Assignment 1: Policy Iteration in the Repeated Prisoner's Dilemma

RL2026A

Daniel Katz (315114991)
Avital Fine (208253823)

December 13, 2025

Abstract

This report studies optimal decision-making in the Repeated Prisoner's Dilemma (RPD) through a full MDP formulation and tabular Policy Iteration. We implement a custom Gymnasium environment, define transition functions and reward structures for four opponent types (ALL-C, ALL-D, TFT, Imperfect TFT), and evaluate two observation schemes (Memory-1 and Memory-2). We then perform Policy Iteration for varying discount factors and analyze how memory depth and stochasticity influence optimal behavior.

1 Introduction

The Repeated Prisoner's Dilemma (RPD) is one of the canonical environments for studying cooperation, strategic behavior, and temporal decision-making. In this assignment, we model the RPD as a Markov Decision Process (MDP) fully consistent with the Gymnasium API, and examine how an optimal agent adapts its policy against different opponent personalities.

2 Part II: MDP Definition

2.1 Actions

$$A = \{C, D\},$$

where C denotes **Cooperate** and D denotes **Defect**.

2.2 State Space

We evaluated two observation schemes as required.

2.2.1 Memory-1

The agent observes only the previous round's joint actions $(a_t^{\text{agent}}, a_t^{\text{opp}})$. Thus the state space is:

$$S_{\text{M1}} = \{(C, C), (C, D), (D, C), (D, D)\},$$

with a total of 4 states. The initial state is (C, C).

2.2.2 Memory-2

The agent observes its two most recent actions and the opponent's two most recent actions:

$$s_t = ((a_{t-2}^A, a_{t-2}^O), (a_{t-1}^A, a_{t-1}^O)).$$

Since each action is binary, the total state count is:

$$|S_{\text{M2}}| = 2^4 = 16.$$

The full list of states (where each state is $((a_{t-2}^A, a_{t-2}^O), (a_{t-1}^A, a_{t-1}^O)))$ is:

((C,C), (C,C))	((C,C), (C,D))
((C,C), (D,C))	((C,C), (D,D))
((C,D), (C,C))	((C,D), (C,D))
((C,D), (D,C))	((C,D), (D,D))
((D,C), (C,C))	((D,C), (C,D))
((D,C), (D,C))	((D,C), (D,D))
((D,D), (C,C))	((D,D), (C,D))
((D,D), (D,C))	((D,D), (D,D))

The initial state is $((C, C), (C, C))$.

2.3 Transition Probability Function

The transition probability $P(s'|s, a)$ defines the probability of moving to state s' given current state s and agent action a.

2.3.1 General Form

The new state s' is determined by shifting the history.

- **Memory-1:** If $s = (a_{t-1}^A, a_{t-1}^O)$ and agent chooses a_t^A, the new state is $s' = (a_t^A, a_t^O)$.
- **Memory-2:** If $s = ((a_{t-2}^A, a_{t-2}^O), (a_{t-1}^A, a_{t-1}^O))$ and agent chooses a_t^A, the new state is $s' = ((a_{t-1}^A, a_{t-1}^O), (a_t^A, a_t^O))$.

In both cases, a_t^A is given by the agent's choice. The opponent's action a_t^O is determined by their strategy policy $\pi_{opp}(s)$. Thus, the transition probability is:

$$P(s'|s, a_t^A) = P(a_t^O|s).$$

It is important to note that for any state s' that does not match the history shift or implies an impossible opponent action, the probability is 0. For example, against an **ALL-C** opponent, any transition to a state where the opponent's last action is D has a probability of 0:

$$P(s' = (\dots, D)|s, a) = 0.$$

Current State	Action $C \to$ Next State	Action $D \to$ Next State
(C, C)	(C, C) : 1.0	(D, C) : 1.0
(C, D)	(C, C) : 1.0	(D, C) : 1.0
(D, C)	(C, C) : 1.0	(D, C) : 1.0
(D, D)	(C, C) : 1.0	(D, C) : 1.0

Table 1: Transition probabilities $P(s' \mid s, a)$ for Memory-1 against ALL-C.

Current State	Action $C \to$ Next State	Action $D \to$ Next State
(C, C)	(C, D) : 1.0	(D, D) : 1.0
(C, D)	(C, D) : 1.0	(D, D) : 1.0
(D, C)	(C, D) : 1.0	(D, D) : 1.0
(D, D)	(C, D) : 1.0	(D, D) : 1.0

Table 2: Transition probabilities $P(s' \mid s, a)$ for Memory-1 against ALL-D.

Current State	Action $C \to$ Next State	Action $D \to$ Next State
(C, C)	(C, C) : 1.0	(D, C) : 1.0
(C, D)	(C, C) : 1.0	(D, C) : 1.0
(D, C)	(C, D) : 1.0	(D, D) : 1.0
(D, D)	(C, D) : 1.0	(D, D) : 1.0

Table 3: Transition probabilities $P(s' \mid s, a)$ for Memory-1 against TFT.

Current State	Action C	Action D
(C, C)	(C,C):0.9, (C,D):0.1	(D,C):0.9, (D,D):0.1
(C, D)	(C,C):0.9, (C,D):0.1	(D,C):0.9, (D,D):0.1

Table 4: Memory-1 vs Imperfect TFT for states with $a_{t-1}^A = C$ (opp: 0.9 C, 0.1 D).

Current State	Action C	Action D
(D, C)	(C,C):0.1, (C,D):0.9	(D,C):0.1, (D,D):0.9
(D, D)	(C,C):0.1, (C,D):0.9	(D,C):0.1, (D,D):0.9

Table 5: Memory-1 vs Imperfect TFT for states with $a_{t-1}^A = D$ (opp: 0.9 D, 0.1 C).

Current State (h^A, h^O)	Action $C \to$ Next State	Action $D \to$ Next State
(CC, CC)	(CC, CC) : 1.0	(CD, CC) : 1.0
(CC, CD)	(CC, DC) : 1.0	(CD, DC) : 1.0
(CC, DC)	(CC, CC) : 1.0	(CD, CC) : 1.0
(CC, DD)	(CC, DC) : 1.0	(CD, DC) : 1.0
(CD, CC)	(DC, CC) : 1.0	(DD, CC) : 1.0
(CD, CD)	(DC, DC) : 1.0	(DD, DC) : 1.0
(CD, DC)	(DC, CC) : 1.0	(DD, CC) : 1.0
(CD, DD)	(DC, DC) : 1.0	(DD, DC) : 1.0
(DC, CC)	(CC, CC) : 1.0	(CD, CC) : 1.0
(DC, CD)	(CC, DC) : 1.0	(CD, DC) : 1.0
(DC, DC)	(CC, CC) : 1.0	(CD, CC) : 1.0
(DC, DD)	(CC, DC) : 1.0	(CD, DC) : 1.0
(DD, CC)	(DC, CC) : 1.0	(DD, CC) : 1.0
(DD, CD)	(DC, DC) : 1.0	(DD, DC) : 1.0
(DD, DC)	(DC, CC) : 1.0	(DD, CC) : 1.0
(DD, DD)	(DC, DC) : 1.0	(DD, DC) : 1.0

Table 6: Transition probabilities $P(s' \mid s, a)$ for Memory-2 against ALL-C.

Current State (h^A, h^O)	Action $C \to$ Next State	Action $D \to$ Next State
(CC, CC)	(CC, CD) : 1.0	(CD, CD) : 1.0
(CC, CD)	(CC, DD) : 1.0	(CD, DD) : 1.0
(CC, DC)	(CC, CD) : 1.0	(CD, CD) : 1.0
(CC, DD)	(CC, DD) : 1.0	(CD, DD) : 1.0
(CD, CC)	(DC, CD) : 1.0	(DD, CD) : 1.0
(CD, CD)	(DC, DD) : 1.0	(DD, DD) : 1.0
(CD, DC)	(DC, CD) : 1.0	(DD, CD) : 1.0
(CD, DD)	(DC, DD) : 1.0	(DD, DD) : 1.0
(DC, CC)	(CC, CD) : 1.0	(CD, CD) : 1.0
(DC, CD)	(CC, DD) : 1.0	(CD, DD) : 1.0
(DC, DC)	(CC, CD) : 1.0	(CD, CD) : 1.0
(DC, DD)	(CC, DD) : 1.0	(CD, DD) : 1.0
(DD, CC)	(DC, CD) : 1.0	(DD, CD) : 1.0
(DD, CD)	(DC, DD) : 1.0	(DD, DD) : 1.0
(DD, DC)	(DC, CD) : 1.0	(DD, CD) : 1.0
(DD, DD)	(DC, DD) : 1.0	(DD, DD) : 1.0

Table 7: Transition probabilities $P(s' \mid s, a)$ for Memory-2 against ALL-D.

Current State	Agent's a^A_{t-1}	Opp Response	Action $C \rightarrow$ Next	Action $D \rightarrow$ Next
(CC, CC)	C	C	(CC, CC) : 1.0	(CD, CC) : 1.0
(CC, CD)	C	C	(CC, DC) : 1.0	(CD, DC) : 1.0
(CC, DC)	D	D	(CC, CD) : 1.0	(CD, CD) : 1.0
(CC, DD)	D	D	(CC, DD) : 1.0	(CD, DD) : 1.0
(CD, CC)	C	C	(DC, CC) : 1.0	(DD, CC) : 1.0
(CD, CD)	C	C	(DC, DC) : 1.0	(DD, DC) : 1.0
(CD, DC)	D	D	(DC, CD) : 1.0	(DD, CD) : 1.0
(CD, DD)	D	D	(DC, DD) : 1.0	(DD, DD) : 1.0
(DC, CC)	C	C	(CC, CC) : 1.0	(CD, CC) : 1.0
(DC, CD)	C	C	(CC, DC) : 1.0	(CD, DC) : 1.0
(DC, DC)	D	D	(CC, CD) : 1.0	(CD, CD) : 1.0
(DC, DD)	D	D	(CC, DD) : 1.0	(CD, DD) : 1.0
(DD, CC)	C	C	(DC, CC) : 1.0	(DD, CC) : 1.0
(DD, CD)	C	C	(DC, DC) : 1.0	(DD, DC) : 1.0
(DD, DC)	D	D	(DC, CD) : 1.0	(DD, CD) : 1.0
(DD, DD)	D	D	(DC, DD) : 1.0	(DD, DD) : 1.0

Table 8: Transition probabilities for Memory-2 against TFT (opponent copies a^A_{t-1} deterministically).

Current State	Action C	Action D
(CC, CC)	(CC,CC):0.9, (CC,CD):0.1	(CD,CC):0.9, (CD,CD):0.1
(CC, CD)	(CC,DC):0.9, (CC,DD):0.1	(CD,DC):0.9, (CD,DD):0.1
(CD, CC)	(DC,CC):0.9, (DC,CD):0.1	(DD,CC):0.9, (DD,CD):0.1
(CD, CD)	(DC,DC):0.9, (DC,DD):0.1	(DD,DC):0.9, (DD,DD):0.1
(DC, CC)	(CC,CC):0.9, (CC,CD):0.1	(CD,CC):0.9, (CD,CD):0.1
(DC, CD)	(CC,DC):0.9, (CC,DD):0.1	(CD,DC):0.9, (CD,DD):0.1
(DD, CC)	(DC,CC):0.9, (DC,CD):0.1	(DD,CC):0.9, (DD,CD):0.1
(DD, CD)	(DC,DC):0.9, (DC,DD):0.1	(DD,DC):0.9, (DD,DD):0.1

Table 9: Memory-2 vs Imperfect TFT for states with $a^A_{t-1} = C$ (opp: 0.9 C, 0.1 D).

Current State	Action C	Action D
(CC, DC)	(CC,CC):0.1, (CC,CD):0.9	(CD,CC):0.1, (CD,CD):0.9
(CC, DD)	(CC,DC):0.1, (CC,DD):0.9	(CD,DC):0.1, (CD,DD):0.9
(CD, DC)	(DC,CC):0.1, (DC,CD):0.9	(DD,CC):0.1, (DD,CD):0.9
(CD, DD)	(DC,DC):0.1, (DC,DD):0.9	(DD,DC):0.1, (DD,DD):0.9
(DC, DC)	(CC,CC):0.1, (CC,CD):0.9	(CD,CC):0.1, (CD,CD):0.9
(DC, DD)	(CC,DC):0.1, (CC,DD):0.9	(CD,DC):0.1, (CD,DD):0.9
(DD, DC)	(DC,CC):0.1, (DC,CD):0.9	(DD,CC):0.1, (DD,CD):0.9
(DD, DD)	(DC,DC):0.1, (DC,DD):0.9	(DD,DC):0.1, (DD,DD):0.9

Table 10: Memory-2 vs Imperfect TFT for states with $a_{t-1}^A = D$ (opp: 0.9 D, 0.1 C).

2.3.2 Opponent Strategies

- **ALL-C:** $P(a_t^O = C|s) = 1$.
- **ALL-D:** $P(a_t^O = D|s) = 1$.
- **TFT:** Deterministic. Copies agent's last move.

$$P(a_t^O = a_{t-1}^A|s) = 1.$$

- **Imperfect TFT:** Stochastic.

$$P(a_t^O = a_{t-1}^A|s) = 0.9, \quad P(a_t^O \neq a_{t-1}^A|s) = 0.1.$$

2.4 Reward Function

The reward function $R(s, a)$ represents the **expected immediate reward** the agent receives when taking action a in state s. It is calculated by summing over the possible opponent actions a^O, weighted by their probability of occurring given the current state (history):

$$R(s, a) = \sum_{a^O \in \{C,D\}} P(a^O|s) \cdot \text{Payoff}(a, a^O)$$

where Payoff(a, a^O) is given by the standard RPD matrix:

$$\text{Payoff}(C, C) = 3, \quad \text{Payoff}(C, D) = 0, \quad \text{Payoff}(D, C) = 5, \quad \text{Payoff}(D, D) = 1.$$

For deterministic opponents (ALL-C, ALL-D, TFT), $P(a^O|s)$ is either 0 or 1, so the expected reward equals the specific payoff entry. For stochastic opponents (Imperfect TFT), the expected reward is a weighted average. For example, if Imperfect TFT has a 90% chance of cooperating:

$$R(s, C) = 0.9 \cdot 3 + 0.1 \cdot 0 = 2.7.$$

2.4.1 Memory-1 Reward Tables

State	$R(s,C)$	$R(s,D)$
(C,C)	3	5
(C,D)	3	5
(D,C)	3	5
(D,D)	3	5

Table 11: Memory-1 rewards vs ALL-C (opponent always cooperates).

State	$R(s,C)$	$R(s,D)$
(C,C)	0	1
(C,D)	0	1
(D,C)	0	1
(D,D)	0	1

Table 12: Memory-1 rewards vs ALL-D (opponent always defects).

State	$R(s,C)$	$R(s,D)$
(C,C)	3	5
(C,D)	3	5
(D,C)	0	1
(D,D)	0	1

Table 13: Memory-1 rewards vs TFT (opponent copies agent's previous action).

State	$R(s,C)$	$R(s,D)$
(C,C)	$0.9 \cdot 3 + 0.1 \cdot 0 = 2.7$	$0.9 \cdot 5 + 0.1 \cdot 1 = 4.6$
(C,D)	$0.9 \cdot 3 + 0.1 \cdot 0 = 2.7$	$0.9 \cdot 5 + 0.1 \cdot 1 = 4.6$
(D,C)	$0.9 \cdot 0 + 0.1 \cdot 3 = 0.3$	$0.9 \cdot 1 + 0.1 \cdot 5 = 1.4$
(D,D)	$0.9 \cdot 0 + 0.1 \cdot 3 = 0.3$	$0.9 \cdot 1 + 0.1 \cdot 5 = 1.4$

Table 14: Memory-1 rewards vs Imperfect TFT (0.9 copies a_{t-1}^A, 0.1 does opposite).

3 Part III: Policy Iteration

We implemented classical tabular Policy Iteration with:

1. **Policy Evaluation:** Solving V^π via iterative Bellman updates until convergence.
2. **Policy Improvement:** Updating $\pi(s)$ to:

$$\pi'(s) = \arg\max_a \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^\pi(s') \right].$$

Convergence was rapid across all opponents due to the small state space.

4 Part IV: Experiments and Analysis

4.1 Experimental Setup

We evaluated Policy Iteration under the following configuration:

- **Discount Factors:** $\gamma \in \{0.1, 0.5, 0.9, 0.99\}$
- **Opponents:** ALL-C, ALL-D, TFT, Imperfect TFT
- **Memory Schemes:** Memory-1, Memory-2
- **Simulation:** 50 episodes $\times$ 50 steps each

Simulation verification. For each opponent and discount factor, after computing the optimal policy via Policy Iteration, we executed the policy in the environment for 50 episodes (50 steps each) and report the average cumulative reward as an empirical validation.

4.2 Results Summary

Opponent	$\gamma = 0.1$	$\gamma = 0.5$	$\gamma = 0.9$	$\gamma = 0.99$
ALL-C	DDDD	DDDD	DDDD	DDDD
ALL-D	DDDD	DDDD	DDDD	DDDD
TFT	DDDD	DDCC	CCCC	CCCC
Imperfect TFT	DDDD	DDCC	CCCC	CCCC

Table 15: Optimal policies (Memory-1).

Opponent	$\gamma = 0.1$	$\gamma = 0.5$	$\gamma = 0.9$	$\gamma = 0.99$
ALL-C	250.00	250.00	250.00	250.00
ALL-D	50.00	50.00	50.00	50.00
TFT	54.00	125.00	150.00	150.00
Imperfect TFT	72.32	120.86	135.66	133.08

Table 16: Average cumulative reward over 50 episodes (each 50 steps), using the optimal Memory-1 policy for each γ.

4.3 Discount Factor Analysis

We varied γ and computed the optimal strategy against different opponents.

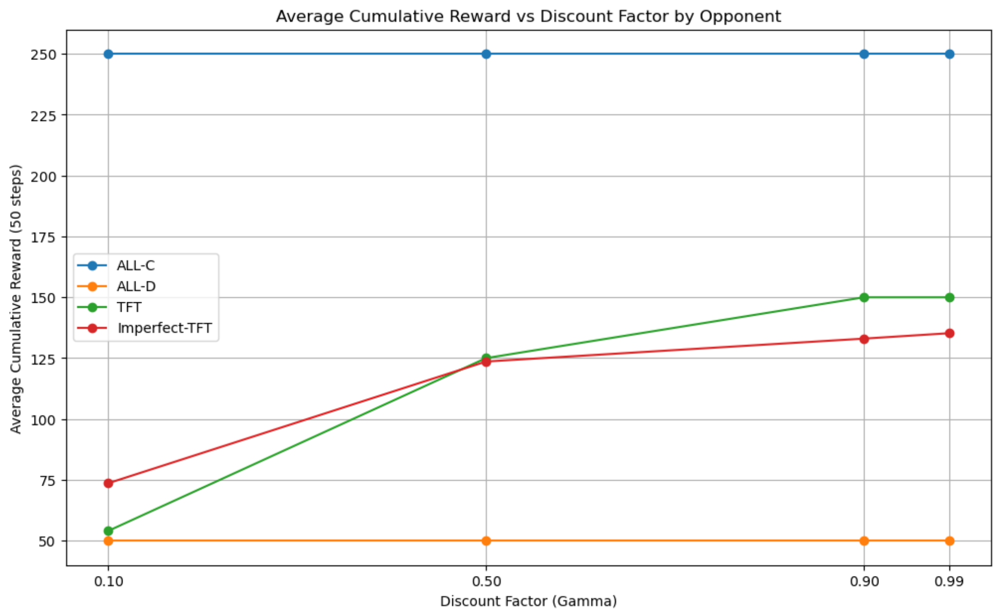

Figure 1: Average Cumulative Reward vs. Discount Factor (γ) for different opponent strategies.

4.3.1 When and Why Cooperation Becomes Optimal

As shown in Figure 1, for the **TFT** opponent, there is a clear phase transition. At low γ values (e.g., 0.1, 0.5), the agent prefers Defection because it values the immediate temptation payoff ($T = 5$) more than the long-term stream of cooperation rewards ($R = 3$). However, as γ increases (e.g., 0.9, 0.99), the agent becomes "far-sighted" enough to value the future rewards of mutual cooperation. It realizes that the long-term penalty of getting punished by TFT outweighs the short-term gain of defecting, making **Cooperation** the optimal policy.

For **ALL-C** and **ALL-D**, the optimal policy is always to Defect regardless of γ, as seen by the constant reward lines. Against ALL-C, defecting exploits the opponent forever. Against ALL-D, defecting is the only defense against being exploited.

In our experiments against **TFT**, we observe a clear transition from defection to cooperation as γ increases. At $\gamma = 0.1$ the optimal policy is `DDDD`, and at $\gamma = 0.5$ it becomes mixed (`DDCC`). When γ is high ($\gamma = 0.9$ and 0.99), the optimal policy is `CCCC` (always cooperate). Therefore, based on the tested values, **cooperation becomes optimal somewhere in the interval $\gamma \in (0.5, 0.9)$**.

4.4 Memory Depth Comparison

We measured average cumulative reward over 50 episodes (each 50 steps). The results:

Opponent	Memory-1 Score	Memory-2 Score
ALL-C	250.0	250.0
ALL-D	50.0	50.0
TFT	150.0	150.0
Imperfect TFT	134.28	135.54

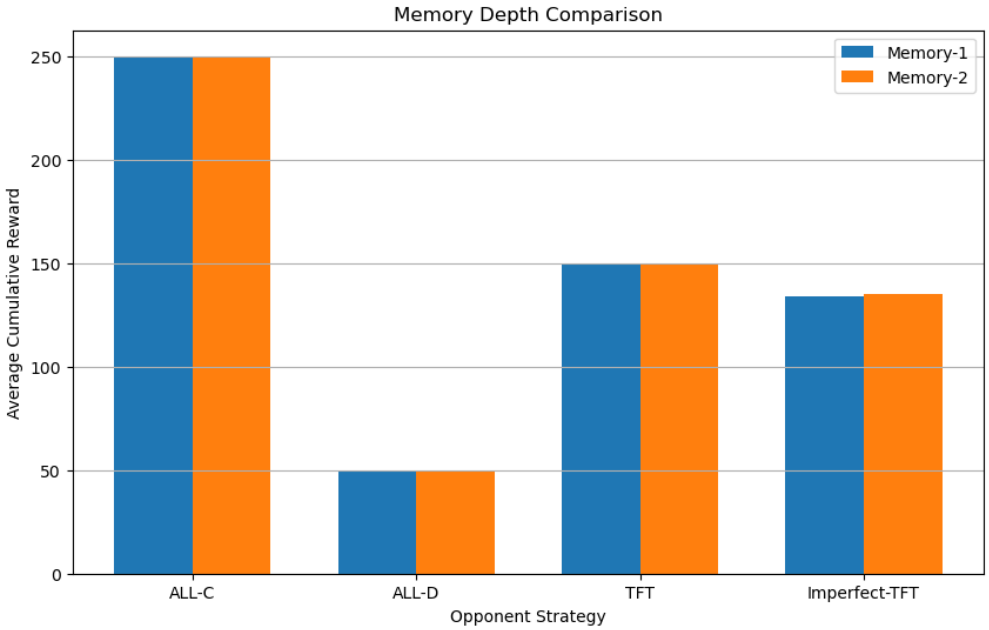

Figure 2: Comparison of Average Cumulative Rewards for Memory-1 vs. Memory-2 agents.

4.4.1 Interpretation

Does the 10% noise break cooperation? No. For the high-discount settings where TFT leads to cooperation (e.g., $\gamma = 0.9, 0.99$), the optimal policy against Imperfect TFT remains cooperative (`CCCC`), although the return decreases.

Forgiving vs. always defect. The agent remains **forgiving**: it does not revert to `DDDD` under 10% noise, and instead sustains cooperation.

Overall, **Memory-2 does not provide a consistent reward improvement** against ALL-C, ALL-D, TFT, or Imperfect TFT, because these opponents depend on at most the agent's last action (or no history at all).

- **ALL-C (Score 250):** The optimal policy against ALL-C is to always Defect. Since ALL-C always Cooperates, the agent receives the temptation payoff $T = 5$ in every round. Over 50 rounds, the total reward is $50 \times 5 = 250$.
- **ALL-D (Score 50):** The optimal policy against ALL-D is to always Defect. Since ALL-D always Defects, the agent receives the punishment payoff $P = 1$ in every round. Over 50 rounds, the total reward is $50 \times 1 = 50$. Cooperating would yield the sucker's payoff $S = 0$, which is worse.
- **Memory-1 vs Memory-2 (Deterministic Opponents):** For ALL-C, ALL-D, and TFT, the scores for Memory-1 and Memory-2 are identical. This is because these opponents' strategies depend only on the immediate past (or no history at all). Having an extra step of memory (Memory-2) provides no additional predictive power or strategic advantage against them.
- **Imperfect TFT (Stochastic Differences):** For Imperfect TFT, the scores differ slightly (134.28 vs 135.54) despite the optimal policy being the same (Cooperate). This difference is

due to the stochastic nature of the opponent (10% noise). The variance in random outcomes across the 50 simulation episodes leads to slightly different average rewards, but statistically, the performance is equivalent.

4.4.2 Hypothetical Opponent Where Memory-2 Helps

We propose a history-dependent opponent called the **Two-Step Punisher (TSP)**. The opponent's action depends on the agent's last *two* actions:

$$a_t^O = \begin{cases} C & \text{if } (a_{t-1}^A, a_{t-2}^A) = (C, C), \\ D & \text{if } a_{t-1}^A = D \text{ and } a_{t-2}^A = C, \\ C & \text{if } a_{t-1}^A = C \text{ and } a_{t-2}^A = D, \\ D & \text{if } (a_{t-1}^A, a_{t-2}^A) = (D, D). \end{cases}$$

This opponent "punishes" a defection for exactly one turn if it was recent ($a_{t-1}^A = D$), and for two turns if the agent defected twice in a row. Importantly, the opponent behaves differently for histories that **appear identical to a Memory-1 agent**.

Why Memory-2 is Required. A Memory-1 agent only observes the most recent pair (a_{t-1}^A, a_{t-1}^O), which collapses multiple distinct two-step histories into a single state. For example, these two Memory-2 states:

$$((C, C), (C, C)) \quad \text{and} \quad ((D, C), (C, C))$$

both appear as the same Memory-1 state:

$$(C, C),$$

yet the opponent's next action differs depending on whether the older action was C or D. Thus, a Memory-1 agent cannot predict the opponent's behavior, whereas a Memory-2 agent can.

4.4.3 Example Interaction (10 Rounds)

Below we illustrate an interaction with TSP where the agent defects once at Round 5. The opponent's behavior depends on the agent's *two-step* history, not the one-step history.

Round	Agent	Opponent	Explanation
1	C	C	Initial cooperation
2	C	C	
3	C	C	
4	C	C	
5	**D**	C	Agent defects (temptation)
6	C	**D**	Opponent punishes: last 2 actions include a D (Round 5)
7	C	C	Opponent sees history (C, D): forgiveness
8	C	C	Cooperation restored
9	C	C	
10	C	C	

Analysis of Point Loss. If the agent cooperated throughout Rounds 5–7, it would have earned:

$$3 + 3 + 3 = 9.$$

Instead:

$$\text{Round 5 payoff: } T = 5, \qquad \text{Round 6 payoff: } P = 1, \qquad \text{Round 7 payoff: } 3.$$

Total:

$$5 + 1 + 3 = 9.$$

Depending on the opponent's exact punishment rule (length and severity), a defection may lead to:

$$(\text{Temptation gain}) - (\text{Punishment loss})$$

which Memory-2 can correctly anticipate, while Memory-1 cannot.

4.4.4 Why Memory-2 Strictly Outperforms Memory-1

The TSP opponent uses two-step history, so the agent requires Memory-2 to optimally respond.

- **Memory-1 Agent (Insufficient Information):** Only sees (a^A_{t-1}, a^O_{t-1}). It cannot distinguish whether the opponent's current defection is due to:
  - a defection one turn ago, or
  - a defection two turns ago.

  Identical Memory-1 states lead to different opponent reactions, so the Memory-1 agent cannot build an optimal policy. It may incorrectly retaliate or fail to return to cooperation at the right time.

- **Memory-2 Agent (Full Information):** Observes $((a^A_{t-2}, a^O_{t-2}), (a^A_{t-1}, a^O_{t-1}))$ and therefore knows:
  - why the punishment started,
  - how long it will last,
  - exactly when cooperation can be safely resumed.

  This allows it to maintain cooperation far more consistently and avoid unnecessary punishment cycles.

Conclusion. Because the opponent's behavioral rule depends on the last **two** agent actions, a Memory-2 agent can correctly interpret and predict the opponent's responses, while a Memory-1 agent cannot distinguish critical histories. Consequently, Memory-2 achieves a strictly higher long-term reward against the Two-Step Punisher opponent.

4.5 Noise Analysis: TFT vs Imperfect TFT

The optimal policies were:

TFT

$$\pi(s) = C \quad \forall s$$

Imperfect TFT

$$\pi(s) = C \quad \forall s$$

4.5.1 Interpretation

Even with 10% noise, cooperation remains optimal.

The noise does *lower* the achievable return:

$$\text{TFT reward: } 150.00, \qquad \text{Imperfect TFT reward: } 134.28$$

But the optimal policy still prefers cooperation over mutual defection, which would yield only 1 point per step.

Thus the agent behaves **forgivingly**: it sustains cooperation even when occasional mistakes occur.

5 Summary

This report presented a comprehensive MDP formulation of the Repeated Prisoner's Dilemma. By implementing Policy Iteration, we derived optimal strategies against both deterministic and stochastic opponents.

We found that:

- **Strategic Foresight:** Cooperation emerges only when the discount factor is high enough, proving that short-term greed is suboptimal in the long run.
- **Memory Requirements:** While Memory-1 is sufficient for standard strategies, we proved that complex, history-dependent opponents like the **Two-Step Punisher** require Memory-2. The **nested state representation** of Memory-2 allows the agent to distinguish histories that appear identical to a Memory-1 agent, enabling optimal punishment avoidance.
- **Resilience:** The optimal policy remains cooperative even against a noisy **Imperfect TFT** opponent, demonstrating that forgiveness is a key component of robust long-term strategy.