Section 1 - Specification Table

Table 1: Model Overview

Category	**Details**
Base model	Tennis Game with mixed strategy Nash equilibrium: Players choose serve/defend directions, requiring strategic randomization to avoid predictability.
Extension assumptions	• **Game complexity**: Extends Tennis Game's simultaneous mixed strategies to Kuhn Poker's sequential betting (check/bet/fold/call) with imperfect information (hidden cards) • **Strategy space**: Expands from 2×2 payoff matrix to continuous probability distributions over 12 information sets • **Information structure**: Introduces hidden information (private cards) unlike Tennis Game's complete information • **Equilibrium multiplicity**: Multiple Nash equilibria exist in Kuhn Poker vs. unique mixed equilibrium in Tennis Game
Techniques showcased	• **Game theory**: Mixed strategy Nash equilibrium analysis (Tennis base) and support enumeration (Kuhn extension) • **Heuristics**: Evolutionary algorithm with tournament selection, crossover, and adaptive mutation • **Monte Carlo method**: Strategy evaluation through repeated game simulation and exploitability testing
Modelling question 1	Can evolutionary algorithms discover the continuum of mixed strategy equilibria in sequential games with imperfect information?
Modelling question 2	How closely do heuristically evolved strategies approximate game-theoretic Nash equilibria in asymmetric games?

Section 2 - Introduction

Strategic games often require players to randomize their actions to remain unpredictable. The classic Tennis Game exemplifies this: a server must vary between serving left and right to prevent the receiver from perfectly anticipating and countering every serve. I am fascinated by more complex strategic games where players can win through multiple viable strategy without relying on a single optimal strategy. In turn-based games like Poker, skilled players blend prediction, psychology, and calculated risk-taking, gaining advantages by reading opponents or strategically bluffing.

This project investigates how artificial evolution can discover and maintain strategic diversity in increasingly complex game environments. I begin with the Tennis Game as my base model—a simple 2×2 simultaneous game where both players must adopt mixed strategies in equilibrium. The server randomizes serve direction while the receiver randomizes defensive positioning, creating a unique mixed Nash equilibrium.

This project explores how artificially evolved strategies can discover competitive playstyles in such complex environments, specifically examining two core questions:

(1) How closely do heuristically evolved strategies approximate game-theoretic Nash equilibria?

Can evolutionary algorithms discover the continuum of mixed strategy equilibria that exist in complex sequential games?

I extend this framework to Kuhn Poker. This is a simplified Poker game where the deck consist of King, Queen and Jack, and two players are dealt two of the cards. This game allows us to explore:

- imperfect information: Private cards create uncertainty about opponent holdings
- Sequential decisions: Players act in turns rather than simultaneously
- Strategic depth: Bluffing, calling, and folding create rich strategy spaces
- Multiple equilibria: Unlike Tennis Game's unique mixed equilibrium, Kuhn Poker has infinitely many optimal mixed strategies

Using evolutionary algorithms, I'll evolve strategies through selection and mutation, then evaluate them against two benchmarks: (1) theoretical Nash equilibria via game theory, and (2) exploitability via Monte Carlo simulation. Markov chain analysis will reveal whether strategies converge to stable equilibria or diverge along the continuum of optimal play. Through this approach, I aim to demonstrate how heuristic methods can discover nuanced solutions in imperfect-information games.

Section 3 - Model Description

The foundational model examines strategic decision-making in professional tennis through a simplified two-player, zero-sum game. Players choose serve and return directions simultaneously, with payoffs derived from empirical tennis match data. The model captures the essential strategic tension between predictability and surprise in competitive sports. The tennis game represents an analytical, linear, discrete, deterministic,

static model. Solutions are obtained through closed-form analysis using support enumeration methods, with linear expected payoffs and fixed action spaces. The game is characterized by payoff matrices where the

server's matrix A and receiver's matrix B satisfy A + B = constant:

$$A = \begin{bmatrix} 58 & 79 \\ 73 & 49 \end{bmatrix}, \quad B = \begin{bmatrix} 42 & 21 \\ 27 & 51 \end{bmatrix}$$

Mixed strategies are represented as x = (θ, 1-θ) and y = (ϕ, 1-ϕ), where θ and ϕ denote the probability of choosing the first action. The Nash equilibrium conditions require mutual best responses, yielding the indifference equations:

- Server indifference: $58\phi + 79(1\text{-}\phi) = 73\phi + 49(1\text{-}\phi)$
- Receiver indifference: $42\theta + 27(1\text{-}\theta) = 21\theta + 51(1\text{-}\theta)$

Solution Algorithm

The support enumeration method identifies Nash equilibrium through systematic analysis:

- Support Identification: Determine which actions appear in equilibrium with positive probability
- Indifference Equation Solution: Solve the linear system for equilibrium probabilities
- Nash Verification: Confirm no profitable unilateral deviations exist

This yields the unique Nash equilibrium: $\theta^* = 8/15$, $\phi^* = 2/3$.

Model Extension: Kuhn Poker

The extension transforms the static, perfect information tennis model into a dynamic, imperfect information card game. This transition introduces sequential decision-making, private information, and stochastic elements while maintaining the strategic essence of mixed strategy equilibrium.

Kuhn Poker represents a numerical, non-linear, discrete, stochastic, dynamic model. The analytical complexity necessitates evolutionary simulation methods, with non-linear payoff dependencies and sequential action structures.

Extension Assumptions

- Imperfect Information: Players observe opponent actions but not private card holdings
- Sequential Action Structure: Player 1 acts first, followed by Player 2's response
- Stochastic Card Distribution: Cards (King, Queen, Jack) are dealt uniformly at random
- Asymmetric Information Advantage: Player 2 observes Player 1's initial action before deciding
- Bounded Rationality: Players learn through evolutionary processes rather than analytical optimization

0.1 Pure Strategy Representation

Pure strategies are encoded as ordered triples for systematic analysis. Player 1 strategies take the form (x_1, x_2, x_3) where $x_i \subset$ {0, 1, 2} represents the decision rules for card i. Player 2 strategies are represented as (y_1, y_2, y_3) where $y_i \subset$ {0, 1, 2, 3}.

Player 1 Encoding: Each x_i value is decoded through binary expansion, where the first bit indicates first-round action and the second bit indicates second-round response (0 = pass/fold, 1 = bet/call). For example, Strategy(2,0,1) = (10_2, 00_2, 01_2) instructs Player 1 to:

- Card 1 (Jack): Bet initially (first bit = 1), pass if opponent bets (second bit = 0)
- Card 2 (Queen): Always pass (00_2)
- Card 3 (King): Pass initially (first bit = 0), call if opponent bets (second bit = 1)

Player 2 Encoding: Each y_i value decodes to responses against different Player 1 actions, with the first bit governing action after Player 1 passes, and the second bit after Player 1 bets. For example, Strategy (2, 1, 3) = (10_2, 01_2, 11_2) instructs Player 2 to:

- Card 1 (Jack): Bet when confronted by a pass(first bit =1), and fold when confronted by a bet (second bit = 0)
- Card 2 (Queen): Pass when confronted by a pass(first bit =0), and call when confronted by a bet (second bit = 1)
- Card 3 (King): Always bet when confronted by a pass or bet (first and second bit = 1)

0.2 Payoff Matrix Construction

This encoding generates 3^3 = 27 pure strategies for Player 1 and 4^3 = 64 pure strategies for Player 2, yielding a 27×64 payoff matrix. Each matrix entry represents the expected payoff calculated across all possible card distributions

The game tree contains five terminal outcomes with path-dependent payoffs:

Table 2: Decision Paths and Corresponding Payoffs in Kuhn Poker

Decision Path	**Payoff (Player 1, Player 2)**	**Outcome Condition**
Check $\rightarrow$ Check	± 1	Based on card comparison
Check $\rightarrow$ Bet $\rightarrow$ Fold	(-1, 1)	Player 1 folds after bet
Check $\rightarrow$ Bet $\rightarrow$ Call	± 2	Based on card comparison
Bet $\rightarrow$ Fold	(1, -1)	Player 1 folds after bet
Bet $\rightarrow$ Call	± 2	Based on card comparison

The expected payoff is calculated as:

$$E[\text{Payoff}] = \frac{1}{6} \sum_{\text{card combinations}} \sum_{\text{decision paths}} \text{outcome} \times P(\text{path})$$

Matrix Reduction Through Dominated Strategy Elimination

The initial 27×64 matrix contains 1,728 strategy combinations. An experienced Poker player can easily see that some strategies are optimal. For example:

- Player 1 Never folds with King when facing a call/bet from Player 2
- Player 2 Never bets with Jack, and Always fold with Jack when confronted with a bet.

However it is much easier to Iteratively eliminate dominated Strategies. A Player 1 Strategy A_i dominates A_j if payoff(A_i, B) $\geq$ payoff(A_j, B) for all Player 2 Strategies B, with strict inequality for at least one B. Similarly, a Player 2 strategy B_i dominates B_j if payoff(A, B_i) $\leq$ payoff(A, B_j) for all Player 1 strategies A, again with strict inequality for at least one A, since Player 2's objective is to minimize Player 1's payoff. This will reduce the 1728-entry matrix into a manageable form suitable for equilbrium analysis, which I will show in the next section.

Nash Equilibrium Calculation

I will be using nashpy to calculate the nash equilibrium as the regular method shown in the applied will not be feasible. The result will show that Player 1 will have many different strategies whereas Player 2 will have one optimal Strategy.

0.3 Evolutionary Algorithm Framework

I chose an evolutionary algorithm because it naturally models strategic adaptation in competitive settings like Kuhn Poker, where optimal strategies emerge through mutual competition rather than static optimization. Key components of my implementation are:

- **Fitness Calculation:** Each strategy is evaluated against a random sample of 20 opponents (500 games per matchup), reducing computational complexity from O(n^2) to O(n) . Fitness reflects average expected payoff, enabling coevolutionary dynamics where both populations improve by competing directly.
- **Mutation:** Mutation operates in logit space using Gaussian noise (σ=0.1), ensuring valid probabilities after perturbation. A 10% mutation rate balances exploration of new strategies with preservation of successful ones. This approach enables smooth exploration of the strategy space while respecting probability constraints.
- **Crossover:** Uniform crossover combines parent strategies at the decision-point level, preserving independent betting and calling tendencies. This allows offspring to inherit the strongest traits from multiple parents for improved performance.
- **Selection:** Tournament selection with tournament size 5 provides a good balance between selection pressure and diversity maintenance. Additionally, elitism preserves the best individual from each generation, ensuring that good solutions are never lost while allowing the rest of the population to explore new areas.

Section 4 - Results

0.4 Payoff Matrix Analysis

0.4.1 Construction

I generate the pure strategies for Player 1 (X) and Player 2 (Y) using simple loop structures. Each strategy is encoded as a tuple representing actions for different card holdings. For example, Strategy (2, 1, 3) = (10_2, 01_2, 11_2).

Using these strategies, I construct the full 27×64 game matrix, where each cell corresponds to the expected payoff when a specific Player 1 strategy faces a specific Player 2 strategy. The payoff in each cell is computed by simulating all possible card combinations and decision paths according to the game rules, then totaling the outcomes. Here is an example calculation:

- Strategy A = Strategy(2,0,1) = (10_2, 00_2, 01_2)
- Strategy B = Strategy (2, 1, 3) = (10_2, 01_2, 11_2)

We simlulate the game mechanics for each possible card combination:

- **Case 1: A holds Jack** A always bets on the first turn. B calls with both Queen and King. In both cases, A loses and pays 2 points.
- **Case 2: A holds Queen** A checks or folds in response. B bets with Jack and King, causing A to fold in both cases. Each fold costs A 2 points.
- **Case 3: A holds King** A checks and then bets in response. If B bets with Jack, A raises, and B calls $\rightarrow$ A gains 2 points. If B folds with Queen $\rightarrow$ A gains 1 point.

Summing up A's total earnings:

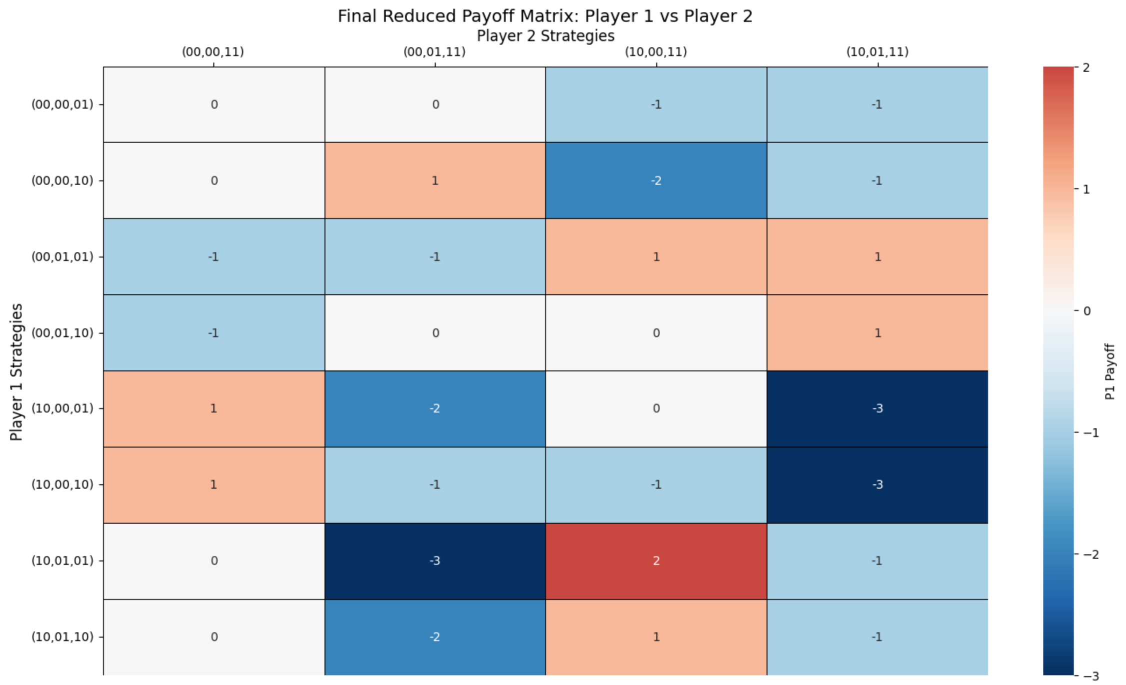
$$Total = (+2 + 1) - 2 - 2 = -1$$

Thus, the expected payoff for this pair of strategies is -1, indicating a slight disadvantage for Player 1 in this matchup.

By repeating this process across all 27×64 strategy pairs, I populate the complete payoff matrix for further analysis.

0.4.2 Iteratively eliminating Dominated Strategies.

I now iterate through the matrix to eliminate the dominated strategies and obtained a 8x4 reduced matrix. This indicates Player 1 has more viable strategies then player 2.



	(00,00,11)	(00,01,11)	(10,00,11)	(10,01,11)
(00,00,01)	0	0	-1	-1
(00,00,10)	0	1	-2	-1
(00,01,01)	-1	-1	1	1
(00,01,10)	-1	0	0	1
(10,00,01)	1	-2	0	-3
(10,00,10)	1	-1	-1	-3
(10,01,01)	0	-3	2	-1
(10,01,10)	0	-2	1	-1

Figure 1: Reduced payoff matrix

After eliminating strictly dominated strategies, we can now analyze the optimal behavior for both players across different card holdings.

Player 1

- (Card 1: Jack): Remaining strategies are 00 (check/fold) and 10 (bet/fold). Player 1 always folds if raised, but may choose to bet or check initially — suggesting bluffing with Jack is part of an optimal mixed strategy.
- (Card 2: Queen) Remaining strategies are 00 (check/fold) and 01 (check/call). Player 1 always checks first and may call or fold in response to a bet, reflecting the need for unpredictability with a medium-strength hand.
- (Card 3: King) Remaining strategies are 01 (check/call) and 10 (bet/call). Player 1 can either bet outright or check and call if raised — allowing them to either extract value or induce bluffs.

Player 2

- (Card 1: Jack): Remaining strategies are 00 (check/check) and 10 (bet then fold). If Player 1 checks, Player 2 may choose to bet or check . However, if Player 1 bets, Player 2 always folds , as calling would result in a guaranteed loss.
- (Card 2: Queen) Remaining strategies are 00 (check/check) and 01 (check/call). If Player 1 checks, Player 2 also checks . If Player 1 bets, Player 2 may call or fold , depending on the equilibrium. This cautious behavior reflects the marginal strength of the Queen.
- (Card 3: King) Only one strategy remains: 11 (bet then call). Regardless of Player 1's action, Player 2 bets if checked to and calls if raised . This is consistent with strong, aggressive play — maximizing expected value with the strongest possible hand.

For an experienced Poker player, the result makes a lot of sense. For example, Player 2 should always bet in both cases with King, as Player 2 will always win. Player 1 should always fold with Jack when confronted with a bet, as player 1 will lose no matter and should minimize his losses.

An interesting discovery is that Player 1 can choose to bet with a Jack, and check with a King. This suggests that Player 1 can bluff by betting with Jack, and see Player 2's response when checking with a King. Checking with a King allows the chance for player 2 to bet, which can increase Player 1's earnings by calling in response to Player 2.

While both our poker-inspired game and the classic tennis game rely on mixed strategy equilibria to prevent exploitation, they differ significantly in structure and strategic depth. The tennis game is a simple simultaneous-move setting where players randomize actions to remain unpredictable. In contrast, our game involves sequential decision-making under incomplete information , where players must balance bluffing, value betting, and calling across different private holdings. This leads to more complex strategies that involve not just momentary randomness, but consistent behavior across multiple scenarios, reflecting richer, more nuanced gameplay akin to real-world strategic reasoning in games like poker.

I can calculate the expected payoff of Player 1 and Player 2. For player 1:

$E[\text{payoff}] = \frac{1}{32}(-1 + (-1) + 1 + (-2) + (-1) + (-1) + (-1) + 1 + 1 + (-1) + 1 + 1 + (-2) + (-3) + 1 + (-1) + (-1) + (-3) + (-3) + 2 + (-1) + (-2) + 1 + (-1))$
$= -0.5$

This indicates that player 1 loses 0.5 on average. Conversely, because this is a zero-sum game, player 2 wins 0.5 on average.

0.4.3 Nash Equilibrium Calculation

I used nashpy and found several Nash equilibria in the game. Below are five representative mixed-strategy equilibria:

My results suggest that Player 2 Has one optimal strategy, whereas Player 1 has many. This can be explained by the fact that Player 1 has zero information about the opponent, and thus they can choose to bluff. In equilibrium 1, Player 1 has strategies 2, 4 and 6 which correspond to (00, 00, 10), (00, 01, 10), and (10, 01, 01). This results in:

Table 3: Selected Nash Equilibria

Equilibrium	Player I Strategy	Player II Strategy
1	$[0.000, 0.417, 0.000, 0.333, 0.000, 0.000, 0.250, 0.000]$	$[0.333, 0.333, 0.333, 0.000]$
2	$[0.556, 0.000, 0.000, 0.333, 0.000, 0.000, 0.111, 0.000]$	$[0.333, 0.333, 0.333, 0.000]$
3	$[0.000, 0.400, 0.467, 0.000, 0.133, 0.000, 0.000, 0.000]$	$[0.333, 0.333, 0.333, 0.000]$
4	$[0.000, 0.444, 0.333, 0.000, 0.000, 0.000, 0.000, 0.222]$	$[0.333, 0.333, 0.333, 0.000]$
5	$[0.000, 0.167, 0.000, 0.583, 0.250, 0.000, 0.000, 0.000]$	$[0.333, 0.333, 0.333, 0.000]$

- $\frac{5}{12}$ chance to check and fold with Jack; check and fold with Queen; bet with King.
- $\frac{1}{3}$ chance to check and fold with Jack; check and bet with Queen; bet with King.
- $\frac{1}{4}$ chance to bet with Jack; check and bet with Queen; check and bet with King.

This simplifies to:

- $\frac{3}{4}$ to bet with King; $\frac{1}{4}$ to check and bet with King
- $\frac{1}{4}$ to bet with Jack; $\frac{3}{4}$ to check and fold with King
- $\frac{7}{12}$ to check and bet with Queen; $\frac{5}{12}$ to check and fold with Queen.

The observed relationship between the probabilities reveals a strategic balance in Player 1's behavior: betting with the King occurs three times as often as bluffing with the Jack, while check-betting with Queen happens at a rate of $\frac{7}{12}$, just slightly favoring aggression. This suggests that Player 1 uses strong value betting with the King to support occasional bluffs with the Jack, maintaining a balanced strategy that prevents Player 2 from exploiting predictable patterns. The structure of these mixed strategies highlights how incomplete information leads to calculated randomization. Bluffing is not arbitrary, but carefully proportioned to mirror the strength distribution of one's actual hand.

0.5 Evolutionary Algorithm to Find Optimal Strategy

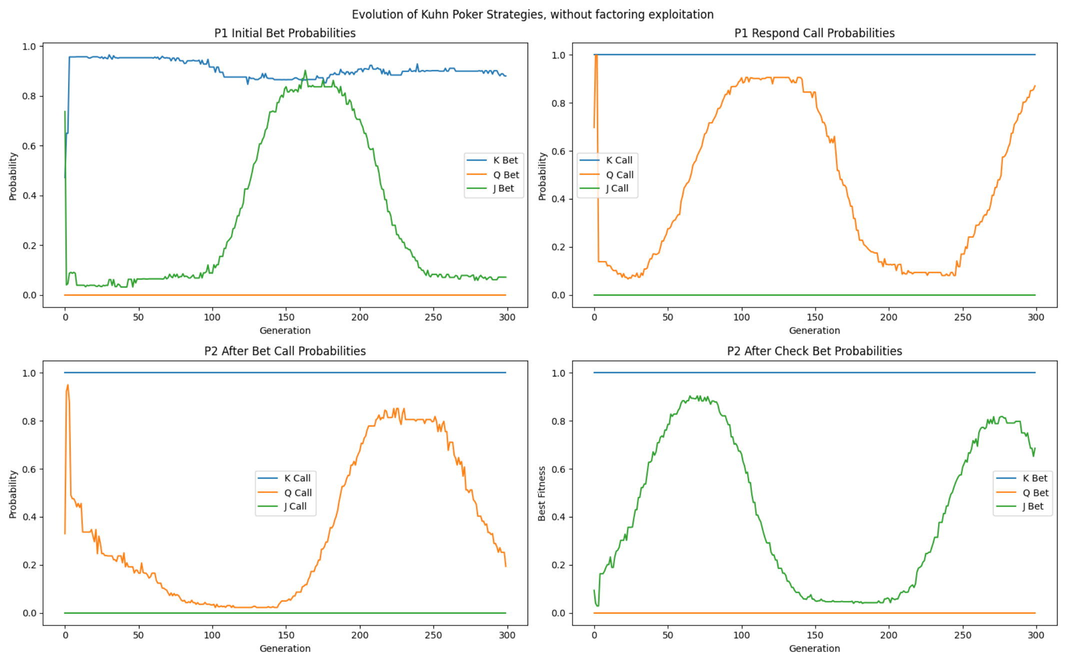


Figure 2: Evolution of Strategies without factoring Exploitation

After identifying strictly dominant strategies, I fixed certain decision probabilities based on optimal behavior — for example, Player 2 always bets with King, so its probability of doing so is set to 1. For non-dominant decisions — such as whether Player 1 should bet or check with Jack — I allowed the strategy probabilities to be randomized.

I then ran an evolutionary algorithm with mutation and crossover rates set at 0.08, enabling exploration of the mixed strategy space. The results revealed a phenomenon known as the Red Queen effect : as Player 1 adapts to exploit Player 2's tendencies, Player 2 evolves counter-strategies, leading to a continuous cycle of adaptation without settling into a stable equilibrium.

Specifically, in the early generations (up to generation 150), Player 2 rarely calls with Queen when faced with a bet. This allows Player 1 to exploit the gap by bluffing more frequently with Jack. However, from generation 150 onward, Player 2 begins to call with Queen more often, effectively countering Player 1's increased bluffing frequency.

A similar dynamic occurs in response to checking: up to generation 150, Player 2 frequently bets with Jack after Player 1 checks, exploiting the fact that Player 1 rarely calls with Queen. After generation 150, Player 1 begins calling more often with Queen, neutralizing Player 2's betting strategy.

Importantly, most of the dominant strategies remain unchanged throughout evolution , reflecting their stability:

- Player 2 never calls with Jack — doing so always results in a loss.
- Player 1 never bets with Queen — Player 2 will always respond aggressively with King.
- Player 1 never calls with Jack — it always results in a loss.
- Player 2 never bets with Queen — Player 1 will either call with King or fold with Jack, both of which minimize Player 2's expected gain.

These findings illustrate how evolutionary dynamics can approximate game-theoretic equilibria , but also highlight the challenge of convergence in games with mixed equilibria and adaptive opponents.

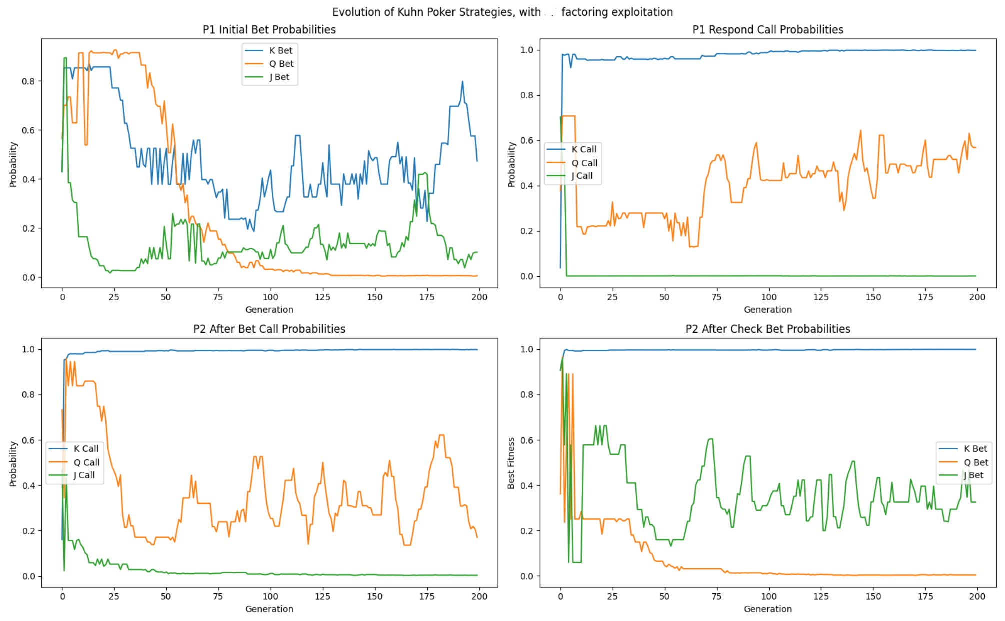


Figure 3: Evolution of Strategies with Exploitation

To better assess strategic robustness, I revised the fitness function to measure exploitability — how much an opponent can gain by optimally countering a strategy. This is calculated by simulating matches against a diverse population, identifying the best-response adversary, and comparing performance to the population baseline. Lower exploitability indicates stronger, more balanced play that performs consistently across varied strategies, rather than specializing against familiar ones.

The graph shows reduced oscillation and now stabilizes within a consistent range, indicating convergence toward an optimal strategy profile. I calculated the mean and median values to confirm alignment with the Nash equilibrium:

Table 4: Player Strategy Probabilities (Median vs. Mean)

Action	**Median Probability**	**Mean Probability**
P1_initial_K_bet	0.4423	0.4813
P1_initial_Q_bet	0.0316	0.2589
P1_initial_J_bet	0.1158	0.1390
P1_respond_K_call	0.9891	0.9764
P1_respond_Q_call	0.4348	0.4076
P1_respond_J_call	0.0011	0.0095
P2_after_check_K_bet	0.9961	0.9959
P2_after_check_Q_bet	0.0108	0.0725
P2_after_check_J_bet	0.3255	0.3574
P2_after_bet_K_call	0.9942	0.9889
P2_after_bet_Q_call	0.3097	0.3684
P2_after_bet_J_call	0.0085	0.0241

Player 2 now bets with Jack approximately one-third of the time and calls with Queen against a bet also about one-third of the time — consistent with the theoretically derived equilibrium..

Similarly, Player 1 bets with King roughly three times as often as with Jack, and calls with Queen at a frequency close to the probability of betting with Jack plus $\frac{1}{3}$. These relationships reflect the strategic balance predicted by the Nash equilibrium.

On repeated runs, the probabilities of Player 1 would defer by quite a lot, but is still consistent with the relationship, showcasing the various optimal strategies player 1 could employ. Similarly, Player 2's strategy oscillate in the same region, showing that theres only one optimal strategy. (not enough space to show a second graph)

0.6 Monte Carlo Methods

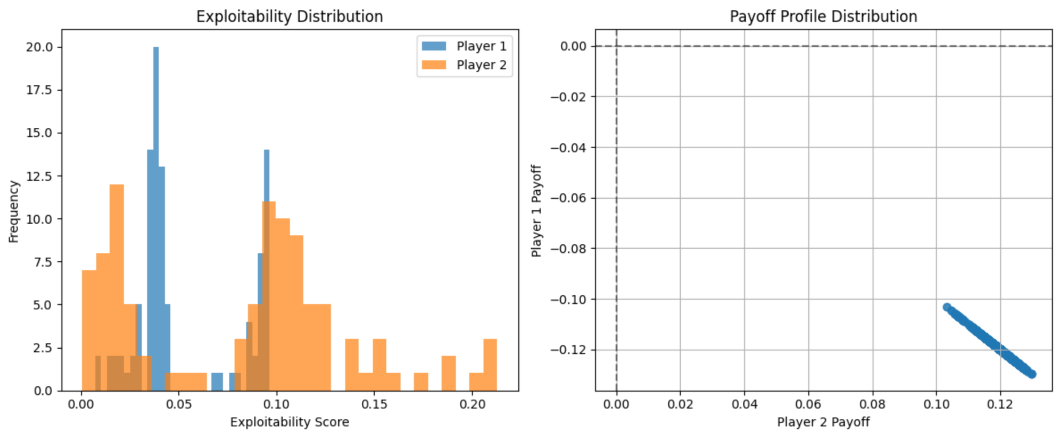


Figure 4: Exploitation and Payoff Distribution using Monte Carlo Simulations

0.6.1 Exploitability Estimation via Sampling

To evaluate how far evolved strategies deviate from Nash equilibrium, we employ Monte Carlo (MC) sampling to estimate exploitability—the worst-case performance gap against an optimal counter-strategy.

Methodology:

For a given strategy, we sample 1,000 opponent strategies from:

- The current evolutionary population (to test robustness against coevolving strategies).
- A Hall of Fame (historical strong strategies, to avoid cyclical overfitting).
- Each opponent plays 500 games against the test strategy, and we record:

$$Exploitability = max(Opponent'sAveragePayoff)$$

Player 2's exploitability has a wider range of values, demonstrating how player 1 has many different strategies to poke holes in player 2 strategy.

Player 1's has a shorter range, which demonstrates how Player 2 does not deviate from its strategy as much. I

0.6.2 Discovering Mixed-Strategy Continuums

Nash equilibria in Kuhn Poker often involve continuums of mixed strategies. To identify these, we use MC sampling to:

- Sample 10,000 combinations of Player 1 and Player 2 strategies from the evolved population.
- For each pair (s_1, s_2), calculate the exact expected payoffs (u_1, u_2)
- Use scatter plots to visualize (u_1, u_2) payoffs.

Interpretation

- A continuum appears as a linear cluster of points
- The line y=-x indicates zero-sum consistency , where one player's gain equals the other's loss
- Deviations from this line suggest exploitable imbalances or off-equilibrium behavior

Section 5- List of Algorithms and Concept

- Iterative removal of Dominated Strategies: A method for simplifying games by eliminating strategies that are strictly worse than others, regardless of the opponent's behavior. This was used to reduce the original 27×64 Kuhn Poker payoff matrix into a more manageable 8×4 size, preserving only strategically relevant options.
- Support Enumeration (nashpy): An algorithmic technique used to compute Nash equilibria in two-player normal-form games. We applied this using the nashpy library to find mixed-strategy equilibria in the reduced payoff matrix. It systematically tests combinations of strategy supports to identify stable equilibrium pairs.
- Evolutionary Algorithm: A population-based optimization method inspired by natural selection. In our context, it was used to evolve player strategies over successive generations through mechanisms such as mutation, crossover, and tournament selection. This allowed us to explore complex mixed-strategy spaces and approximate Nash-like behaviors in a dynamic learning environment.
- Exploitation: A metric used to assess how close an evolved strategy is to a true Nash equilibrium. It quantifies the maximum advantage an optimal adversary can gain when targeting a given strategy. Lower exploitability indicates greater strategic robustness and balance.
- Monte Carlo Simulation: A computational technique that uses random sampling to estimate outcomes in probabilistic settings. We applied it to evaluate expected payoffs and test strategy performance across large numbers of simulated games. It also helped visualize payoff continuums and analyze exploitability across evolved strategies

Usage of Generative AI

I used AI to clean alot of my sentences and explanation. I also used it to debug my code and help me autocomplete some sections of it.