Navigating the N-Person Prisoner's Dilemma: From the Tragic Valley to the Collaborative Hill

Chris Tcaci and Chris Huyck[1]

[1] Middlesex University, London NW4 4BT UK
M00674787@mdx.ac.uk
[2] c.huyck@mdx.ac.uk
https://cwa.mdx.ac.uk/chris/chrisroot.html

Abstract. The N-Person Iterated Prisoner's Dilemma (N-IPD) is an excellent environment to explore collaboration. Static agent decision policies lead to cooperative results, with high payout when each agent can vote against each other agent individually, and the results descend into low cooperation when each agent only gets one decision. A similar result is shown with agents that learn via reinforcement learning.

Keywords: N-Person Prisoner's Dilemma · Reinforcement Learning · Emergence of Cooperation · Multi-Agent Systems.

1 Introduction

The Prisoner's Dilemma (PD) serves as a foundational paradigm in game theory, illustrating the conflict between individual rational self-interest and mutually beneficial collective action [**?**]. In its simplest form, two individuals, unable to communicate, must independently choose whether to cooperate or defect. While mutual cooperation yields a good outcome for both, each player has an individual incentive to defect, leading to a suboptimal outcome if both choose to do so. The Iterated Prisoner's Dilemma (IPD), where the game is played repeatedly, opens the door for cooperation to emerge through strategies based on reciprocity, as famously demonstrated by Axelrod's tournaments where Tit-for-Tat proved remarkably successful [**?**].

However, many real-world social and economic dilemmas—ranging from managing common resources to international climate agreements—involve more than two interacting parties. The N-Person Iterated Prisoner's Dilemma (N-IPD) generalizes the IPD to scenarios with N participants [**?**,**?**].

This paper will first provide a brief background on the N-IPD and relevant learning approaches (Section **??**). It then describes

2 Background and Related Work

This section briefly reviews some key concepts from game theory, focusing on the N-Person Prisoner's Dilemma (N-IPD), and introduces the Agent-Based Modelling (ABM) and Multi-Agent Reinforcement Learning (MARL) approaches relevant to this paper.

2.1 The Prisoner's Dilemma

The Prisoner's Dilemma [?,?] is widely known problem used to study collaboration. There are two prisoners and they are given the option to turn the other in. So both have the option to collaborate, and not turn the other in, or defect. That is each has a choice C or D. Each is given a reward based on their decision and the decision of the other. If they both collaborate, they both are given R, a reward. If both defect, they are both given P, a punishment. If one defects, and the other collaborates, the defector is given T, a temptation, and the collaborator is given S, a sucker's payoff.

		Prisoner 1	
		Collaborate	Defect
Prisoner 2	Collaborate	$R_{1,2}$	T_1S_2
	Defect	T_2S_1	$P_{1,2}$

Table 1. The table represents the outcomes for the four scenarios when two prisoners vote. When both collaborate, both get R, and when both defect, both get P. When one collaborates and one defects, the defector gets T and the collaborator gets S payoff.

Axelrod and Hamilton restrict the values so that equations **??** and **??** are followed. A standard set of values is $T = 5$, $R = 3$, $P = 1$ and $S = 0$.

$$T > R > P > S \tag{1}$$

$$R > (S + T)/2 \tag{2}$$

The IPD merely plays the tournament over and over. This gives the agents a chance to develop their own policy.

The Tit-for-Tat (TFT) strategy proved to be most successful. The strategy is to collaborate when the opponent collaborates, then defect in the round after they defect. Note that the TFT strategy is not a Nash equilbrium [?], indicating that it can make cooperation difficult to maintain [?].

2.2 The N-Person Prisoner's Dilemma (N-IPD)

The Iterated Prisoner's Dilemma (IPD) serves as a model for understanding cooperation. While Axelrod's work highlighted the success of strategies like Tit-for-Tat in two-player encounters, extending this to multi-agent scenarios (N-Person IPD or N-IPD) reveals a more complex strategic landscape [?].

The overall reward increases linearly as the number of collaborators increase [?]. However, the individual reward is greater when the agent defects (see section **??**).

2.3 Core Interaction Models

When there are N agents (and they all have only one vote), an individual agent's payoff is determined by its own choice and the total number of other agents in the group who chose to cooperate. This structure represents scenarios with diffuse payoffs, where individual actions contribute to a shared outcome, and direct one-to-one reciprocity is obscured. The payoff for an agent who cooperates is calculated as $S + (R - S) \times (n_{oc}/(N - 1))$, and for a defector as $P + (T - P) \times (n_{oc}/(N-1))$, where n_{oc} is the number of other cooperators. T, R, P, and S are from table **??**, and the simulations below use the default values, $T = 5$, $R = 3$, $P = 1$ and $S = 0$.

There are two distinct interaction models in N-IPD environments. The first is the pairwise voting model. In this setup, agents can vote for each of the other $N - 1$ players. This is in esence a series of independent 2-player IPD games. Each round each of the N players plays against the $N - 1$ other players. An agent's total score is the sum of payoffs from all its interactions. This model emphasizes direct, one-to-one accountability, where the actions of one agent in a pair directly affect the other, and responses can be specifically targeted.

The second is neighbourhood voting model. All N agents make a single choice (cooperate or defect) simultaneously as part of one collective group. In this case payoff comes directly from the individual interactions of table **??**.

2.4 Agent Strategies and Adaptations

There are several static polcies that are used. They are static in the sense that they perform by the same rules each time. These are the always collaborate strategy, the always defect strategy and the Tit-for-Tat strategy. In the N-Person game the Tit-for-Tat (TFT) strategy makes a probabilistic decision based on the number of collaborators in the last round. That is, if the number of collaborating agents in the prior round is C, and there are $N-1$ other agents, the TFT agents randomly select collaborate $C/(N - 1)$ of the time. An additional variant of the TFT agent, the TFT-E agent, explores; that is a given percentage of the time, no matter what the other agents do, the TFT-E agent merely flips a coin to determine whether it collaborates or defects.

2.5 Agent-Based Modelling (ABM) for Social Dilemmas

Agent-Based Modelling (ABM) offers a powerful computational methodology for studying complex social systems from the bottom up [**?**,**?**]. By simulating the actions and interactions of autonomous, heterogeneous agents according to predefined rules within a specified environment, ABM allows researchers to observe emergent macroscopic phenomena, such as the rise or fall of cooperation. It is particularly well-suited for exploring the N-IPD due to its ability to model local interactions, diverse agent strategies (including learning), and the non-linear dynamics that often characterize social dilemmas. Axelrod's pioneering tournaments using ABM for the 2-player IPD provided early insights into the conditions favoring cooperative strategies [**?**].

2.6 Reinforcement Learning in Multi-Agent Systems (MARL)

Reinforcement Learning (RL) is a class of machine learning where agents learn to make sequences of decisions by interacting with an environment and receiving feedback in the form of rewards or punishments [?]. Standard Q-learning is a foundational RL algorithm that learns the value of taking a particular action in a given state. However, when applied to multi-agent systems, where multiple agents are learning simultaneously, standard RL algorithms face significant challenges, primarily due to the non-stationarity of the environment: each agent's policy changes as it learns, thereby changing the environment from the perspective of other agents [?]. This can destabilize learning and prevent convergence to cooperative equilibria. To address these issues within the N-IPD context, prior work has explored more advanced multi-agent reinforcement learning (MARL) techniques like Hysteretic Q-learning [?] and WoLF-PHC [?], which aim to endow agents with more sophisticated learning capabilities.

Q-learning [?] is a system that learns by building a table of results from prior experience, called a Q-table. For example, in the section **??**, three agents participate, and make decisions. If one is a Q-learning agent, it can build a table of, for example, the last two moves. Each step move has eight possible outcomes 2^3, so there are 64 cells to fill. Additionally, the Q-learning agent typically has an explore option, so that no matter what the tables say, it will try a random move a small percentage of the time.

undone discount factor

3 The Collaborative Hill and the Tragic Valley

The simplest extension to the two person IPD is the three person IPD. Simulations on this task show that the voting mechanism largely determines whether the agents converge on a cooperative solution (the Collaborative Hill) or whether they defect (the Tragic Valley).

The first set of simulations uses pairwise voting, and tournaments are run with agents with static policies. Fifty iterations are performed on all of the combinations of agents with the static policies of always defect, always collaborate, Tit-for-Tat (TFT), Tit-for-Tat with exploration (TFT-E), and random.

Figure **??** show the results of four different sets of three agents with static policies: three TFT agents, two TFT with 10% exploration (TFT-E) and one always defect, two TFT agents and one always defect agent, and two TFT-E agents and one always collaborate agent. The vertical axis refers to how often the two or three TFT or TFT-E agents voted collaborate. The Tit-for-Tat agents start off by collaborating, and the three TFT system continues to collaborate. The two TFT agents with the always defect agent always collaborate with each other but (undone, after an initial collaboration) always defect against the always defect agent. This leads to a lower average payout, but is still largely collaborative. Both of these give horizontal lines in figure **??**.

The TFT-E agents behave more stochastically, as sometimes they change their decisions. The pair with the collaborative agent largely collaborate, and

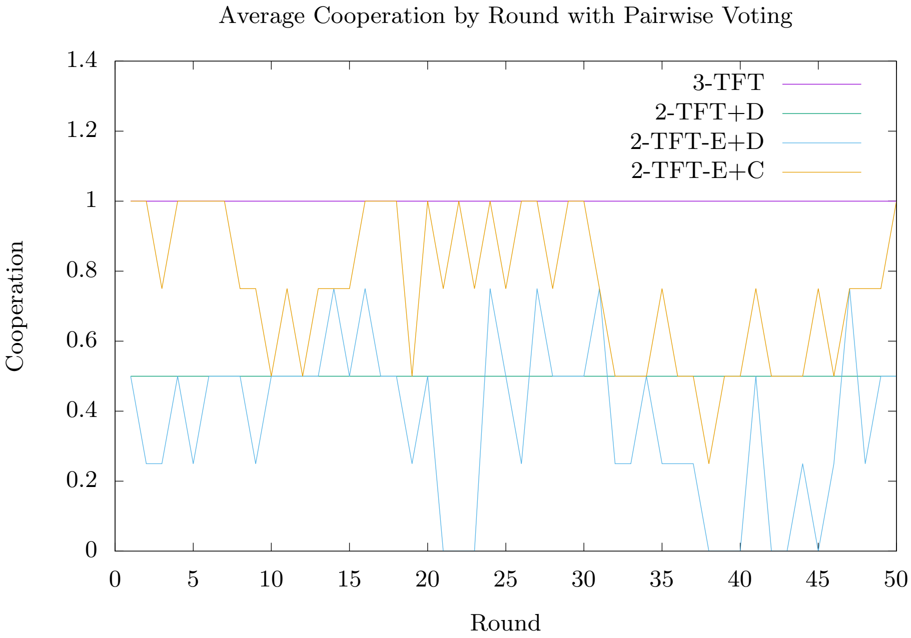

Fig. 1. Results from four static 50 round tournaments with neighbourhood voting. 3-TFT refers to 3 Tit-for-Tat agents that start cooperating; note that they continue to cooperate. 2-TFT+D refers to two TFT agents with an agent that always defects; this leads to system where the TFT agents collaborate with each other and defect against the defecting agent. 2-TFT-E+D refers to two TFT agents with 10% exploration and an always defect agent; this leads to a system where the TFT agents have some collaboration but mostly defect. 2-TFT-E+C refers to two TFT agents with 10% exploration and an agent that always cooperates; the TFT agents mostly collaborate, but there is some defection.

the pair with the always defect agent largely defect. Below (figure **??**) it is shown that the ratio is 75% and 25%.

The second set of simulations uses neighbourhood voting, with each agent getting one vote. The results are shown in figure **??**. The same four sets of agents, using static policies, are used. The three TFT agents continue to collaborate, as do the two TFT agents with the always collaborate agent (not shown in figure **??**). However, the TFT agents with the always defect initially vote to collaborate, but then quickly move to always defect. They descend into the Tragic Valley.

The TFT-E systems move up and down, with the pair with the collaborative agent being more collaborative, and the pair with the defecting agent being less collaborative. Below (figure **??**) it is shown that the ratio is 80% and 20% .

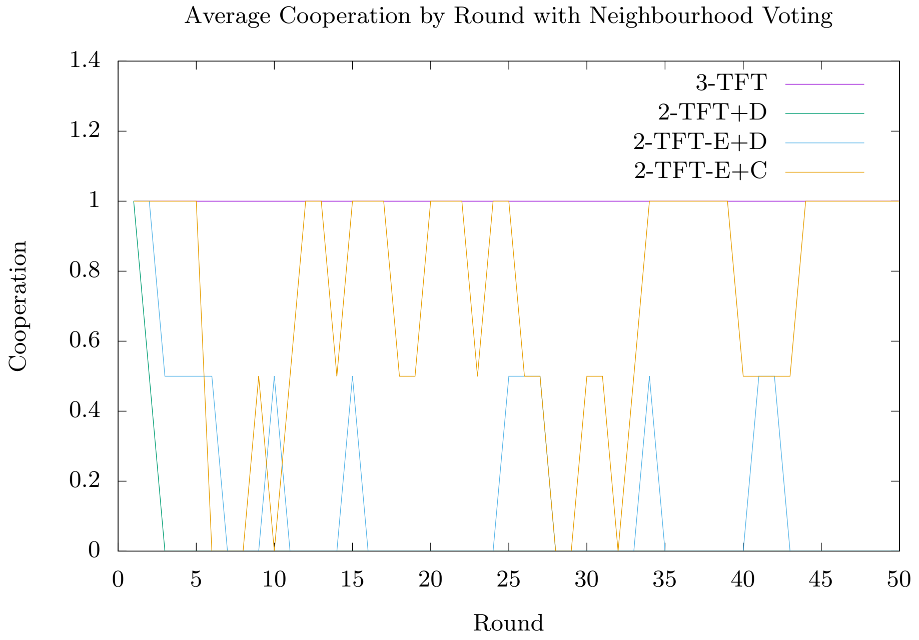

Fig. 2. Results from four 50 round tournaments with static agents with neighbourhood voting. 3-TFT refers to 3 tit-for-tat agents that start cooperating; they continue to cooperate. 2-TFT+D refers to two TFT agents with an agent that always defects; this quickly descends to no cooperation. 2-TFT-E+D refers to two TFT agents with 10% exploration and an always defect agent; this system has some cooperations but mostly defects; note that it defects more than when there is pairwise voting. 2-TFT-E+C refers to two TFT agents with 10% exploration and an agent that always cooperates; this cooperates about half the time, which is less than when there is pairwise voting.

undone something about TFT-E and variability. replace 3TFT with 3-TFT-E on the second figure

The reason behind the difference between the two is that the reward space for each agent reinforces collaboration with with pairwise voting. Each agent can punish particular defectors and reward particular collaborators. This is why the agents can all get better results and ascend the Collaborative Hill. On the other hand, with neighbourhood voting, each agent can only reward or punish in aggregate. The reward structure is that each agent does better by defecting, and any collaboration tends to give worse immediate results. This reward space and the agents' ability to only weakly influence other agents draws the agents into the Tragic Valley where agents do not cooperate.

Note that without exploration, the static policies move into an attractor state. When all three agents are TFT, if they all collaborate, they will continue to do so; not shown is the case when the initial choice is random. In the one of eight times when all three agents defect, the system starts in the attractor state when they all defect, and continue to defect. In the case where there is mixed voting, the system will move about until it gets to one of the two attractor states.

Figure **??** compares the Tit-for-Tat agents with exploration and voting mechanisms. These are similar to the runs in figures **??** and **??**, but here the results reflect an average of 100 runs. It indicates that the pairswise voting strategy is more likely to collaborate in the against always defect though this is eventually always collaboration between the two Tit-for-Tat agents.

Fig. 3. Results from the average of 100 50 round tournaments with four different sets of agents. All the sets have two Tit-for-Tat (TFT) agents with 10% exploration. Two have a third agent that always defects, and two have one that always cooperates. The other variable is voting (pairwise or neighbourhood).

4 Reinforcement Learning Agents

Using standard Q-learning agents with only two steps of history, agents become cooperative with pairwise voting, and defect with neighbourhood votign. Figure **??** shows this. Over time the neighbourhood voting descends toward 20% cooperation while the pairwise voting agents remain at around 80 %. The static TFT agents follow their counterparts that have learned policies. Undone are these true Variance to 80 and 20% are due to the exploratory behaviour of the Q-Learning agents. When the exploration gradually decays the cooperation values go to 100 and 0%.

Fig. 4. Results from the average of 100 500 round tournaments with two different sets of agents. Both sets are two Q-Learning agents with one Tit-for-Tat agent with exploration. They differ by pairwise vs. neighbourhood voting, and the average of the Q-Learning agents have a line, and the Tit-for-Tat agent has a line. This clearly shows the pairwise agents remaining collaborative while the neighbourhood agents descend into the Tragic Valley.

undone results from 5 7 19 and 25 agents

undone payout vs collaboration in figures

undone results of q-learning agents vs. always cooperate pairwise and neighbour

Table 2. Sample Results

Voting	Agent 1 & 2 Type	Avg. Cooperation	Agent 3 Type	Avg. Cooperation
Pairwise	Q Learning	64.1%	Random	50%
Neighbour	Q Learning	19.6%	Random	50%
Pairwise	Q Learning	52.7%	Defect	0%
Neighbour	Q Learning	31.8%	Defect	0%
Pairwise	Q Defect	0%	Q Learning	17.1%
Neighbour	Q Defect	0%	Q Learning	17.6%

The authors were suprised that they have found no papers explicitly stating that pairwise voting led to largely collaborative performance. However, this may be due to pairwise voting being largely equivalent to $(N-1)*(N-2)$ individual tournaments. Thus, the original work on the two person Prisoner's Dilemma holds.

5 Escaping the Tragic Valley with Enhanced Reinforcement Learning

The preceding sections have established a critical dichotomy: the pairwise interaction model creates a Collaborative Hill where simple reciprocity can thrive, while the neighbourhood voting model leads to a Tragic Valley of mutual defection for both static agents (Section **??**) and basic reinforcement learners (Section **??**). The diffuse nature of rewards in the neighbourhood setting obscures the one-to-one accountability needed for simple learning algorithms to foster cooperation.

This raises a crucial question: is the Tragic Valley an inescapable feature of the N-IPD's neighbourhood structure, or can it be overcome by more sophisticated agents? To investigate this, an Enhanced Q-Learning (EQL) agent, designed to better perceive and react to the dynamics of group behaviour, was developed.

5.1 The Enhanced Q-Learning Agent

The standard Q-learning agent, which bases its state on a simple discretization of the previous rounds' cooperation, struggles to identify meaningful patterns. The Enhanced Q-Learning agent incorporates several key improvements inspired by advancements in MARL to create a more sophisticated learning mechanism:

- **Richer State Representation:** Instead of just the previous rounds' cooperation level, the EQL agent's state can incorporate its own action history and the trend of group cooperation over multiple rounds. For example, a state might capture not only that cooperation is *high*, but also that it is *stable* or *increasing*, and whether the agent itself has been cooperating or defecting. This allows the agent to learn the consequences of its actions on the group's trajectory.

- **Optimistic Initialisation:** The EQL agent's Q-table is initialised with optimistic values for unexplored state-action pairs [?]. This encourages the agent to thoroughly explore its options, particularly cooperative actions, before committing to a potentially suboptimal defect-heavy strategy.
- **Adaptive Exploration:** The agent employs a decaying exploration strategy. It explores more at the beginning of a tournament and gradually reduces its random exploration rate to exploit the knowledge it has gained, allowing for convergence to a stable policy.

These enhancements transform the agent from a purely reactive learner into one that can perceive and respond to the emergent dynamics of the system.

5.2 Comparative Results: Learning to Cooperate

To test the EQL agent's capabilities, a series of tournaments with neighbourhood voting, mirroring those in the previous sections, is described below. The results demonstrate a clear and significant ability to escape the Tragic Valley.

The most striking result is observed when placing a single EQL agent in a group with two Tit-for-Tat agents. As established, a basic Q-learner in this scenario learns to exploit the initially cooperative Tit-for-Tats, leading to a downward spiral of defection. The EQL agent, however, behaves entirely differently.

As shown in Figure **??**, the EQL agent learns to reciprocate the Tit-for-Tats' cooperative nature. Its cooperation rate climbs and stabilises at a high level, resulting in a system of mutual cooperation. Crucially, its cumulative score rises in lockstep with the Tit-for-Tat agents, indicating it has discovered a high-payoff, cooperative equilibrium. This emergence of sustained cooperation, driven by a higher score, is precisely the intelligent behaviour sought.

Fig. 5. Performance of a single Enhanced Q-Learning (EQL) agent with two TFT agents in the neighbourhood setting (average of 500 runs). Left: The EQL agent's cooperation rate rises to match the TFT agents. Right: The EQL agent achieves a high cumulative score, comparable to the cooperative TFT agents, demonstrating it has learned a beneficial, cooperative policy.

The EQL agent also demonstrates rational behaviour in less cooperative environments. When paired with two always defect agents, it quickly learns that cooperation is futile and its cooperation rate drops to the baseline exploration level (Figure **??**, left panel). This confirms the agent is not simply a blind cooperator; it is learning an appropriate, context-dependent strategy.

Interestingly, when placed with one always defect and one always cooperate, the EQL agent learns to defect. (Figure **??**, right panel). It learns that cooperating does not influence either agent.

These results show that the structural barrier of the Tragic Valley is not insurmountable. An agent equipped with a sufficiently rich state representation

Fig. 6. Cooperation rate of an EQL agent in uncooperative neighbourhood settings. Left: Against two AllD agents, the EQL agent learns to defect.

and a robust exploration mechanism can learn to identify and foster cooperative dynamics even in the absence of direct, pairwise reciprocity.

6 General Discussion

The simulations show that the pairwise voting leads to a cooperative system, and the neighbourhood voting leads to systems where the agents defect. This shows the Tragic Valley (neighourhood voting) and the Collaborative Hill (pairwise voting).

The difference stems from the reward space. If the agent knows every other agent's response, it will do better to select defect in almost every circumstance. The only exception to this is with neighbourhood voting, when all of the other agents defect; in this case, defecting or collaborating give the same reward (0). The real difference only occurs when the agent can look back. With the two person version, TFT uses history from the last step to discourage defection. With the $N > 2$ agents, history is still imporant. The static policies that were explored, random, always defect, and always collaborate do not change based on history and there decisions are unaffected by the other agents.

Only TFT uses history, but it behaves differently with pairwise voting and neighbourhood voting because of the reward space. Its history is affected by other agents, individually in pairwise voting, and in aggregate for neighbourhood voting. When it can choose to punish or reward each individually, it climbs he Collaborative Hill. When it can only choose in aggegate, it can only reward or punish in aggregate, so other agent will free ride, and the overall effect will be a descent into the Tragic Valley.

The same is true with simple reinforcement learning agents. The reward space moves these agents up the Collaborative Hill with pairwise voting, and down the Tragic Valley with neighbourhood voting.

undone history and EQL

undone eql fails on pairwise

undone future work explore other games such as the volunteer's dilemma [**?**].

7 Conclusion

undone something