ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS
DEPARTMENT OF DIGITAL SYSTEMS

NCSR DEMOKRITOS
INSTITUTE OF INFORMATICS AND TELECOMMUNICATIONS

Ranking Joint Policies in Dynamic Games using Evolutionary Dynamics

by

Natalia Koliou

Submitted

in partial fulfilment of the requirements for the degree of

Master of Artificial Intelligence

at the

UNIVERSITY OF PIRAEUS

December 2024

Author . . . .

Natalia Koliou
II-MSc “Artificial Intelligence”
December 23, 2024

Certified by. . . . .

George Vouros
Professor
Thesis Supervisor

Certified by. . . . .

Orestis Telelis
Professor
Member of Examination Committee

Certified by. . . . .

Christos Spatharis
Researcher
Member of Examination Committee

Ranking Joint Policies in Dynamic Games using Evolutionary Dynamics

By

Natalia Koliou

Submitted to the II-MSc "Artificial Intelligence" on
December 28, 2024,
in partial fulfillment of the
requirements for the MSc degree

Abstract

Game-theoretic solution concepts, such as the *Nash equilibrium*, have been key to finding stable joint actions in multi-player games. However, it has been shown that the dynamics of agents' interactions, even in simple two-player games with few strategies, are incapable of reaching *Nash equilibria*, exhibiting complex and unpredictable behavior. Instead, evolutionary approaches can describe the long-term persistence of strategies and filter out transient ones, accounting for the long-term dynamics of agents' interactions. Our goal is to identify agents' joint strategies that result in stable behavior, being resistant to changes, while also accounting for agents' payoffs, in dynamic games. Towards this goal, we propose transforming dynamic games into their empirical forms by considering agents' strategies instead of agents' actions, and applying the evolutionary methodology α*-Rank* to evaluate and rank strategy profiles according to their long-term dynamics. This methodology not only allows us to identify joint strategies that are strong through agents' long-term interactions, but also provides a descriptive, transparent framework regarding the high ranking of these strategies. Experiments report on agents that aim to collaboratively solve a stochastic version of the graph coloring problem. We consider different styles of play as strategies to define the empirical game, and train policies realizing these strategies, using the DQN algorithm. Then we run simulations to generate the payoff matrix required by α*-Rank* to rank joint strategies.

Thesis Supervisor: George Vouros
Title: Professor, University of Piraeus

List of Abbreviations

CGT	Classical Game Theory
CNN	Convolutional Neural Network
DQN	Deep Q-Network
EGT	Evolutionary Game Theory
EGTA	Evolutionary Game Theory Analysis
ESS	Evolutionary Stable Strategy
FC	Fully Connected
GCG	Graph Coloring Game
GCP	Graph Coloring Problem
GTA	Game Theory Analysis
MCC	Markov-Conley Chain
NN	Neural Network
SCC	Strongly Connected Component

Acknowledgments

I would like to thank my professor, George Vouros, for his invaluable support and guidance throughout the process of writing this thesis. His insights and feedback have been incredibly helpful and inspiring. I am especially grateful for the opportunity to collaborate with him on a paper, built upon this research, which we have submitted to the 24th International Conference on Agents and Multi-Agent Systems (AAMAS 2025). Working together on this paper has been a truly rewarding experience. I look forward to many more collaborations in the future.

A heartfelt thank you goes to my colleagues at NCSR ”Demokritos”, and especially my boss, Stasinos Konstantopoulos, for his constant patience and support. Balancing my work responsibilities with my master’s studies was challenging, but their understanding made it possible for me to manage both. While our work projects were not directly related to my thesis, the experience I gained was invaluable. It enhanced my research skills, refined my writing, and strengthened my technical abilities, all of which were crucial for completing this thesis.

I would also like to thank my wonderful piano professor at Athens Conservatoire, Theodore Tzovanakis. Throughout this year, he has been an incredible source of emotional and mental support. Music has always been a big part of my life, helping me clear my mind and get inspired. They say inspiration comes from unexpected places, and for me, it comes from my deep love for music. A big thank you for all the beautiful moments we’ve shared together.

Finally, I am deeply grateful to my parents, George and Foteini, for their unconditional love and support. They cared for me not only physically but also emotionally, creating a nurturing environment that allowed me to focus on my work and studies. Whether helping with housework when I was overwhelmed or encouraging me when I felt defeated, they always believed in me and stood by my side. Even during difficult times, they did everything they could to keep the stress away, making sure I stayed commited to my goals. Their strength and encouragement helped me stand back up when I stumbled, and keep pushing forward, even when giving up felt easier. Thank you mom and dad!

Any opinions, findings, conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the «funding body» or the view of University of Piraeus and Institute of Informatics & Telecommunications of NCSR “Demokritos”.

Contents

List of Figures

List of Tables

I Scientific Paper

This section presents the paper accepted to the 24th International Conference on Agents and Multi-Agent Systems (AAMAS 2025). It provides a summary of the theoretical background, including the problem definition, proposed methodology, and experimental results, primarily serving as a snapshot of the research.

The paper is included below in its original form to reflect the work as it was submitted.

Ranking Joint Policies in Dynamic Games using Evolutionary Dynamics

Natalia Koliou
University of Piraeus
Piraeus, Greece
nataliakoliou@iit.demokritos.gr

George Vouros
University of Piraeus
Piraeus, Greece
georgev@unipi.gr

ABSTRACT

Game-theoretic solution concepts, such as the *Nash equilibrium*, have been key to finding stable joint actions in multi-player games. However, it has been shown that the dynamics of agents' interactions, even in simple two-player games with few strategies, are incapable of reaching *Nash equilibria*, exhibiting complex and unpredictable behavior. Instead, evolutionary approaches can describe the long-term persistence of strategies and filter out transient ones, accounting for the long-term dynamics of agents' interactions. Our goal is to identify agents' joint strategies that result in stable behavior, being resistant to changes, while also accounting for agents' payoffs, in dynamic games. Towards this goal, and building on previous results, this paper proposes transforming dynamic games into their empirical forms by considering agents' strategies instead of agents' actions, and applying the evolutionary methodology *α-Rank* to evaluate and rank strategy profiles according to their long-term dynamics. This methodology not only allows us to identify joint strategies that are strong through agents' long-term interactions, but also provides a descriptive, transparent framework regarding the high ranking of these strategies. Experiments report on agents that aim to collaboratively solve a stochastic version of the graph coloring problem. We consider different styles of play as strategies to define the empirical game, and train policies realizing these strategies, using the DQN algorithm. Then we run simulations to generate the payoff matrix required by *α-Rank* to rank joint strategies.

KEYWORDS

Evolutionary Dynamics, Empirical Games, Stochastic Games, Deep Reinforcement Learning, Ranking Joint Strategies

ACM Reference Format:
Natalia Koliou and George Vouros. 2025. Ranking Joint Policies in Dynamic Games using Evolutionary Dynamics. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 10 pages.

1 INTRODUCTION

Game theory studies agents' strategies not only in terms of optimality of performance but also with regard to stability of agents' behavior. Game-theoretic solution concepts, particularly the *Nash equilibrium*, have played an important role in this research. However, solution concepts do not account for the long-term dynamics of agents' interactions, which are important in dynamic settings. In static games, where payoff matrices are known, studying solution concepts is relatively straightforward. For example, consider the payoff matrix for the Rock-Paper-Scissors game in Table 1. The mixed strategy *Nash equilibrium* occurs when both players randomize their choices uniformly across Rock, Paper, and Scissors. In dynamic settings involving sequential decision making, one must account for the dynamics of agents' interactions over time. In these settings, we need to analyze agents' behavior in terms of their payoffs, identifying joint strategies that result into agents' stable behaviors. Evolutionary approaches have shown great potential towards this aim.

Table 1: Payoff matrix for the Rock-Paper-Scissors game.

	Rock	Paper	Scissors
Rock	0,0	-1,1	1,-1
Paper	1,-1	0,0	-1,1
Scissors	-1,1	1,-1	0,0

To study agents' behavior in multi-agent dynamic settings, researchers often train deep learning models to learn joint policies. These models, either in collaborative or competitive settings, are usually trained with the ultimate objective to result into Nash equilibria, aiming to agents' stability of behavior, where no agent has an incentive to deviate from their joint policy. In complex dynamic settings with long-term dynamics of agents' interactions, there is no guarantee of reaching that objective and there is no way to reveal the reasoning behind the agents' choice of a policy instead of another. Although proposals towards explainability and interpretability of models are important, these aim to provide either explanations for the policy as a whole (i.e. agent's style of play) or about individual decisions. In our case, we need a descriptive framework to account for transparency regarding the strength of agents' joint policies, accounting for long-term dynamics.

Our goal is to identify agents' joint policies that result in stable behavior, being resistant to changes, while also accounting for agents' payoffs, in dynamic games. This is motivated by the need to identify joint strategies of agents, whether human or software, that need to act as co-players in a common setting. To address this, we propose using a descriptive evolutionary framework that accounts for long-term agents' interactions. We conjecture that this approach helps agents select strong (i.e. non transient) policies when playing with or against other agents in dynamic settings where their actions affect future states and decisions. Identifying these joint policies and understanding their "superiority" is important, particularly in collaborative scenarios where agents must choose strategies while interacting with humans that use specific styles of problem-solving.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), A. El Fallah Seghrouchni, Y. Vorobeychik, S. Das, A. Nowe (eds.), May 19 – 23, 2025, Detroit, Michigan, USA.

Towards this goal we propose exploiting multiple policy models, each realizing a distinct style of play (*strategy*), and then defining an empirical game for evaluating agents' joint performance when they play using various strategy profiles. This empirical game is exploited by the α-*Rank* evolutionary framework [5] to evaluate the evolutionary dynamics of agents' strategies over time, ultimately identifying which ones prevail in the long run.

Although this work builds on the α-*Rank* framework, it contributes a perspective for evaluating individual agents' strategies in stochastic, sequential decision making settings, when they act with other agents following specific styles of play. In so doing, we do:

- Describe a concise methodology for evaluating and ranking agents' joined policies, accounting for their long-term interactions in dynamic settings, using the α-*Rank* evolutionary framework.
- Demonstrate this methodology in multi-agent graph coloring dynamic games, defining multiple styles of play (i.e strategies) per agent.
- Show how agents' choices of strategies—and the policies realizing them—can be transparently justified, by means of a descriptive framework.

2 BACKGROUND

In this section, we outline the key concepts necessary to follow the proposed approach.

2.1 Dynamic Games

Dynamic games describe agent interactions along the time dimension. As opposed to static games, where players execute single, one-shot actions, dynamic games involve a series of decisions made by each of the players at subsequent points in time. A key property of dynamic games is that the actions taken at any given moment influence the future states of the system and future decisions made. These temporal dependencies require players to consider the long-term consequences of their actions. A dynamic game can be represented as a tuple $G = (S, K, A, T, P)$, where S represents a finite set of states, K is the set of players, and $A = (A^k \times A^{-k})$ is the set of joined actions, with A^k corresponding to the action set available to player k. A^{-k} denotes the action set available to players other than k. The transition matrix T describes how states evolve over time, determining the next state of the system based on the current state and the actions chosen by the players. Finally, $P^k : S \times (A^k \times A^{-k}) \times S \rightarrow \mathbb{R}$ is the payoff function for player k, given the current joint state, the action chosen by player k and the actions of the other agents, and the resulting state.

In this work we focus on stochastic dynamic games, as introduced by L.S. Shapley in 1953 [7]. In stochastic games, the outcome of players' actions is influenced by probabilistic events, making future states of the game uncertain. These games are often referred to as Markov games [8]. Therefore, in stochastic games, the transition function T is defined as a probability distribution over next states. Specifically, $T : S \times A \rightarrow \Delta(S)$, where $\Delta(S)$ is a probability distribution over the states, given a state and joint action. For example, in poker, while players' actions do influence the outcome, the next state of the game also depends on luck, such as drawing a strong hand like a flush or a weak hand like a pair of twos. In such games, players, when planning their actions, must account for both the actions of their opponents and the dynamics of the environment.

In dynamic games, players aim to decide on the course of their joint actions through time (joint policy) to maximize their accumulated rewards over time: $\sum_{s_{t+1} \in S} T(s_t, (a_t^k, a_t^{-k}), s_{t+1}) \cdot P^k(s_t, (a_t^k, a_t^{-k}), s_{t+1})$. Here, T represents the transition from state s_t to the state s_{t+1}, and $P^k(s_t, (a^k, a^{-k}), s_{t+1})$ is the reward the player receives for choosing action a^k, given the actions a^{-k} of the other players, at state s_t, and resulting into state s_{t+1}.

2.2 Empirical Analysis and Empirical Games

Empirical Game Theory Analysis (EGTA) provides a framework that uses empirical methods to analyze player interactions within complex game environments [3]. These methods are used to define game components, such as payoff matrices, based on observed interactions, rather than relying on predefined rules. Simulation is one such method, where agents repeatedly play a game, and payoffs are collected based on the outcomes of these interactions. Other techniques include sampling, where a subset of the action space is explored to approximate the payoffs for a wider set of actions, and machine learning methods to identify players' behavior and estimate outcomes based on historical data [11]. Empirical techniques are applied in cases where the action space is too large and complex to define manually, making payoff matrices impossible to generate from simple rules and assumptions.

An empirical game, also referred as a meta-game, is a *Normal Form Game* of the form $G = (K, \mathcal{S}tr, P)$, where K, $\mathcal{S}tr$ and P specify players, players' strategies, and payoffs, correspondingly. We define empirical games by abstracting the actions and defining the payoffs of players in an underlying dynamic game. The underlying game represents the actual setting where players interact. In the empirical game representation, K is the same as in the underlying game, i.e. the set of players engaged in strategic interactions. Strategies (i.e. styles of playing the game) in empirical games offer an action abstraction and can be derived by identifying distinct behaviors during game-play. The strategy space $\mathcal{S}tr$ consists of distinct agents' styles of play. $\mathcal{S}tr^k$ denotes the strategies of agent k and $\mathcal{S}tr^{-k}$ the set of strategies of agents other than k. The set of strategy profiles, i.e. agents' joint strategies, is defined to be $\mathcal{SP} = \{\mathcal{S}_i\} = \{(str_i^1, str_i^2, \ldots, str_i^K)\}$, where $str_i^k \in \mathcal{S}tr^k$, and $i = 1, \ldots$ the profile index$\}$. The payoff matrix of an empirical game can be generated using empirical analysis techniques. Here, we focus on simulation, where agents engaged in the underlying game act according to policies adhering to specific strategies.

Subsequently, we use the terms *action* and *policy* when speaking about the underlying game, and the term *strategies* or *styles of play* when speaking about the empirical game.

The payoff function P of the empirical game is computed from simulations for each strategy profile as follows:

$$P^k(str^k, str^{-k}) = \frac{1}{N} \cdot \sum_{i=1}^{N} P_i^k(str^k, str^{-k})$$

where N is the number of simulation runs, str^k represents player k's strategy, str^{-k} denotes the strategies of the other players, and

$P_i^k(str^k, str^{-k})$ (with an abuse of notation) represents the payoff player k receives in simulation run i when playing strategy str^k against the strategies of the other players. It must be noted that in contrast to dynamic games the payoff function does not take states as arguments, as the outcomes are determined by agents' joint strategies, i.e. $P^k : (\mathcal{S}tr^k \times \mathcal{S}tr^{-k}) \to \mathbb{R}^K$ [5]. If we aggregate these expected payoffs into a matrix, we get the empirical payoff matrix whose dimensionality is $\prod_{k=1}^{K} \mathcal{S}tr^k$. Each entry represents the expected payoff for strategy str^k against strategy str^{-k}.

2.3 The α-Rank Method

Evolutionary dynamics studies how agents' interactions in multi-agent settings evolve over time. While single-agent systems have acquired a strong foundation over the years [1], multi-agent systems are more challenging to analyze.

Current literature indicates a growing interest in studying the evolutionary dynamics of multi-agent systems [1] [2] [6]. Although one might view evolutionary algorithms as mere tools for agents' hyper-parameter tuning , their contributions extend far beyond that. In the context of games, evolutionary algorithms are widely used to explore game-theoretic concepts. This area of study is also known as *Evolutionary Game Theory*. Building on work done in Evolutionary Game Theory, *α-Rank* [5] introduces a novel game-theoretic approach to provide insights into the long-term dynamics of agents' interactions.

α-Rank is an evolutionary methodology designed to evaluate and rank agents' strategies in large-scale multi-agent interactions, using a new dynamic solution concept called *Markov-Conley chains* (MCCs). Given a K-player game, *α-Rank* considers the empirical game with K player slots, called *populations*, where individual agents correspond to strategies, i.e. to styles of playing the underlying game.

In *α-Rank*, populations of agents interact with each other through an evolutionary process following the dynamics of games. The rewards received from these interactions determine how well each strategy performs and, in turn, how often it is adopted by individuals in the populations. Strategies that perform well have a higher probability of being adopted and carried over to the next generation, while those performing poorly are less likely to be adopted. This process of competition and selection between populations leads to their evolution.

To facilitate evolution, *α-Rank* uses the concept of mutation. Initially, populations are monomorphic, meaning all individuals within them choose the same strategy. During K-wise interactions, individuals have a small probability of mutating into different strategies or choosing to stick with their current one. The probability that the mutant will take over the population, defined to be the fixation probability function ρ, depends on the relative fitness of the mutant and the population being invaded. Fitness is a function that computes the expected reward an individual can receive when adopting a particular strategy, given the strategies of the other individuals. The stronger the fitness, the more likely it is for individuals to mutate, whereas the lower the fitness, the more likely it is for the mutant to go extinct. When the mutation rate is small, we can assume that the fitness for any agent k is $f^k(str^k, str^{-k}) = P^k(str^k, str^{-k})$, where P is the empirical game payoff.

Formally, the probability of a mutant strategy str' fixating in some population where individuals play strategy str is given by:

$$\rho_{str \to str'} = \frac{1 - e^{-\alpha \cdot \Delta f}}{1 - e^{-\alpha \cdot m \cdot \Delta f}} \tag{1}$$

assuming that Δf is non-zero. $\Delta f = f^k(str', str^{-k}) - f^k(str, str^{-k})$ represents the difference in fitness between the mutant strategy str' and the resident strategy str in the focal population k, while the remaining $K-1$ populations are fixed in their monomorphic strategies str^{-k}. Parameter m is the population size and α is the selection intensity. This adjusts the sensitivity of the system to fitness differences: with higher values of α, even small differences in fitness lead to larger changes in ρ. The nominator measures the potential of the mutant to "invade" the resident population solely based on its fitness advantage. Note that, for example, as Δf approaches zero, the probability of the mutant's success decreases. The denominator, on the other hand, normalizes the fixation probability using the population size m, making it more challenging for a mutant to dominate in larger populations. When Δf is zero, the fixation probability comes down to $1/m$, indicating that the mutant strategy has the same probability of taking over as any other strategy in the population. We refer to this probability as the *neutral fixation probability*, denoted by ρ_m.

In the context of K-player games, *α-Rank* creates a Markov transition matrix over strategy profiles. This is an $|\mathcal{S}tr| \times |\mathcal{S}tr|$ matrix that defines the probability of moving from one strategy profile to another based on how likely each population is to change its strategy.

$$C_{str \to str'} = \begin{cases} \eta \cdot \rho_{str \to str'} & \text{if } str \neq str' \\ 1 - \sum_{str \neq str'} C_{str \to str'} & \text{otherwise} \end{cases} \tag{2}$$

Here, C is the strategy-transition matrix where each entry $C_{str \to str'}$ represents the probability of transitioning from strategy str to strategy str'. The first part of the formula, calculates the probability of transitioning from one strategy to a different one, scaled to ensure that the sum of probabilities for all possible transitions from that strategy sums up to 1. The second part of the formula, computes the probability of staying with the same strategy, str, by excluding transitions to all other strategies.

This evolutionary process of competition and selection among players' strategies leads to a unique stationary probability distribution π of dimensionality $|\mathcal{SP}|$, where the mass assigned to a strategy profile indicates how likely it is to resist being "invaded" by other strategies as the dynamics evolve. To evaluate and rank strategy profiles —which is the ultimate goal— the method calculates π over the game's Markov chain, using the strategy-transition matrix C. This distribution indicates how often the system is likely to remain in each profile over time, allowing us to identify the most dominant strategies that are expected to prevail in the long run. Formally, π can be computed from the following equation:

$$\pi C = \pi \Rightarrow \pi(C - \mathbb{I}) = 0 \tag{3}$$

where $\mathbb{I}$ is the identity matrix. This means we are looking for a probability vector π such that when multiplied by the transition matrix C, it remains unchanged. To solve for π, the augmented matrix from $C - \mathbb{I}$ is constructed and a normalization condition to ensure that probabilities sum to 1 is imposed[1]. In this stationary distribution, $\pi = (\pi_1, \pi_2, ..., \pi_{|(SP)|})$, each π_i represents the average time the system spends in strategy profile i.

3 PROBLEM STATEMENT

As already stated, we aim at identifying (human and software) agents' strong joint strategies, in terms of stability and joint performance, to solve problems in dynamic settings, accounting for agents' long-term dynamics of interactions. Stability implies non-transient strong strategies, persisting in time, as they fit better to the objective of the agents given the structure of the game and payoffs received. However, in dynamic games, we need to define the payoff matrix and exploit this to determine strategies stability. Even if we manage to estimate payoffs, the computation of solution concepts like the *Nash equilibrium* imposes a high computational cost in these settings, does not guarantee convergence, and fails to scale to large games. Beyond identifying stable joint strategies, it is important to transparently justify/describe what makes one joint strategy better than another. This requires more than just providing rankings of strategy profiles; it requires providing evidence for the rankings.

We could, therefore, consider our problem as follows: Given a dynamic game G with K players, our goal is to identify styles of playing G, and thus, the set of strategy profiles $\mathcal{SP}$, and rank these profiles based on how stable they are over time, considering long-term agents' interactions towards achieving their objectives. Specifically, we aim to define a ranking function $\mathcal{R} : \mathcal{SP} \to \mathbb{R}$, where $\mathcal{R}(\mathcal{S}_i) > \mathcal{R}(\mathcal{S}_j)$ (resp. $\mathcal{R}(\mathcal{S}_i) \geq \mathcal{R}(\mathcal{S}_j)$) indicates that the strategy profile $\mathcal{S}_i$ is strictly (resp. weakly) preferred over $\mathcal{S}_j$, using a descriptive framework $\mathcal{D} : \mathcal{SP} \times \mathcal{SP} \to \mathbb{R}$ that provides transparency on how rankings are decided.

It must be noted that empirical game strategies are realized by agents' policies adhering to these strategies in the underlying game. Thus, identifying stable joint strategies in the empirical game translates to identifying stable joint policies adhering to these strategies in the underlying dynamic game.

4 PROPOSED METHOD

To address the challenge of identifying stable joint policies in dynamic games, we propose an approach that combines concepts from *Empirical Game Theory* and *Evolutionary Dynamics*, using *α-Rank*, providing transparency to rankings of agent's styles of play.

Given that the set of agents' policies in dynamic games can be infinitely large we focus on a subset of policies that adhere to concrete and well-defined styles of play. A way to identify styles of play is to observe how players behave in the underlying game or exploit demonstrations of game playing. For instance, human experts performing a task usually follow a distinct set of specific styles based on well-established practices, preferences and experience. Having determined the game playing strategies, we can transform the dynamic game into its empirical form, defining the meta-game, as specified in Section 2.2: By (a) identifying empirical game strategies, and (b) training policies for agents to play the underlying game according to these strategies, (c) defining the empirical game payoff matrix, through simulations, exploiting the trained policies.

Having defined the meta-game, we need to define the function $\mathcal{R}$, which ranks joint strategies based on agents' long-term dynamics and objectives. In our approach, we propose using the evolutionary *α-Rank* methodology to determine these rankings. The rankings are based on each strategy profile evolutionary success, which is reflected in the probability of that profile being selected over time. This probability is captured by the stationary distribution π, which *α-Rank* computes in the limit of infinite ranking intensity α. As demonstrated by [5], a large α limit suffices. Therefore the long-term behavior is captured by the unique stationary distribution π under the large α limit. As it is proved in [5], the Markov chain associated with a generalized multi-population model, coincides with the MCC solution concept. MCCs can be found efficiently in all games and can be identified by the sink strongly connected components of a response graph, whose vertices correspond to pure strategies' profiles and a directed edge from strategy profile $\mathcal{S}_i$ to a strategy profile $\mathcal{S}_j$ specifies that $\mathcal{S}_j$ is weakly a better response than $\mathcal{S}_i$ for player k.

To compute π over strategy profiles, *α-Rank* requires the payoff matrix of the empirical game P. Along with the stationary distribution π, *α-Rank* also outputs the fixation probability function $\rho_{\mathcal{S}_i \to \mathcal{S}_j}$, where $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{SP}$. One could abstractly illustrate *α-Rank* as a function:

$$\alpha\text{-Rank}(P) \to (\pi, \rho) \tag{4}$$

While the stationary distribution π provides valuable insight into the long-term behavior of strategies, it alone does not help us fully understand how strategies transition between one another. The fixation probability function ρ, which measures the likelihood of transitioning from one strategy profile $\mathcal{S}_i$ to another $\mathcal{S}_j$, fills this gap. Based on this, the descriptive framework $\mathcal{D}$ can be adequately represented by π and ρ, which are constituents of the response graph, which provides a complete view of the empirical game dynamics.

Overall, building on the *α-Rank* descriptive framework, the method proposed here for computing strategy profile rankings in dynamic games is as follows:

(1) Identify players' styles of play.
(2) Define the strategies of the empirical game based on those styles.
(3) Train policies realizing the defined strategies.
(4) Run game simulations to create the empirical payoff matrix P.
(5) Apply *α-Rank* to define $\mathcal{R}$ and $\mathcal{D}$:
   (a) Calculate the Markov transition matrix C.
   (b) Find the unique stationary distribution π.
   (c) Rank joint strategies by ordering the masses of π.
   (d) Describe the rankings through the response graph.
   (e) Study the effect of different α values on π.

[1]The system $\pi(C - \mathbb{I}) = 0$ by itself does not have a unique solution, as there are infinitely many vectors π that satisfy it. To get a unique solution $\pi = (\pi_1, \pi_2, ..., \pi_{|(SP)|})$, it must hold that $\sum_i \pi_i = 1$.

5 EXPERIMENTS AND RESULTS

5.1 The Graph Coloring Problem

The *Graph Coloring Problem* (GCP) is one of the most well-known problems in graph theory. It involves assigning colors to vertices in a graph such that no two adjacent vertices share the same color, and using the minimum number of colors, also known as the chromatic number [10].

In this study, we shift our focus from finding the chromatic number across graph configurations to solving the multi-agent problem of assigning colors to vertices of a dynamic graph with respect to the constraints: In doing so, we define the graph coloring problem as a dynamic game that allows us to study the evolutionary dynamics in multi-agent interactions.

Through this problem, we aim to demonstrate how we can gain insights into individuals' joint strategies, i.e. into the effectiveness of playing the dynamic game when individual styles of play are combined.

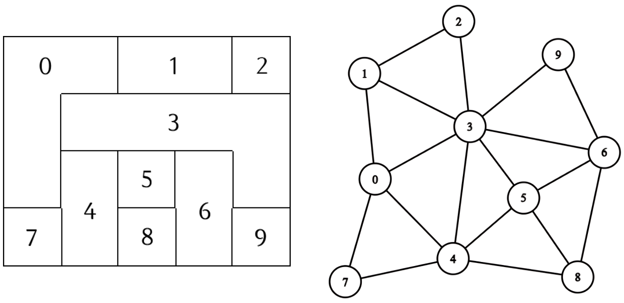


Figure 1: A snaphsot of the game environment in grid and graph forms.

We consider the underlying graph coloring dynamic game to be a two-player game executed in rounds. The graph corresponds to a grid comprising blocks of cells: A block comprises one or more merged cells. Each vertex of the graph corresponds to a block, and the adjacency relation between blocks specifies the edges in the graph. At the beginning of the game, the grid is initialized with a random number of rows and columns ($n \times m$). In our experimental setup we assume a 4×5 grid. The environment is initialized by randomly combining cells to create the blocks. The resulting configuration remains the same throughout the entire game. A snapshot of such a configuration with 10 blocks is the one shown in Figure 1, together with the corresponding graph. Merging cells is important as it allows for complex neighboring relationships to be defined, expanding beyond the standard constraints between adjacent blocks. Blocks are either (a) colored by the agents, (b) white (free to be colored) or (c) hidden (their colors cannot be observed and they can not be re-colored by the agents). Let B the set of blocks corresponding to graph vertices, and CR the set of possible colors that an agent can use for coloring blocks in B. The game unfolds over multiple rounds in which agents simultaneously choose their actions. At the beginning of each round, the environment reveals the color of some of the hidden blocks, if any. The number of blocks that get un-hidden is random, which implies that the state of the graph is influenced not only by the agents' actions, but also by the environment. We therefore consider the game to be stochastic. As soon as all blocks in B are uncovered and colored, the game ends.

Preference Dimension	*Value*
Color Tone	warm (W) vs. cool (C)
Block Coloring Difficulty	small (L) vs. large (A)
Coloring Approach	minimalistic (M) vs. extravagant (E)

Table 2: Dimensions specifying agents' strategies

The set of agents' actions A is defined to be the Cartesian product of the set of blocks B and the set of the available colors CR, denoted as:

$$A = B \times CR = \{(b, c) \mid b \in B,\ c \in CR\} \tag{5}$$

To specify states, let CR^* include the elements of CR, and two additional elements representing hidden and white blocks: $CR^* = CR \cup \{\text{hidden}, \text{white}\}$. A state s is as follows:

$$s = \{(b_i, c_i), i = 1, \dots, |B|\},$$
$$s.t. \forall b \in B,\ \exists \text{a unique } c \in CR^*, \text{ with } (b, c) \in s$$

Regarding the reward function, it is a sum of gains, penalties, sanctions, delays and adopted preferences. Given that actions are represented as vectors of shape $(b, c) \in B \times CR$, an agent receives a gain point (+1) for each neighbor of b that has a different color than the chosen color c. On the contrary, an agent receives a penalty point (-2) for each neighbor that shares the same color c. Sanction is a big negative reward (-10) that an agent receives when it attempts to color a hidden block or a block that has already been colored. Delay (-1) is a small negative reward that both agents receive when they try to color the same block b, causing a brief pause in the game to determine which agent will eventually color b. Last but not least, there is the preference-adoption reward, which agents receive regardless of whether their action is good, bad, or forbidden. This reward helps agents to be trained so as to adhere to specific preferences, or what we call *styles of play*. We will elaborate shortly on these in the following section.

5.2 Defining the Empirical Game

Transforming the underlying dynamic graph-coloring game into its empirical form involves two key steps: (1) identifying agents' strategies and (2) constructing the empirical game payoff matrix.

5.2.1 Agents' Strategies. Agents' strategies define distinct styles of play, usually revealed by preferences in playing the game. In our experiments, we specify different styles across three main dimensions: color tone (preference for which colors to use), block difficulty (preference for the types of blocks to choose), and coloring approach (preference for the number of colors to use), as shown in Table 2. Through the combination of preferences in each of these dimensions, a style can range from complete indifference, where none of the dimensions hold any influence (denoted by "I"), to specific preferences in all dimensions.

Policies corresponding to specific strategies are represented using convolutional neural networks. To train these policies, we assign specific values in the three dimensions of the game's *preference* reward. These values, range from -1 to 1, where 1 indicates a strong preference for a particular dimension. For example, a value of 0.7 for warm colors indicates a relatively high preference for warm tones. In our experimental setting, we define 11 distinct styles: I, C, W, E, M, L, A, AE, CA, LE and WL, given "I" and combinations of preference values specified in Table 2.

Assuming no inherent bias among the players of the empirical game, we allow populations to sample from the same list of strategies.

5.2.2 Training the agents. All policy models share the same underlying architecture and training setup. Although hyperparameter tuning is typically recommended, it doesn't make much difference in this case, as these models are relatively easy to optimize when trained in small settings. Regarding the convolutional neural network architecture, it consists of four convolutional layers, each defined with a kernel size of 3, stride of 1, and padding of 1, meant to extract spatial features from the input. The input tensor has dimensions 10×12, where $|B| = 10$ represents the number of blocks in the state and $|CR^*| = 12$ represents the number of possible colors a block can have. Each block is encoded using one-hot encoding, meaning that each color is represented as a binary vector of length 12. The output is then flattened and passed through two fully connected layers, which process the data to produce the final output, as shown in Figure 2.

Policy models are trained individually (with no co-players) in the underlying game using the deep Q-learning reinforcement learning algorithm specified in Algorithm 1. We set γ to 0.7. To optimize the model parameters, we use the smooth L1 loss function with β=1.0 and the Adam optimizer with a learning rate of 5e-4 and weight decay of 1e-5 to prevent over-fitting. To further enhance the learning process, we incorporate experience replay, with a memory that stores up to 10 million experiences [4]. A target network alongside the main policy network, is being used according to the Double-DQN approach [9]. To update the target network we apply a soft update with a factor τ=5e-3. This gradually brings the target network closer to the policy network, balancing learning speed and stability. With a batch size of 64, we train the models for 10000 episodes.

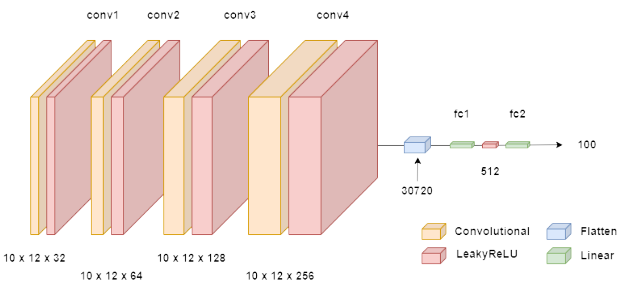


Figure 2: Convolution Policy Network Architecture

Algorithm 1: Double Deep Q-Learning with Experience Replay

$Q_\theta, Q_{\theta'} \leftarrow Q_\theta, M$ ▷ Initialize policy/target nets & memory
for episode **do**
  $s \leftarrow s_0$
  for step **do**
    $a \leftarrow \text{argmax}_a Q_\theta(s)$ ▷ Select ϵ-greedy action
    $(s, a, r, s') \in M$ ▷ Store experience
    if $|M| >$ batch size **then**
      for each (s, a, r, s') in M **do** ▷ Sample memory
        $y \leftarrow r + \gamma \max_{a'} Q_{\theta'}(s')$
        $L \leftarrow \text{Loss}(Q_\theta(s), y)$
        $\theta \leftarrow \theta - \alpha \nabla_\theta L$
      end for
    end if
    $Q_{\theta'} \leftarrow \tau Q_\theta + (1 - \tau) Q_{\theta'}$ ▷ Soft update
    $s \leftarrow s'$
  end for
end for

5.2.3 Empirical Game Payoff Matrix. We generate the empirical payoff matrix by simulating each strategy profile over multiple games. These payoffs represent how well different styles of play perform jointly, according to the game's rules.

The values in the payoff matrix are computed in terms of the delay and the quality of the solution according to the game's constraints (gain, penalty, and sanction), excluding preferences. This ensures a common ground for distinct strategies, evaluating solutions solely based on the game's rules. For each pair of strategies, we simulate the game over 5,000 repeats and calculate the average payoff for each strategy. These values are then organized into the payoff matrix, which is provided in Table 4 (Appendix A). From this matrix, we observe that (L, WL) and its symmetric counterpart (WL, L) both with payoffs of (3.15, 3.21) and (3.21, 3.15) respectively, are the only Nash equilibria. It is important to note here that these equilibria prescribe agents' strategies given that they do play the game with rational co-players, but they do not capture the overall dynamics of the game, considering the long-term effects of agents' interactions.

5.3 Evaluation and Ranking

Given the payoff matrix derived from the empirical analysis, we apply the α*-Rank* method to evaluate the performance of strategy profiles over time in terms of the MCC solution concept. Specifically, we ran the method 1000 times, using values of α within the range $[0.1, 10]$ with step=0.01, while assuming populations of size $m = 100$. We provide as input the strategies defined in Section 5.2.1 and the empirical game payoff matrix. We focus on the rankings of the top 6 strategy profiles, to identify the stronger ones across different values of α.

As we observe from the rankings in Table 3, the strategy profile that prevails in the long run is (WL, CA); this is the primary component of the MCC. Although the table was derived using an α value of 2, the rankings remain consistent even when α is set to 10. We choose $\alpha = 2$ over $\alpha = 10$, to display the rankings of lower-performing strategy profiles, which would otherwise drop to

Agent	Rank	Score
(WL, CA)	1	0.42
(W, CA)	2	0.13
(M, CA)	3	0.12
(CA, M)	4	0.08
(CA, W)	5	0.08
(CA, LE)	6	0.01

Table 3: Rankings for $\alpha = 2$

zero. First, it is worth mentioning that the Nash equilibria (L, WL) and (WL, L) don't appear among the top-ranked strategy profiles. This is because MCC components are defined based on how well strategies perform when interacting with other strategies, based on long-term agents interactions. The individual strategies within the Nash equilibrium profile, either WL or L, may not result in favorable interactions with other strategies. As a result, the profile (WL, L) is ranked lower that others.

To further support our observations regarding the misalignment between the two solution concepts, let's examine why (CA, WL) is part of the MCCs, while (L, WL), the Nash equilibrium, is not. A closer look at the payoff matrix in Table 4 reveals that L appears to be the worst-performing strategy for the row player, with an average payoff of 3.13. In this case, being in the Nash equilibrium means the player is stuck with a strategy that gives low rewards, making it the best among other options, rather than a strong choice. If it happens to play this strategy, it would expect its rational opponent to play WL. Strategy CA on the other hand, is the best-performing strategy for the row player, with an average payoff of 3.18. Combined with WL, which is the best performing strategy for the column player, with an average payoff of 3.18, they make profile (CA, WL) becomes the top ranked strategy profile in the ranking Table 3.

Rankings within the MCC are also very intuitive. For example, strategies that prefer different color tones, such as (WL, CA) or (W, CA), tend to result into fewer conflicts since, they naturally avoid selecting the same colors. Similarly, strategies that prefer different blocks based on their difficulty, such as (WL, CA) or (CA, LE), tend to provide solutions with minimal delay, as they naturally avoid coloring the same blocks. Notably, profiles with mixed preferences across these dimensions demonstrate the most promising performance, which explains why (WL, CA), as such a profile, is a key component of the MCC. However, not all profile rankings can be easily explained through the game's rules alone; the expected influence of certain strategies on the quality of the solutions remains ambiguous. For example, profiles with strategies like M and E are more difficult to analyze.

The response graph provides a visualization to interpret the *α-Rank* results. This graph illustrates the MCC, using the strategy profiles' masses from the stationary distribution, π, along with the fixation probability function ρ provided by *α-Rank*. Figure 5d shows the response graph for $\alpha = 6.4$. We consider it to be part of the descriptive framework $\mathcal{D}$, as it offers insights into how rankings were derived. Additional graphs for $\alpha = 0.4$, 1.3, and 1.9 are available in Figure 5 (Appendix B).

The response graph describes the overall dynamics of the strategy profiles in the empirical game. One prominent feature is the primary component of the MCC, specifically the profile (WL, CA). This profile, indicated by a dark blue color, has multiple graph edges leading to it, while none from it, indicating that strategies in this profile are non-transient. This is further supported by the large fixation probabilities along the edges. A particularly prominent example is the cluster (CA, LE)-(CA, M)-(CA, W), which consists of three strongly connected profiles, indicating that once a player adopts one of these profiles, they will likely remain within their cluster. These components reflect stable regions in the game's strategy dynamics, where transitions between profiles become locked into a cycle.

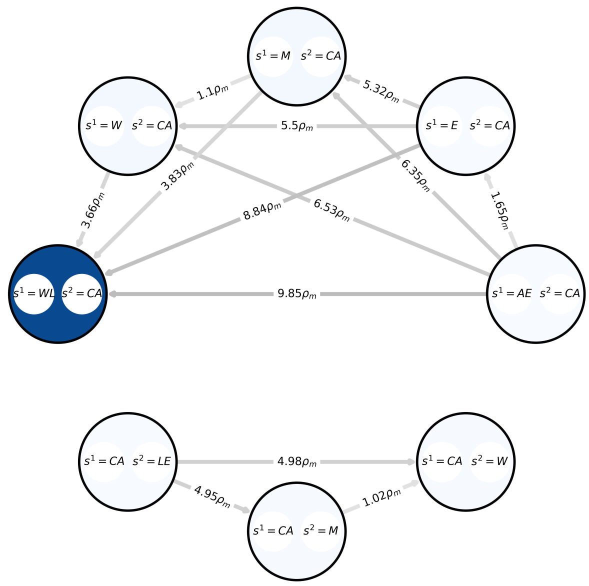
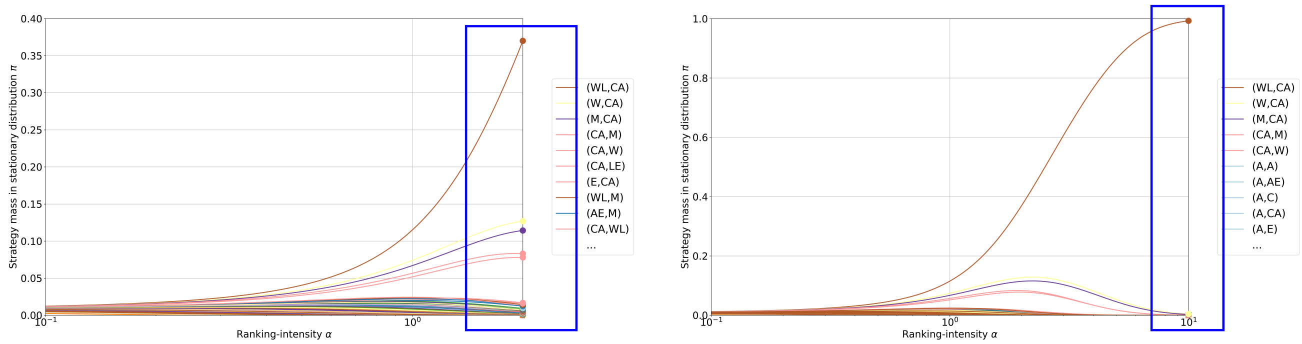


Figure 3: Response graph for $\alpha = 6.4$.

To further investigate the effect of α on profile dominance, we plotted the stationary distribution π across all α values used in the experiments, for the top-performing strategy profiles (see Figure 4). This visualization -also part of $\mathcal{D}$- helps us understand how the stationary distribution changes as the selection intensity increases. The x-axis represents the different α values, ranging from 0.1 to 3 in Figure 4a, and from 0.1 to 10 in Figure 4b, while the y-axis in both figures shows the mass of each strategy profile in the stationary distribution π. As α increases, the distribution converges, indicating that the selection process stabilizes. The final mass distributions are highlighted in boxed regions. The legend on the right side of the plot displays the top-performing joint strategies, with the stronger ones appearing at the top.

We plot two such graphs to observe how the mass of strategy profiles is distributed in the MCCs across different α values. In the stationary distribution resulting from a bigger α, the dominant strategy profile (WL, CA) in the MCC achieves a mass of 1, with all other profiles dropping to 0. This is clearly illustrated in the second



(a) Mass across $\alpha \in [0.1, 3]$

(b) Mass across $\alpha \in [0.1, 10]$

Figure 4: Effect of ranking intensity α on strategy profile mass in the stationary distribution π.

plot (see Figure 4b). However, regarding the mass distribution for a smaller range of α, depicted n the first plot, the game has not yet converged to the final MCC.

6 CONCLUSIONS

In this study, we developed a methodology for identifying strong joint-strategies in dynamic multi-agent games, accounting for stability and performance, using the *α-Rank* evolutionary algorithm. The methodology is applied on a stochastic version of the *Graph Coloring Problem*, in which players work together to color a graph while ensuring that neighboring vertices are assigned different colors. According to the methodology, first we transformed the game into its empirical form, by defining strategies (styles of play). We then designed and trained Deep Q-Learning policy models that realize those styles of play in the underlying game, and run simulations to generate the empirical payoff matrix. *α-Rank*, applied to this matrix, results into a unique stationary distribution over strategy profiles that defines the empirical game's MCC. The *α-Rank* not only helped us identify stable strategy profiles resistant to changes but also provided a descriptive framework for understanding why certain profiles prevail in the long run, based on the underlying dynamics of the game. Through this approach, we successfully described a concise methodology for evaluating and ranking agents' joint policies, considering their long-term interactions in dynamic settings, while also explaining how strategy profiles are defined within the MCC.

Future work involves (a) applying the methodology in more complex and large-scale settings, accounting for strategy profiles of multiple stakeholders that may collaborate and/or compete, (b) using machine learning methods to identify different styles of play from demonstrations and specifying the empirical game, (c) exploring advanced models able to adapt their strategies based on observed behaviors based on the behavior of co-players, and (d) applying the methodology into real-world settings where agents need to align with human preferences in dynamic settings.

ACKNOWLEDGMENTS

xxxxxxxxxxx

REFERENCES

[1] Daan Bloembergen, Karl Tuyls, Daniel Hennes, and Michael Kaisers. 2015. Evolutionary dynamics of multi-agent learning: a survey. *J. Artif. Int. Res.* 53, 1 (May 2015), 659–697.

[2] Omid E. David, H. Jaap van den Herik, Moshe Koppel, and Nathan S. Netanyahu. 2014. Genetic Algorithms for Evolving Computer Chess Programs. *IEEE Transactions on Evolutionary Computation* 18, 5 (Oct. 2014), 779–789. https://doi.org/10.1109/tevc.2013.2285111

[3] Michael Levet. 2016. Game Theory : Normal Form Games. https://api.semanticscholar.org/CorpusID:131771375

[4] Long-Ji Lin. 1992. Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching. *Mach. Learn.* 8, 3–4 (may 1992), 293–321. https://doi.org/10.1007/BF00992699

[5] Shayegan Omidshafiei, Christos Papadimitriou, Georgios Piliouras, Karl Tuyls, Mark Rowland, Jean-Baptiste Lespiau, Wojciech M. Czarnecki, Marc Lanctot, Julien Perolat, and Remi Munos. 2019. α-Rank: Multi-Agent Evaluation by Evolution. arXiv:1903.01373 [cs.MA]

[6] Sheryl Paul and Jyotirmoy V. Deshmukh. 2022. Multi Agent Path Finding using Evolutionary Game Theory. arXiv:2212.02010 [cs.MA] https://arxiv.org/abs/2212.02010

[7] Lloyd S. Shapley. 1953. Stochastic Games*. *Proceedings of the National Academy of Sciences* 39 (1953), 1095 – 1100. https://api.semanticscholar.org/CorpusID:263414073

[8] Yoav Shoham and Kevin Leyton-Brown. 2008. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge University Press.

[9] Hado van Hasselt, Arthur Guez, and David Silver. 2015. Deep Reinforcement Learning with Double Q-learning. arXiv:1509.06461 [cs.LG]

[10] George Watkins, Giovanni Montana, and Juergen Branke. 2023. Generating a Graph Colouring Heuristic with Deep Q-Learning and Graph Neural Networks. arXiv:2304.04051 [cs.LG]

[11] Michael P. Wellman, Karl Tuyls, and Amy Greenwald. 2024. Empirical Game-Theoretic Analysis: A Survey. arXiv:2403.04018 [cs.GT] https://arxiv.org/abs/2403.04018

A EMPIRICAL PAYOFF MATRIX

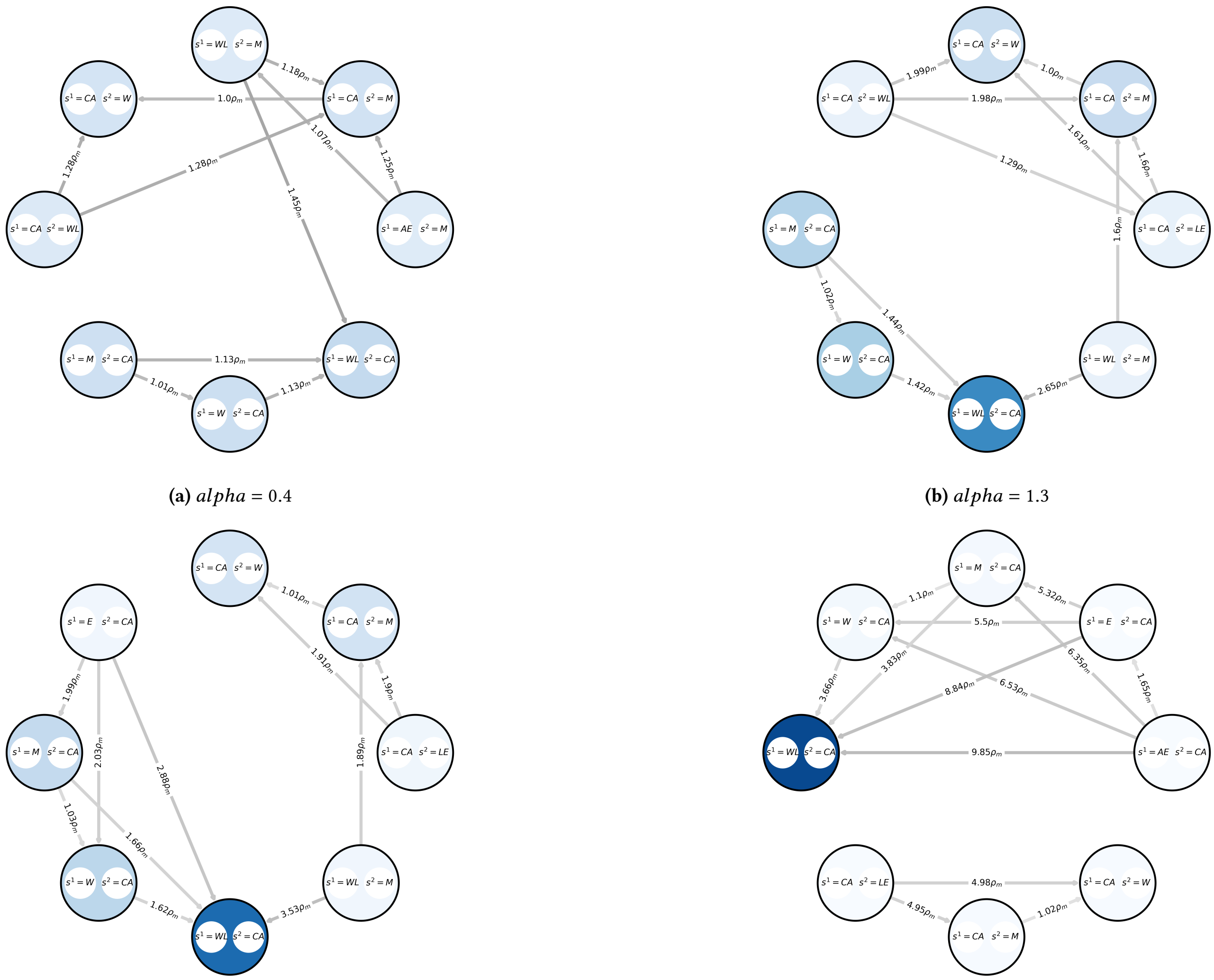
This is the empirical payoff matrix derived from simulations of the *Graph Coloring Game* using policies trained to adhere to specific styles of play. Each entry in the matrix represents the payoffs of strategies in the corresponding profile, with the first value indicating the payoff of the row strategy and the second value of the column player. The Nash equilibria are highlighted in bold, while nine of the top-ranked strategy profiles in the MCC are shaded in gray.

	A	AE	C	CA	E	I	L	LE	M	W	WL
A	(3.12, 3.11)	(3.15, 3.16)	(3.17, 3.17)	(3.14, 3.17)	(3.16, 3.17)	(3.16, 3.15)	(3.22, 3.13)	(3.19, 3.16)	(3.15, 3.18)	(3.16, 3.17)	(3.21, 3.18)
AE	(3.17, 3.17)	(3.11, 3.11)	(3.18, 3.17)	(3.15, 3.17)	(3.17, 3.16)	(3.19, 3.16)	(3.23, 3.12)	(3.19, 3.16)	(3.15, 3.18)	(3.17, 3.17)	(3.20, 3.16)
C	(3.17, 3.16)	(3.16, 3.17)	(3.10, 3.10)	(3.14, 3.17)	(3.15, 3.15)	(3.18, 3.15)	(3.22, 3.12)	(3.17, 3.14)	(3.14, 3.17)	(3.17, 3.16)	(3.20, 3.17)
CA	(3.17, 3.15)	(3.17, 3.15)	(3.17, 3.14)	(3.11, 3.11)	(3.18, 3.15)	(3.18, 3.14)	(3.24, 3.13)	(3.21, 3.16)	(3.16, 3.16)	(3.19, 3.16)	(3.22, 3.15)
E	(3.15, 3.16)	(3.16, 3.16)	(3.15, 3.16)	(3.15, 3.17)	(3.10, 3.10)	(3.18, 3.16)	(3.22, 3.12)	(3.19, 3.14)	(3.15, 3.17)	(3.16, 3.17)	(3.19, 3.17)
I	(3.14, 3.16)	(3.16, 3.18)	(3.16, 3.18)	(3.15, 3.19)	(3.16, 3.17)	(3.12, 3.12)	(3.22, 3.14)	(3.18, 3.16)	(3.14, 3.19)	(3.16, 3.18)	(3.19, 3.18)
L	(3.14, 3.22)	(3.11, 3.22)	(3.12, 3.22)	(3.13, 3.23)	(3.12, 3.22)	(3.13, 3.22)	(3.12, 3.12)	(3.14, 3.20)	(3.11, 3.21)	(3.14, 3.23)	**(3.15, 3.21)**
LE	(3.15, 3.19)	(3.14, 3.18)	(3.14, 3.18)	(3.15, 3.21)	(3.15, 3.19)	(3.16, 3.17)	(3.20, 3.14)	(3.11, 3.11)	(3.14, 3.22)	(3.15, 3.18)	(3.18, 3.19)
M	(3.17, 3.14)	(3.17, 3.15)	(3.17, 3.15)	(3.16, 3.17)	(3.16, 3.14)	(3.18, 3.14)	(3.23, 3.11)	(3.20, 3.14)	(3.06, 3.08)	(3.18, 3.15)	(3.20, 3.16)
W	(3.17, 3.17)	(3.17, 3.18)	(3.16, 3.18)	(3.16, 3.20)	(3.17, 3.17)	(3.18, 3.16)	(3.21, 3.13)	(3.18, 3.15)	(3.15, 3.18)	(3.08, 3.09)	(3.19, 3.15)
WL	(3.17, 3.20)	(3.17, 3.19)	(3.17, 3.19)	(3.17, 3.22)	(3.17, 3.19)	(3.18, 3.19)	**(3.21, 3.15)**	(3.19, 3.17)	(3.16, 3.20)	(3.16, 3.19)	(3.13, 3.13)

Table 4: Empirical Payoff Matrix for the Graph Coloring Game

B RESPONSE GRAPH

These are four response graphs illustrating the dynamics of strategy profiles in the empirical *Graph Coloring Game* for different α values. Each node in the graph represents a unique strategy profile in the MCC, while the edges indicate transitions between them. The values on the edges show the fixation probabilities normalized by the neutral fixation probability, denoted as ρ_m. The nodes and edges are color-coded. Darker blue nodes represent more strong joint profiles, while lighter blue nodes represent transient ones. Similarly, bold arrows suggest a strong advantage in shifting between the nodes, whereas faint ones suggest less of an advantage.



(a) $alpha = 0.4$

(b) $alpha = 1.3$

(c) $alpha = 1.9$

(d) $alpha = 6.4$

Figure 5: Response graphs of strategy profiles' dynamics.

II Literature Study

This section forms the main body of the thesis, providing a detailed explanation of the content of the related paper. Although much of the material is based on the paper, this section is included to offer a clear breakdown of the research work.

The first section introduces the key concepts of *Game Theory*, mainly focusing on dynamic games. The second section presents α-*Rank*, the core evolutionary method behind the proposed methodology. This section provides a detailed definition of *Markov-Conley Chains* and explains how they model strategy evolution. The third section defines the problem we aim to solve and outlines the proposed methodology. The fourth section introduces the *Graph Coloring Game*, which is used to demonstrate the proposed methodology. Here, we explain the game's state and action spaces, as well as its reward function. The fifth section applies the proposed methodology to the *Graph Coloring Game*, demonstrating its practical implementation. Finally, the conclusion summarizes our findings and suggests potential directions for future work.

1 Introduction to Game Theory

Game theory is the mathematical study of strategic decision-making in situations where independent, self-interested agents interact with one another [16]. It provides a structured way to model strategic behaviors, with the goal of understanding how choices affect the agents' outcomes. The key assumption in game theory is that agents are rational, meaning they make decisions that maximize their individual payoffs. By making this assumption, we can identify equilibrium points—strategies that no agent has an incentive to deviate from. In Game Theory, the focus can be either on direct outcomes, involving *actions*, or indirect outcomes, involving *strategies* and *policies*.

Actions refer to the actual decisions made at a particular point in the game. For example, in chess, e5xd6[1] is an action. Strategies, on the other hand, refer to the different ways in which a player might approach a game—often called *styles of play*. They define tendencies in actions that a player might take. In the same chess example, a strategy could involve a player capturing an opponent's pawn with a 70% probability when given the opportunity. Finally, policies represent the probability distribution over actions given a state of the game. In chess, a policy would describe the probabilities of each possible action a player might take at every state of the game.

The study of strategic decision-making is divided into two main fields: *Classical Game Theory* (CGT) and *Evolutionary Game Theory* (EGT). CGT focuses on games with actual players and their actions. A key solution concept in CGT is the *Nash equilibrium*, which identifies the strategy profiles in which no player can improve their outcome by unilaterally changing their action, assuming the actions of others remain unchanged [11]. On the other hand, EGT examines indirect outcomes that occur when players adopt strategies, also known as styles of play, rather than specific strategies. Here, equilibrium is based on the evolution of behaviors over time [19], often described using concepts like the *Evolutionarily Stable Strategy* (ESS). An ESS is a stable strategy that a player cannot easily replace with another strategy to achieve better long-term outcomes. In EGT, stability is determined by how well strategies perform in populations of players over time, with the most successful strategies being those that resist invasion and persist through the evolutionary process. Another solution concept based on the idea of ESSs is the recently introduced *Markov-Conley Chain* (MCC).

In static games, where payoff matrices are known, finding *Nash equilibria* is possible; at least theoretically if not practically due to being an NP-complete problem in terms of complexity [3]. For example, consider the payoff matrix for the Rock-Paper-Scissors

[1]In chess notation, e5xd6 represents a move where a pawn from the e5 square captures a piece on the d6 square.

game in Table 1. The mixed-strategy *Nash equilibrium* occurs when both players randomize their choices uniformly across Rock, Paper, and Scissors.

Table 1: Payoff matrix for the Rock-Paper-Scissors game.

	Rock	Paper	Scissors
Rock	0,0	-1,1	1,-1
Paper	1,-1	0,0	-1,1
Scissors	-1,1	1,-1	0,0

However, in dynamic settings, considering also sequential decision making problems, one must account for the dynamics of agents' interactions over time. In these settings, we need to analyze agents' behavior in terms of their payoffs, identifying joint strategies that result into stable behaviors. Although CGT provides a robust foundation for understanding static interactions, its solution concepts cannot easily reveal equilibria across sequences of actions; the number of possible policies is extremeley difficult to define, especially in games that unfold over a large number of rounds. Thus, Nash equilibrium in such settings can be theoretically impossible to define, let alone practically compute. Evolutionary approaches have shown great potential toward addressing this challenge.

The main idea in EGT is to approximate the otherwise intractable dynamics of a dynamic game by transforming it into an empirical game—an abstract representation derived from sampled interactions among strategies. This approach enables researchers to explore equilibria, within the strategy space, sidestepping the computational complexity of evaluating every possible action in large-scale games with many players that unfold in stages over time. Instead of analyzing the entire action space, EGT focuses on strategy -and ultimately policy- evaluation, identifying behaviors that prove to be stable over multiple interactions.

1.1 Dynamic Games

Dynamic games are mathematical models that describe the interactions between agents controlling a system whose state evolves over time [7]. These systems rely on the current state and the actions of the agents to determine future states. Dynamic games are particularly useful for studying scenarios where the consequences of decisions unfold progressively and agents plan their actions accordingly. Examples of dynamic games include economic competition, military strategy, and even board games like chess, where each move shapes the future course of the game. The complexity of these games arises from the need to account for the temporal dependencies of actions, making it necessary for players to consider long-term consequences in their decision-making process.

Formally, a dynamic game can be represented as a tuple:

$$G = (S, K, A, T, P) \tag{1}$$

where S represents a finite set of states, K is the set of players, and $A = (A^k \times A^{-k})$ is the set of joined actions, with A^k corresponding to the action set available to player k. A^{-k} denotes the action set available to players other than k. The transition function T is a probability function that determines the next state of the system based on the current state and the actions chosen by the players. Finally, $P^k : S \times (A^k \times A^{-k}) \times S \to \mathbb{R}$ is the payoff function for player k, given the current joint state, the action chosen by player k and the actions of the other agents, and the resulting state.

This study primarily focuses on stochastic dynamic games, a concept initially introduced by L.S. Shapley in 1953 [15]. In stochastic games, the outcome of players' actions is influenced by probabilistic events, introducing an element of uncertainty in the future states of the game. These games are commonly referred to as Markov games [16], as the system's state at any given time depends not only on the players' decisions but also on the inherent randomness of the environment. The transition function T in a stochastic game is defined as a probability distribution over next states. Specifically, $T : S \times A \to \Delta(S)$, where $\Delta(S)$ is a probability distribution over the states, given a state and joint action. For example, in a game like poker, while a player's strategy influences the course of the game, the outcome is also affected by random events, such as drawing a high-value hand (flush) or a low-value hand (pair of twos). In such games, players, when planning their actions, must account for both the actions of their opponents and the dynamics of the environment.

In dynamic games, players aim to decide on the course of their joint actions over time, referred to as a joint policy, to maximize their accumulated rewards:

$$\sum_{s_{t+1} \in S} T(s_t, (a_t^k, a_t^{-k}), s_{t+1}) \cdot P^k(s_t, (a_t^k, a_t^{-k}), s_{t+1}) \tag{2}$$

Here, T represents the transition from state s_t to the state s_{t+1}, and $P^k(s_t, (a^k, a^{-k}), s_{t+1})$ is the reward the player receives for choosing action a^k, given the actions a^{-k} of the other players, at state s_t, and resulting into state s_{t+1}.

1.2 Empirical Games

Empirical Game Theory Analysis (EGTA) provides a framework that uses empirical methods to analyze player interactions within complex game environments [8]. These methods are used to define game components, such as payoff matrices, based on observed interactions, rather than relying on predefined rules.

Simulation is one such method, where agents repeatedly play a game, and payoffs are collected based on the outcomes of these interactions. A similar approach can be found in the early work of [18], where they used simulations to generate a payoff matrix modeling the outcomes of different strategies in animal conflict, aiming to identify evolutionarily stable strategies. Other techniques include sampling, where a subset of the action space is explored to approximate the payoffs for a wider set of actions, and machine learning methods to identify players' behavior and estimate outcomes based on historical data [22]. Empirical techniques are applied in cases where the action space is too large and complex to define manually, making payoff matrices impossible to generate from simple rules and assumptions.

An empirical game, also known as a meta-game, provides an abstract representation of strategic interactions derived from an underlying dynamic game. Formally, it is a *normal form game* defined as:

$$G = (K, \mathcal{S}tr, P) \tag{3}$$

where K represents the set of players, $\mathcal{S}tr$ is the set of strategies available to them, and P is the payoff function.

In the context of empirical games, the strategies in $\mathcal{S}tr$ do not correspond to specific actions in the underlying game, but rather to higher-level behaviors —referred to as *styles of play*. This abstraction simplifies the analysis by focusing on the aggregate outcomes rather than the detailed sequence of individual actions. $\mathcal{S}tr^k$ denotes the strategies of agent k and $\mathcal{S}tr^{-k}$ the set of strategies of agents other than k. The set of strategy profiles, i.e. agents' joint strategies, is defined to be:

$$\mathcal{SP} = \{\mathcal{S}_i\} = \{(str_i^1, str_i^2, \ldots, str_i^K)\}. \tag{4}$$

where $str_i^k \in \mathcal{S}tr^k$, and $i = 1, \ldots,$ represents the profile index.

The payoff matrix P of an empirical game can be generated using empirical analysis techniques. Here, we focus on simulation, where agents engaged in the underlying game act according to policies adhering to specific strategies. Subsequently, we use the terms *action* and *policy* when speaking about the underlying game, and the term *strategies* or *styles of play* when speaking about the empirical game. The matrix is computed for each strategy profile as follows:

$$P^k(str^k, str^{-k}) = \frac{1}{N} \cdot \sum_{i=1}^{N} P_i^k(str^k, str^{-k}) \tag{5}$$

where N is the number of simulation runs, str^k represents player k's strategy, str^{-k} denotes the strategies of the other players, and $P_i^k(str^k, str^{-k})$ (with an abuse of notation) represents the payoff player k receives in simulation run i when playing strategy str^k against the strategies of the other players. It must be noted that in contrast to dynamic games the payoff function does not take states as arguments, as the outcomes are determined by agents' joint strategies, i.e. $P^k : (\mathcal{S}tr^k \times \mathcal{S}tr^{-k}) \rightarrow \mathbb{R}^K$ [13]. If we aggregate these expected payoffs into a matrix, we get the empirical payoff matrix whose dimensionality is $\prod_{k=1}^{K} \mathcal{S}tr^k$. Each entry represents the expected payoff for strategy str^k against strategy str^{-k}.

2 The α-Rank Method

Evolutionary dynamics studies how agents' interactions in multi-agent settings evolve over time. While single-agent systems have acquired a strong foundation over the years [1], multi-agent systems are more challenging to analyze.

Current literature indicates a growing interest in studying the evolutionary dynamics of multi-agent systems. Although one might view evolutionary algorithms as mere tools for agents' hyper-parameter tuning [17][5], their contributions extend far beyond that. In the context of games, evolutionary algorithms are widely used to explore game-theoretic solution concepts. This area of study is also known as *Evolutionary Game Theory* (EGT). An example of research in this field is the work reported in [14], which introduced an evolutionary algorithm for multi-agent path-finding in stochastic environments. This approach showed significant improvements in minimizing path length and computational efficiency, outperforming state-of-the-art reinforcement learning algorithms. In another work [4], a novel approach was introduced for evolving the key components –mainly the evaluation function and search mechanism– of a chess program from randomly initialized values using genetic algorithms. By learning from databases of grand-master games, the program managed to outperform a world chess champion computer.

Building on work done in EGT, the evolutionary methodology *α-Rank* [13] introduces a novel game-theoretic approach to provide insights into the long-term dynamics of agents' interactions. At its core, *α-Rank* is designed to evaluate and rank strategy profiles in large-scale multi-agent interactions, using a new dynamic solution concept called *Markov-Conley chains* (MCCs). It achieves this by calculating a stationary distribution from the transition probabilities between strategy profiles, reflecting how much time agents spend using each profile. This stationary distribution is then used to rank the strategy profiles, with the ranking intensity α adjusting the sensitivity of the rankings to the stability of the strategies over time.

2.1 Markov-Conley Chains

Markov-Conley chains (MCCs) are a dynamic solution concept that extends the traditional idea of Nash equilibrium by considering the evolution of strategies over time, rather than focusing solely on fixed points. MCCs model the long-term behavior of agents' interactions within a dynamical system to identify stable components where agents reach equilibria. They provide a more dynamic perspective on stability by shifting the focus from static points, where no player benefits from unilaterally changing their strategy, to trajectories that define how equilibrium is reached over time.

To better understand the structure of MCCs, we can view the dynamics of agent interactions as flows within a topological space, where each strategy is represented as a point in this space. These flows describe how the system evolves over time, with the state of the system at any given moment depending solely on the current strategy, as defined by the Markov property:

$$P(str_{t+1} = str_j \mid str_t = str_i) = P(str_{t+1} = str_j \mid str_t = str_i, str_{t-1}, \dots) \tag{6}$$

where $P(str_{t+1} = str_j \mid str_t = str_i)$ represents the probability of transitioning from strategy str_i to str_j at time $t+1$, independent of past strategies. These dynamics can be expressed mathematically using a flow ϕ_t defined on a topological space X:

$$\phi_t : X \to X \tag{7}$$

For each time step $t \in \mathbb{R}$, the flow maps a point $x \in X$ to another point in the space, $\phi_t(x)$. This mapping represents the evolution of the agent's strategy over time, with x being the current strategy and $\phi_t(x)$ the updated one at time t.

This concept is further supported by *Conley's Fundamental Theorem of Dynamical Systems*, which divides the state space into recurrent sets, representing stable behaviors, and transient points that eventually lead to recurrent sets [2][12]. These recurrent sets, including fixed points, periodic orbits, and limit cycles, correspond to different forms of equilibrium. In the context of multi-agent games, the long-term dynamics of agent interactions can be modeled similarly, where each point in the space corresponds to a joint strategy (strategy profile). Specifically, these dynamics can be visualized through a graph, known as the *response graph*, where each node represents a strategy profile and edges represent transitions between them. The main structures in this graph are the *strongly connected components* (SCCs), which correspond to the recurrent sets in the topological space. Once players enter these components, they tend to remain, indicating an equilibrium [13].

2.1.1 Chain Recurrent Set The *chain recurrent set* $\mathcal{R}_\phi$ of the flow ϕ_t is the set of all points $x \in X$ that are chain recurrent under the flow ϕ_t. A point x is chain recurrent if there exists an (ϵ, T)-chain from x to itself, meaning there exists a sequence of points $(x_0, x_1, \dots, x_n)$ connecting back to x, with each step being arbitrarily close to the previous one.

Definition 2.1 (Chain recurrent point). *An* (ϵ, T)*-chain from* x *to itself, with respect to the flow* ϕ_t *and distance function* d*, is a sequence* $(x_0, x_1, \dots, x_n)$ *such that:*

$$d(\phi_{t_i}(x_i), x_{i+1}) < \epsilon \quad \textit{for each} \quad i = 0, 1, \dots, n-1, \quad t_i \geq T$$

where $\epsilon > 0$ represents the allowed perturbation at each step, and $T > 0$ is the minimum time-step between transitions. If such a chain exists from x back to itself, the point x is chain recurrent.

2.1.2 Transient Points

A point $x \in X$ is called *transient* if it is not chain recurrent. This means that there does not exist an (ϵ, T)-chain from x to itself, and trajectories starting from a transient point eventually leave every neighborhood of x without returning. Transient points do not exhibit any form of cyclic behavior; they eventually *escape* from their initial region.

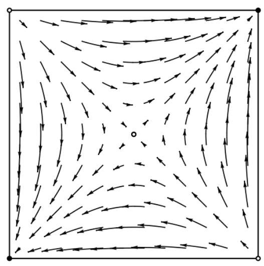
Definition 2.2 (Transient point). *A point $x \in X$ is transient if $x \notin \mathcal{R}_\phi$.*

To visualize the flow representation of strategy evolution in agent interactions, consider the *Coordination Game* in Table 1a. This is a cooperative game where both players benefit from coordinating on the same strategy, A or B, rather than playing different strategies. Figure 1 illustrates the evolutionary dynamics of the agents' strategies in a graph, as designed by the authors in [13].

Each point in the graph corresponds to a mixed strategy profile for the two players, while the arrows represent the flow of strategy profiles over time. The recurrent points in the graph (bottom-left and top-right corners) correspond to the two stable profiles: (A, A) and (B, B). These two recurrent points represent the attractors of the system, where the flows converge. On the other hand, (A, B) and (B, A) (top-left and bottom-right corners) are the transient points of the graph. The flow field directs trajectories away from these points, indicating their instability. Finally, the middle point in the graph represents a mixed strategy profile where both players assign equal probability to strategies A and B (i.e., 50% for each in the probability distribution). Trajectories near this point diverge, moving towards one of the two recurrent points. This behavior represents the mixed-strategy Nash equilibrium of the *Coordination Game*.

	A	B
A	1,1	-1,-1
B	-1,-1	1,1

(a) Payoff matrix.



(b) Response Graph.

Figure 1: Payoff matrix (a) and Replicator dynamics (b) in the Coordination Game.

To distinguish between recurrent sets and transient points in a dynamical system, the creators of α-*Rank* use a complete *Lyapunov function* [10]. This function, $\gamma : X \to \mathbb{R}$, assigns values to points in the space such that:

- For transient points, the value of $\gamma(\phi_t(x))$ strictly decreases over time as the system evolves along the flow ϕ_t. This reflects how transient points are driven towards the recurrent part of the space.
- For points within the recurrent set $\mathcal{R}_\phi$, $\gamma(\phi_t(x))$ remains constant. This signifies that all points within the same chain component share equivalent dynamic properties.

MCCs offer a discrete-time approximation of the continuous-time dynamics described by *Conley's Fundamental Theorem*. In the context of MCCs, recurrent sets correspond to SCCs in the response graph, where strategy profiles remain stable over time. On the other hand, transient points are strategy profiles that are not part of any SCCs and eventually escape the system's flow.

The stability of strategy profiles in MCCs is determined by the transition probabilities, similarly to the way the *Lyapunov function* tracks the evolution of a dynamical system. Strategy profiles within a SCC are considered stable because the transition probabilities between them are balanced. This means that, in the SCC, the probability of transitioning from one profile to another is not biased in any particular direction. Each strategy profile has an equal chance of moving to another one within the same SCC. This balance in transition probabilities creates a situation where, once an agent enters the SCC, they are likely to remain within the same set of strategy profiles over time. In contrast, strategy profiles that are not part of any SCC are considered unstable because their transition probabilities are structured in a way that makes moving to a different profile more likely than staying in the current one. This leads them to eventually escape the system's flow.

2.2 Strategy Evolution Process

Given a K-player game, α-*Rank* considers the empirical game with K player slots, called $populations$, where individual agents correspond to strategies, i.e. to styles of playing the underlying game. In α-*Rank*, populations of agents interact with each other through an evolutionary process following the dynamics of games. The rewards received from these interactions determine how well each strategy performs and, in turn, how often it is adopted by individuals in the populations. Strategies that perform well have a higher probability of being adopted and carried over to the next generation, while those performing poorly are less likely to be adopted. This process of competition and selection between populations leads to their evolution.

To facilitate evolution, α-*Rank* uses the concept of mutation. Initially, populations are monomorphic, meaning all individuals within them choose the same strategy. During K-wise interactions, individuals have a small probability of mutating into different strategies or choosing to stick with their current one. The probability that the mutant will take over the population, defined to be the fixation probability function ρ, depends on the relative fitness of the mutant and the population being invaded. Fitness is a function that computes the expected reward an individual can receive when adopting a particular strategy, given the strategies of the other individuals. The stronger the fitness, the more likely it is for individuals to mutate, whereas the lower the fitness, the more likely it is for the mutant to go extinct. When the mutation rate is small, the fitness for any agent k is:

$$f^k(str^k, str^{-k}) = P^k(str^k, str^{-k}), \tag{8}$$

where P represents the empirical game payoff. Formally, the probability of a mutant strategy str' fixating in some population where individuals play strategy str is given by:

$$\rho_{str \to str'} = \frac{1 - e^{-\alpha \cdot \Delta f}}{1 - e^{-\alpha \cdot m \cdot \Delta f}} \tag{9}$$

assuming that Δf is non-zero. $\Delta f = f^k(str', str^{-k}) - f^k(str, str^{-k})$ represents the difference in fitness between the mutant strategy str' and the resident strategy str in the focal population k, while the remaining $K-1$ populations are fixed in their monomorphic strategies str^{-k}. Parameter m is the population size and α is the selection intensity. This adjusts the sensitivity of the system to fitness differences: with higher values of α, even small differences in fitness lead to larger changes in ρ. The nominator measures the potential of the mutant to "invade" the resident population solely based on its fitness advantage. Note that, for example, as Δf approaches zero, the probability of the mutant's success decreases. The denominator, on the other hand, normalizes the fixation probability using the population size m, making it more challenging for a mutant to dominate in larger populations. When Δf is zero, the fixation probability equals to $1/m$, indicating that the mutant strategy has the same probability of taking over as any other strategy in the population. This probability is referred to as the *neutral fixation probability*, denoted by ρ_m.

2.3 Modeling Dynamics through MCCs

In the context of K-player games, α-*Rank* creates a Markov transition matrix over strategy profiles. This is an $|\mathcal{S}tr| \times |\mathcal{S}tr|$ matrix that defines the probability of moving from one strategy profile to another based on how likely each population is to change its strategy.

$$C_{str \to str'} = \begin{cases} \eta \cdot \rho_{str \to str'} & \text{if } str \neq str' \\ 1 - \sum_{str \neq str'} C_{str \to str'} & \text{otherwise} \end{cases} \tag{10}$$

Here, C is the strategy-transition matrix where each entry $C_{str \to str'}$ represents the probability of transitioning from strategy str to strategy str'. The first part of the formula, calculates the probability of transitioning from one strategy to a different one, scaled to ensure that the sum of probabilities for all possible transitions from that strategy sums up to 1. The second part of the formula, computes the probability of staying with the same strategy, str, by excluding transitions to all other strategies.

This evolutionary process of competition and selection among players' strategies leads to a unique stationary probability distribution π of dimensionality $|\mathcal{SP}|$, where the mass assigned to a strategy profile indicates how likely it is to resist being "invaded" by other strategies as the dynamics evolve. To evaluate and rank strategy profiles —which is the ultimate goal— the method calculates π over the game's Markov chain, using the strategy-transition matrix C. This distribution indicates how often the system is likely to remain in each profile over time, allowing us to identify the most dominant strategies that are expected to prevail in the long run. Formally, π can be computed from the following equation:

$$\pi C = \pi \Rightarrow \pi(C - \mathbb{I}) = 0 \tag{11}$$

where $\mathbb{I}$ is the identity matrix (see Equation 12 for the corresponding linear system representation). This means we are looking for a probability vector π such that when multiplied by the transition matrix C, it remains unchanged. To solve for π, the augmented matrix from $C - \mathbb{I}$ is constructed and a normalization condition to ensure that probabilities sum to 1 is imposed[2]. In this stationary distribution, $\pi = (\pi_1, \pi_2, ..., \pi_{|(SP)|})$, each π_i represents the average time the system spends in strategy profile i.

$$\begin{bmatrix} C_{11}-1 & C_{12} & \cdots & C_{1n} & | & 0 \\ C_{21} & C_{22}-1 & \cdots & C_{2n} & | & 0 \\ \vdots & \vdots & \ddots & \vdots & | & 0 \\ C_{n1} & C_{n2} & \cdots & C_{nn}-1 & | & 0 \\ 1 & 1 & \cdots & 1 & | & 1 \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_n \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{12}$$

[2]The system $\pi(C - \mathbb{I}) = 0$ by itself does not have a unique solution, as there are infinitely many vectors π that satisfy it. To get a unique solution $\pi = (\pi_1, \pi_2, ..., \pi_{|(SP)|})$, it must hold that $\sum_i \pi_i = 1$.

3 Identifying Strong Joint Policies

In this section, we address the challenge of identifying strong joint policies in dynamic games, focusing on strategies that are stable over time and robust to fluctuations. We begin by defining the core problem: how to identify stable joint strategies that persist across multiple iterations of the game. This involves a deeper understanding of how agents' behaviors evolve and interact within the context of dynamic decision-making. We will present the problem statement that outlines the key challenges in modeling and analyzing strategies in such games.

Following this, we propose an approach designed to identify and evaluate stable joint strategies in dynamic games. The methodology uses α-*Rank* to provide insights into the stability and effectiveness of strategies based on the long-term interactions between agents.

3.1 Problem Statement

In dynamic games, understanding the long-term effect of agents' behaviors is crucial for identifying stable and effective joint strategies. In this context, stability refers to strategies that persist over time—strategies that are robust to fluctuations and deviations. These strategies are considered strong because they are well-aligned with the game's structure and the agents' payoff expectations during interactions. To identify such stable strategies, one must define and analyze the payoff matrix.

Defining the payoff matrix in static games is relatively straightforward. For example, in Rock-Paper-Scissors, where strategies are individual actions, the payoffs can easily be determined by the game's rules, such as rock beating scissors (see Table 1). However, in dynamic games, where policies consist of sequences of actions, defining the payoff matrix is more complex. Even if we manage to estimate payoffs, computing solution concepts like the *Nash equilibrium* can be computationally expensive and may not guarantee convergence, especially in complex or large games. Beyond simply identifying stable joint strategies, it is also crucial to explain why one strategy is better than another. This involves more than just ranking strategies; it requires providing clear evidence for why some strategy profiles are preferred [20], ensuring transparency in the decision-making process.

We could, therefore, consider our problem as follows: Given a dynamic game G with K players, our goal is to identify styles of playing G, and thus, the set of strategy profiles $\mathcal{SP}$, and rank these profiles based on how stable they are over time, considering long-term agents' interactions towards achieving their objectives. Specifically, we aim to define a ranking function:

$$\mathcal{R} : \mathcal{SP} \rightarrow \mathbb{R} \tag{13}$$

where $\mathcal{R}(\mathcal{S}_i) > \mathcal{R}(\mathcal{S}_j)$ (resp. $\mathcal{R}(\mathcal{S}_i) \geq \mathcal{R}(\mathcal{S}_j)$) indicates that the strategy profile $\mathcal{S}_i$ is strictly (resp. weakly) preferred over $\mathcal{S}_j$. In conjunction to that, we aim at providing a descriptive framework to promote transparency on how rankings are decided:

$$\mathcal{D} : \mathcal{SP} \times \mathcal{SP} \rightarrow \mathbb{R} \tag{14}$$

Empirical game strategies are realized by agents' policies adhering to these strategies in the underlying game. Thus, identifying stable joint strategies in the empirical game translates to identifying stable joint policies adhering to these strategies in the underlying dynamic game.

3.2 Proposed Methodology

To address the challenge of identifying stable joint policies in dynamic games, we propose an approach that combines concepts from *Empirical Game Theory* and *Evolutionary Dynamics*, using α-*Rank*, providing transparency to rankings of agent's styles of play.

Given that the set of agents' policies in dynamic games can be infinitely large we focus on a subset of policies that adhere to concrete and well-defined styles of play. A way to identify styles of play is to observe how players behave in the underlying game or exploit demonstrations of game playing. For instance, human experts performing a task usually follow a distinct set of specific styles based on well-established practices, preferences and experience. Having determined the game playing strategies, we can transform the dynamic game into its empirical form, defining the meta-game by:

(a) Identifying empirical game strategies.

(b) Training policies for agents to play the underlying game according to these strategies.

(c) Defining the empirical game payoff matrix, through simulations, exploiting the trained policies.

Once the meta-game is defined, the next step is to define the function $\mathcal{R}$, which ranks joint strategies based on agents' long-term dynamics and objectives. To achieve this, we propose using the evolutionary methodology α-*Rank*, which provides rankings by assessing the evolutionary success of each strategy profile. This is reflected in the probability of a given strategy profile being selected over time. This probability is captured by the stationary distribution π, which is computed by α-*Rank* in the limit of infinite ranking intensity α. As demonstrated earlier, once α reaches a sufficiently large value, the rankings stabilize, accurately capturing the system's long-term behavior.

To compute the stationary distribution π over strategy profiles, the α-*Rank* methodology requires the payoff matrix P of the empirical game. Along with the stationary distribution π, α-*Rank* also outputs the fixation probability function $\rho_{\mathcal{S}_i \to \mathcal{S}_j}$, where $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{SP}$. One could abstractly illustrate α-*Rank* as a function:

$$\alpha\text{-Rank}(P) \to (\pi, \rho) \tag{15}$$

While the stationary distribution π provides valuable insight into the long-term behavior of strategies, it alone does not help us fully understand how strategies transition between one another. The fixation probability function ρ, which measures the likelihood of transitioning from one strategy profile $\mathcal{S}_i$ to another $\mathcal{S}_j$, fills this gap. Based on this, the descriptive framework $\mathcal{D}$ can be adequately represented by π and ρ, which are constituents of the response graph, providing a complete view of the empirical game's dynamics.

Overall, building on the α-*Rank* descriptive framework, the method proposed here for computing strategy profile rankings in dynamic games is as follows:

Algorithm 1 Ranking Joint Policies in Dynamic Games

```
Identify players' styles of play.
Define the strategies of the empirical game based on those styles.
Train policies realizing the defined strategies.
Run game simulations to create the empirical payoff matrix P.
Apply α-Rank to define R and D:
    Calculate the Markov transition matrix C.
    Find the unique stationary distribution π.
    Rank joint strategies by ordering the masses of π.
    Describe the rankings through the response graph.
    Study the effect of different α values on π.
```

4 The Graph Coloring Game

The *Graph Coloring Problem* (GCP) is one of the most well-known problems in graph theory. It involves assigning colors to vertices in a graph such that no two adjacent vertices share the same color, and using the minimum number of colors, also known as the chromatic number [21].

In this study, we shift our focus from finding the chromatic number across graph configurations to a slightly different approach. Rather than solving the GCP itself, we use the concept of graph coloring to define a dynamic game that allows us to study the evolutionary dynamics in multi-agent interactions. Nonetheless, the concept of the chromatic number will reappear in our analysis, not as a problem to be solved but as a parameter to define a style of play. Through this, we aim to gain insights into whether individuals with this preference towards minimal solutions in terms of color actually ensure good collaboration, or if other styles of play are more effective when combined. To provide a consistent framework for our study, we assume a fixed grid configuration and a fixed number of colors available to the agents, as a static base to study agent interactions. Specifically, we set the number of colors to match the total number of blocks in the grid, allowing agents to consider even the most extravagant coloring approach, in cases when assigning a unique color to each block is the only valid solution.

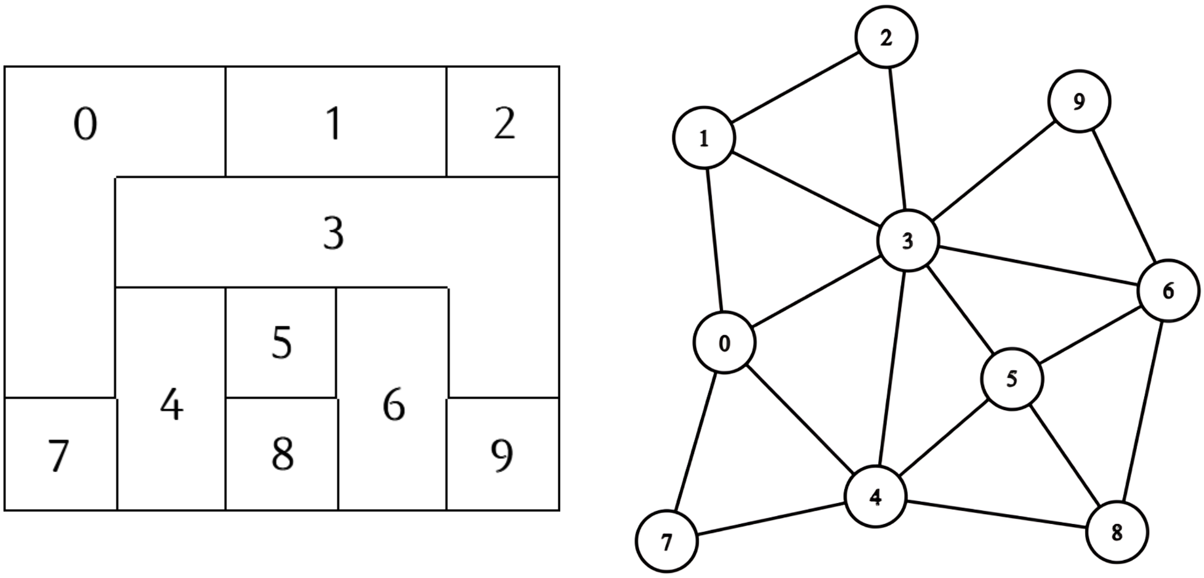


Figure 2: A snaphsot of the game environment in grid and graph forms.

4.1 State and Action Space

The graph in the game corresponds to a grid of blocks, where each block is defined as one or more merged cells within the two-dimensional matrix. At the beginning of the game, the grid is initialized with a random number of rows and columns ($n \times m$), and this configuration remains the same throughout the entire game. The environment then randomly combines cells to create blocks. Merging cells is important as it allows for complex neighboring relationships to be defined, expanding beyond the standard constraints between adjacent blocks. In our experimental setup we assume a 4×5 grid with 10 blocks as illustrated in Figure 2.

Blocks can be either: (a) colored by the agents, (b) white i.e., available to be colored and (c) hidden i.e., cannot yet be colored until revealed by the environment. Initially all blocks in the grid are hidden and symbolically represented in gray. Let G denote the grid environment, B the set of blocks in that environment and CR the set of possible colors that an agent can use for coloring blocks. Then our first assumption is that every block b in B can potentially be assigned any color c from CR:

$$\forall b \in B,\, \forall c \in CR,\, (b, c) \in R \tag{16}$$

The set of agents' actions A is defined to be the Cartesian product of the set of blocks B and the set of the available colors CR, denoted as:

$$A = B \times CR = \{(b, c) \mid b \in B,\, c \in CR\} \tag{17}$$

To specify states, let CR^* include the elements of CR, and two additional elements representing hidden and white blocks: $CR^* = CR \cup \{\text{hidden}, \text{white}\}$. We define state s as the Cartesian product of B and CR^*, denoted as:

$$s = \{(b_i, c_i), i = 1, \ldots, |B|\}, s.t. \forall b \in B,\, \exists \text{a unique } c \in CR^*, \text{with } (b, c) \in s \tag{18}$$

We consider the *Graph Coloring Game* (GCG) to be a two-player game that unfolds over multiple rounds. At the beginning of each round, the environment reveals the color of some of the hidden blocks, if any. The number of blocks that get un-hidden is random, which implies that the state of the graph is influenced not only by the agents' actions, but also by the environment. We therefore consider the game to be stochastic. After the state update from the environment, the agents simultaneously choose their actions. As soon as all blocks in B are uncovered and colored, the game ends.

4.2 Reward Function

The reward function in the GCG is designed to encourage agents to adopt effective strategies for coloring blocks while discouraging undesirable ones. The total reward that an agent receives during the game is the sum of multiple components, including gains, penalties, sanctions, delays and adopted preferences.

These components are defined as follows:

1. **Gains:** An agent receives a gain of +1 for each neighboring block b' of b, such that $c \neq c'$ where c is the color of b and c' is the color of b'. This reward encourages agents to color neighboring blocks with different colors, in line with the rules of the GCP, where adjacent blocks must have distinct colors.

2. **Penalties:** An agent receives a penalty of -2 for each neighboring block b' of b, such that $c = c'$. This reward serves the same purpose of ensuring proper coloring by discouraging agents from coloring adjacent blocks with the same color, which would violate the rules of the GCP.

3. **Sanctions:** An agent receives a sanction, a large negative reward of -10, when it attempts to color a hidden block or a block that has already been colored. This reward ensures that agents follow the game's rules and do not waste their actions on illegal moves.

4. **Delays:** If both agents attempt to color the same block b during the same round, a small negative reward of -1 is applied to both agents. This reward represents the time spent in resolving the conflict between the two agents over who will color the block.

5. **Preference-Adoption:** Regardless of whether an agent's action is successful, unsuccessful, or forbidden, the agent receives a preference-adoption reward for adhering to specific preferences or styles of play. This reward encourages agents to adopt behaviors that align with desired goals.

5 Identifying strong joint policies in the Graph Coloring Game

In this section, we apply our methodology to the *Graph Coloring Game* (GCG) defined in Section 4, to identify strong joint policies. This analysis allows us to provide an explainable link between the observed policies and the actions they refer to.

We begin by defining the different styles of play and modeling them as policies using *neural network* (NN) models. Next, we define the empirical game payoff matrix, which measures how well these policies interact with each other over the long run. Finally, we apply α-*Rank* to evaluate and rank the joint policies, gaining insights into the game's dynamics.

5.1 Defining Styles of Play

The process of transforming the underlying dynamic GCG into its empirical form begins by understanding the different ways in which agents play the game. This involves two essential steps:

1. Identifying agents' strategies.
2. Constructing the empirical game payoff matrix.

Agents' strategies in the game can be thought of as different styles of play, which are typically revealed through preferences and behaviors in response to the game's structure. In our experiments, we specify different styles across three main dimensions: color tone (preference for which colors to use), block difficulty (preference for the types of blocks to choose), and coloring approach (preference for the number of colors to use), as shown in Table 2. These dimensions are combined to define a player's overall style, which can range from complete indifference—where no specific preference is observed in any of the dimensions (denoted by "I")—to highly specific preferences across all dimensions.

Table 2: Dimensions specifying agents' strategies.

Preference Dimension	*Value*
Color Tone	warm (W) vs. cool (C)
Block Coloring Difficulty	small (L) vs. large (A)
Coloring Approach	minimalistic (M) vs. extravagant (E)

5.2 Realizing the Empirical Game Strategies

Policies corresponding to specific strategies are realized using *convolutional neural networks* (CNNs), which are trained to adapt to different styles of play in the GCG. To guide the training of these policies, we assign specific values to each of the three dimensions of the game's *preference reward* —color tone, block difficulty, and coloring approach. These values, range from -1 to 1, where 1 indicates a strong preference for a particular dimension. For example, a value of 0.7 for warm colors suggests a relatively strong preference for warm tones, while values closer to 0 a tendency towards indifference.

In our experimental setting, we define a total of 11 distinct styles of play: I, C, W, E, M, L, A, AE, CA, LE and WL, given "I" and combinations of preference values specified in Table 2. Assuming no inherent bias among the players of the empirical game, we allow populations to sample from the same list of strategies.

All the policy models used to realize the empirical game strategies share a common underlying architecture and training setup. Although hyperparameter tuning is typically recommended, it doesn't make much difference in this case, as these models are relatively easy to optimize when trained in small settings. Regarding the CNN architecture, it consists of four convolutional layers, each defined with a kernel size of 3, stride of 1, and padding of 1. These parameters are chosen to ensure that the model is capable of extracting spatial features from the input data, which in this case is a grid representation of the game state.

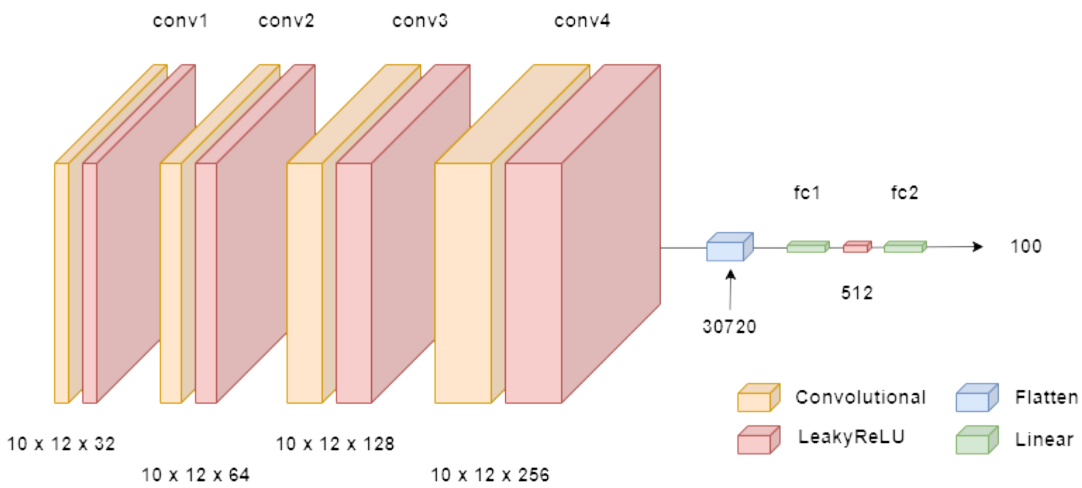


Figure 3: Convolution Policy Network Architecture.

The input tensor has dimensions 10×12, where $|B| = 10$ represents the number of blocks in the state and $|CR^*| = 12$ represents the number of possible colors a block can have. Each block is encoded using one-hot encoding, meaning that each color is represented as a binary vector of length 12. The output is then flattened and passed through two *fully connected* (FC) layers, which process the data to produce the final output, as shown in Figure 3.

Policy models are trained individually in the underlying game using the deep Q-learning reinforcement learning algorithm specified in Algorithm 2. We set γ to 0.7. To optimize the model parameters, we use the smooth L1 loss function with β=1.0 and the Adam optimizer with a learning rate of 5e-4 and weight decay of 1e-5 to prevent over-fitting. To further enhance the learning process, we incorporate experience replay, with a memory that stores up to 10 million experiences [9]. A target network alongside the main policy network, is being used according to the *Double-DQN* approach [6]. To update the target network we apply a soft update with a factor τ=5e-3. This gradually brings the target network closer to the policy network, balancing learning speed and stability. With a batch size of 64, we train the models for 10,000 episodes.

Algorithm 2 Double Deep Q-Learning with Experience Replay

Q_θ, $Q_{\theta'} \leftarrow Q_\theta$, M ▷ Initialize policy/target nets & memory
for episode **do**
  $s \leftarrow s_0$
  for step **do**
    $a \leftarrow \text{argmax}_a Q_\theta(s)$ ▷ Select ϵ-greedy action
    $(s, a, r, s') \in M$ ▷ Store experience
    if $|M| >$ batch size **then**
      for each (s, a, r, s') in M **do** ▷ Sample memory
        $y \leftarrow r + \gamma \max_{a'} Q_{\theta'}(s')$
        $L \leftarrow \text{Loss}(Q_\theta(s), y)$
        $\theta \leftarrow \theta - \alpha \nabla_\theta L$
      end for
    end if
    $Q_{\theta'} \leftarrow \tau Q_\theta + (1 - \tau) Q_{\theta'}$ ▷ Soft update
    $s \leftarrow s'$
  end for
end for

In this setup, the agents are trained without any co-players, meaning that each model learns in isolation. This approach is intentional, as our goal is to develop agents that play optimally on their own, rather than in collaboration with others. The idea is to explore how different styles of independent players interact with each other. We expect that some styles will lead to more conflicts than others, and this behavior is key to our analysis. If we had trained the agents together, for example using a *multi-agent reinforcement learning* (MARL) approach designed for collaborative settings, the resulting agents would have learned joint policies, which would defeat the very purpose of evaluating how their individual policies affect collaboration.

Let us consider, for instance, the simulation statistics of two relatively compatible play styles that we expect to perform well together in the game: Player W (a robot player with preferences for warm color tones) and player C (a human with preferences for cool color tones). In Figure 4, we observe how these players' individual preferences shape their interactions.

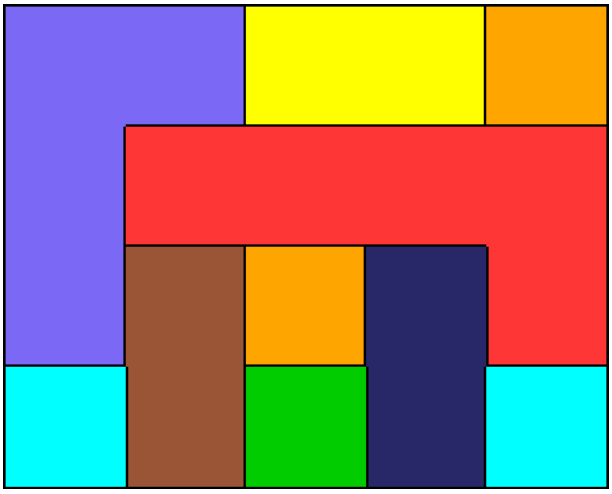


Figure 4: Example solution to the game showing how C player and W player interact based on their preferences.

The statistics in Figure 5 reveal the most frequently selected actions in terms of blocks and colors. Player C prefers colors like blue, purple, and green, while player W tends to choose colors like brown, orange, and red. Given these preferences, conflicts are unlikely to arise from color selection alone. Even in scenarios where they may choose to color neighboring blocks, it is highly impossible that both players will choose the same color.

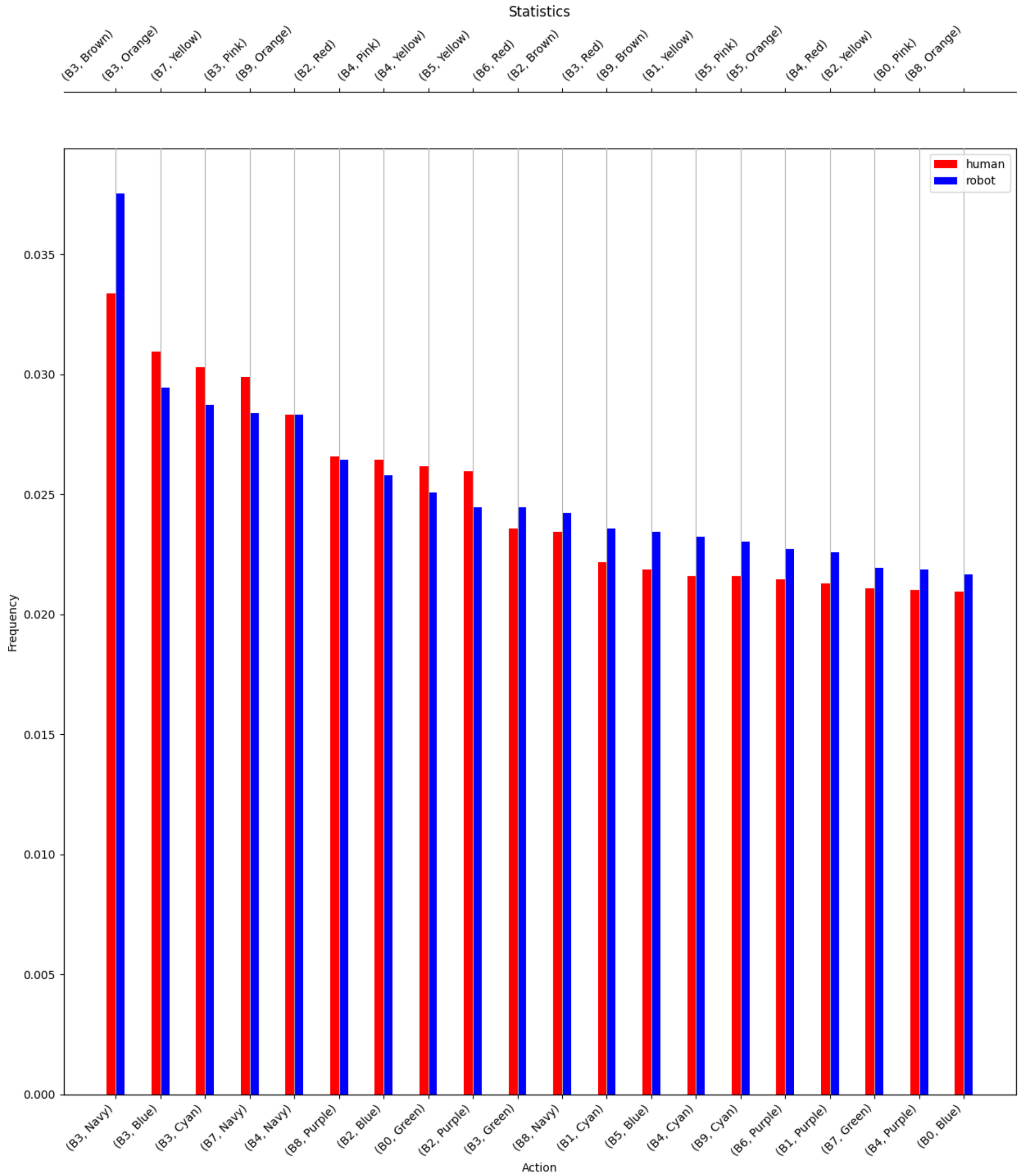


Figure 5: Statistical analysis of the most frequently selected actions by C player and W player.

On the other hand, in Figure 6, we observe the interactions between two C players. In this case, we expect more conflicts, as both players share similar color preferences. This increases the likelihood of both players selecting the same color for neighboring blocks.

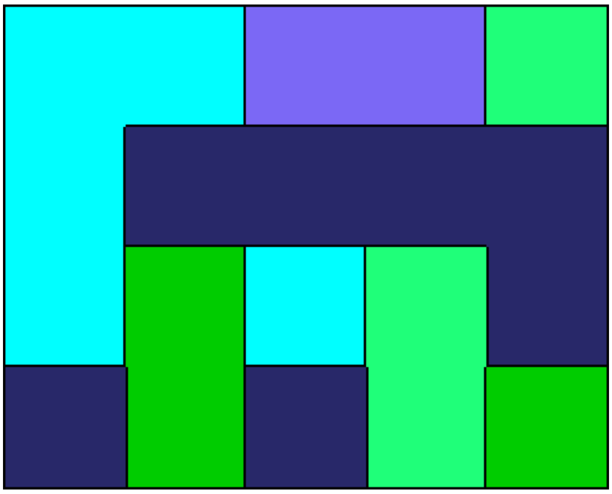


Figure 6: Example solution to the game showing how two C players interact based on their preferences.

The statistics shown in in Figure 7 further support this, as they reveal a high probability of color overlap, primarily due to the dominance of cool colors in both action spaces. However, these conflicts are not a result of insufficient training, but rather stem from the inherent similarity in preferences.

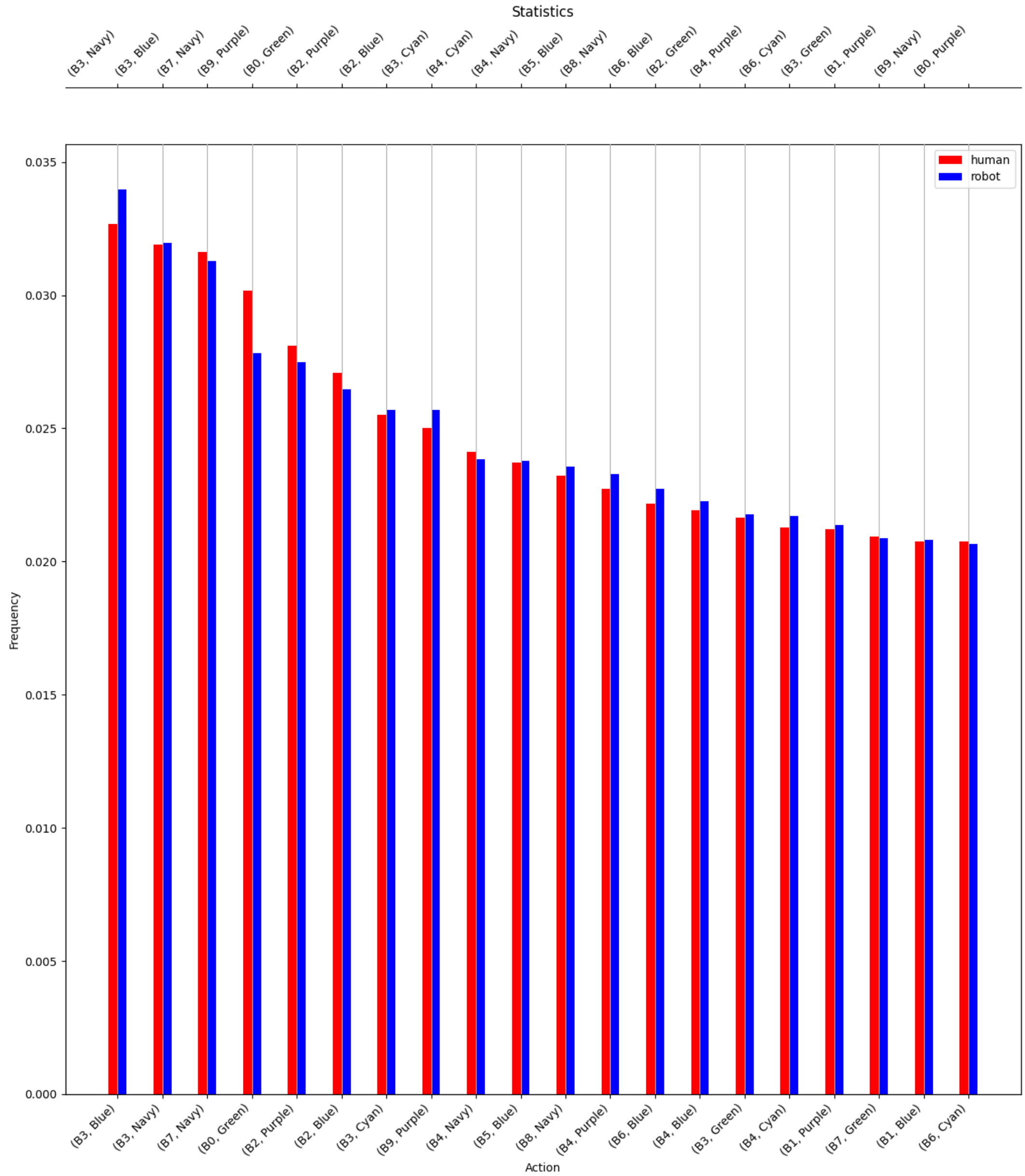


Figure 7: Statistical analysis of the most frequently selected actions by two C players.

Other than that, both agents have been thoroughly trained in isolation, exploring a wide range of states and ultimately converging to stable policies. This allows them to respond effectively and optimally, even when paired with agents with conflicting styles of play. Therefore, even in the case of the most incompatible pairings, the number of mistakes remains minimal, and these mistakes are not due to inadequate exploration, but rather to the inherent conflicts in the agents' preferences.

5.3 Extracting the Empirical Payoff Matrix

We generate the empirical payoff matrix by simulating each strategy profile over multiple games. These payoffs represent how well different styles of play perform jointly, according to the game's rules.

The values in the payoff matrix are computed in terms of the delay and the quality of the solution according to the game's constraints (gain, penalty, and sanction), excluding preferences. This ensures a common ground for distinct strategies, evaluating solutions solely based on the game's rules. For each pair of strategies, we simulate the game over 5,000 repeats and calculate the average payoff for each strategy. These values are then organized into the payoff matrix, which is provided in Table 3. Each entry in the matrix represents the payoffs of strategies in the corresponding profile, with the first value indicating the payoff of the row player and the second value of the column player. The Nash equilibria are highlighted in bold, while nine of the top-ranked strategy profiles in the MCC are shaded in gray.

Table 3: Empirical Payoff Matrix for the Graph Coloring Game.

	A	AE	C	CA	E	I	L	LE	M	W	WL
A	(3.12, 3.11)	(3.15, 3.16)	(3.17, 3.17)	(3.14, 3.17)	(3.16, 3.17)	(3.16, 3.15)	(3.22, 3.13)	(3.19, 3.16)	(3.15, 3.18)	(3.16, 3.17)	(3.21, 3.18)
AE	(3.17, 3.17)	(3.11, 3.11)	(3.18, 3.17)	(3.15, 3.17)	(3.17, 3.16)	(3.19, 3.16)	(3.23, 3.12)	(3.19, 3.16)	(3.15, 3.18)	(3.17, 3.17)	(3.20, 3.16)
C	(3.17, 3.16)	(3.16, 3.17)	(3.10, 3.10)	(3.14, 3.17)	(3.15, 3.15)	(3.18, 3.15)	(3.22, 3.12)	(3.17, 3.14)	(3.14, 3.17)	(3.17, 3.16)	(3.20, 3.17)
CA	(3.17, 3.15)	(3.17, 3.15)	(3.17, 3.14)	(3.11, 3.11)	(3.18, 3.15)	(3.18, 3.14)	(3.24, 3.13)	(3.21, 3.16)	(3.16, 3.16)	(3.19, 3.16)	(3.22, 3.15)
E	(3.15, 3.16)	(3.16, 3.16)	(3.15, 3.16)	(3.15, 3.17)	(3.10, 3.10)	(3.18, 3.16)	(3.22, 3.12)	(3.19, 3.14)	(3.15, 3.17)	(3.16, 3.17)	(3.19, 3.17)
I	(3.14, 3.16)	(3.16, 3.18)	(3.16, 3.18)	(3.15, 3.19)	(3.16, 3.17)	(3.12, 3.12)	(3.22, 3.14)	(3.18, 3.16)	(3.14, 3.19)	(3.16, 3.18)	(3.19, 3.18)
L	(3.14, 3.22)	(3.11, 3.22)	(3.12, 3.22)	(3.13, 3.23)	(3.12, 3.22)	(3.13, 3.22)	(3.12, 3.12)	(3.14, 3.20)	(3.11, 3.21)	(3.14, 3.23)	**(3.15, 3.21)**
LE	(3.15, 3.19)	(3.14, 3.18)	(3.14, 3.18)	(3.15, 3.21)	(3.15, 3.19)	(3.16, 3.17)	(3.20, 3.14)	(3.11, 3.11)	(3.14, 3.22)	(3.15, 3.18)	(3.18, 3.19)
M	(3.17, 3.14)	(3.17, 3.15)	(3.17, 3.15)	(3.16, 3.17)	(3.16, 3.14)	(3.18, 3.14)	(3.23, 3.11)	(3.20, 3.14)	(3.06, 3.08)	(3.18, 3.15)	(3.20, 3.16)
W	(3.17, 3.17)	(3.17, 3.18)	(3.16, 3.18)	(3.16, 3.20)	(3.17, 3.17)	(3.18, 3.16)	(3.21, 3.13)	(3.18, 3.15)	(3.15, 3.18)	(3.08, 3.09)	(3.19, 3.15)
WL	(3.17, 3.20)	(3.17, 3.19)	(3.17, 3.19)	(3.17, 3.22)	(3.17, 3.19)	(3.18, 3.19)	**(3.21, 3.15)**	(3.19, 3.17)	(3.16, 3.20)	(3.16, 3.19)	(3.13, 3.13)

From this matrix, we observe that (L, WL) and its symmetric counterpart (WL, L) both with payoffs of (3.15, 3.21) and (3.21, 3.15) respectively, are the only Nash equilibria. It is important to note here that these equilibria prescribe agents' strategies given that they do play the game with rational co-players, but they do not capture the overall dynamics of the game, considering the long-term effects of agents' interactions.

5.4 Evaluating and Ranking Joint Policies

Given the payoff matrix derived from the empirical analysis, we apply the α-*Rank* method to evaluate the performance of strategy profiles over time in terms of the MCC solution concept. Specifically, we ran the method 1,000 times, using values of α within the range $[0.1, 10]$ with step=0.01, while assuming populations of size $m = 100$. We provide as input the strategies defined in Section 5.1 and the empirical game payoff matrix. We focus on the rankings of the top 6 strategy profiles, to identify the stronger ones across different values of α.

As we observe from the rankings in Table 4, the strategy profile that prevails in the long run is (WL, CA); this is the primary component of the MCC. Although the table was derived using an α value of 2, the rankings remain consistent even when α is set to 10. We choose $\alpha = 2$ over $\alpha = 10$, to display the rankings of lower-performing strategy profiles, which would otherwise drop to zero. First, it is worth mentioning that the Nash equilibria (L, WL) and (WL, L) don't appear among the top-ranked strategy profiles. This is because MCC components are defined based on how well strategies perform when interacting with other strategies, based on long-term agents interactions. The individual strategies within the Nash equilibrium profile, either WL or L, may not result in favorable interactions with other strategies. As a result, the profile (WL, L) is ranked lower that others.

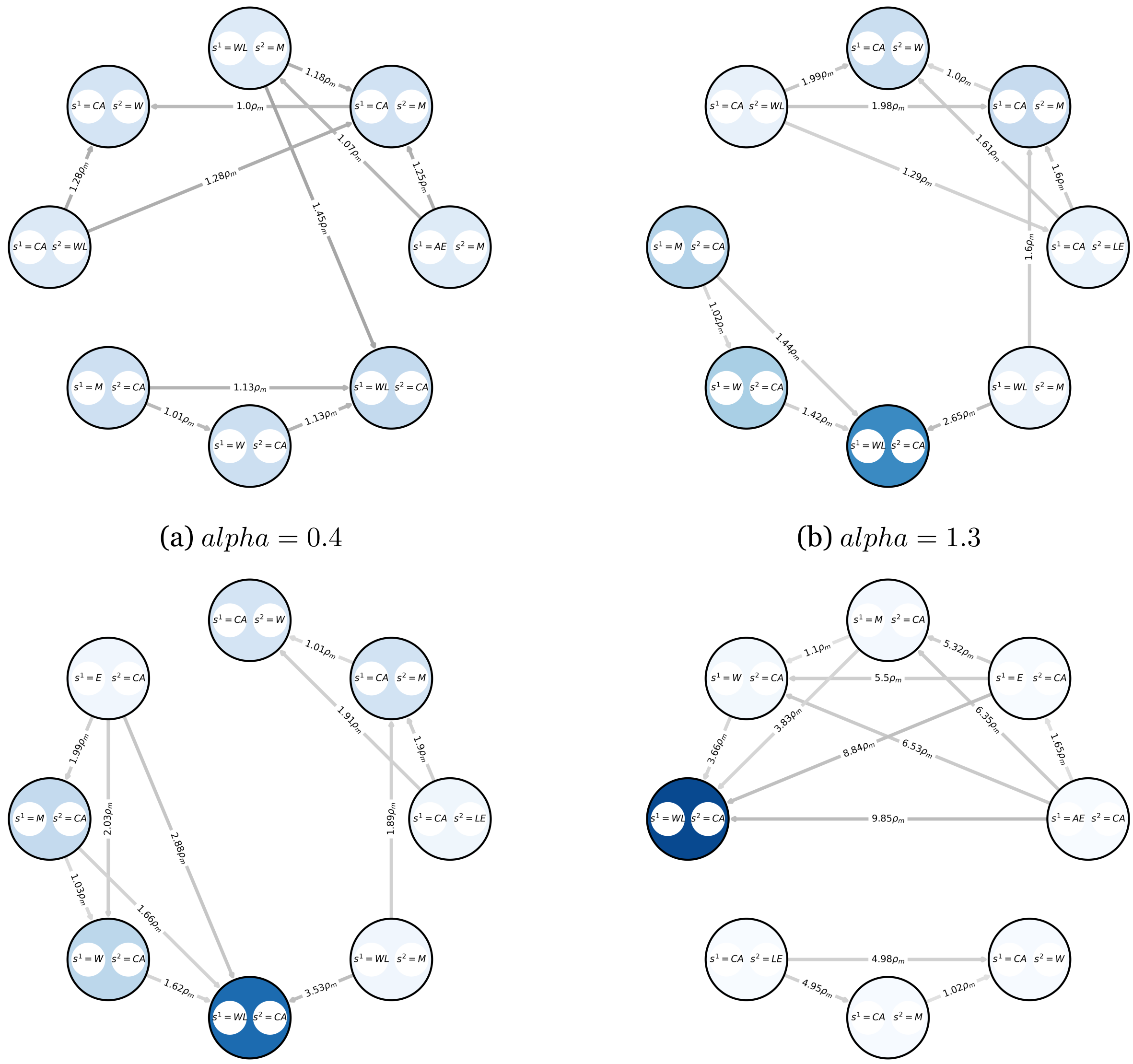
Table 4: Rankings for $\alpha = 2$.

Agent	**Rank**	**Score**
(WL, CA)	1	0.42
(W, CA)	2	0.13
(M, CA)	3	0.12
(CA, M)	4	0.08
(CA, W)	5	0.08
(CA, LE)	6	0.01

To further support our observations regarding the misalignment between the two solution concepts, let's examine why (CA, WL) is part of the MCCs, while (L, WL), the Nash equilibrium, is not. A closer look at the payoff matrix in Table 3 reveals that L appears to be the worst-performing strategy for the row player, with an average payoff of 3.13. In this case, being in the Nash equilibrium means the player is stuck with a strategy that gives low rewards, making it the best among other options, rather than a strong choice. If it happens to play this strategy, it would expect its rational opponent to play WL. Strategy CA on the other hand, is the best-performing strategy for the row player, with an average payoff of 3.18. Combined with WL, which is the best performing strategy for the column player, with an average payoff of 3.18, they make profile (CA, WL) becomes the top ranked strategy profile in the ranking Table 4.

Rankings within the MCC are also very intuitive. For example, strategies that prefer different color tones, such as (WL, CA) or (W, CA), tend to result into fewer conflicts since, they naturally avoid selecting the same colors. Similarly, strategies that prefer different blocks based on their difficulty, such as (WL, CA) or (CA, LE), tend to provide solutions with minimal delay, as they naturally avoid coloring the same blocks. Notably, profiles with mixed preferences across these dimensions demonstrate the most promising performance, which explains why (WL, CA), as such a profile, is a key component of the MCC. However, not all profile rankings can be easily explained through the game's rules alone; the expected influence of certain strategies on the quality of the solutions remains ambiguous. For example, profiles with strategies like M and E are more difficult to analyze.

The response graph provides a visualization to interpret the α-*Rank* results. This graph illustrates the MCC, using the strategy profiles' masses from the stationary distribution, π, along with the fixation probability function ρ provided by α-*Rank*. Figure 8 shows the response graphs for $\alpha = 0.4$, 1.3, 1.9 and 6.4. We consider it to be part of the descriptive framework $\mathcal{D}$, as it offers insights into how rankings were derived.



(a) $alpha = 0.4$ (b) $alpha = 1.3$

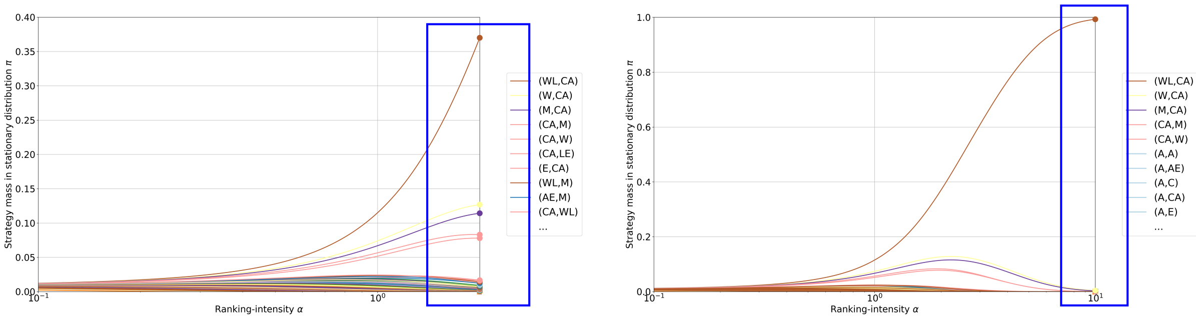
(c) $alpha = 1.9$ (d) $alpha = 6.4$

Figure 8: Response graphs of strategy profiles' dynamics.

Each node in the graph represents a unique strategy profile in the MCC, while the edges indicate transitions between them. The values on the edges show the fixation probabilities normalized by the neutral fixation probability, denoted as ρ_m. The nodes and edges are color-coded. Darker blue nodes represent more strong joint profiles, while lighter blue nodes represent transient ones. Similarly, bold arrows suggest a strong advantage in shifting between the nodes, whereas faint ones suggest less of an advantage.

The response graph describes the overall dynamics of the strategy profiles in the empirical game. One prominent feature is the primary component of the MCC, specifically the profile (WL, CA). This profile, indicated by a dark blue color, has multiple graph edges leading to it, while none from it, indicating that strategies in this profile are non-transient. This is further supported by the large fixation probabilities along the edges. A particularly prominent example is the cluster (CA, LE)-(CA, M)-(CA, W), which consists of three strongly connected profiles, indicating that once a player adopts one of these profiles, they will likely remain within their cluster. These components reflect stable regions in the game's strategy dynamics, where transitions between profiles become locked into a cycle.

To further investigate the effect of α on profile dominance, we plotted the stationary distribution π across all α values used in the experiments, for the top-performing strategy profiles (see Figure 9). This visualization —also part of $\mathcal{D}$— helps us understand how the stationary distribution changes as the selection intensity increases. The x-axis represents the different α values, ranging from 0.1 to 3 in Figure 9a, and from 0.1 to 10 in Figure 9b, while the y-axis in both figures shows the mass of each strategy profile in the stationary distribution π. As α increases, the distribution converges, indicating that the selection process stabilizes. The final mass distributions are highlighted in boxed regions. The legend on the right side of the plot displays the top-performing joint strategies, with the stronger ones appearing at the top.



(a) Mass across $\alpha \in [0.1, 3]$.

(b) Mass across $\alpha \in [0.1, 10]$.

Figure 9: Effect of ranking intensity α on strategy profile mass in the stationary distribution π.

We plot two such graphs to observe how the mass of strategy profiles is distributed in the MCCs across different α values. In the stationary distribution resulting from a bigger α, the dominant strategy profile (WL, CA) in the MCC achieves a mass of 1, with all other profiles dropping to 0. This is clearly illustrated in the second plot (see Figure 9b). However, regarding the mass distribution for a smaller range of α, depicted n the first plot, the game has not yet converged to the final MCC.

6 Conclusion

In this study, we developed a methodology for identifying strong joint-strategies in dynamic multi-agent games, accounting for stability and performance, using the α-*Rank* evolutionary algorithm. The methodology is applied on a stochastic version of the *Graph Coloring Problem*, in which players work together to color a graph while ensuring that neighboring vertices are assigned different colors. According to the methodology, first we transformed the game into its empirical form, by defining strategies (styles of play). We then designed and trained Deep Q-Learning policy models that realize those styles of play in the underlying game, and run simulations to generate the empirical payoff matrix. α-*Rank*, applied to this matrix, results into a unique stationary distribution over strategy profiles that defines the empirical game's MCC. The α-*Rank* not only helped us identify stable strategy profiles resistant to changes but also provided a descriptive framework for understanding why certain profiles prevail in the long run, based on the underlying dynamics of the game. Through this approach, we successfully described a concise methodology for evaluating and ranking agents' joint policies, considering their long-term interactions in dynamic settings, while also explaining how strategy profiles are defined within the MCC.

Future work involves (a) applying the methodology in more complex and large-scale settings, accounting for strategy profiles of multiple stakeholders that may collaborate and/or compete, (b) using machine learning methods, such as imitation learning and inverse reinforcement learning, to identify different styles of play from demonstrations and specifying the empirical game, (c) exploring advanced models able to adapt their strategies based on observed behaviors based on the behavior of co-players, (d) testing the methodology across various configurations (e.g., different graph structures and sparsities) to evaluate robustness, and (e) applying the methodology into real-world settings where agents need to align with human preferences in dynamic settings.

References

[1] Daan Bloembergen et al. “Evolutionary dynamics of multi-agent learning: a survey”. In: *J. Artif. Int. Res.* 53.1 (May 2015), pp. 659–697. ISSN: 1076-9757.

[2] C.C. Conley and Conference Board of the Mathematical Sciences. *Isolated Invariant Sets and the Morse Index*. Regional conference series in mathematics. Conference Board of the Mathematical Sciences, 1978. ISBN: 9780821816882. URL: https://books.google.gr/books?id=2hn1zAEACAAJ.

[3] Konstantinos Daskalakis. “The complexity of nash equilibria”. AAI3353239. PhD thesis. USA, 2008. ISBN: 9781109097825.

[4] Omid E. David et al. “Genetic Algorithms for Evolving Computer Chess Programs”. In: *IEEE Transactions on Evolutionary Computation* 18.5 (Oct. 2014), pp. 779–789. ISSN: 1941-0026. DOI: 10.1109/tevc.2013.2285111. URL: http://dx.doi.org/10.1109/TEVC.2013.2285111.

[5] Keshav Ganapathy. *A Study of Genetic Algorithms for Hyperparameter Optimization of Neural Networks in Machine Translation*. 2020. arXiv: 2009.08928 [cs.NE]. URL: https://arxiv.org/abs/2009.08928.

[6] Hado van Hasselt, Arthur Guez, and David Silver. *Deep Reinforcement Learning with Double Q-learning*. 2015. arXiv: 1509.06461 [cs.LG].

[7] Jacek Krawczyk. “An Introduction to Dynamic Games”. In: (Jan. 1998).

[8] Michael Levet. “Game Theory : Normal Form Games”. In: 2016. URL: https://api.semanticscholar.org/CorpusID:131771375.

[9] Long-Ji Lin. “Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching”. In: *Mach. Learn.* 8.3–4 (May 1992), pp. 293–321. ISSN: 0885-6125. DOI: 10.1007/BF00992699. URL: https://doi.org/10.1007/BF00992699.

[10] A.M. Lyapunov. *The General Problem of the Stability of Motion*. Общая задача об устойчивости движения. Moscow: Gostekhizdat, 1950. URL: http://libarch.nmu.org.ua/handle/GenofondUA/63856.

[11] John F. Nash. “Equilibrium points in n-person games”. In: *Proceedings of the National Academy of Sciences* 36.1 (1950), pp. 48–49. DOI: 10.1073/pnas.36.1.48. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.36.1.48. URL: https://www.pnas.org/doi/abs/10.1073/pnas.36.1.48.

[12] Douglas E. Norton. “The fundamental theorem of dynamical systems”. eng. In: *Commentationes Mathematicae Universitatis Carolinae* 36.3 (1995), pp. 585–597. URL: http://eudml.org/doc/247765.

[13] Shayegan Omidshafiei et al. *α-Rank: Multi-Agent Evaluation by Evolution*. 2019. arXiv: 1903.01373 [cs.MA].

[14] Sheryl Paul and Jyotirmoy V. Deshmukh. *Multi Agent Path Finding using Evolutionary Game Theory*. 2022. arXiv: 2212.02010 [cs.MA]. URL: https://arxiv.org/abs/2212.02010.

[15] Lloyd S. Shapley. "Stochastic Games*". In: *Proceedings of the National Academy of Sciences* 39 (1953), pp. 1095–1100. URL: https://api.semanticscholar.org/CorpusID:263414073.

[16] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008.

[17] Ankur Sinha and Paritosh Pankaj. "A Linear Programming Enhanced Genetic Algorithm for Hyperparameter Tuning in Machine Learning". In: *2023 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, July 2023, pp. 1–8. DOI: 10.1109/cec53210.2023.10254162. URL: http://dx.doi.org/10.1109/CEC53210.2023.10254162.

[18] J. Maynard Smith and G. R. Price. "The Logic of Animal Conflict". In: *Nature* 246.5427 (Nov. 1973), pp. 15–18. ISSN: 1476-4687. DOI: 10.1038/246015a0. URL: https://doi.org/10.1038/246015a0.

[19] György Szabó and Gábor Fáth. "Evolutionary games on graphs". In: *Physics Reports* 446.4–6 (July 2007), pp. 97–216. ISSN: 0370-1573. DOI: 10.1016/j.physrep.2007.04.004. URL: http://dx.doi.org/10.1016/j.physrep.2007.04.004.

[20] George A. Vouros. "Explainable Deep Reinforcement Learning: State of the Art and Challenges". In: *ACM Computing Surveys* 55.5 (Dec. 2022), pp. 1–39. ISSN: 1557-7341. DOI: 10.1145/3527448. URL: http://dx.doi.org/10.1145/3527448.

[21] George Watkins, Giovanni Montana, and Juergen Branke. *Generating a Graph Colouring Heuristic with Deep Q-Learning and Graph Neural Networks*. 2023. arXiv: 2304.04051 [cs.LG].

[22] Michael P. Wellman, Karl Tuyls, and Amy Greenwald. *Empirical Game-Theoretic Analysis: A Survey*. 2024. arXiv: 2403.04018 [cs.GT]. URL: https://arxiv.org/abs/2403.04018.