

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

RL2026A

Student 1 (ID1)

Student 2 (ID2)

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:

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

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:

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

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

Initial states follow the assignment convention:

M1 initial: (C, C) , 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 $s = (a_{t-1}^A, a_{t-1}^O)$ and agent chooses a_t^A , the new state is $s' = (a_t^A, a_t^O)$.
- **Memory-2:** If $s = (a_{t-1}^A, a_{t-1}^O, a_{t-2}^A, a_{t-2}^O)$ and agent chooses a_t^A , the new state is $s' = (a_t^A, a_t^O, a_{t-1}^A, a_{t-1}^O)$.

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:

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

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 $\text{Payoff}(a, a^O)$ 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 $\gamma \geq 0.71$.

For $\gamma < 0.71$, the agent is too short-sighted and defects to exploit the immediate temptation payoff $T = 5$.

4.2 Memory Depth Comparison

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

Opponent	Memory-1	Memory-2	Better?
ALL-C	250.00	250.00	No
ALL-D	50.00	50.00	No
TFT	150.00	150.00	No
Imperfect TFT	136.02	135.12	No

4.2.1 Interpretation

Memory-2 did not outperform Memory-1 for any opponent. This matches theoretical expectations: none of the four opponents use a history longer than one step, so additional memory provides no advantage.

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.

4.3 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:

TFT reward: 150.00, 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.

5 Executive Summary

- Cooperation becomes optimal against TFT when $\gamma \geq 0.71$.
- Memory-2 provided no advantage against any of the four assignment opponents.
- A hypothetical “Grim-Trigger-2” opponent would make Memory-2 strictly better.
- Imperfect TFT reduces long-term reward but does *not* break the optimal cooperative policy.

References

References

- [1] RL2026A Assignment 1: Policy Iteration in the Repeated Prisoner's Dilemma.
Provided PDF: RL2026A-Assignment1.pdf [?].