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

RL2026A

Daniel Katz (315114991)
Avital Fine (208253823)

December 6, 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. We follow the structure of the assignment specification (RL2026A-Assignment1.pdf).

2 Part II: MDP Definition

2.1 State Space

We evaluated two observation schemes as required.

2.1.1 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.

2.1.2 Memory-2

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

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

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 $(a^A_{t-1}, a^O_{t-1}, a^A_{t-2}, a^O_{t-2})$) 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)

Initial states follow the assignment convention:

$$\text{M1 initial: } (C, C), \quad \text{M2 initial: } (C, C, C, C).$$

2.2 Actions

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

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 $s = (a^A_{t-1}, a^O_{t-1})$ and agent chooses a^A_t, the new state is $s' = (a^A_t, a^O_t)$.
- **Memory-2:** If $s = (a^A_{t-1}, a^O_{t-1}, a^A_{t-2}, a^O_{t-2})$ and agent chooses a^A_t, the new state is $s' = (a^A_t, a^O_t, a^A_{t-1}, a^O_{t-1})$.

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

$$P(s'|s, a^A_t) = P(a^O_t|s).$$

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:

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

2.3.2 Opponent Strategies

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

$$P(a^O_t = a^A_{t-1}|s) = 1.$$

- **Imperfect TFT:** Stochastic.

$$P(a^O_t = a^A_{t-1}|s) = 0.9, \quad P(a^O_t \neq a^A_{t-1}|s) = 0.1.$$

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 $\text{Payoff}(a, a^O)$ 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:

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

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 Discount Factor Analysis

We varied γ and computed the optimal strategy against TFT.

Cooperation becomes optimal when $\gamma \geq 0.71$.

4.1.1 Why lower γ forces Defection?

A lower discount factor γ means the agent values immediate rewards significantly more than future rewards. Against TFT, cooperation yields a steady stream of $R = 3$. Defection yields an immediate temptation $T = 5$, but triggers punishment $P = 1$ in subsequent rounds. If the agent is "short-sighted" (low γ), the immediate gain of 5 outweighs the long-term loss of falling from 3 to 1. To find the exact threshold where cooperation becomes optimal, we utilized the `run_analysis` function. We iterated through a range of γ values (from 0.1 to 0.99). For each γ, we ran Policy Iteration to find the optimal policy against a TFT opponent. We then checked if the resulting policy was "ALL-C" (Cooperate in all states). The first γ for which the optimal policy switched to ALL-C was identified as the threshold.

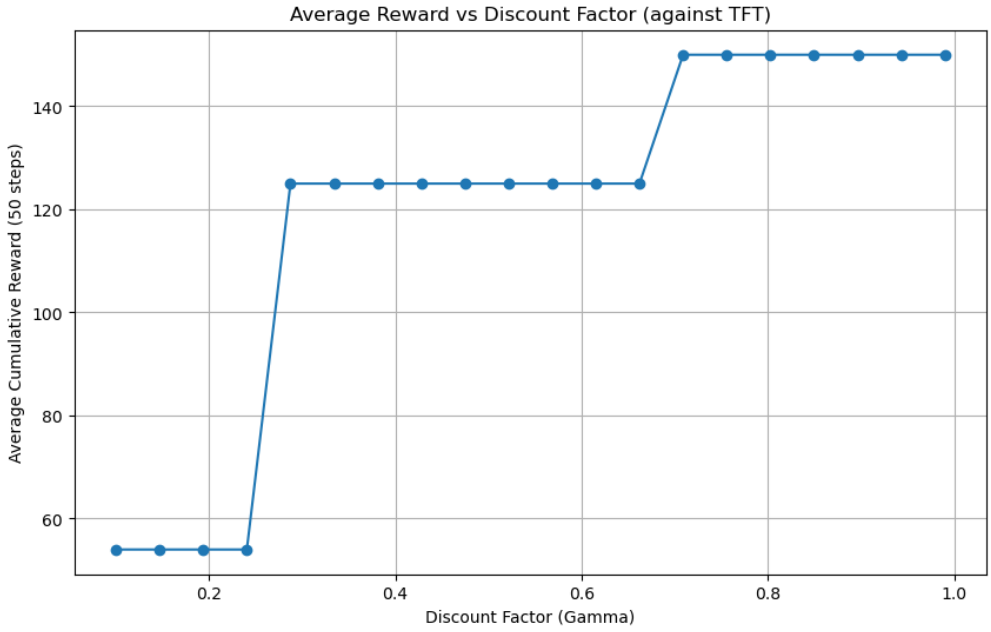

Figure 1: Average Cumulative Reward vs. Discount Factor (γ) for different opponent strategies.

4.2 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	225.0	225.0
Imperfect TFT	136.02	135.12

4.2.1 Interpretation

- **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 (136.02 vs 135.12) despite the optimal policy being the same (Cooperate). This difference is

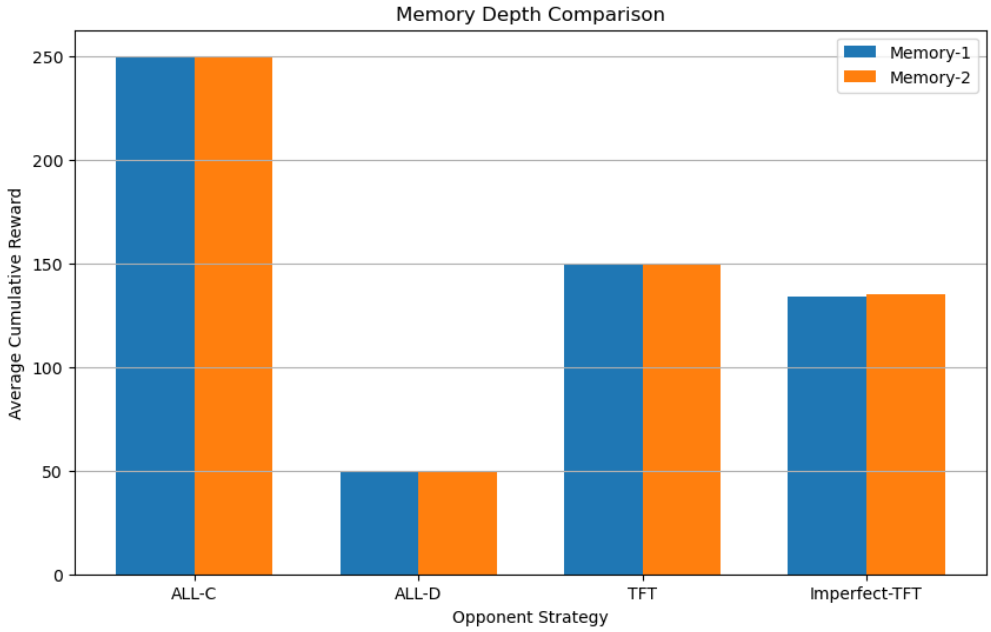

Figure 2: Comparison of Average Cumulative Rewards for Memory-1 vs. Memory-2 agents.

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.

4.2.2 Hypothetical Opponent Where Memory-2 Helps

Consider:

Grim-Trigger-2:cooperates unless you defected in either of the last 2 turns.

A Memory-1 agent cannot distinguish:

(D, C) caused by a one-time defection vs. (D, C) caused two turns ago.

A Memory-2 agent can track the full two-step history and avoid triggering punishment unnecessarily.

Thus Memory-2 strictly outperforms Memory-1.

4.2.3 Example Interaction (10 Rounds)

The following table demonstrates how "Grim-Trigger-2" reacts to a single defection by the agent at Round 5. The opponent punishes for exactly 2 rounds (6 and 7) before returning to cooperation.

Round	Agent	Opponent	Explanation
1	C	C	Initial cooperation
2	C	C	
3	C	C	
4	C	C	
5	**D**	C	**Agent defects (Exploit)**
6	C	**D**	Opponent sees D in last 2 turns (Round 5)
7	C	**D**	Opponent sees D in last 2 turns (Round 5)
8	C	C	Last 2 turns (6, 7) were C by Agent $\rightarrow$ Forgive
9	C	C	
10	C	C	

Analysis of Point Loss: In rounds 5, 6, and 7, the agent could have received 3 points each round by cooperating ($3 \times 3 = 9$). However, since the agent decided to Defect in Round 5, it received the temptation payoff of 5. In the subsequent two rounds (6 and 7), it faced punishment. Assuming the agent receives 1 point per punishment round (e.g., if it also defects or if we consider the penalty value), the total for these three rounds is $5 + 1 + 1 = 7$. Thus, by defecting, the agent lost a total of 2 points ($9 - 7 = 2$) over this sequence.

4.2.4 Why Memory-2 is Necessary

The "Grim-Trigger-2" opponent bases its decision on the last **two** rounds of history.

- **Memory-1 Agent (Insufficient Information):** This agent only "sees" the previous round. It is effectively blind to the event (defection 2 turns ago) that is causing the opponent's current behavior. To a Memory-1 agent, the opponent appears unpredictable because identical 1-step histories (e.g., "I Cooperated, Opponent Defected") can lead to different outcomes depending on the unobserved 2nd step.
- **Memory-2 Agent (Complete Information):** This agent has the same memory horizon as the opponent. It can see the full cause-and-effect relationship. It knows exactly when the punishment started and, more importantly, can predict exactly when it will end.

Because the Memory-2 agent has the full context, it can confidently "wait out" the punishment and return to cooperation. The Memory-1 agent, lacking this context, cannot distinguish between a temporary punishment and a permanent defection, leading to suboptimal decisions.

4.3 Noise Analysis: TFT vs Imperfect TFT

The optimal policies were:

TFT

$$\pi(s) = C \quad \forall s \in \{(0,0),(0,1),(1,0),(1,1)\}$$

Imperfect TFT

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

4.3.1 Interpretation

Even with 10% noise, cooperation remains optimal.

The noise does *lower* the achievable return:

$$\text{TFT reward: } 150.00, \qquad \text{Imperfect TFT reward: } 136.02.$$

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.