Manipulation of an Agent using Physical Deception

Beatriz Gomes
Instituto Superior Técnico
Lisbon, Portugal
107432

Francisco Ferreira
Instituto Superior Técnico
Oeiras, Portugal
96861

Sofia Pinho
Instituto Superior Técnico
Lisbon, Portugal
99272

ABSTRACT

In this environment, agents aim to efficiently navigate towards a target landmark while avoiding an adversary. Good agents are rewarded for proximity to the target landmark and penalized for adversary proximity to that same target. Meanwhile, the adversary seeks the target without knowledge of its identity. Rewards are based on unscaled Euclidean distances, motivating good agents to strategically distribute themselves to mislead the adversary.

ACM Reference Format:
Beatriz Gomes, Francisco Ferreira, and Sofia Pinho. 2024. Manipulation of an Agent using Physical Deception. In . ACM, Lisbon, Lx, Portugal, 5 pages.

1 INTRODUCTION

1.1 Problem Definition

Drawing inspiration from seminal work such as the *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments* [1] paper presented at the Neural Information Processing Systems (NIPS) conference in 2017, we opted to construct a scenario known as ***Physical Deception***. In this environment, a group of agents collaborates to reach a single target landmark from multiple potential landmarks while contending with the presence of an adversary. The objective is clear: minimize the distance of any agent to the target landmark. However, the adversary's lack of knowledge regarding the true target location adds a strategic twist, compelling the cooperative agents to employ tactics of deception and misdirection to confound its pursuit.

Through this environment, agents learn not only to collaborate effectively to achieve their shared goal but also to develop sophisticated strategies to outmaneuver the adversary. By rewarding agents based on their proximity to the target landmark and penalizing them for the adversary's closeness to the target, we incentivize strategic dispersion among potential landmarks, fostering a dynamic interplay of cooperation and competition.

1.2 Motivation

Our project, inserted in the Autonomous Agents and Multi Agent Systems course, aims to delve deep into this environment, unraveling the complexities of cooperative behavior and adversarial interaction within multi-agent systems. By leveraging reinforcement learning techniques, we seek to develop models capable of navigating the intricate dynamics of Physical Deception.

, ,

2024. ACM ISBN

1.3 Implementation

Regarding the implementation, we decided to use the lab code as a foundation. Therefore, the base game was implemented in *Python*, using the libraries *gym*, *ma_gym* and *PIL* for the environment and visual aspect. For the Q-Learning model we used *PyTorch*. Finally, to compare results and learning progress, we used *Matplotlib* and *IPython*.

2 SYSTEM ARCHITECTURE

2.1 Multi-Agent System

There are two different type of agents:

- **Good Agents:** Comprise a team of two. Aim to maximize their distance to the target landmark while maximizing the distance between the adversary and the target. They know the target landmark's location and must work together to deceive the adversary.
- **Adversary:** Lone agent. Attempts to find and approach the target landmark without knowing its identity, relying on the movements of the good agents to make inferences.

The core objective for the good agents is to develop strategies that leverage deception and cooperation to mislead the adversary, ensuring the latter moves towards incorrect landmarks. The adversary's goal is to infer the target landmark based on the good agents' behavior and approach it.

2.2 Environment

The environment is represented as a (11, 11) board with two landmarks. Each agent and adversary can perform five actions: stay, move up, move down, move left, or move right. After a set of 20 steps, the teams' scores are calculated and the one with the highest score is the winner.

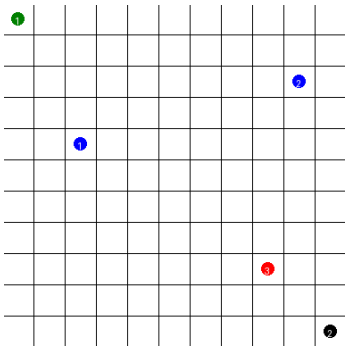

Figure 1: Board with two distinct landmarks (real one in green), featuring agents depicted in blue and adversary in red

2.3 Game Scores

The score system we designed is crafted to ensure a fair and competitive game between the teams.

2.3.1 Notation and Definitions.

- **Agents' Positions:**
  - $\mathbf{a}_i$: Position of good agent i
  - $\mathbf{b}$: Position of the adversary (adversary)
  - $\mathbf{l}_R$: Position of the real landmark
- **Distance Function:** The Euclidean distance between two positions $\mathbf{x}$ and $\mathbf{y}$ is defined as:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

- **Maximum Distance:** The maximum possible distance in the grid, used for normalization, is calculated as:

$$D_{\max} = \sqrt{\text{grid\_width}^2 + \text{grid\_height}^2}$$

2.3.2 Score Calculation.

(1) **Adversary Score:** The score for the adversary is inversely proportional to its distance to the real landmark. This ensures that the closer the adversary is to the landmark, the higher its score, making it a tangible objective for the adversary:

$$S_B = -d(\mathbf{b}, \mathbf{l}_R)$$

(2) **Good Agents Score:** The score for the good agents is based on the closest good agent's distance to the real landmark. This ensures that the closer the good agents are to the landmark, the higher their score, creating a competitive dynamic between the agents:

$$d_{\text{closest\_good}} = \min\{d(\mathbf{a}_i, \mathbf{l}_R)\}$$

$$S_G = -d_{\text{closest\_good}}$$

(3) **Good Agents Penalty:** To maintain fairness and challenge, the good agents receive a penalty based on the adversary's proximity to the real landmark. This penalty increases as the adversary gets closer, making it harder for the good agents to score if the adversary is too close:

$$P_G = -\log(D_{\max} - d(\mathbf{b}, \mathbf{l}_R) + 1)$$

(4) **Combined Good Agents Score:** The total score for the good agents is a combination of their proximity-based score and the penalty for the adversary's proximity. This ensures that the good agents' score reflects both their success in approaching the landmark and their effectiveness in keeping the adversary away:

$$S_{\text{combined\_good}} = S_G + P_G$$

2.3.3 Summary. The score system is designed to maintain the fairness and competitiveness of the game by:

- Rewarding the adversary for getting closer to the real landmark.
- Rewarding the good agents based on their proximity to the real landmark.
- Penalizing the good agents when the adversary gets too close to the landmark.

3 ADAPTING Q-LEARNING

In our project, we used *PyTorch* to train ***Q-learning***, in order to develop a strategy for the good agents. The key components of this adaptation include defining the states, actions, rewards, and the policy used by the agents.

We chose *Q-learning* as our learning algorithm due to 2 main factors:

- Since *Q-learning* is an off-policy algorithm, the agents were trained maximizing the expected future reward, independently of the action chosen by the current policy. On the other hand, *SARSA* is on-policy. meaning it updates the Q-values strictly based on the current policy.
- *Q-learning* reaches the most optimal policy after some time, because it uses a more aggressive approach, while *SARSA* may converge to a sub-optimal policy, because it is conservative.

3.0.1 States. In our environment, the states s includes boolean values indicating the relative directions between the agent and:

- The real landmark;
- The fake landmark;
- The adversary;
- The other good agent.

Given the small board setup, the state space is manageable but still complex enough to require strategic decision-making.

3.0.2 Actions. Each agent and adversary can perform one of five actions at any time step:

- Stay in the current position;
- Move up;
- Move down;
- Move left;
- Move right.

These actions allow agents to navigate the board and adjust their positions relative to the landmarks and each other.

3.0.3 Greedy Adversary. The greedy adversary is a baseline agent used to evaluate the performance of the learning agents. Its strategy is straightforward: it moves towards the landmark that has the closest good agent. This approach ensures that the adversary always targets the landmark that is most contested, providing a constant challenge for the good agents.

The greedy adversary determines its actions by following these steps:

(1) It identifies the positions of all agents and landmarks on the board.
(2) For each landmark, it calculates the distance between the landmark and the closest good agent.
(3) It then selects the landmark that has the closest good agent to it. If the distances are equal a landmark is picked at random.
(4) The adversary moves towards this selected landmark, choosing the direction that minimizes the distance to the landmark.

This method ensures that the adversary is always focused on disrupting the most immediate threat from the good agents. By constantly moving towards the most contested landmark, the greedy adversary provides a robust adversarial environment that helps in evaluating and improving the strategies of the good agents.

3.0.4 Rewards. The reward system is designed to encourage strategic behavior in the good agents:

Good Agents' Rewards:

- Positive reward for reducing the distance to the target landmark and winning.
- Negative reward if the adversary gets closer to the target landmark and losing.

$$\text{reward} = \begin{cases} 10 + S_{\text{agents}} - S_{\text{adversary}} & \text{if } S_{\text{agents}} > S_{\text{adversary}} \\ -10 + S_{\text{agents}} - S_{\text{adversary}} & \text{otherwise} \end{cases} \quad (1)$$

The reward structure incentives the good agents to balance between getting close to the target and misleading the adversary.

3.0.5 Convergence. *Q-learning*'s iterative process helps our agents converge towards optimal policies. Over many episodes, as agents repeatedly experience different states and actions, the Q-values stabilize, representing the expected rewards of taking specific actions from specific states. This convergence is critical for developing reliable strategies in our competitive environment.

3.0.6 Policy Execution. After sufficient training, the policy for each agent is derived by selecting the action with the highest Q-value in each state. The good agents then use this policy to minimize their distance to the target while maximizing the adversary's distance, effectively implementing the decoy and distribution strategies.

4 EVALUATION

4.1 Random Agents vs Greedy Adversary

In this scenario, the agents start on random positions on the grid each game, with the landmarks fixed in the corners.

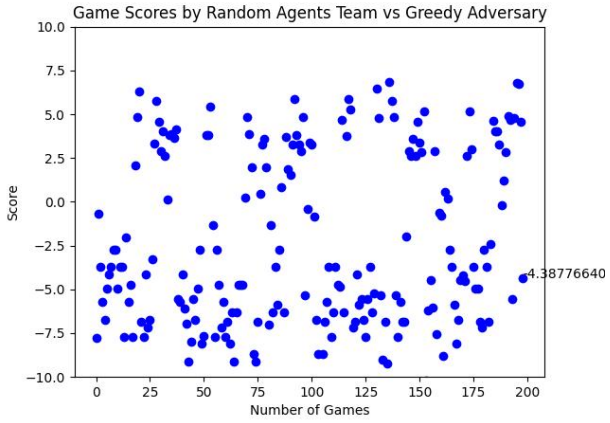

Figure 2: Progress Score by Random Agents Team

Figure 2 displays the scores of 200 games where a team of agents performing random movements competes against a greedy adversary. The key observations from this plot are:

High Variability: The scores fluctuate widely between -10 and +10, indicating a high variability in game outcomes.

Lack of Trend in Performance: There is no discernible upward or downward trend over the number of games. The average score appears to hover around zero, but with a slight negative bias, suggesting that the greedy adversary often has a slight advantage.

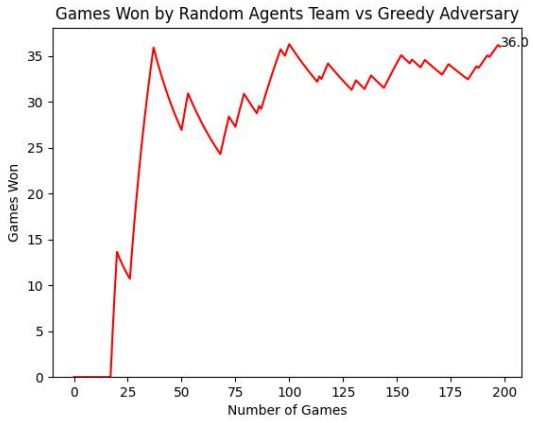

Figure 3: Games Won by Random Agents Team

Figure 3 tracks the cumulative number of games won by the random agents over 200 games.

The random agents win some games early on, indicating that random actions occasionally lead to favorable outcomes against the greedy adversary. By the end of 200 games, the random agents have won 36 games, translating to an 18% win rate. This relatively low win rate indicates that the random strategy is generally ineffective against the greedy adversary.

4.2 Learning Agents vs Greedy Adversary

4.2.1 Fixed Landmarks and Fixed Starting Agent Position. In this scenario, the agents all start in position [5,5] on the grid, with the landmarks fixed in the corners. We can see that this position is initially advantageous for the adversary, since in the first 10 games the win-rate tends towards 0%.

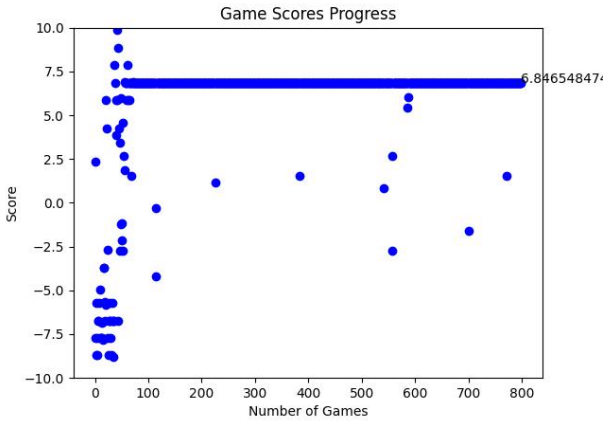

Figure 4: Progress Score by Learning Agents Team

Early Score Volatility: In the first 100 games, there is significant volatility in the scores, reflecting the agents' early exploration phase. During this period, the agents are experimenting with various strategies, resulting in a wide range of scores.

Strategy Development: After the initial phase, from approximately game 100 onwards, the annotation shows that the score stabilises at around 6.8, reflecting the agents' ability to maintain a high average performance after the initial learning period.

There are occasional outliers even after the stabilization phase, suggesting that while the dominant strategy is highly effective, there are still instances where the adversary can disrupt the agents' performance.

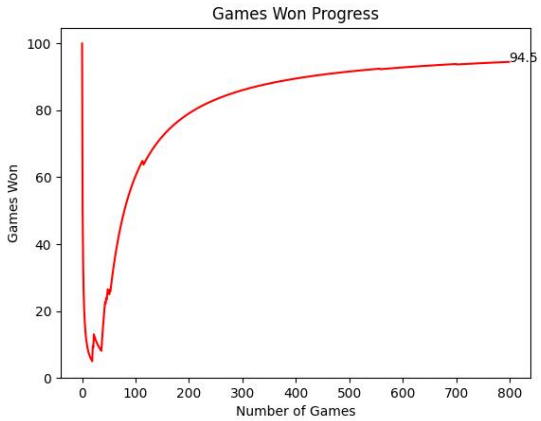

Figure 5: Games Won by Learning Agents Team

Figure 5 tracks the cumulative number of games won by the agents over the same 800-game span.

Interestingly, in the first 10 games, the win rate drops sharply to nearly 0%. This suggests that the initial starting position ([5,5] on the grid) is advantageous for the adversary, leading to early losses for the agents.

Following the initial losses, the agents quickly learn and adapt their strategies, leading to a steep increase in the number of wins. By game 100, the agents have significantly improved their performance.

From game 100 onwards, the win rate plateaus at around 94.5%. This indicates that the agents have developed a robust strategy that allows them to win over 90% of the games consistently.

4.2.2 Fixed Landmarks and Random Starting Agent Position. In this scenario, the agents start on random positions on the grid each game, with the landmarks fixed in the corners. This proves to be a more challenging learning experience.

Figure 6 illustrates the progress of game scores over 1500 games, where the score for each game is plotted against the number of games played.

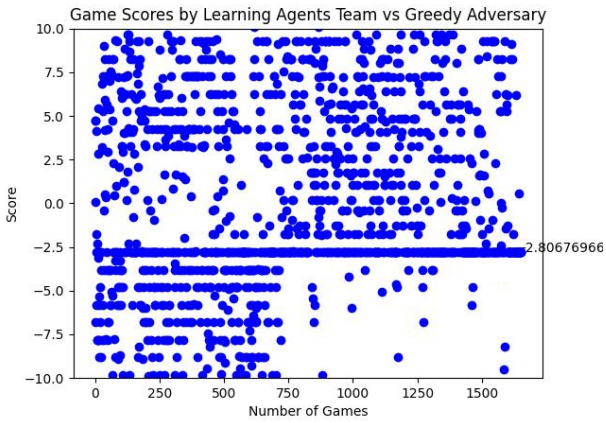

Figure 6: Progress Score by Learning Agents Team

The data reveals several patterns, which can be observed in the training run represented in Figure 6:

8**Initial Volatility:** The initial games exhibit a wide range of scores, reflecting the agents' early exploration phase. During this period, the agents are experimenting with various strategies, resulting in significant score fluctuations.

Stabilization: After approximately 500 games, the scores begin to stabilize around a central value, -2.8 and upwards. This is the highest score the agents team can reach if the adversary reaches the correct landmark. Which indicates that at this point the agents have devised a strategy in which at least one of them always stays in the correct landmark position, even if it leads the adversary to it.

The absence of higher positive scores implies that there is still room for improvement and that the agents have not yet discovered a more optimal strategy that could lead to consistently higher scores.

Occasional Dips: These outliers suggest that agents occasionally attempt new strategies. This is indicative of ongoing fine-tuning and adaptation in response to the adversary's behavior.

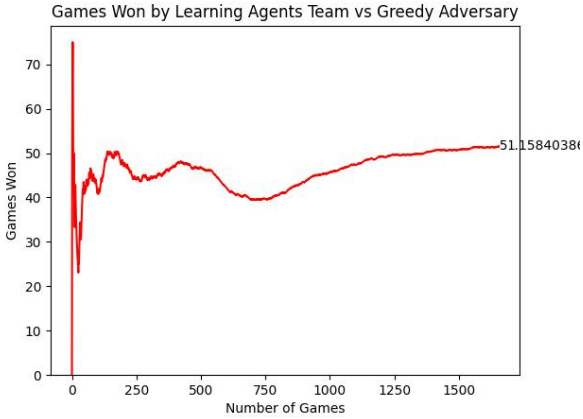

Figure 7: Games Won by Learning Agents Team

Figure 7 tracks the cumulative number of games won by the agents over the same 1500-game span.

Initial Exploration: In the first 200 games, the agents engage in extensive exploration. During this period, the cumulative wins

increase as the agents are experimenting with different actions and learning the environment dynamics.

Wins Decrease: Between 200 and 500 games, there is a noticeable decrease in the win rate. This phase corresponds to the period where the agents' have the *First Stabilization* described before.

Steady Increase: After 500 games, the agents' win rate starts to increase steadily. The win rate plateaus at around 50%. This period marks the convergence towards more effective strategies, where the agents have learned to consistently outmaneuver the adversary and secure wins.

4.3 Comparison with *Q-Learning* Agents' Performance

In contrast, when using *Q-learning* agents, the performance shows significant improvement over time, as seen in the results provided earlier.

Unlike random agents, *Q-learning* agents show a clear upward trend in performance, indicating learning and adaptation to the environment.

The average game scores for *Q-learning* agents tend to improve, often leading to positive scores, suggesting that the *Q-learning* agents are effectively optimizing their actions to achieve better outcomes.

Q-learning agents achieve a higher win rate compared to random agents. This is evident from the gradual and consistent increase in games won, reflecting the effectiveness of the learning algorithm in overcoming the greedy adversary.

5 CONCLUSION

In this project, we developed an environment where cooperative agents aim to minimize their distance to a target landmark while contending with an adversary attempting to deduce and approach the target based on the agents' behavior. Our primary goal was to investigate the effectiveness of *Q-learning*.

Through our experiments, we demonstrated the significant advantages of *Q-learning* over random action selection in this multi-agent scenario. The *Q-learning* agents showed a clear trend of improvement over games, learning to effectively deceive the adversary and optimize their approach to the target landmark. This learning was evidenced by an upward trend in game scores and a higher win rate compared to random agents.

The comparison with random agents, who displayed highly variable performance without any improvement over games, highlighted the importance of learning algorithms in such environments. *Q-learning* enabled the agents to develop and refine their strategies, achieving a balance between approaching the target and misleading the adversary, as seen in the steady increase in their performance metrics.

6 FUTURE WORK

Developing adversaries that also learn and adapt over time would create a more challenging and dynamic environment. This would test the robustness and adaptability of the good agents' strategies, ensuring that they can handle evolving threats and maintain their effectiveness under pressure. Moreover, expanding the environment to include more agents and landmarks would increase the complexity and scalability of the system, helping to understand how well the agents' strategies scale with the problem size.

Lastly, developing a *Q-learning* model where agents learn to navigate and strategize with randomly placed landmarks would enhance the agents' adaptability and generalization. This approach prepares them for more unpredictable and varied environments, ensuring that their strategies remain effective even when the conditions change.

By addressing these future directions, we can deepen our understanding of multi-agent interactions and enhance the practical applicability of reinforcement learning in complex environments.

REFERENCES

[1] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, Igor Mordatch (2017). Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. Neural Information Processing Systems (NIPS) conference (2017). Retrieved from https://doi.org/10.48550/arXiv.1706.02275