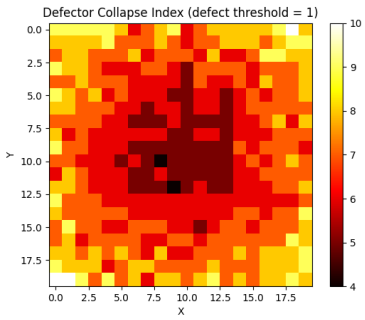


(c) Defective threshold: 4

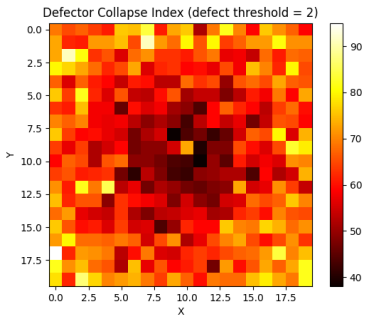


(d) Defective threshold: 5

Figure 11: Evolution of cooperation percentage for different initial numbers of defectors.



(a) x=1



(b) x=2

Figure 12: Heat map of collapse indices based on initial defector location.

4.7 Experiment 7: Incentives Determine Survival Fitness

In this experiment, we implement a credit system, essentially a scorecard for individuals, with an offset representing the initial credit allocated before the game commences. The primary goal of this setup is to evaluate the survival fitness of various meta-strategies. Specifically, players whose credits fall below zero will be eliminated. Moreover, by adjusting the payoff matrix, we can take on an authoritative role to selectively incentivize certain strategy switchers, thereby influencing their survival rates. Additionally, through experimentation with the payoff matrix, our aim is to identify the range of its values within which the evolution of a system comprised of mixed strategies tends to favor cooperation.

1. **Initial conditions:** Begin with an equal distribution of cooperative and defective

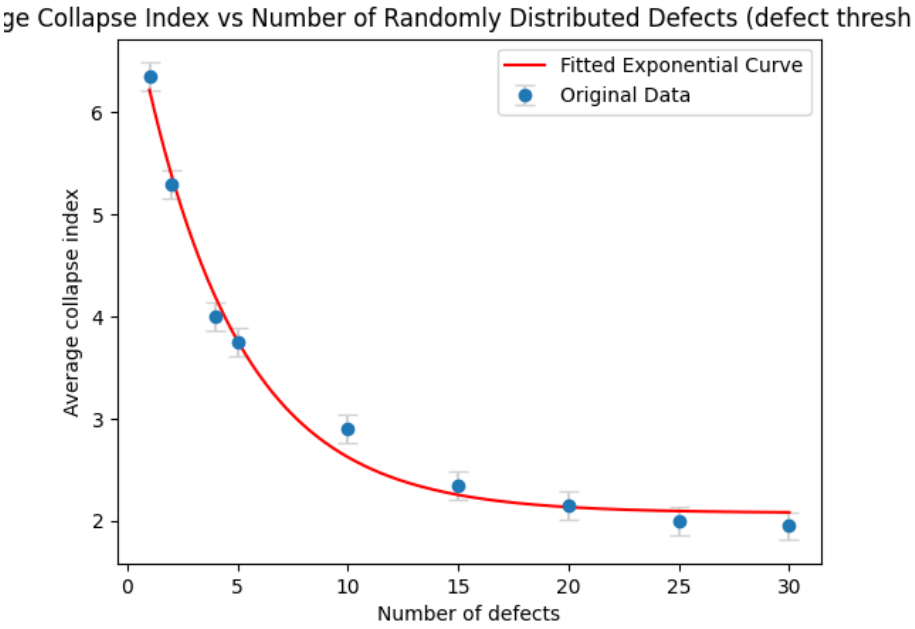


Figure 13: Collapse index (averaged over random location) vs. Number of initial defectors.

players. Each player is assigned a random strategy switcher for alternating between strategies.

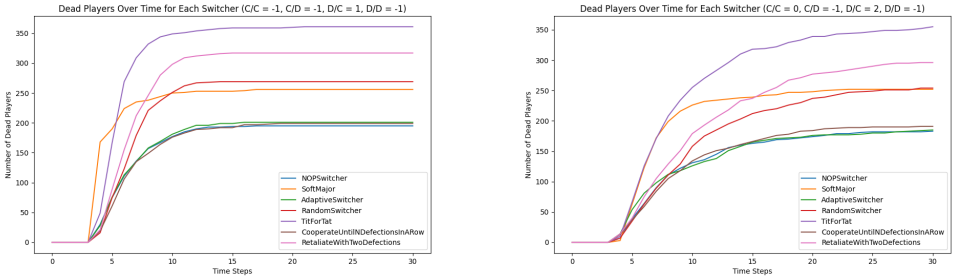
2. **Equal Credit Allocation:** Every player starts with an identical credit allotment.
3. **Survival Rate Analysis:** Conduct a study on the endurance of various strategy switchers over time, considering diverse payoff matrices.

This experiment aims to capture the phenomenon where the effectiveness of tactics is heavily dependent on cultural norms and environmental incentives. For instance, business strategies are often shaped by economic and market conditions, while dating approaches may be influenced by cultural factors.

(a) Before evolution. **(b)** After evolution.

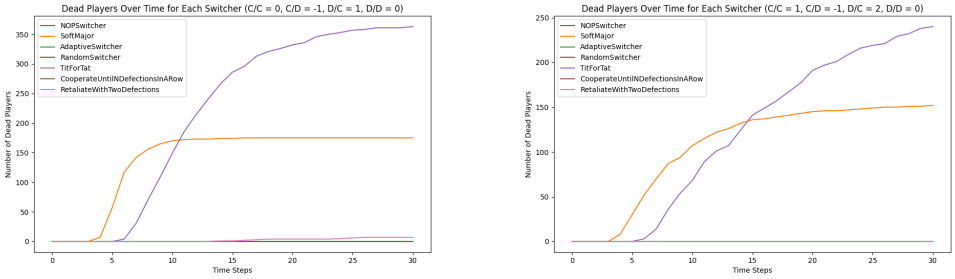
Figure 14: An appropriate payoff matrix helps remove the defective players while limiting elimination of agents.

In Figure 14, we illustrate that with an appropriate payoff matrix, it is possible to effectively eliminate the defectors while preserving the cooperative and adaptive players.
Examining Figure 15 and Figure 16, we discern two distinct behavioral patterns in the elimination rates of players based on different meta-strategies, contingent on the configuration of our payoff matrix. The first type is characterized by negative score penalties



(a) $R(C|C) = -1, R(D|D) = -1$ **(b)** $R(C|C) = 0, R(D|D) = -1$

Figure 15: Eliminated players of all meta-strategies over time steps (type 1).



(a) $R(C|C) = 0, R(D|D) = 0$ **(b)** $R(C|C) = 1, R(D|D) = 0$

Figure 16: Eliminated players of all meta-strategies over time steps (type 2).

for mutual defections, $R(D|D) < 0$, while the second type involves no penalty for mutual defections, $R(D|D) \geq 0$. In the first category, the cynical strategy `TitForTat` exhibits the highest elimination rate, followed by `SoftMajor`, `RandomSwitcher`, and `NOPSwitcher`. In these conditions, logical players employing Bayesian deduction (`AdaptiveSwitcher`) perform the best. Conversely, in the second group (Figure 16), all strategies maintain a 100% survival rate, except for the pessimistic or naive strategies `TitForTat` and `SoftMajor`. These findings suggest the potential to tailor our payoff matrix to favor logical and optimistic strategies.

5 Conclusion and Outlook

Our model introduces several components that go beyond classical Prisoner's Dilemma problems, incorporating concealed history, community knowledge, and a credit system. These additions, inspired by real-world systems, aim to capture the intricate interactive dynamics among strategically selfish agents.

In our framework, we initially explore various factors contributing to the system's inclination toward defective behaviors, often rooted in the cynical, pessimistic, and self-preserving nature of individual agents. However, we find that the extent of these behaviors is contingent upon parameters under the control of the game designer. The model successfully passes numerous benchmark tests, yielding predictable and intuitive outcomes, while also revealing some unexpected features. Each model element, combined with our chosen setup parameters, mirrors essential social, economic, and political phenomena. The experimental results suggest the feasibility of biasing the model towards a more optimistic, cooperative population by altering the incentives.

Despite its strengths, our model has three primary limitations due to the project's time

constraints and scope. First, while we can restrict access to information (opponents' history) to influence the system's behavior, this analytical direction was not fully explored. Second, the simplicity of the credit system may not fully capture the complexities of strategy performance fluctuations, particularly if strategies are given a second chance after hitting zero, reflecting real-world scenarios where second chances are common. Third, our primary network model—a lattice where agents connect via adjacent edges in four or eight directions—effectively represents geographical connections. However, to more accurately mirror the connectivity in our real-world analogues, other network topologies might be better suited. Exploring these topologies could provide deeper insights into the dynamics of complex systems and enhance the applicability of our model to a broader range of scenarios.

> If only it were all so simple! If only there were evil people somewhere insidiously committing evil deeds, and it were necessary only to separate them from the rest of us and destroy them. But the line dividing good and evil cuts through the heart of every human being. And who is willing to destroy a piece of his own heart?

Additionally, as the above quote by Aleksandr Solzhenitsyn in *The Gulag Archipelago* highlights, there is complexity inherent in understanding and addressing human behavior. While our model suggests the potential for influencing cooperative or defective behaviors by adjusting parameters and incentives, reality is often far more intricate. The quote speaks to the challenge of categorizing individuals into rigid binaries of good and evil. The suggestion that destroying the supposed "evil" elements is not a straightforward solution resonates with the idea that addressing complex social dynamics requires a more profound understanding of the intricacies involved in human and societal behaviour. In essence, the model and the quote both caution against oversimplification and emphasize the need for a nuanced approach in understanding and navigating the intricate landscape of human behavior and societal dynamics.
In conclusion, while our model provides valuable insights into the interplay of strategic behaviors, incorporating more nuanced elements and exploring diverse network topologies could further enrich its applicability and predictive power.

References

[1] Robert Axelrod and William D Hamilton. The evolution of cooperation. *science*, 211(4489):1390–1396, 1981.

[2] Zhiqiang Gou and Ya Li. Prisoner's dilemma game model based on historical strategy information. *Scientific Reports*, 13(1):1, 2023.

[3] Clay Halton. Iterated prisoner's dilemma: Definition, example, strategies, Jun 2022.

[4] Garrett Hardin. The tragedy of the commons. *Science*, 162(3859):1243–1248, 1968.

[5] Christoph Hauert and Olaf Stenull. Simple adaptive strategy wins the prisoner's dilemma. *Journal of Theoretical Biology*, 218(3):261–272, 2002.

[6] John F Nash Jr. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

[7] William Poundstone. *Prisoner's dilemma: John von Neumann, game theory, and the puzzle of the bomb*. Anchor, 1993.

[8] William Poundstone. *Prisoner's dilemma: John von Neumann, game theory, and the puzzle of the bomb*, page 118. Anchor, 1993.