JiangJun: Mastering Xiangqi by Tackling Non-Transitivity in Two-Player Zero-Sum Games

Anonymous authors
Paper under double-blind review

Abstract

In this paper we present an empirical study of non-transitivity in perfect-information games by studying Xiangqi, a traditional board game in China with similar game-tree complexity to chess and shogi. After analyzing over 10,000 human Xiangqi playing records, we demonstrate that the game's strategic structure contains both transitive and non-transitive components. To address non-transitivity, we propose the JiangJun algorithm, which combines Monte-Carlo Tree Search (MCTS) with Policy Space Response Oracles (PSRO) to find an approximate Nash equilibrium. We evaluate the algorithm empirically using a WeChat mini program and achieve a Master level with a 99.39% win rate against human players. The algorithm's effectiveness in overcoming non-transitivity is confirmed by relative population performance and visualization results.

1 Introduction

Multi-Agent Reinforcement Learning (MARL) has demonstrated remarkable success in various games such as Hide and Seek (Baker et al., 2019), Go (Silver et al., 2016), StarCraft II (Vinyals et al., 2019b), Dota 2 (Berner et al., 2019), and Stratego Perolat et al. (2022). However, algorithms such as AlphaZero (Silver et al., 2018) and AlphaGo (Silver et al., 2016) that train against the most recent opponent can potentially cycle in games that have non-transitive structure. While this problem has been well-studied in imperfect-information games (Lanctot et al., 2017; Perolat et al., 2022; McAleer et al., 2022a;c; 2021; Fu et al., 2022; Brown et al., 2019; Heinrich & Silver, 2016; Steinberger et al., 2020; Perolat et al., 2021; Hennes et al., 2019), it has been less studied in perfect-information games.

Conquering non-transitivity in perfect-information games remains an open research direction. Recent works (Balduzzi et al., 2019b; McAleer et al., 2020; Liu et al., 2021; McAleer et al., 2022c;b) have focused on finding (approximate) Nash equilibria using the Policy Space Response Oracles (PSRO) algorithm (Lanctot et al., 2017), which is developed from the Double Oracle (DO) algorithm (McMahan et al., 2003). These approaches deal with non-transitivity by mixing over a population of policies, but have to our knowledge not been studied in perfect-information games.

This study aims to investigate Xiangqi, a prevalent two-player zero-sum game with moderate game-tree complexity of 10^{150} (between Chess (10^{128} game-tree complexity) and Go (10^{360} game-tree complexity)), which has not received much attention in previous research. Although Xiangqi's complexity presents a challenge, it is not an insurmountable obstacle, and its accessibility makes it an excellent candidate for exploring the geometrical landscape of board games and non-transitivity.

This paper explores the geometry of Xiangqi, using more than 10,000 game records from human players as the basis of our analysis. Our research uncovers a spinning top structure within the game and reveals non-transitivity in the middle region of the ELO rating score. In addition, we identify strategic cycles in policy checkpoints by employing self-play training, which provides empirical evidence for the existence of non-transitivity in Xiangqi. Building upon these findings, we introduce the JiangJun algorithm to address the non-transitive problem in Xiangqi. The JiangJun algorithm consists of two modules: the MCTS actor and populationer, which work together to approximate Nash equilibria on the population using MCTS. We train JiangJun to the Master level using 90 V100 GPUs on the Huawei Cloud ModelArt platform. To evaluate the

Opponent Selection Strategy	Algorithms
Latest	AlphaZero(Silver et al., 2018),Suphx (Li et al., 2020), ACH (Fu et al., 2022), DouZero (Zha et al., 2021), PerfectDou (Guan et al., 2022)
History Random	AlphaGo (Silver et al., 2016)
History Best	AlphaGo Zero(Silver et al., 2017)
History Top K	AlphaHoldem (Zhao et al., 2022)
Population-Based	AlphaStar (Vinyals et al., 2019b)
Mixed	OpenAI Five (Berner et al., 2019), Jue Wu (Wu et al., 2018)

Table 1: The presented summary table provides an overview of the opponent selection strategies used by 11 leading algorithms, which are classified into six distinct categories: the latest strategy, random/best strategy in history, top-k strategies in history, population-based strategy, and a mixed sampling method that combines the latest and historical random strategies.

effectiveness of JiangJun, we develop a Xiangqi program on WeChat and collect over 7,000 game records between JiangJun and human players over six months. Moreover, JiangJun is capable of tackling the complex endgames of Xiangqi. Our results show that JiangJun achieves an impressive 99.35% win rate against human players and effectively overcomes non-transitivity, as evidenced by the Nash distribution and low-dimension gamescape visualization of the first 2-dims embedding.

In summary, this paper presents three key contributions: 1) an analysis of the geometrical landscape of Xiangqi using over 10,000 real game records, uncovering a spinning top structure and non-transitivity in real-world games; 2) the proposal of the JiangJun algorithm to conquer non-transitivity and master Xiangqi; and 3) the empirical evaluation of JiangJun with human players using a WeChat mini program, achieving over a 99.35% win rate and demonstrating the efficient overcoming of non-transitivity as shown by other metrics such as relative population performance.

The remaining sections of this paper are organized as follows. Section 2 summarizes related work in the field. In Section 3, we give a brief introduction to the Xiangqi game. Section 4 presents our non-transitive analysis method and the results of real-world Xiangqi data analysis. Next, in Section 5, we introduce our proposed JiangJun algorithm, while Section 6 presents the experimental results. Finally, we draw our conclusions in Section 7.

2 Related Work

MARL has achieved a huge breakthrough in the field of games such as StarCraft, DOTA, Majiang, and Doudizhu. To solve those complex games, many methods are based on self-play or training against previous strategies. As shown in Table 1, we provide an overview of opponent selection strategies used by 11 state-of-the-art algorithms, which we've grouped into 5 categories. The categories are 1) Latest: selects the most recent opponent strategy, such as AlphaZero (Silver et al., 2018), Suphx (Li et al., 2020), ACH (Fu et al., 2022), DouZero (Zha et al., 2021) and Perfect Dou (Guan et al., 2022), 2) History Random: randomly samples a historical opponent's strategy like AlphaGo (Silver et al., 2016), 3) History Best: selects the opponent strategy with the best performance like AlphaGo Zero (Silver et al., 2017), 4) History Top K: selects the top-k historical opponent strategies like AlphaHoldem (Zhao et al., 2022), 5)Population-Based: select the opponent strategy from a population of strategies such as AlphaStar (Vinyals et al., 2019b), and 6) Mixed category: combine the Latest and History Random strategies, with opponents selected according to a weighted combination of the two strategies (80% Latest and 20% History Random).

Recent research has examined the non-transitivity and corresponding reinforcement learning solutions. The spinning top hypothesis proposed by Czarnecki et al. (2020) describes the Game of Skill geometry, where the strategy space of real-world games consists of transitivity and non-transitivity, resembling a spinning top structure. The transitive strength gradually diminishes as game skill increases upward to the Nash Equilibrium or evolves downward to the worst possible strategies.

To address the non-transitivity in real-world games, especially zero-sum games, Double Oracle (DO) (McMahan et al., 2003)-based reinforcement learning methods have been proposed. These algorithms attempt to find the best response against a previous aggregated policy at pre-iteration. The DO algorithm is designed to derive Nash Equilibrium in normal-form games, which is guaranteed by finding a true best-response oracle(Yang et al., 2021).

For large-scale games, policy-space response oracles (PSRO) (Lanctot et al., 2017) naturally developed from the DO algorithm, finding an approximate best response instead of a true best response. In PSRO with two players, each player maintains a population $\mathcal{B}_{1,2} = \{\pi^1_{1,2}, \ldots, \pi^n_{1,2}\}$. When player $i \in \{1, 2\}$ tries to add a new policy π_i^{n+1}, the best response to the mixture of opponents will be obtained by following the equation.

$$\mathrm{Br}(\mathcal{B}_{-i}) = \max_{\pi_i^{n+1}} \sum_j \sigma^j_{-i} E_{\pi_i^{n+1}, \pi^j_{-i}}[r_i(s, \mathbf{a})], \tag{1}$$

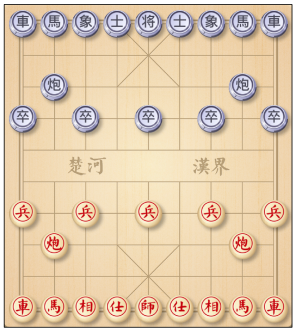
where (σ_i, σ_{-i}) is the Nash equilibrium distribution over policies in $\mathcal{B}_i$ and $\mathcal{B}_{-i}$. Variants of PSRO have been proposed, including PSRO with a Nash rectifier ($PSRO_{rN}$)(Balduzzi et al., 2019b) to explore strategies with positive Nash support to improve strategy strength, AlphaStar (Vinyals et al., 2019a) for computing the best response against a mixture of opponents, and Pipeline PSRO (McAleer et al., 2020) for parallelizing PSRO by maintaining a hierarchical pipeline of reinforcement learning workers. Behavioral diversity is believed to be linked to non-transitivity, observed in both human societies and biological systems (Kerr et al., 2002; Reichenbach et al., 2007). As a result, some researchers have studied behavioral diversity to solve non-transitivity problems (Yang et al., 2021; Liu et al., 2021). However, the quantification and measurement of behavioral diversity are still not well understood (Yang et al., 2021). Some works define behavioral diversity as the variance in rewards (Lehman & Stanley, 2008; 2011), the convex hull of a gamescape (Czarnecki et al., 2020), and the discrepancies of occupancy measures (Liu et al., 2021).

While PPAD-Hard for computing Nash equilibrium (Balduzzi et al., 2019a), $\alpha-$PSRO (Muller et al., 2019) proposes a polynomial-time solution $\alpha-$Rank for general-sum games instead of Nash equilibrium. Other works focus on maximizing diversity in deriving the best response, including Diverse-PSRO (Liu et al., 2021), which offers a unified view for diversity in the PSRO framework by combining both the behavioural diversity and the response diversity of a strategy, and DPP-PSRO (Nieves et al., 2021), which combines a novel diversity metric, determinantal point processes, and best-response dynamics for solving normal-form and open-ended games. PSRO has also recently been applied to robust RL, where it can find policies that are robust to all feasible environments (Lanier et al., 2022).

3 Xiangqi: the game

Xiangqi, also known as Chinese chess, is an ancient and widely popular board game played worldwide. Its history dates back around 3,500 years to a game called Liubo. Over the centuries, the game evolved into the Xiangqi, played today, a two-player board game on a 9x10 board. The game's goal is for each player, red and black, to capture the other's king. The red player always plays first, and the two players alternate moves. Each player controls seven types of pieces for a total of sixteen pieces: one king (帅/将, K/k), two advisors (仕/士, A/a), two bishops (相/象, B/b), two rooks (車/車, R/r), two knights (馬/馬, N/n), two cannons (炮/炮, C/c), and five pawns (卒/兵, P/p). The first Chinese character or uppercase, represents pieces for the red player, and the second Chinese character or lowercase, represents pieces for the black player.

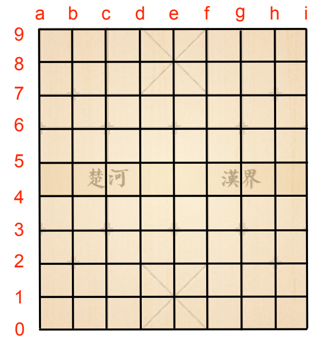
The game of Xiangqi is introduced in Figure 1, which provides the initial board setup and the corresponding English name and symbol. The game board is divided into two territories, red and black, by a horizontal line known as the "River". The Palace, marked with an "X" on each side, is located in the center bottom board and restricted to Kings and Advisors' movement only. Kings can move one step at a time in the horizontal or vertical direction, while Advisors are restricted to one-step diagonal movements within the Palace. If a King faces the other King, it will be captured, resulting in an automatic loss. The Bishops on either side of the Palace can only move diagonally within a 2×2 square on their respective side. Similarly, the Horses can



(a)

Symbol	Piece	Chinese
P/p	Pawns	卒/兵
C/c	Cannons	炮/炮
R/r	Rooks	車/車
K/k	Knights	馬/馬
B/b	Bishops	相/象
A/a	Advisors	仕/士
S/s	Kings	帅/将

(b)



(c)

Figure 1: Introduction of Xiangqi. Left figure (a) shows the initial position of Xiangqi. To better describe Xiangqi, we use corresponding English characters (as shown in sub-figure (b)) and 9×10 (width×height) grid matrix (as shown in sub-figure (c)) to present pieces and Xiangqi board, respectively.

move diagonally within a 1×2 rectangle. However, the movement of Bishops and Horses is blocked by other pieces. Bishops cannot move if a piece is positioned in the center of the 2×2 square, and a piece located along the long edge of a Horse prevents it from moving along the corresponding 1×2 rectangle. The Rooks and Cannons move freely in vertical or horizontal directions, with Cannons being capable of capturing other pieces by jumping over a single piece. Rooks, on the other hand, can only capture the first piece in their path. The game also features five Pawns on each side, which can move one space forward within their territory. Once they cross the River, they can also move one space forward, left, or right.

To provide a standardized way to describe the positions of the pieces in Xiangqi, we encode the Xiangqi game into a 9×10 matrix, as shown in Fig. 1 (c). The board matrix has been numbered using nine letters from a to i for the length and ten numbers from 0 to 9 for the width. This allows us to represent the current state and actions using a string of characters based on the Universal Chinese chess Protocol (UCCI). For example, the initial position shown in Fig. 1 (b) can be represented by the string "rnbakabnr/9/1c5c1/p1p1p1p1p/9/9/P1P1P1P1P/1C5C1/9/RNBAKABNR", where the numbers represent the number of empty spaces. The action of each piece can also be described using a tuple consisting of four characters, such as b0c2, which means that the piece moves from column b row 0 to column c row 2.

4 Real-World Xiangqi Data Analysis

In this section, we will present the analysis method and results of real-world Xiangqi data obtained from the Play OK game platform[1], a global gaming platform offering a variety of games to play with online players from around the world. Our analysis reveals that the geometric structure of the real-world Xiangqi game resembles that of a spinning top, in accordance with the Game of Skill hypothesis (Czarnecki et al., 2020). This finding suggests that non-transitivity exists in real-world Xiangqi gameplay.

4.1 Preliminaries for Measuring Non-Transitivity

To begin with, assuming that the probability p of one player winning over another can be estimated or computed, the Xiangqi game can be represented by a tuple $(n, \mathcal{W}, \mathcal{M})$. Here, $\mathcal{W} = \{w_1, w_2, \cdots, w_n\}$ refers to the n players or agents, who can be either human players or neural network-based agents. Furthermore, the payoff matrix $\mathcal{M} \in \mathbb{R}^{n \times n}$ is defined, where each element in $\mathcal{M}$ is calculated by a payoff function $\phi : w_i \times w_j \to \mathbb{R}$. The function ϕ is an antisymmetric function $\phi(w_i, w_j) = -\phi(w_j, w_i)$ for $i, j < n$ and $i \neq j$, and a higher value of $\phi(w_i, w_j)$ represents a better outcome for player w_i. The payoff function can be transformed into a win/loss probability via $\phi(w_i, w_j) := p(w_i, w_j) - 1/2$. Wins, losses, and ties for w_i are represented by $\phi(w_i, w_j) > 0, \phi(w_i, w_j) < 0, \phi(w_i, w_j) = 0$, respectively. Therefore, the Xiangqi game

[1]www.playok.com

$(n, \mathcal{W}, \mathcal{M})$ is a symmetric zero-sum functional-form game (FFG) Balduzzi et al. (2019b), where players seek to maximize their own payoff while minimizing the payoff of their opponents.

Nash equilibrium. The Nash equilibrium concept is widely used as a solution concept in symmetric zero-sum games (Nash, 1951). It describes a state where no player is incentivized to change their strategy from the equilibrium strategy unilaterally. However, multiple Nash equilibria can exist in a game. To guarantee the uniqueness of the Nash equilibrium, we can solve for it using the maximum entropy method (Ortiz et al., 2007). To obtain the unique maximum entropy Nash equilibrium, we can solve a Linear Programming problem as follows.

$$\begin{aligned} \mathfrak{p}^\star &= \arg\max_{\mathfrak{p}} \sum_{j \in k} -\mathfrak{p}_j \log \mathfrak{p}_j, \\ s.t. \quad & \mathcal{M}\mathfrak{p} \leq \mathbf{0}_k, \\ & \mathfrak{p} \geq \mathbf{0}_k, \\ & \mathbf{1}_k^T \mathfrak{p} = 1. \end{aligned} \tag{2}$$

The symbols $\mathbf{0}_k$ and $\mathbf{1}_k$ represent vectors with k entries, where all entries in $\mathbf{0}_k$ are equal to 0 and all entries in $\mathbf{1}_k$ are equal to 1.

Nash Clustering. The concept of Nash clustering is an important tool for measuring non-transitivity. This method is based on the layered game geometry proposed in (Czarnecki et al., 2020). According to this geometry, a game's strategy space can be partitioned into an ordered list of layers.

Definition 4.1 (The Layered Game Geometry (Czarnecki et al., 2020)) *Given a game, if the set of strategies* Π *can be factorized into* k *layers* L_i *such that* $\cup_i L_i = \Pi$*, and for any layers* $L_i, L_j \in L (i \neq j)$, $L_i \cap L_j = \emptyset$ *, then layers are fully transitive and there exists* $z \in \mathbb{R}$ *such that for each* $i < z$ *we have* $|L_i| \leq |L_i + 1|$ *and for each* $i \geq z$ *we have* $|L_i| \geq |L_i + 1|$.

The phenomenon of non-transitivity typically does not occur within a single layer of the layered game geometry. Therefore, to measure non-transitivity, the Nash clustering method was proposed to identify mixed Nash equilibria across multiple layers.

Definition 4.2 (Nash Clustering) *Given a finite two-player zero-sum symmetric game and corresponding strategy set* Π*, Nash clustering* $C := (N_j : j \in \mathbb{N} \bigwedge N_j \neq \emptyset)$*, where for each* $i \geq 1$, $N_{i+1} = supp(Nash(\mathcal{M}|\Pi \backslash \bigcup_{j \leq i} N_j))$ *for* $N_0 = \emptyset$.

The size of strategies in each Nash cluster could be used to evaluate non-transitivity.

Rock-Paper-Scissor Cycles. The formation of cycles among the strategies in the strategy space characterizes non-transitive games. A common and efficient method to measure non-transitivity is by calculating the length of the longest cycle in the strategy space. However, this is a known NP-hard problem in computing the longest directed paths and cycles in a directed graph (Björklund et al., 2004). As an alternative, we can measure non-transitivity by finding cycles of length three, known as the Rock-Paper-Scissors cycles.

Theorem 4.1 (Rock-Paper-Scissor Cycles) *Given a payoff matrix* $\mathcal{M}$*, we can obtain the corresponding adjacency matrix* $\mathcal{A}$ *from* $\mathcal{M}$*. For each element* $\mathcal{A}_{i,j}$ *in* $\mathcal{A}$*,*

$$\mathcal{A}_{i,j} = \begin{cases} 1, & \mathcal{M}_{i,j} > 0, \\ 0, & otherwise. \end{cases} \tag{3}$$

$(\mathcal{A}^k)_{i,j}$ *is the number of* k*-length paths from node* n_i *to node* n_j*. The number of Rock-Paper-Scissor (RPS) Cycles in a strategy set can be obtained by the diagonal of* $\mathcal{A}^3$*.*

Elo rating system. The Elo rating system (Elo, 1978) is frequently used to assess the relative skill levels of players in zero-sum games such as Chess or Xiangqi. It is based on the assumption that the performance of a game player is a random variable that follows a normal distribution, forming a bell-shaped curve over time. As a result, the mean value of players' performances remains at a constant level and changes gradually over time. The calculation details of the Elo rating system are provided in Appendix A.

4.2 Non-transitivity Analysis

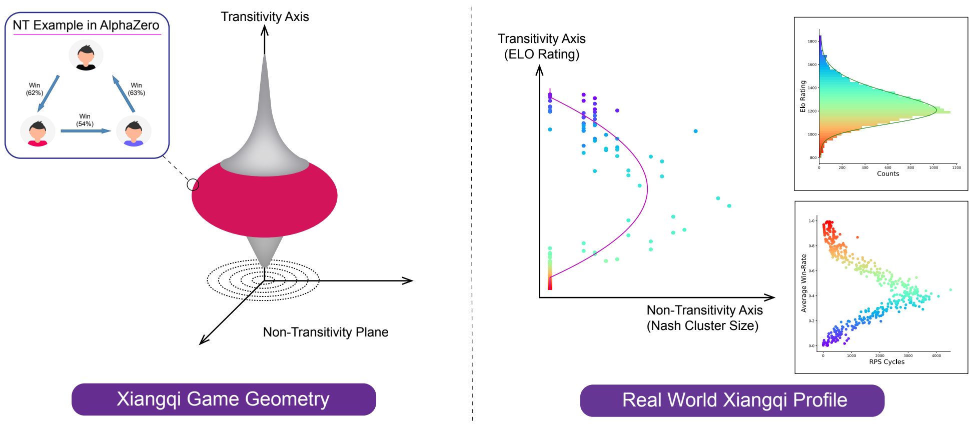


Figure 2: The Spinning Top Geometry and Profile of Xiangqi. The left figure illustrates the game geometry of Xiangqi, which resembles a spinning top structure. The upper-left corner of the left figure provides a non-transitive example of training checkpoints in Xiangqi AlphaZero. The right figures depict the real-world Xiangqi profile. In the middle sub-figure, the x-axis shows the Nash cluster size, which reflects the strength of non-transitivity, and the y-axis shows the ELO rating, which reflects the strength of transitivity. The histogram of ELO ratings (upper-right corner) and RPS cycles (lower-right corner) compares the degree of transitivity and the degree of non-transitivity, where the number of RPS cycles measures the latter.

Real-world games, including Chess and AlphaStar, are believed to have game geometrical structures that resemble a spinning top geometry, which allows the games to be split into transitive and non-transitive parts Czarnecki et al. (2020). Interestingly, we discovered that non-transitivity also exists in the self-play of Xiangqi AlphaZero, as demonstrated in the upper-left corner of Fig. 2. The three cartoon characters, represented by different colors (black, red, and blue), correspond to the different AlphaZero agents generated sequentially by self-play. The three agents exhibit non-transitivity, where the black player wins against the red player with a probability of 0.62, and the red player wins against the blue player with a probability of 0.54. Still, the red player wins against the black player with a probability of 0.63. To calculate the win probability, we played each agent against each other at least 100 times.

To thoroughly analyze Xiangqi's geometry, we obtained a dataset consisting of over 10,000 records of real-world Xiangqi games, which were sourced from the Play Ok game platform. This online gaming platform allows users to play various games, including Chess, Xiangqi, and Checkers, with players from all over the world. Our first step in analyzing this dataset was constructing a payoff matrix by discretizing the entire strategy space, a process we outline in Algorithm 1. As the data was sourced from human players, we used binned ELO ratings to measure the strength of transitivity. Specifically, we consider a two-way match-up to occur between any two ELO rating bins b_i and b_j, where $b_k = [b_k^L, b_k^H]$ for $k \in i, j$, if one player's ELO rating falls into b_i, the other player's rating falls into b_j, and each player plays both black and white.

In the initial stage, we select all relevant records ($\mathcal{D}^\star$) from the dataset $\mathcal{D}$, which satisfy the Elo rating condition for players belonging to bins b_i and b_j. In case none of the records satisfy the condition, we approximate the expected payoff score using the winning probability predicted by the Elo rating, i.e.,

$$E_{ij} = 2 \times p(i > j) - 1 = 2 \times \frac{1}{1 + \exp\left(-\frac{ln(10)}{400}\left(\frac{b_i^H - b_i^L}{2} - \frac{b_j^H - b_j^L}{2}\right)\right)} - 1. \tag{4}$$

If all the records satisfy the condition, we calculate the expected payoff score using the average game score of the games played. The game score is assigned a value of 1, 0, and -1 for a win, tie, and loss, respectively. Subsequently, we exchange players i and j and calculate the expected payoff score E_{ji}. The value of players i and j in the payoff matrix $\mathcal{M}$ for the two-way match-up between bin b_i and bin b_j is determined by averaging the expected payoff score of both cases. Specifically, we can write $\mathcal{M}i, j = \frac{Eij + E_{ji}}{2}$ and $\mathcal{M}j, i = -\mathcal{M}i, j$.

Therefore, the skew-symmetric payoff matrix can be constructed from real-world Xiangqi records by traversing every possible pair of bins.

Algorithm 1: Algorithm for building the payoff matrix $\mathcal{M}$

Data: Dataset: $\mathcal{D}$, m bins $B = \{b_1, b_2, \cdots, b_m\}$, $\mathcal{M} = \mathbf{0}_{m\times m}$.
Result: Payoff Matrix: $\mathcal{M}$.
initialization;
for $i \in [1, 2, \cdots, m]$ **do**
  for $j \in [i+1, \cdots, m]$ **do**
    if $\mathcal{D}^\star = \emptyset$ **then**
      $R_i, R_j = \frac{b_i^H - b_i^L}{2}, \frac{b_j^H - b_j^L}{2}$;
      $E_{ij} = 2 \times p(i, j) - 1.$;
    else
      $E_{ij} = 0, n = 0$;
      for *data in* $\mathcal{D}^\star$ **do**
        score = 1 if i wins, 0 if ties, -1 if i losses;
        $E_{ij} = E_{ij} +$ score;
        $n = n + 1$;
      end
      $E_{ij} = E_{ij}/n$;
    end
    Exchange players i, j, repeat the above steps and obtain E_{ji};
    $\mathcal{M}_{i,j} = \frac{E_{ij} + E_{ji}}{2}$;
    $\mathcal{M}_{j,i} = -\mathcal{M}_{i,j}$;
  end
end

The results of the analysis of the payoff matrix $\mathcal{M}$ from real-world Xiangqi records are presented in Fig. 2. The left sub-figure depicts the Xiangqi Game Geometry, which bears a resemblance to a spinning top structure. The non-transitivity plane is represented by the x-y plane, while the z-axis represents the transitivity level. Non-transitivity in the plane is increased by radiation from the origin to the outside in sequence. The right sub-figures describe the details of the real-world Xiangqi profile. The main subfigure shows the strength of non-transitivity at each ELO level, where the size of the Nash cluster size quantizes the non-transitivity. The blue curve, which illustrates the non-transitivity game profile, is obtained by fitting a skewed-normal curve to the Nash cluster size. It verifies the spinning top game geometry hypothesis. Similar to the 3D game geometry, the middle region of the ELO rating in the transitivity axis is accompanied by much stronger non-transitivity. Furthermore, the strength of non-transitivity (Nash cluster size) gradually diminishes with an increase or decrease in ELO rating. Similar evidence is revealed in the right sub-figures of the Real World Xiangqi Profile. These subfigures show the comparison between the histogram of transitivity strength (right-up corner) and the non-transitivity strength (right-bottom corner) at each average win rate, which is measured by ELO rating and the number of Rock-Paper-Scissor cycles. The middle region between 0.4 and 0.6 indicates a higher degree of non-transitivity, which is consistent with the spinning top game geometry and real-world Xiangqi profile presented in Fig. 2. As the average win rate increases or decreases, the strength of non-transitivity (the number of Rock-Paper-Scissor cycles) gradually diminishes.

5 JiangJun: the method

Through the analysis of human Xiangqi gameplay data, we have discovered that Xiangqi exhibits a spinning top game geometry structure, which reveals the presence of non-transitivity. To address this non-transitivity issue, we propose the JiangJun algorithm. As depicted in Figure 3, JiangJun is comprised of two primary modules, the MCTS Actor and the Populationer. The MCTS Actor module generates training data, while

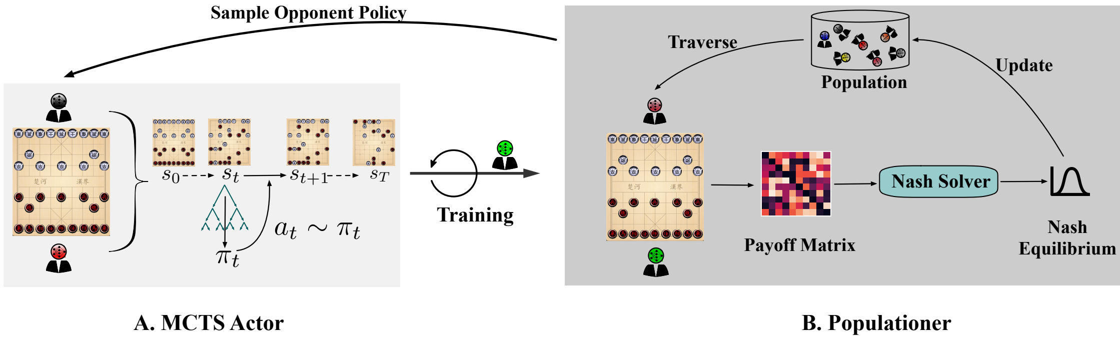


Figure 3: The proposed JiangJun algorithm consists of two main modules: *MCTS Actor* and *Populationer*. The *MCTS Actor* is the inference module that generates trajectories used for training by sampling policies from the population according to the Nash equilibrium distribution. The resulting best response policies are then fed back to the *Populationer*. In the *Populationer*, new policies are tested against each policy in the Population, and the payoff matrix is completed. We use a maximum entropy Nash solver to derive the Nash equilibrium from the payoff matrix, which is then used for Population update and policy selection.

the Populationer module maintains a Population consisting of varied policies, an inference module, and a maximum entropy Nash solver.

Populationer. The proposed *Populationer* module maintains a population of JiangJun policies and selects opponents that are approximate best responses to the Nash mixture of the population. The algorithmic details of *Populationer* are presented in Algorithm 2 in Appendix D. Specifically, *Populationer* keeps track of two entities: a population of previous JiangJun policies, and a Nash buffer for storing game results between the latest best response $\mathfrak{J}$ and all agents in the population. The information of the game result is stored as a tuple (n_0, n_1, n_2), where n_0 denotes the game score for the red player, which is assigned as $1, 0, -1$ for the wins, ties, and losses of the red, respectively. n_1 and n_2 represent the names of the red and black players. During inference, the red and black players alternate positions in each game until k games are played.

Once the Nash buffer is filled, the payoff matrix $\mathcal{M}$ is constructed by loading every tuple of the game's result into the buffer. For each tuple (n_0, n_1, n_2), we add n_0 to the value of $\mathcal{M}_{n_1,n_2}$ and subtract n_0 from the value of the symmetric position $\mathcal{M}_{n_2,n_1}$. Thus, the payoff matrix is anti-symmetric. We obtain the maximum entropy Nash equilibrium of $\mathcal{M}$ by using the maximum entropy Nash Solver and Equation 2. The resulting equilibrium distribution is used to select an appropriate opponent from the top-n agents in the population with a higher probability. Additionally, the JiangJun agent with the lowest probability is eliminated from the population to manage computing resources and replaced by the latest JiangJun agent $\mathfrak{J}$. The proposed *Populationer* module is illustrated in the right subfigure of Fig. 8.

MCTS Actor. The ***MCTS Actor*** module plays a crucial role in our proposed method. Once an opponent has been selected, the latest updated agent $\mathfrak{J}$ and the opponent agent are provided as inputs to the module, as illustrated in the left subfigure of Fig. 3. The latest JiangJun agent uses the Monte Carlo Tree Search (MCTS) algorithm to play against the opponent agent for the current state s. The position information of each piece is represented using a $0, 19 \times 10 \times 14$ matrix. The schematic diagram of state s can be found in Appendix B. Starting from the root node s, *MCTS Actor* begins a sequence of simulations. At each turn t, the *MCTS Actor* selects an action a_t in the current state s_t until the leaf node is reached. Here, a ResNets-based neural network within the JiangJun agent predicts the action probability $\mathbf{p}$ and value v with parameters θ as follows: $(\boldsymbol{p}, v) = f\theta(s)$, where f_θ is the JiangJun network, s represents the state, $\boldsymbol{p} = [p_{a_1}, p_{a_2}, \cdots]$ denotes a vector of action probabilities, and v approximates the expected outcome z of the Xiangqi game from state s. Each prior probability p_a is assigned according to the corresponding predicted action probability, and the leaf node is expanded.

After a series of simulations from the root node s, the *MCTS Actor* generates the actions probability distribution $\boldsymbol{\pi}$, where each component represents the probability of each action. Subsequently, a move is chosen according to $a \sim \boldsymbol{\pi}$, and the game proceeds until its conclusion. Finally, the final score z is determined, with +1 indicating a win, 0 indicating a tie, and −1 indicating a loss. Consequently, each trajectory contains information from every turn, including the state s, the search probabilities $\boldsymbol{\pi}$, the expected result z, and the predicted value v and the predicted action probabilities $\boldsymbol{p}$. Each trajectory is saved in the Replay Buffer after the game.

Training. Within our method, we employ a ***Training*** module to update the parameters of the JiangJun agent by simultaneously sampling the trajectories of the replay buffer and participating in a training process between *MCTS Actor* and *Populationer*. To achieve this, we employ a mean-square loss function that minimizes the error between the predicted value v and the actual outcome z of the game and a cross-entropy loss function that maximizes the similarity between the probabilities of predicted actions $\boldsymbol{p}$ and the search probabilities $\boldsymbol{\pi}$. This process can be expressed mathematically as follows.

$$l = (z - v)^2 - \alpha \boldsymbol{\pi}^T \log \boldsymbol{p} + \beta \|\theta\|^2, \tag{5}$$

where α, β are balance constants between 0 and 1, and $\|\theta\|^2$ is the L_2 weight regularization of JiangJun agent.

Evaluation Metrics. In a non-transitive game, the performance improvement of one player may not be informative, as other players can still beat that player. To address this issue, we propose a relative population ELO rating (RP-ELO) measure to evaluate the JiangJun agent's progress. At the beginning of the study, the initial ELO rating of all agents in the population and the updated JiangJun agent $\mathfrak{J}$ then is set to 1500. The JiangJun agent $\mathfrak{J}$ plays 100 games with each agent in the population to update their ELO ratings. As such, the RP-ELO provides a more comprehensive reflection of the population's improvement, rather than relying solely on the last agent, as it is non-transitive, but the ELO rating increases.

In addition, we employ the quantitative metric Relative Population Performance (RPP) Balduzzi et al. (2019b) to assess the performance of the population and the degree to which JiangJun successfully addressed the non-transitivity issue.

Definition 5.1 *Consider two populations $\mathcal{A}$ and $\mathcal{B}$, where each population comprises a set of agents. Let (p, q) be the Nash equilibrium for a zero-sum game on the payoff matrix $\mathcal{M}_{\mathcal{A},\mathcal{B}}$. The relative population performance is*

$$\mathbf{v}(\mathcal{A}, \mathcal{B}) := p^T \cdot \mathcal{M}_{\mathcal{A},\mathcal{B}} \cdot q. \tag{6}$$

Suppose $\mathbf{v}(\mathcal{A}, \mathcal{B})$ is greater than 0. In that case, it suggests that population $\mathcal{A}$ has achieved a significant performance improvement in the absence of non-transitivity when compared to population $\mathcal{B}$.

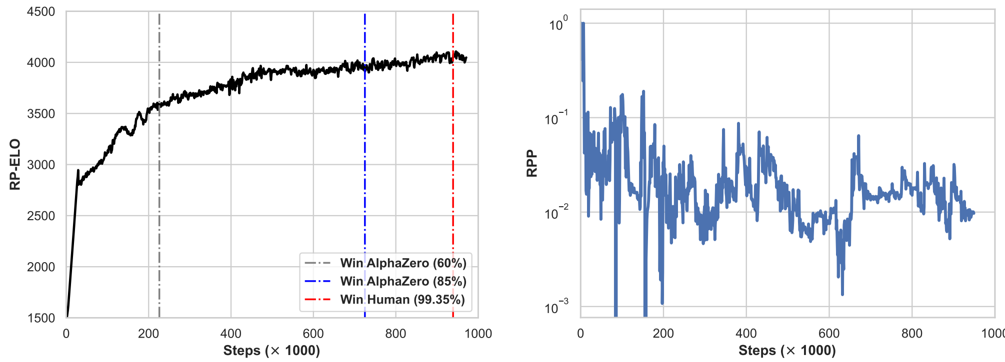
6 Experiment Results

The training of the JiangJun algorithm was performed on the Huawei Cloud ModelArt platform, utilizing a total of 90 V100 GPUs. Specifically, 78 of these GPUs were allocated for the *MCTS Actor*, 4 GPUs were used for the *Training*, and 8 GPUs were dedicated to the *Populationer*. A detailed description of the training framework can be found in Appendix C.

To comprehensively evaluate JiangJun, we developed an interactive WeChat mini program that allows human players to compete against the agent in Xiangqi games. WeChat mini-programs are sub-applications that operate within the WeChat ecosystem, which currently boasts over a million, daily active users. Appendix E provides more details on the JiangJun mini program. In Section 6.1, we present experimental results demonstrating that JiangJun can overcome the non-transitivity problem. Furthermore, in Section 6.2, we report on the agent's performance against human players in the mini-program.

6.1 Experiment results: conquer non-transitivity

The training progress of the JiangJun algorithm is presented in Fig.4, where we utilize the Relative Population ELO Rating (RP-ELO) and Relative Population Performance (RPP) to evaluate its performance. The left figure in Fig.4 shows the increase of RP-ELO with the training steps (×1000). We also indicate three vertical dashed lines from left to right as the milestones of JiangJun strength. The first dashed line indicates that the JiangJun agent at 227k steps could win Xiangqi AlphaZero with a 60% win probability. The win probability



(a) RP-ELO Curve

(b) RPP Curve

Figure 4: The figure illustrates the training progress of the JiangJun algorithm, with the Relative Population ELO Rating (RP-ELO) and Relative Population Performance (RPP) plotted in the left and right subfigures, respectively. The vertical dashed lines on the RP-ELO curve indicate the specific stages when the JiangJun agent competes with AlphaZero and human players with different win rates. The RPP curve shows the performance improvement of the population as a function of training steps ($\times 1000$). A positive RPP value indicates that the population has meaningfully improved without non-transitivity.

increases to 85% at 726k steps, shown as the middle vertical dashed line. At 940k steps, JiangJun agent can win 99.35% human players in our JiangJun mini-program. The win probability with human players is calculated based on nearly 1300 games.

Furthermore, we examine the RPP performance in the right figure of Fig. 4. The RPP value rapidly rises at the beginning of this phase and then stabilizes at a level always greater than 0, except at 13k and 14k training steps. We can observe from the RPP value curve that our proposed JiangJun effectively overcomes the non-transitivity and makes the population performance consistent in improving.

6.2 Experiment results: With human and challenging endgame

To comprehensively evaluate the efficacy of our proposed JiangJun algorithm, we have developed a JiangJuan mini-program on the WeChat platform. Details of the mini-program are available in Appendix E. Over six months, we have collected 7061 game records with human players on the mini-program, which are presented in Table 2. The monthly statistics in the table include the number of JiangJun's wins, draws, losses, total games played, and the agent's win rate for each month. Additionally, the stage flag used in the table distinguishes between two deployment stages of the JiangJun algorithm. The developing stage signifies that the deployed weights undergo training, while the done stage uses the well-trained weight. During the developing stage, our agent achieved an average win rate of 98.16% in the first three months, highlighting its high level of performance even as it continued to learn and improve. As we concluded our training and promoted the mini-program, the second three months saw an average of nearly 1700 games played each month, with JiangJun's win rate remaining stable at 99.39%. These findings demonstrate the agent's exceptional ability to compete effectively against human players in the game of Xiangqi.

Endgame in Xiangqi is a crucial phase that marks the culmination of the game, wherein both sides engage in a final struggle for survival. As this stage features fewer pieces on the board and less variation in tactics, it becomes increasingly difficult for AI models to master fundamental yet practical skills needed to solve the endgame. Despite these challenges, our analysis of the JiangJun algorithm's trajectories reveals that it has succeeded in learning how to solve the Xiangqi endgame effectively.

For instance, the classical endgame of "three Pawns V.S. the full Advisors and Bishops" is one such example where JiangJun has demonstrated its prowess in winning the endgame by keeping one Pawn in a higher position and using the other two to form a left and right pincer attack. The trajectories of JiangJun's successful endgame strategies are displayed in Appendix F, highlighting the trajectories of the "double

Month	Stage	Wins	Ties	Losses	Total	Win Rate
Month 1	Developing	717	11	8	736	97.42%
Month 2	Developing	724	0	17	741	97.71%
Month 3	Developing	462	0	3	465	99.35%
Month 4	Done	2212	2	6	2220	99.64%
Month 5	Done	1594	1	15	1610	99.01%
Month 6	Done	1283	0	6	1289	99.53%

Table 2: Monthly statistics of the JiangJun mini-program over a six-month period are presented in this table. The data is divided into two stages: developing and done. The developing stage denotes the period during which the deployed weights undergo training, while the done stage uses well-trained weights. The table provides information on the number of games won, tied, and lost by JiangJun against human players, as well as the total number of games played and the win rate for each month.

Knight and Pawn V.S. double Knight and single Bishop" endgame. Our analysis of JiangJun's trajectories demonstrates that it has successfully learned the key strategies to win Xiangqi endgames.

6.3 Case Study

This section presents two case studies of the JiangJun training process, demonstrating its effectiveness in overcoming non-transitivity in Xiangqi. The first case study presents the visualization results of our approach to conquering non-transitivity, as shown in Figure 5. The figure displays three sets of low-dimensional gamescapes as the training progresses from left to right. The top row of each set represents the payoff matrix of populations consisting of 21 agents. The color gradient from blue to red represents the average game score of the corresponding agent against other agents, where a bluer color indicates a lower win rate (i.e., a score closer to -0.5), and a redder color indicates a higher win rate (i.e., a score closer to 0.5). The bottom row of each set shows the corresponding 2D embedding of the Shur decomposition, and the dashed-dotted line denotes the second-order linear regression. Outliers in the linear regression are identified and removed using the Z-score technique.

In an ideally transitive game, the gamescape will appear as a line in the 2D embedding figure, while a non-transitive game will form a cycle Balduzzi et al. (2019b). As illustrated in Figure 5, the second order linear regression is almost a straight line across the three different training steps, suggesting that the JiangJun agent's training is approaching transitivity. This finding indicates that our approach effectively conquers non-transitivity, which ensures consistent performance improvement across the population.

The probability distributions of the Nash equilibria of AlphaZero and JiangJun are illustrated in Figure 6. Each row of subfigures represents the Nash distribution of a single population at approximately 115k training steps, with the agent index increasing from left to right. The left subfigure in Fig.6 shows that the Nash probabilities are nearly uniformly distributed when AlphaZero is trained for about 115k steps. In contrast, the right subfigure of Fig.6 indicates that the Nash probabilities of the latest agent updated by the JiangJun algorithm dominate the entire population of agents.

7 Conclusion

This paper provides a detailed analysis of the geometric structure of Xiangqi based on real-world human game data, revealing a significant degree of non-transitivity in the middle region of transitivity, which resembles the geometry of a spinning top. Additionally, we identify strategy cycles in the training checkpoints of AlphaZero for Xiangqi. To address the non-transitivity issue, we propose the JiangJun algorithm, which selects an opponent by Nash response for training instead of playing with agent-self in AlphaZero. Our empirical results show that JiangJun has achieved a master level of Xiangqi, achieving a 99.39% win rate against human players and solving complex Xiangqi endgames. Furthermore, the relative population performance and visualization results confirm that JiangJun effectively overcomes the non-transitivity issue in Xiangqi. Overall, this work

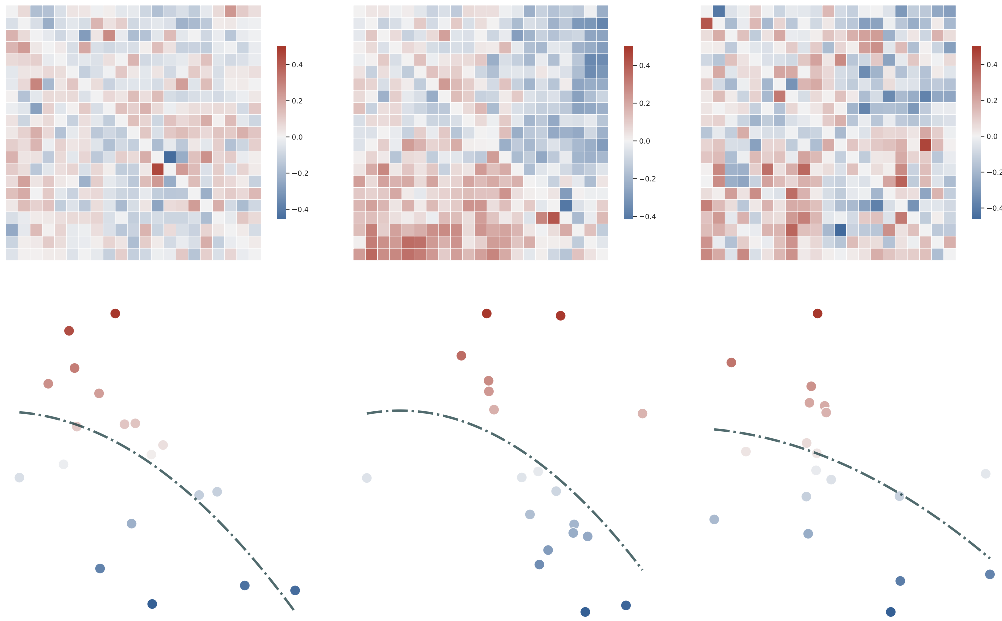


Figure 5: This figure displays the Low Dimension Gamescapes of JiangJun, with the top row of each column showing the payoff matrix of populations consisting of 21 agents. The color variation, ranging from blue (-0.5) to red (0.5), corresponds to the average game score of an agent against each other. The bottom row of each column presents the points corresponding to the first two-dimensional embedding of Shur decomposition. Additionally, a dash-dotted line is plotted for the second-order linear regression, with its outliers detected and deleted by Z-Score.

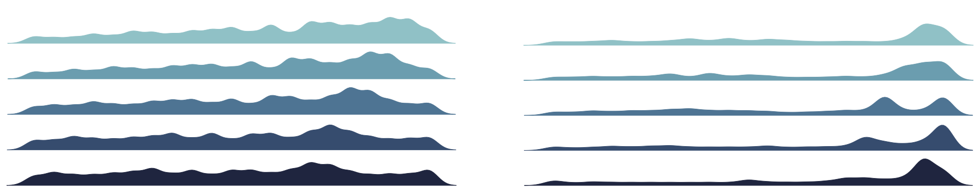


Figure 6: Comparison of Nash equilibrium distributions between AlphaZero (on the left) and JiangJun (on the right). Each row of subfigures represents the Nash distribution of a single population at approximately 115k training steps, with the agent index increasing from left to right.

provides a significant contribution to understanding the game geometry of Xiangqi and demonstrates the effectiveness of the proposed JiangJun algorithm for solving the non-transitivity problem in Xiangqi.

References

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*, 2019.

David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech Czarnecki, Julien Perolat, Max Jaderberg, and Thore Graepel. Open-ended learning in symmetric zero-sum games. In *International Conference on Machine Learning*, pp. 434–443. PMLR, 2019a.

David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech M. Czarnecki, Julien Perolat, Max Jaderberg, and Thore Graepel. Open-ended learning in symmetric zero-sum games, 2019b.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Christopher Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *CoRR*, abs/1912.06680, 2019. URL http://arxiv.org/abs/1912.06680.

Andreas Björklund, Thore Husfeldt, and Sanjeev Khanna. Approximating longest directed paths and cycles. volume 3142, pp. 222–233, 07 2004. ISBN 978-3-540-22849-3. doi:10.1007/978-3-540-27836-8_21.

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *International conference on machine learning*, pp. 793–802. PMLR, 2019.

Wojciech Marian Czarnecki, Gauthier Gidel, Brendan Tracey, Karl Tuyls, Shayegan Omidshafiei, David Balduzzi, and Max Jaderberg. Real world games look like spinning tops, 2020.

Arpad E. Elo. *The Rating of Chessplayers, Past and Present*. Arco Pub., New York, 1978. ISBN 0668047216 9780668047210. URL http://www.amazon.com/Rating-Chess-Players-Past-Present/dp/0668047216.

Haobo Fu, Weiming Liu, Shuang Wu, Yijia Wang, Tao Yang, Kai Li, Junliang Xing, Bin Li, Bo Ma, Qiang Fu, et al. Actor-critic policy optimization in a large-scale imperfect-information game. In *International Conference on Learning Representations*, 2022.

Yang Guan, Minghuan Liu, Weijun Hong, Weinan Zhang, Fei Fang, Guangjun Zeng, and Yue Lin. Perfectdou: Dominating doudizhu with perfect information distillation. *arXiv preprint arXiv:2203.16406*, 2022.

Johannes Heinrich and David Silver. Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121*, 2016.

Daniel Hennes, Dustin Morrill, Shayegan Omidshafiei, Remi Munos, Julien Perolat, Marc Lanctot, Audrunas Gruslys, Jean-Baptiste Lespiau, Paavo Parmas, Edgar Duenez-Guzman, et al. Neural replicator dynamics. *arXiv preprint arXiv:1906.00190*, 2019.

Benjamin Kerr, Margaret Riley, Marcus Feldman, and Brendan Bohannan. Local dispersal promotes biodiversity in a real game of rock-paper-scissors. *Nature*, 418:171–4, 08 2002. doi:10.1038/nature00823.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Perolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning, 2017.

John Banister Lanier, Stephen McAleer, Pierre Baldi, and Roy Fox. Feasible adversarial robust reinforcement learning for underspecified environments. *arXiv preprint arXiv:2207.09597*, 2022.

Joel Lehman and Kenneth O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, 2008.

Joel Lehman and Kenneth O Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011.

Junjie Li, Sotetsu Koyamada, Qiwei Ye, Guoqing Liu, Chao Wang, Ruihan Yang, Li Zhao, Tao Qin, Tie-Yan Liu, and Hsiao-Wuen Hon. Suphx: Mastering mahjong with deep reinforcement learning. *CoRR*, abs/2003.13590, 2020. URL https://arxiv.org/abs/2003.13590.

Xiangyu Liu, Hangtian Jia, Ying Wen, Yaodong Yang, Yujing Hu, Yingfeng Chen, Changjie Fan, and Zhipeng Hu. Unifying behavioral and response diversity for open-ended learning in zero-sum games. *CoRR*, abs/2106.04958, 2021. URL https://arxiv.org/abs/2106.04958.

Stephen McAleer, John B. Lanier, Roy Fox, and Pierre Baldi. Pipeline PSRO: A scalable approach for finding approximate nash equilibria in large games. *CoRR*, abs/2006.08555, 2020. URL https://arxiv.org/abs/2006.08555.

Stephen McAleer, John Lanier, Pierre Baldi, and Roy Fox. Xdo: A double oracle algorithm for extensive-form games. In *NeurIPS*, 2021.

Stephen McAleer, Gabriele Farina, Marc Lanctot, and Tuomas Sandholm. Escher: Eschewing importance sampling in games by computing a history value function to estimate regret. *arXiv preprint arXiv:2206.04122*, 2022a.

Stephen McAleer, JB Lanier, Kevin Wang, Pierre Baldi, Roy Fox, and Tuomas Sandholm. Self-play psro: Toward optimal populations in two-player zero-sum games. *arXiv preprint arXiv:2207.06541*, 2022b.

Stephen McAleer, Kevin Wang, Marc Lanctot, John Lanier, Pierre Baldi, and Roy Fox. Anytime optimal psro for two-player zero-sum games. *arXiv preprint arXiv:2201.07700*, 2022c.

H. Brendan McMahan, Geoffrey J. Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pp. 536–543. AAAI Press, 2003. ISBN 1577351894.

Paul Muller, Shayegan Omidshafiei, Mark Rowland, Karl Tuyls, Julien Perolat, Siqi Liu, Daniel Hennes, Luke Marris, Marc Lanctot, Edward Hughes, et al. A generalized training approach for multiagent learning. *arXiv preprint arXiv:1909.12823*, 2019.

John F. Nash. Non-cooperative games. *Annals of Mathematics*, 54:286, 1951.

Nicolas Perez Nieves, Yaodong Yang, Oliver Slumbers, David Henry Mguni, Ying Wen, and Jun Wang. Modelling behavioural diversity for learning in open-ended games. *arXiv preprint arXiv:2103.07927*, 2021.

Luis E. Ortiz, Robert E. Schapire, and Sham M. Kakade. Maximum entropy correlated equilibria. In Marina Meila and Xiaotong Shen (eds.), *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pp. 347–354, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR. URL `https://proceedings.mlr.press/v2/ortiz07a.html`.

Julien Perolat, Remi Munos, Jean-Baptiste Lespiau, Shayegan Omidshafiei, Mark Rowland, Pedro Ortega, Neil Burch, Thomas Anthony, David Balduzzi, Bart De Vylder, et al. From poincaré recurrence to convergence in imperfect information games: Finding equilibrium via regularization. In *International Conference on Machine Learning*, pp. 8525–8535. PMLR, 2021.

Julien Perolat, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T Connor, Neil Burch, Thomas Anthony, et al. Mastering the game of stratego with model-free multiagent reinforcement learning. *Science*, 378(6623):990–996, 2022.

Tobias Reichenbach, Mauro Mobilia, and Erwin Frey. Mobility promotes and jeopardizes biodiversity in rock–paper–scissors games. *Nature*, 448(7157):1046–1049, Aug 2007. ISSN 1476-4687. doi:10.1038/nature06095. URL `http://dx.doi.org/10.1038/nature06095`.

David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016. URL `http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html`.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018. doi:10.1126/science.aar6404. URL `https://www.science.org/doi/abs/10.1126/science.aar6404`.

Eric Steinberger, Adam Lerer, and Noam Brown. Dream: Deep regret minimization with advantage baselines and model-free learning. *arXiv preprint arXiv:2006.10410*, 2020.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, L. Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom Le Paine, Caglar Gulcehre, Ziyun Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, pp. 1–5, 2019a.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michael Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P Agapiou, Max Jaderberg, Alexander S Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 521:1–5, 2019b.

Bin Wu, Qiang Fu, Jing Liang, Peng Qu, Xiaoqian Li, Liang Wang, Wei Liu, Wei Yang, and Yongsheng Liu. Hierarchical macro strategy model for MOBA game AI. *CoRR*, abs/1812.07887, 2018. URL `http://arxiv.org/abs/1812.07887`.

Yaodong Yang, Jun Luo, Ying Wen, Oliver Slumbers, Daniel Graves, Haitham Bou-Ammar, Jun Wang, and Matthew E. Taylor. Diverse auto-curriculum is critical for successful real-world multiagent learning systems. *CoRR*, abs/2102.07659, 2021. URL `https://arxiv.org/abs/2102.07659`.

Daochen Zha, Jingru Xie, Wenye Ma, Sheng Zhang, Xiangru Lian, Xia Hu, and Ji Liu. Douzero: Mastering doudizhu with self-play deep reinforcement learning. In *International Conference on Machine Learning*, pp. 12333–12344. PMLR, 2021.

Enmin Zhao, Renye Yan, Jinqiu Li, Kai Li, and Junliang Xing. Alphaholdem: High-performance artificial intelligence for heads-up no-limit poker via end-to-end reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 4689–4697, 2022.

A Elo rating system

Denote the players of Xiangqi as w and b. If w and b has a rating of R^w and R^b respectively, then the expected scores are given by:

$$\begin{aligned} E_w &= \frac{1}{1+10^{(R_b-R_w)/400}}, \\ E_b &= \frac{1}{1+10^{(R_w-R_b)/400}}. \end{aligned} \tag{7}$$

The expected scores also could be expressed as

$$\begin{aligned} E_{\mathrm{w}} &= \frac{Q_{\mathrm{w}}}{Q_{\mathrm{w}}+Q_{\mathrm{b}}}, \\ E_{\mathrm{b}} &= \frac{Q_{\mathrm{b}}}{Q_{\mathrm{w}}+Q_{\mathrm{b}}}, \end{aligned} \tag{8}$$

where $Q_{\mathrm{w}} = 10^{R_{\mathrm{w}}/400}$, and $Q_{\mathrm{b}} = 10^{R_{\mathrm{b}}/400}$. Besides, K-factor is another important variable for Elo rating system. The value of K represents the maximum possible adjustment per game. Therefore, rating update formula of player w, b are

$$\begin{aligned} R_w &= R_w + K\left(S_w - E_w\right), \\ R_b &= R_b + K\left(S_b - E_b\right), \end{aligned} \tag{9}$$

where $S_{w/b}$ is the actually scored points. After obtaining the ELO rating, we could approximate the winning probability p by

$$p(w,b) \approx \frac{1}{1+\exp\left(-\frac{ln(10)}{400}(R_w - R_b)\right)}. \tag{10}$$

B Representation of game state

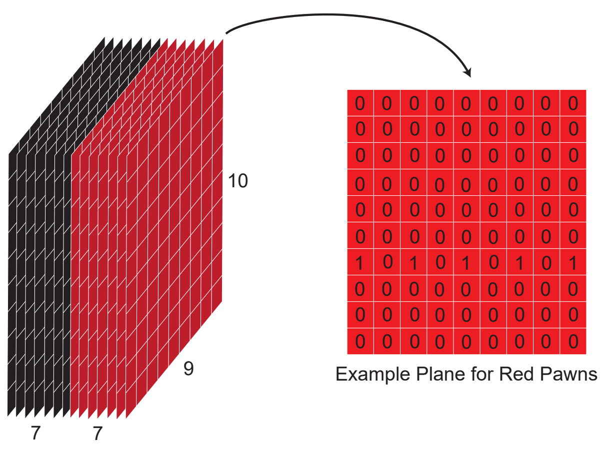


Figure 7: Schematic diagram of the game state.

The schematic diagram of state s is given in Fig 7. The composition of state s. s is a $9 \times 10 \times 14$ matrix, where each plane presents one type of piece's position. with each component of 0 (represents absence) and 1 (represents presence), as shown in the left sub-figure. In the figure, a plane with red or black color represents the piece position of red player or black player respectively. The right sub-figure shows an example of red Pawns, where the Pawn locates in a position of 1. Taking red Pawns as an example, the right figure of Figure 7 represents the initial state of red Pawns. Besides, the first 7 planes are assigned for red player and the last 7 planes are for black player.

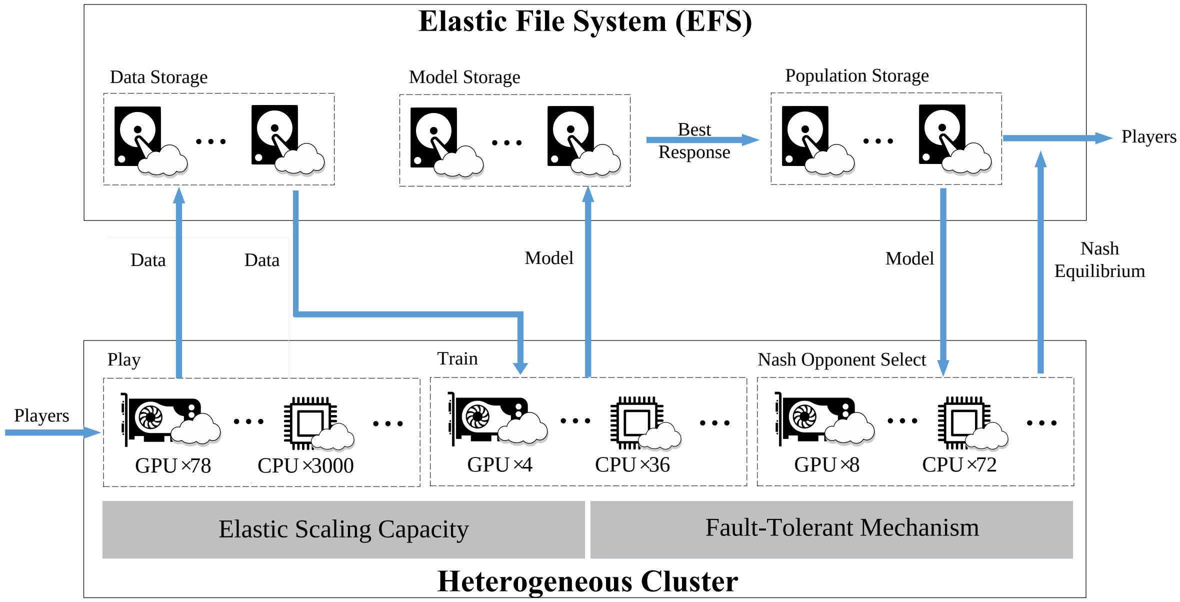


Figure 8: JiangJun Cloud-Native Training Framework.

C JiangJun: Training framework and details

C.1 Training framework

We firstly introduce the training framework, which is conducted on the basis of Huawei Cloud ModelArt. Figure 8 gives a briefly introduction of our could-native JiangJun training framework, consisting of two main modules, i.e., heterogenous cluster (HC) and elastic file system (EFS). HC module aims to provide GPUs, CPUs heterogeneous distributed computing services. In our whole resource pool, we have 90 32GB NVIDIA V100 GPUs and over 3000 CPU cores. How to make so many resources run efficiently, safely and stably is one of the important issues of the whole HC design. Consequently, Elastic Scaling Capacity and Fault-Tolerant Mechanism are provided to ensure the efficient, secure and stable run of GPUs and CPUs.

C.2 Training details

The hyperparameters of the network and training are provided as follows.

- network filters: 192,
- network layers: 10,
- batch size: 2048,
- sample games: 500,
- c_puct : 1.5,
- saver step: 400,
- learning rate : [0.03, 0.01, 0.003, 0.001, 0.0003, 0.0001, 0.0003, 0.001, 0.003, 0.01],
- minimum games in one block: 5000,
- maximum training blocks: 100,
- minimum training blocks: 3,
- number of the process: 10.

D Algorithm of *Populationer*.

Algorithm 2 gives the details of *Populationer* module.

Algorithm 2: Algorithm for *Populationer*

Input: Latest agent $\mathcal{J}$ updated by *Train*. Population $\mathfrak{A}$ with k agents. Nash replay buffer $\mathfrak{B}$.

```
for agent 𝔞 in 𝔄 do
    // Play with every agent in Play
    for i = 1, 2, ..., k do
        if i%2 == 0 then
            red ⟵ 𝒥.
            black ⟵ 𝔞.
        else
            red ⟵ 𝔞.
            black ⟵ 𝒥.
        end
        // Play returns game result information
        t =(score n_0 for red (win=1,tie=0,loss=-1), red index n_1, black index n_2) ⟵ red plays with black.
        Store game information tuple t in 𝔅.
    end
end
Initialize payoff matrix ℳ = 0_{k×k}.
for tuple (n_0, n_1, n_2) in 𝔅 do
    ℳ_{n_1,n_2} = ℳ_{n_1,n_2} + n_0.
    ℳ_{n_2,n_1} = ℳ_{n_2,n_1} − n_0.
end
p ⟵ Nash on ℳ.
P_o ⟵ Sample from top-n agents with higher probability.
𝔄 ⟵ 𝔄 ∪ {𝒥}\{agent with lowest probability.}
```

Output: Opponent agent P_o, New population $\mathfrak{A}$.

E JiangJun: Mini Program

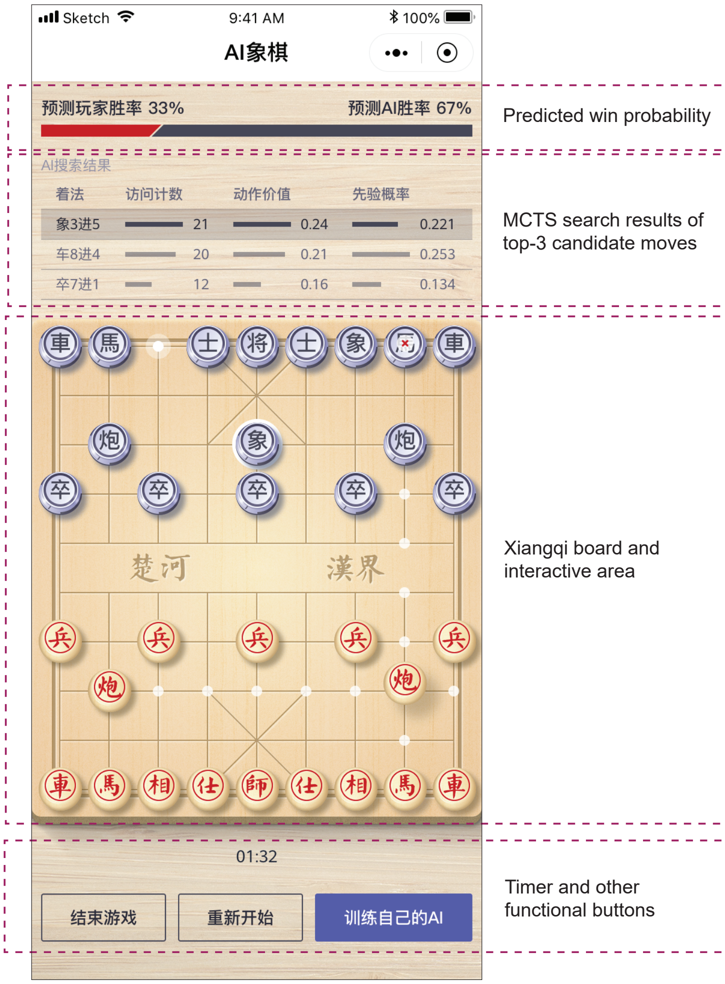


Figure 9: The JiangJun WeChat mini-program is composed of four distinct regions, including the win probability region, MCTS search information region, Xiangqi board region, and functional button region, which are all depicted in the provided screenshots.

The JiangJun mini program, depicted in Figure 9, allows users to play Xiangqi against our trained JiangJun agent. The top of the program displays the predicted win probability. The next section presents the top three candidate actions with their corresponding prior probabilities obtained from the Monte Carlo Tree Search (MCTS). For each action, we show its name in Chinese and key items used in action selection, including the visit count of the parent node $N(s)$, the action value $Q(s,a)$, and the prior probability $P(s,a)$ of selecting action a in state s_t. The main area of the mini program is where the human player interacts with JiangJun. Finally, the bottom region contains a game timer and three function buttons: finish, restart, and train your own JiangJun AI in ModelArt.

F Trajectories of JiangJun playing endgames

F.1 Game A : Three Pawns V.S. the full Advisors and Bishops

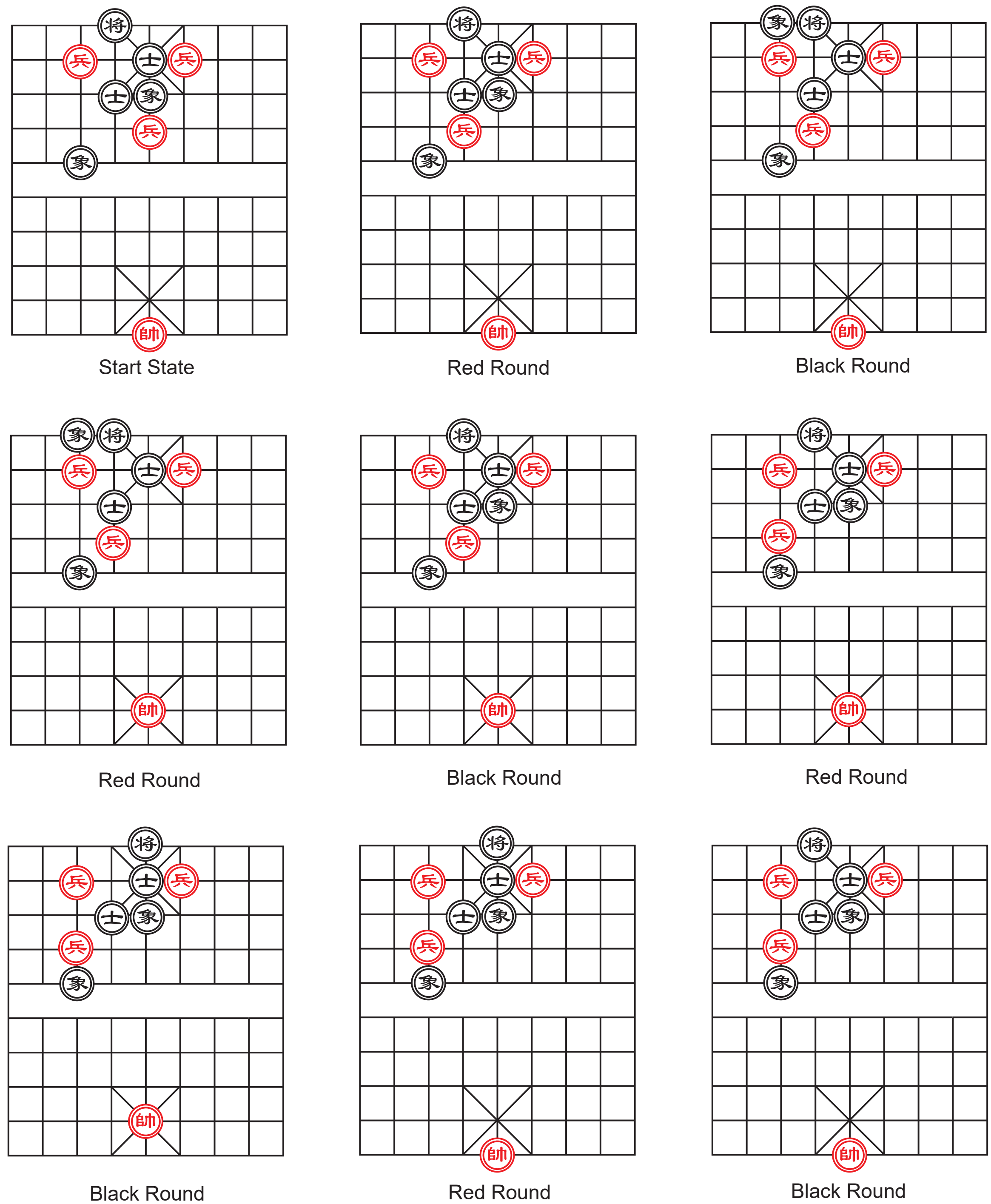


Figure 10: Game Trajectories of JiangJun playing endgame "Three Pawns V.S. the full Advisors and Bishops".

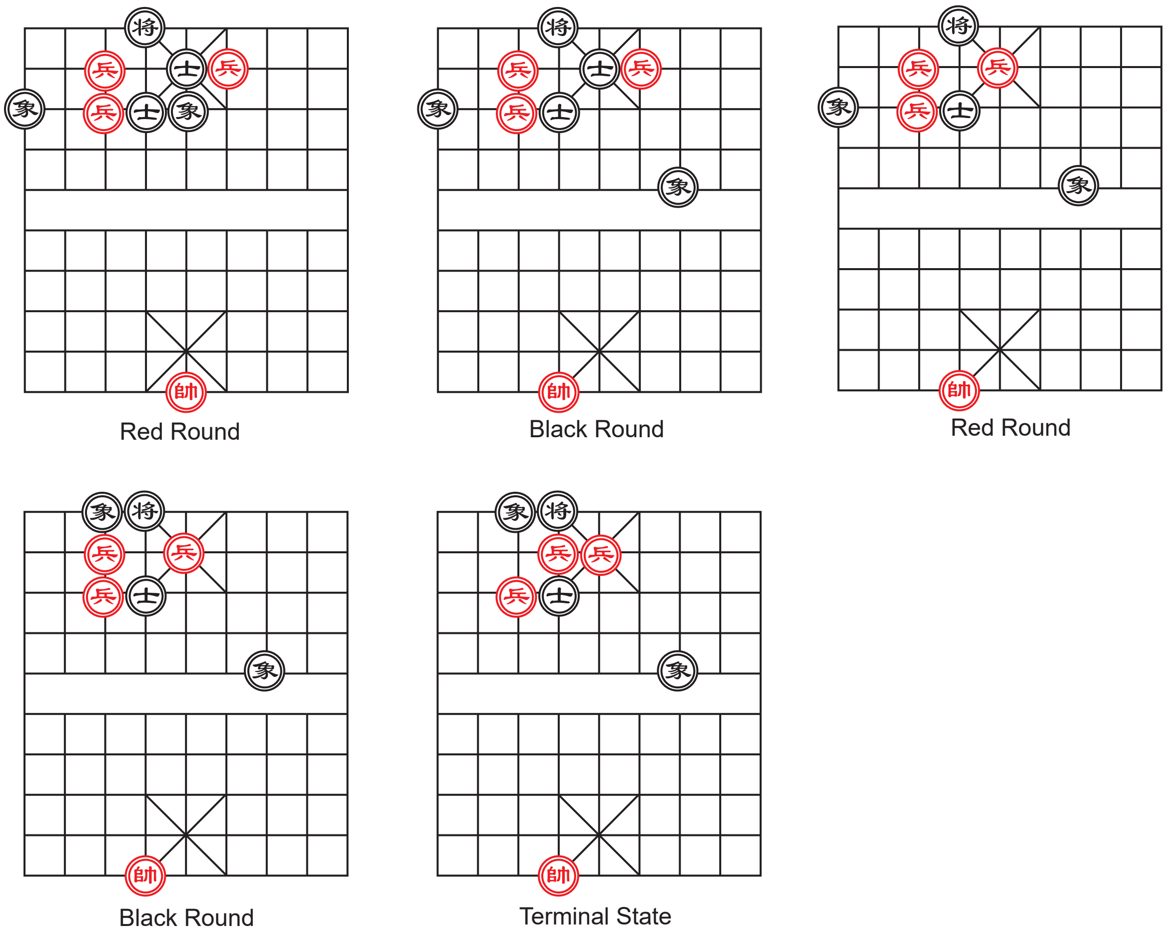


Figure 11: Game Trajectories of JiangJun playing endgame "Three Pawns and the full Advisors and Bishops".

F.2 Game B: double Knight and Pawn V.S. double Knight and single Bishop

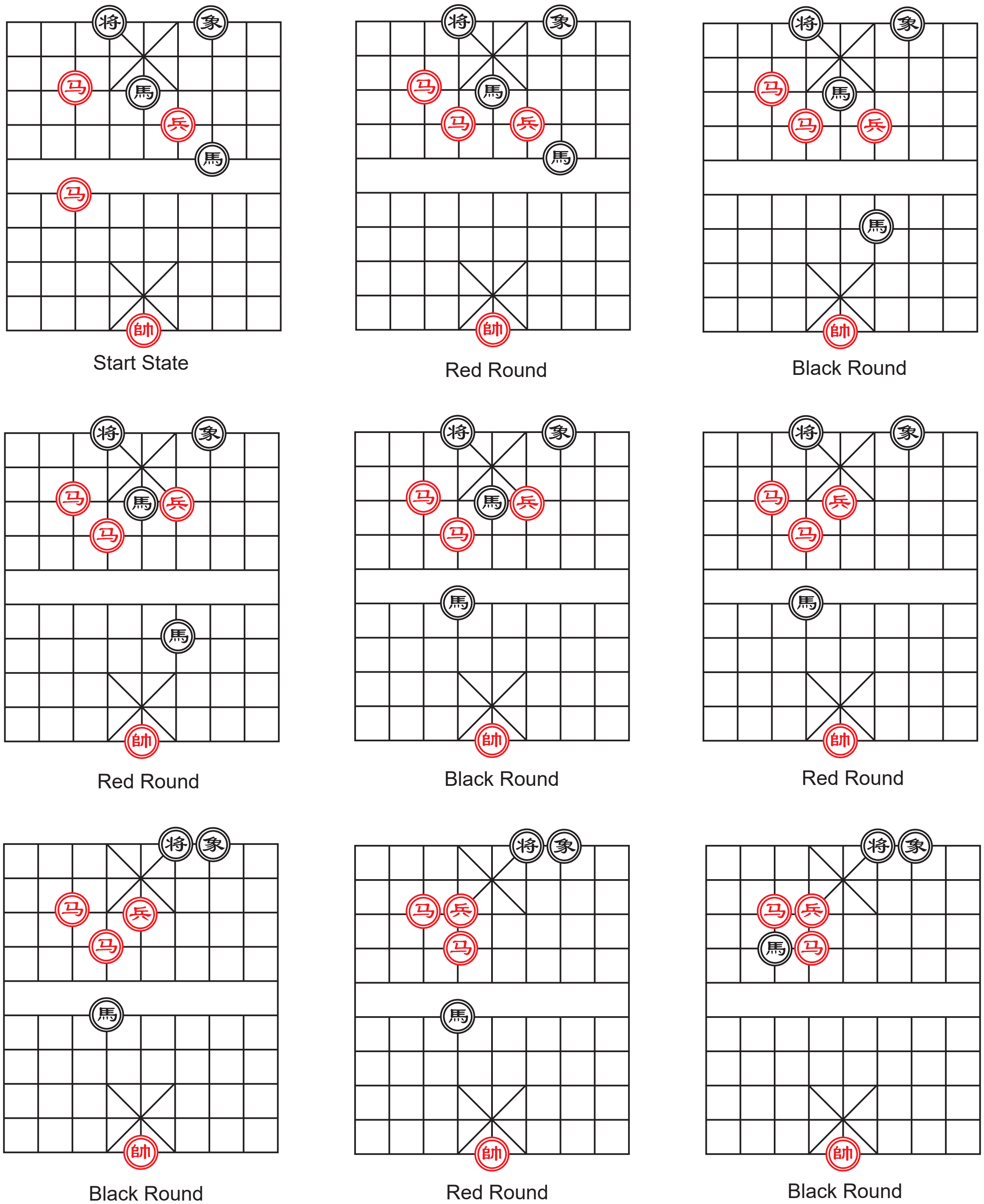


Figure 12: Game Trajectories of JiangJun playing endgame "double Knight and Pawn V.S. double Knight and single Bishop"

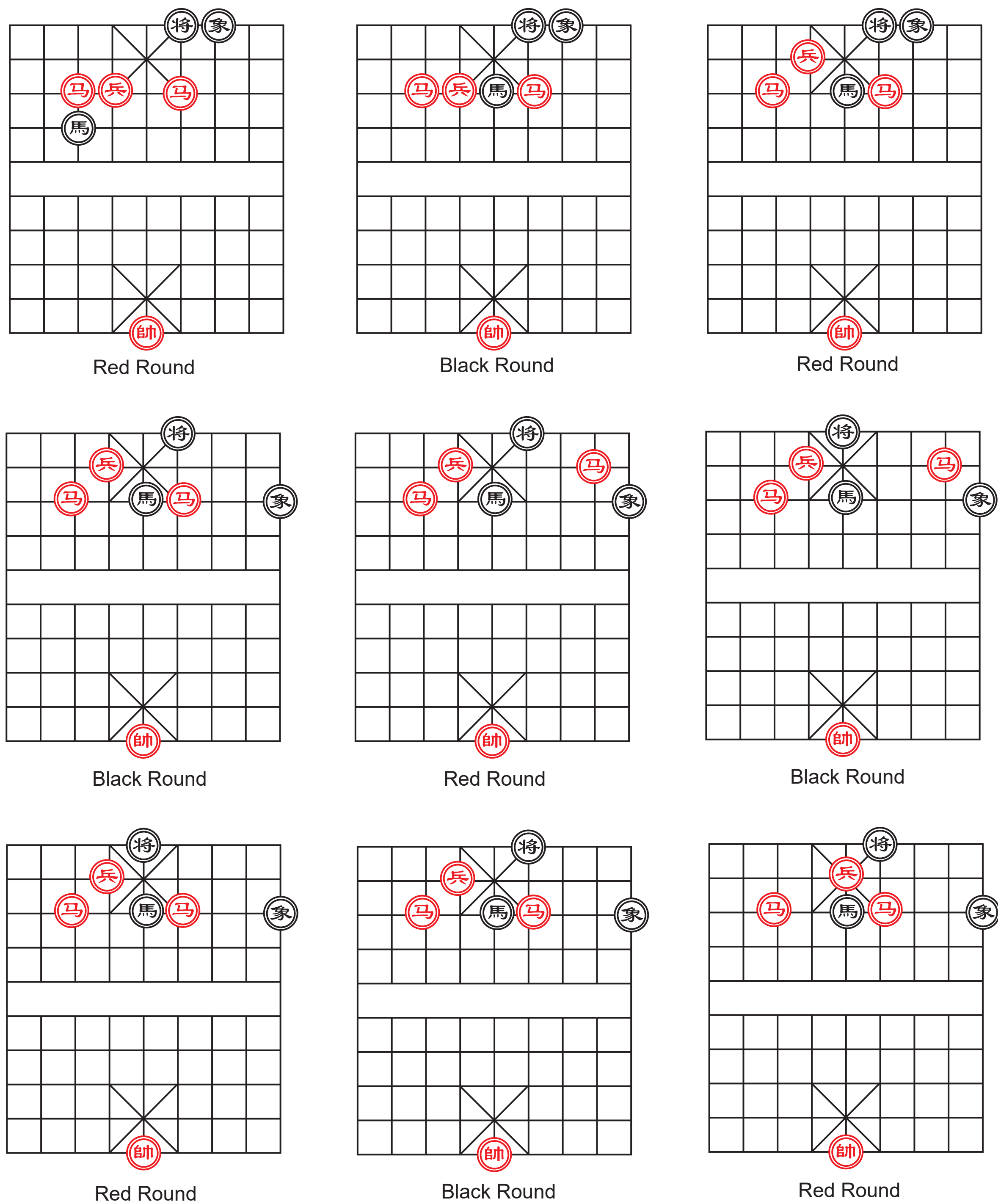


Figure 13: Game Trajectories of JiangJun playing endgame "double Knight and Pawn V.S. double Knight and single Bishop"

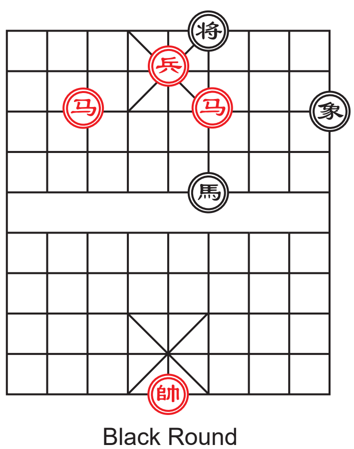


Figure 14: Game Trajectories of JiangJun playing endgame "double Knight and Pawn V.S. double Knight and single Bishop"