Ranking of Contract Bridge Players

Yuran Wang

Ranking of Contract Bridge Players

by

Yuran Wang

To obtain the degree of Master of Science
in Applied Mathematics at Delft University of Technology,
to be defended publicly on Tuesday April 15^{th}, 2025 at 10:00 AM.
Faculty of Electrical Engineering, Mathematics and Computer Science.

Student number:	6058221	
Thesis committee:	Dr. R. Fokkink	TU Delft, Chair and responsible supervisor
	Dr. H. Wang	TU Delft, Daily supervisor
	Dr. C. Kraaikamp	TU Delft, Core member

Project duration: September, 2024 – April, 2025

Acknowledgements

I have been playing Contract Bridge since primary school, and this game has accompanied me through many stages of my life. It has always been my wish to combine my interest with academic research. This thesis could not have been completed without the help and support of many people, and I would like to express my sincere gratitude to all those who have given me guidance, companionship, and encouragement.

First of all, I would like to express my special thanks to my supervisor, Doctor Robbert Fokkink. It has been a great fortune for me to get to know him through the course on Game Theory. He has a great sense of humour, profound professional knowledge, and a strong sense of responsibility towards his students. From the overall research structure of the thesis to the advice for each chapter, from the choice of research methodology to the in-depth academic guidance on the mathematical level, and from advice on the research process to guidance on graduation, he has always been a great help. He also guided me in broadening my academic horizons by combining Contract Bridge with game theory and numerical analysis.

I would like to give special thanks to my Contract Bridge teacher, Mr. Depei Liu. He has been my Bridge mentor for more than ten years, not only teaching me how to play, but also inspiring a deep interest in the mathematical logic and social dynamics behind the game. Thanks to his ongoing support, both for the game itself and its underlying principles, I was able to explore the world of Bridge more deeply and eventually incorporate it into my academic research.

My family's understanding and support have been my greatest source of strength throughout this journey. I would like to thank my father, mother and grandmother. They have always respected my choice, encouraged me to study in the Netherlands and trusted me unconditionally. They supported me fully, both mentally and financially, so that I could focus on my studies without any distractions.

I would also like to thank my friends who have accompanied me along the way. Jiaqin Li, has always been there to share life's joys and sorrows from junior high school, and her support and understanding mean a lot to me. Junyang Zhang, my boyfriend, who stood by me from undergraduate, offered emotional comfort and encouragement during the most stressful times of exam preparation, and together we have witnessed each other's growth. Chengyu Liu, my former roommate, whose optimism was like sunshine on the grey days in the Netherlands. And my most important partners in graduate school, Wei Liu and Hongrui Liu—we worked side by side in the library, supported and encouraged one another; their excellence and perseverance have deeply inspired me and made me more confident in my academic journey and future path.

Finally, I would like to thank all those who have helped and encouraged me in both my academic research and daily life. Because of you, this journey has become richer and more meaningful. Once again, thank you from the bottom of my heart!

Yuran Wang

Delft, April 2025

Abstract

Contract bridge is a challenging card game that combines strategic bidding, teamwork, and probabilistic outcomes. This thesis develops a rigorous mathematical framework for analyzing duplicate bridge tournaments, with a focus on fair and accurate player ranking. We introduce a threshold ranking model that extends the classic Bradley–Terry model by incorporating a tie parameter, allowing us to model the non-negligible probability of tied scores in bridge results. We derive theoretical results for this model, notably, we prove that the total match-points and number of tied boards are sufficient statistics for the estimation of players' skill levels and the tie parameter. Using both a reference dataset from literature and the results of the 2024 China National Bridge Championship, we demonstrate that our model can reliably estimate skill parameters: the inferred rankings align with actual tournament outcomes and distinguish the strongest and weakest pairs. To complement the parametric model, we employ a fuzzy logic approach to categorize board-by-board performance into qualitative grades ("Excellent", "Good", etc.) and compute aggregate fuzzy performance scores. This provides an additional lens to evaluate each pair's consistency and tendency for extreme results. The combined analysis yields a richer understanding of performance: while the statistical model emphasizes overall consistency and head-to-head advantages, the fuzzy analysis highlights variability and exceptional highs or lows. Our results show that both methods concur on the identification of top performers, reinforcing confidence in the findings, and together they reveal nuanced differences among closely ranked competitors. These insights further confirm the fairness of the duplicate format (skill prevails in the long run) while quantifying the impact of chance and strategic diversity on interim rankings.

In summary, the contributions of this thesis are: (1) a new tie-aware mathematical model for ranking players in duplicate bridge, supported by theoretical guarantees and validated by empirical data; (2) the integration of fuzzy logic metrics into bridge performance evaluation, offering practical measures of consistency; and (3) a comprehensive case study on championship data illustrating how our methods can be applied to draw meaningful conclusions about player abilities and game dynamics. This work not only advances the methodology for analyzing bridge tournaments but also provides tools and perspectives that can inform the design of fair competition systems and the training of competitive bridge players.

Contents

1 Introduction

Bridge, or Contract Bridge, is a card game played with a standard 52-card deck(without jokers). The game usually consists of four players, forming two competing partners who sit opposite each other. Bridge is widely regarded as one of the most popular card games in the world [37]. It appeared in 1925 and is largely attributed to the scoring innovations of Harold Stirling Vanderbilt. His innovations included scoring only contract tricks below the game or slam prize line, introducing perishability to increase the complexity of sacrifice strategies, and adjusting scores for balance [41].
Contract Bridge combines skill and luck, with the element of luck arising from the random dealing of cards. There are many different types of Bridges, but most club and tournament games are played in duplicate Bridges, which is also the focus of this article. In the duplicate Bridge, much of the randomness is mitigated by comparing the results of multiple pairs of the same hand. This format requires at least eight players at two or more tables, with hands being saved and passed to other tables. At the end of the session, scores are compared across tables, with the player who performed best on each hand receiving the highest score. This setup measures relative skill while retaining a degree of luck, as all participants are judged on their handling of the same hand.
Duplicate Bridge tournaments can accommodate hundreds of players and are usually held in clubs or large venues. The first officially sanctioned world championship was held in 1935; in 1958, the World Bridge Federation (WBF) was formed to promote Bridge worldwide, coordinate periodic revisions of the rules, and oversee the conduct of world championships [12]. Although the popularity of Bridge has declined since its peak in the 1940s (when 44 per cent of American households participated), the game still maintains a particular popularity. In 2005, the American Bridge Association estimated that 25 million Americans still play Bridge [32].
Many celebrities enjoy contract Bridge. Bill Gates calls it 'the best mind game ever,' and Warren Buffett believes that Bridge hones logic and decision-making skills. Queen Elizabeth II of the United Kingdom regards Bridge as an elegant activity that demonstrates one's intelligence and cool-headedness. Winston Churchill compared the Bridge to the epitome of war, emphasizing its fighting spirit and strategy. Deng Xiaoping, who was intensely passionate about contract Bridge, once stated, "Bridge, like music, is a universal language and should become a Bridge of communication, understanding, and friendship between the Chinese people and people around the world " [33]. For these celebrities, Bridge was a game of leisure and the perfect combination of intellectual challenge and social art.

1.1 The rules of contract Bridge

The game is played through a series of deals, each of which consists of four stages: dealing, bidding, playing, and scoring. Bridge is a zero-sum game played with two pairs of partners, each composed of two players. In a zero-sum game, one side's gain is exactly the other side's loss. This means that every point scored by one side is subtracted from the opponent's score, making the game highly competitive and strategic. The game begins with each of the four card players sorting out the 13 cards that belong to them and then bidding.
The auction determines the contract, specifying the number of tricks a partnership must win and the trump suit (or no trump). The dealer opens the auction, and calls proceed clockwise. A player may pass, bid a contract higher than the last bid, double the opponents' bid, or redouble [21]. Bids represent the number of tricks above six (e.g., a bid of 1♠ commits to seven tricks with spades as trumps). Denominations are ranked in ascending order: ♣, ♢, ♡, ♠, NT (no trump). Each subsequent bid must either increase the number of tricks or specify a higher-ranking suit than the previous bid. The auction phase ends when three consecutive players fold, at which point the final contract is determined.
When bidding, the Bridge players bid on a contract that specifies the number of tricks their partner (i.e., the declarer) will take to score points. During the auction, players communicate information about their hands by bidding, such as card points (as in the Acol bidding system, where A-4 points, K-3 points, Q-2 points, J-1 points, and others-0 points) and suit (spades, hearts, diamonds, clubs) distributions, and any other form of communication is forbidden. The playing phase begins, with

the declarer trying to fulfill the contract and the defender trying to stop it. Scoring depends on the suit won, the contract, the vulnerability of partnerships, and game-specific variations [23]. Bridge combines computational and psychological elements that require logical analysis as well as deception and reasoning, making it both strategically deep and entertaining.

1.1.1 Setup and dealing

In Bridge, four players form two pairs of partners who sit on opposite sides of the table. These partners are designated North-South and East-West. The cards can be freshly shuffled or split in advance in the case of duplicate Bridge for comparative scoring. The base game requires only one deck of cards and one scoring method, although equipment such as duplicate Bridge boards, call boxes or screens are often used in tournaments.
In duplicate Bridge, the cards are distributed manually or by computer in advance. After dealing, the cards are placed into a "board" with designated slots for each player's position (North, East, South, or West). After completing a game, the cards are returned to their slots for use at other tables [1].

1.1.2 Auction and bidding

The auction phase is one of the most complex aspects of the Bridge, as partnerships aim to arrive at a makeable contract. Partners must exchange information about their hands while adhering to restrictions that limit communication to the calls made and later to the cards played. All agreed-upon meanings of calls must also be disclosed to the opponents, adding an element of transparency to the strategy.
A bidding system is a set of agreements between partners about the meanings of calls. It consists of a core framework supplemented by conventions—specialized agreements for handling specific scenarios. Systems range from natural systems like Standard American and Acol to artificial systems like Precision Club.

- **Natural Calls:** Reflect the hand's strength and distribution. For example, a 1NT bid indicates a balanced hand with a specific high-card point (HCP) range.
- **Conventional (Artificial) Calls:** Convey specific pre-agreed meanings, such as Blackwood (asking for the number of aces) or Stayman (inquiring about four-card major suits).

Conventions like preemptive bids serve dual purposes: they communicate a long suit in a weak hand and disrupt opponents' ability to exchange information effectively.

Key principles of bidding

- **Point Count:** High card points (HCP) form the foundation of hand evaluation, with aces, kings, queens, and jacks valued at 4, 3, 2, and 1 points, respectively. A hand with 12-13 points is generally strong enough to open the bidding.
- **Balanced Hands:** A 1NT opening typically indicates a balanced hand with a specific HCP range, often 12-14, 15-17, or 16-18.
- **Preemptive Bids:** High-level bids with long suits disrupt opponents and communicate hand shape, such as an opening bid of 3♠ with a weak hand but a strong spade suit.

Advanced bidding techniques

- **Forcing and Non-Forcing Bids:** Certain bids require a response from the partner, while others do not.
- **Cue Bids:** Bidding the opponents' suit to show strength or a desire for partner input.

- **Takeout Doubles:** Indicate strength but lack a clear suit, asking the partner to choose a suit.

Some conventions and systems also include specialized agreements, such as Jacoby Transfers, Texas Transfers, and Cappelletti over 1NT. Advanced bidding techniques often rely on subtle inferences and predefined agreements between partners.

Doubling increases penalties for undertricks and rewards for making the contract. Redoubling further increases these values. The auction ends when three players pass consecutively after the last bid. Bidding systems help partnerships communicate hand strengths and distributions.
In tournaments, bidding boxes are widely used to eliminate the risk of verbal bidding being heard by nearby tables. These measures ensure fairness by limiting communication and reducing the possibility of unauthorized information sharing. Each box contains all the possible bids a player may make during the auction, and players place their bid cards on the table in order during the auction.

1.1.3 Playing

The player from the declaring side who first bids the final contract's denomination becomes the declarer. The player to the left of the declarer leads the first card, after which the declarer's partner, known as the dummy, lays their cards face-up on the table, organized by suit. The declarer controls the dummy's cards, instructing their play. Play proceeds clockwise, with players following suit if possible. A trick is won by the highest card of the suit led or by the highest trump if trumps are played [2].
Players may claim the remaining tricks at any time, laying their cards face-up and explaining their intended play sequence. Opponents can either accept the claim or dispute it. If disputed in duplicate Bridge, the tournament director resolves the claim.
There are four fundamental ways to take a trick:

1. **Establishing Long Suits:** The last cards in a suit win if opponents run out of that suit.
2. **Playing High Cards:** Using the highest card remaining in play.
3. **Finessing:** Playing for opponents' high cards to be in a specific position.
4. **Trumping:** Using trumps to beat an opponent's high card.

Most advanced techniques are extensions or refinements of the four basic techniques.

1.1.4 Scoring

Points are awarded to the side that completes the pact or to the defending side that successfully defeats the pact. In duplicate Bridge, vulnerability is pre-assigned and affects rewards and penalties. Contract points are awarded for tricks bid and made beyond six [7]:

- 20 points per odd trick for clubs or diamonds,
- 30 points per odd trick for hearts or spades,
- 40 points for the first odd trick in no trump and 30 for subsequent tricks.

These points are doubled or redoubled if the contract was doubled or redoubled.

Bonuses

Bonuses are awarded for slams:

- **Small slam** (12 tricks): 500 points if not vulnerable, 750 if vulnerable.
- **Grand slam** (13 tricks): 1000 points if not vulnerable, 1500 if vulnerable.

For play by another table.

1.2 An illustrative 3NT example

This example demonstrates how a partnership arrives at a 3NT contract. It explains the techniques required to fulfill the contract and describes the associated scoring under various conditions of vulnerability.

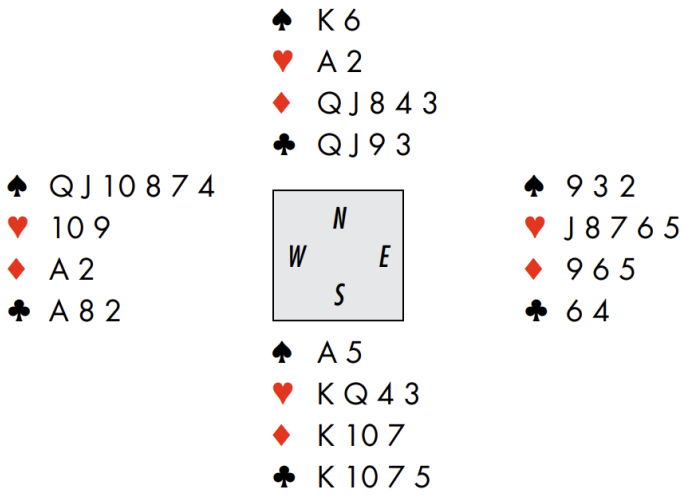
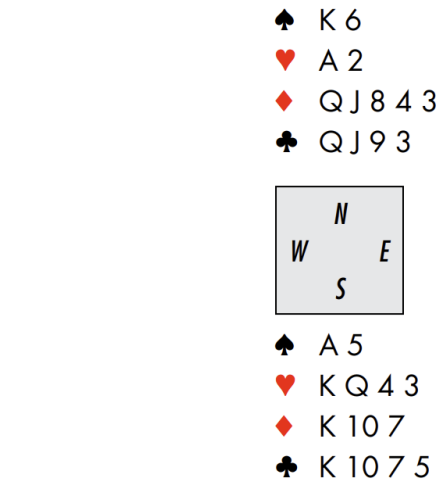


Figure 1: A possible real-life layout of all four players' cards.



West	North	East	South
			1NT
pass	3NT	all pass	

Figure 2: Focus on the North-South hands, along with the complete bidding sequence.

Figure 1 illustrates how the four hands might appear in an actual Bridge deal, showing each player's cards in spades, hearts, diamonds, and clubs. In contrast, figure 2 highlights the specific North-South hands used in this 3NT contract discussion, together with the final bidding sequence, thereby focusing on the essential details for our analysis.

1.2.1 Bidding

South typically holds 15–17 high-card points (HCP) in a balanced or semi-balanced distribution (e.g., 4–3–3–3, 4–4–3–2, or 5–3–3–2) and opens `1NT` to convey this shape and strength. North, recognizing sufficient combined values for a game and finding no four-card major-suit fit, raises directly to `3NT`. This straightforward approach reflects standard practice: once the partnership has at least 25 total points and cannot locate an eight-card major-suit fit, `3NT` is the most efficient contract to secure the game bonus.

1.2.2 Declarer play as a decision tree

The choices the declarer (South) faces in a 3NT contract can be viewed as a complex decision tree. Each decision requires balancing the development of a long suit (to gain extra tricks) with the need to retain sufficient control to avoid surrendering the lead prematurely. There is an important difference between the diamond suit and the club suit. In clubs you have four cards in each hand whereas in diamonds you have five in one hand and three in the other. So by knocking out the **A** you can develop four tricks in diamonds but only three in clubs. Since you need four more tricks for your contract, you must attack diamonds.

Suppose the North-South partnership collectively holds eight cards in diamonds—three in South's hand and five in North's—implying that the two opponents together hold the remaining five diamonds. Assuming the opponents' hands are otherwise unknown and that the remaining 26 cards are equally likely to be distributed, we analyze the possible suit splits using combinatorial enumeration.
Let $P(a\text{-}b)$ denote the probability that the opponents' five diamonds are split as a cards in one hand and b cards in the other (with $a + b = 5$ and $a \geq b$). Since there are two opponents, the labeling of hands is symmetric, and we consider both permutations of each asymmetric split.
The probabilities are given by:

$$P(3\text{-}2) = \frac{\binom{13}{3}\binom{13}{2}}{\binom{26}{5}} \times 2 \approx 67.8\%$$

$$P(4\text{-}1) = \frac{\binom{13}{4}\binom{13}{1}}{\binom{26}{5}} \times 2 \approx 28.3\%$$

$$P(5\text{-}0) = \frac{\binom{13}{5}\binom{13}{0}}{\binom{26}{5}} \times 2 \approx 3.9\%$$

Here, each binomial coefficient $\binom{n}{k}$ represents the number of ways to choose k cards out of n, and the total number of possible 5-card combinations from the opponents' 26 unknown cards is $\binom{26}{5}$. The factor of 2 accounts for the symmetry in assigning the distributions to either opponent.

Although a **5–0** split represents only about 4% of all deals, at higher levels of competition it cannot be disregarded. If one opponent holds all five diamonds, including the **A** and a key card (e.g., the **9**), the declarer risks conceding an additional trick while establishing the suit, thereby disrupting the original plan. Even with a **4–1** split, if the defender who holds four diamonds also has the **A** and **9** immediately behind the dummy, maintaining transportation and control becomes more difficult.
Note that there is not time to develop both suits. You have the ♠A and ♠K, one of which will win the first trick, and the defense will continue spades when they take their ♢A, so you will be exposed in spades after you win that trick. If you lose the lead again the opponents will take four spade tricks, so you cannot profitably knock out the ♣A as well as the ♢A. Playing on diamonds, you take two spades, three hearts and four diamonds and make your contract.

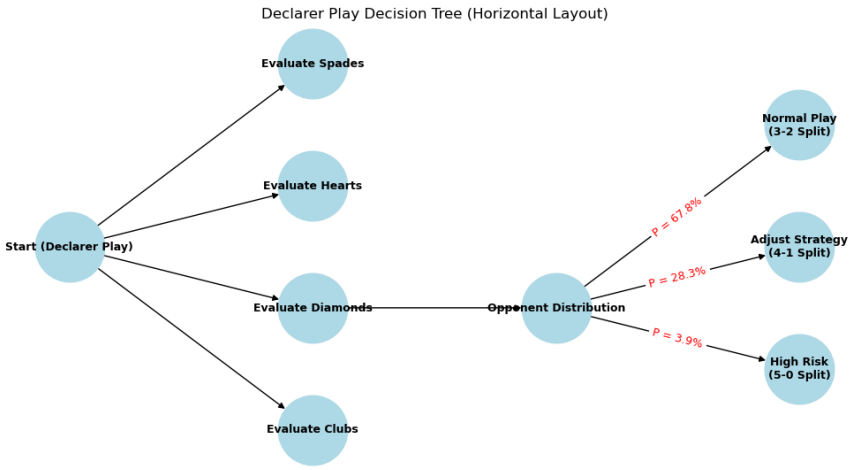


Figure 3: A schematic representation of the declarer's decision-making process in a 3NT contract.

Prior to commencing play, the declarer should approximate the probability of each distribution and formulate a strategy to counter potential setbacks. This could involve forcing out the opponents' **A** of diamonds at an opportune time, while preserving sufficient high cards and entries in the other suits to prevent the opponents from cashing multiple tricks once they gain the lead. Such planning and execution highlight the intellectual depth of Bridge: success depends not only on probability and technical skill, but also on the ability to read opponents' holdings and time plays effectively.

1.2.3 Scoring in this example

If South successfully wins nine or more tricks, the final score depends on the vulnerability status of the declaring side. When non-vulnerable, the contract yields 40 points for the first notrump trick over six and 30 points for each of the subsequent two tricks, totaling 100 contract points. Fulfilling a non-vulnerable 3NT also earns a 300-point bonus, bringing the overall total to 400. Under vulnerable conditions, the same 100 contract points apply, but the game bonus increases to 500, resulting in a total of 600. Conversely, if the defenders defeat the contract, they receive penalty points based on the number of undertricks and on the declarer's vulnerability. This zero-sum dynamic underlines the need for precise bidding and careful play: a single error can drastically shift the outcome in favor of the defenders.

1.3 Movements in Bridge Tournament

In bridge tournaments, rationalising the order of play is essential to ensure a fair and fully competitive game. To solve the problems of balance and efficiency in the pairings, tournaments usually adopt systematic pairing mechanisms. The two most common methods are the 'Mitchell Movement' and the 'Howell Movement'. With their different designs, these two approaches provide effective matchmaking strategies for bridge tournaments.

1.3.1 Mitchell Movement

In the Mitchell Movement mechanism, the arrangement of bridge matches ensures that all bridge pairs compete against different opponents in different rounds, thereby maintaining fairness and symmetry throughout the competition. Specifically, all participating pairs are divided into two groups: one group always sits in the North-South (N/S) direction, and the other always sits in the East-West (E/W) direction. The boards remain fixed at the tables throughout the competition, while the E/W pairs rotate each round. This ensures that each board is played by different partnerships under comparable conditions, allowing objective performance comparisons.
In the example of a three-table Mitchell Movement, suppose there are $n = 6$ bridge pairs and $B = 3$ boards. The pairs are divided into two groups:

- N/S pairs: {1, 2, 3}
- E/W pairs: {4, 5, 6}

t	$\alpha(b,t)$			$\beta(b,t)$		
b	$t=1$	$t=2$	$t=3$	$t=1$	$t=2$	$t=3$
1	1	3	2	4	5	6
2	2	1	3	5	6	4
3	3	2	1	6	4	5

Table 1: A 3-table Mitchell movement

In each round, N/S pairs compete against E/W pairs, while pairs within the same group do not compete against each other. This ensures fairness because multiple partners are involved on each board, and results can be compared based on identical hands, rather than direct competition between pairs. Since the boards remain fixed, only the players rotate, ensuring that no pair plays the same board twice. The specific rotations are as follows:

Rotation Arrangement

- **Round 1**:
  - N/S pairs: 1, 3, 2
  - E/W pairs: 4, 5, 6
  - Matchups:
    - Pair 1 (N/S) vs Pair 4 (E/W)
    - Pair 3 (N/S) vs Pair 5 (E/W)
    - Pair 2 (N/S) vs Pair 6 (E/W)
- **Round 2**:
  - N/S pairs: 2, 1, 3
  - E/W pairs: 5, 6, 4
  - Matchups:
    - Pair 2 (N/S) vs Pair 5 (E/W)
    - Pair 1 (N/S) vs Pair 6 (E/W)
    - Pair 3 (N/S) vs Pair 4 (E/W)
- **Round 3**:
  - N/S pairs: 3, 2, 1
  - E/W pairs: 6, 4, 5
  - Matchups:
    - Pair 3 (N/S) vs Pair 6 (E/W)
    - Pair 2 (N/S) vs Pair 4 (E/W)
    - Pair 1 (N/S) vs Pair 5 (E/W)

Each N/S pair faces different E/W pairs in each round with this rotation arrangement. The key aspect of this mechanism is that N/S pairs always compete against E/W pairs, minimizing the impact of random card distribution on match outcomes. Each group is ranked independently: N/S pairs $1, 2, 3$ are compared among themselves and E/W pairs $4, 5, 6$ are compared among themselves. Therefore, the tournament produces two winners: one in the N/S direction and one in the E/W direction.

1.3.2 Mathematical properties of the Mitchell Movement

The Mitchell Movement is a structured rotation system designed to ensure fairness in duplicate bridge tournaments. It consists of n bridge pairs, denoted as $P = \{1, 2, \ldots, n\}$, competing across t tables, denoted as $T = \{1, 2, \ldots, t\}$, using b boards, denoted as $B = \{1, 2, \ldots, b\}$. In each round, matches occur at fixed tables, where North-South (N/S) pairs remain stationary, while East-West (E/W) pairs rotate systematically. The boards remain fixed at their respective tables, ensuring that the same set of hands is played by different partnerships under comparable conditions.
To formalize this structure, we define two mapping functions: $\alpha : B \times T \to P$, where $\alpha(b, t)$ returns the N/S pair playing board b at table t, and $\beta : B \times T \to P$, where $\beta(b, t)$ returns the E/W pair for the same board and table. The pairing matrix M, defined as $M_{b,t} = (\alpha(b, t), \beta(b, t))$, records all matchups in a tabular form. Since each E/W pair faces a different N/S pair at every table, the resulting pairing matrix exhibits the structure of a Latin square, where each symbol appears exactly once in each row and column [43].
To illustrate, consider a three-table Mitchell Movement where the rotation follows a structured cyclic permutation:

$$M = \begin{bmatrix} (1,4) & (3,5) & (2,6) \\ (2,5) & (1,6) & (3,4) \\ (3,6) & (2,4) & (1,5) \end{bmatrix}$$

Observing this arrangement, each row represents a different table, each N/S pair competes with a unique E/W pair, and each column ensures a unique game between different tables. This fulfils the fundamental property of the Latin square that there are no duplicate elements in any row or column. The structural rotation inherent in the Mitchell movement ensures that each pair of players competes on equal terms while maintaining fairness and balance. This combined arrangement ensures that comparisons between the N/S and E/W pairs of players are unbiased, thus reinforcing the integrity of the outcome of the game.

1.3.3 Howell Movement

The Howell Movement differs from the Mitchell Movement in that it does not divide Bridge pairs into North/South (N/S) and East/West (E/W) groups with fixed positions. Instead, it employs a systematic rotation so that every pair has the opportunity to play against every other pair exactly once over the course of the tournament. As a result, the Howell format typically produces one overall ranking for all participating pairs, rather than separate rankings for N/S and E/W.
Compared to the Mitchell Movement, the Howell Movement can offer a more comprehensive comparison of pairs' skills, because each pair meets every other pair. However, it requires more complex scheduling: for $2n$ total pairs, there are $2n - 1$ rounds so that each pair can face every other pair exactly once. In addition, there is only one overall winner determined by comparing the total match points $(r_i, i = 1, \ldots, 2n)$ among all pairs.
Nevertheless, in certain practical situations (e.g., time constraints or limited boards), organizers may choose a curtailed version of the Howell Movement where not all matchups occur. For the ideal case, though, the full schedule ensures that every pair is compared with every other pair on equal footing.

1.3.4 Mathematical objectives of the Howell Movement

To formalize the Howell Movement, assume there are $2n$ pairs. In a full Howell schedule, we need exactly $2n - 1$ rounds (labeled $r = 1, 2, \ldots, 2n - 1$) so that every pair meets each other pair. Within each round, matches take place at n tables (since $2n$ players split into pairs, each table seats exactly two pairs). We adopt the following notation:

- $\alpha(r, t)$ represents the pair seated in the North-South (N/S) position at table t in round r.
- $\beta(r, t)$ represents the pair seated in the East-West (E/W) position at table t in round r.

The key requirements to ensure a valid (and complete) Howell Movement are:

- **Unique Matchup Rule:** For every *distinct* pair of players $i \neq j$, there must be exactly one round r (and one table t) such that
$$\{\alpha(r,t), \beta(r,t)\} = \{i, j\}.$$
This enforces that each pair faces every other pair exactly once.
- **Injectivity Constraint:** Within the *same* round r, the mappings $t \mapsto \alpha(r,t)$ and $t \mapsto \beta(r,t)$ must be injective, i.e., no pair can appear more than once in a round. This ensures a valid seating arrangement without duplications.

Round-Robin Construction and Example

A well-known method for constructing such a schedule is the round-robin technique, sometimes referred to as the "circle method." Label the $2n$ pairs by $1, 2, \ldots, 2n$. Then:

1. **Fix** one pair in position (say pair 1).
2. **Rotate** the remaining $2n - 1$ pairs circularly each round.

This guarantees that across the $2n - 1$ rounds, each pair meets every other pair exactly once. To illustrate, consider $2n = 8$ (i.e., $n = 4$ tables) and label the pairs as $1, 2, 3, 4, 5, 6, 7, 8$. A possible round-by-round schedule is given in Table 2.

Round	Pairings (each sub-row = a table)	
1	1	8
	2	7
	3	6
	4	5
2	1	7
	8	6
	2	5
	3	4
3	1	6
	7	5
	8	4
	2	3
4	1	5
	6	4
	7	3
	8	2
5	1	4
	5	3
	6	2
	7	8
6	1	3
	4	2
	5	8
	6	7
7	1	2
	3	8
	4	7
	5	6

Table 2: A standard 7-round Howell schedule for 8 pairs

Here, each round lists 4 pairings (one per table). After 7 rounds, every pair has encountered each of the other 7 pairs exactly once, satisfying the Unique Matchup Rule. The Injectivity Constraint holds in each round because no pair is repeated at any table. Hence, this round-robin approach provides a complete and systematic construction for the Howell Movement, and it extends naturally to any even number of pairs $2n$.

In summary, the Howell Movement provides a framework in which all pairs compare against each other. This leads to a single, unified ranking and a more balanced measure of performance than the Mitchell Movement (where pairs stay mostly in fixed directions). Although the schedule can be more

complex to implement, it achieves the desirable property that no pair is advantaged or disadvantaged by facing or missing a particular opponent—every matchup happens exactly once.

1.4 Scientific literature on the game of Bridge

Compared to other fields such as chess or Go, there is not much scientific literature on the game of Bridge. However, Bridge research has still produced fascinating results in the fields of statistics, game theory, artificial intelligence, and social sciences. This section will survey the existing literature and highlight some representative works.

1.4.1 Statistical studies

Statistical methods are important for assessing and measuring the skills of contract Bridge players. Yu and Lam (1996) proposed the use of a paired data approach to analyse scoring points using a permutation model to relate the skills of different opponents in a Bridge match [34]. This is the paper on which I have based my MSc thesis. In addition, Michael Gr. Voskoglou proposed a method to assess the overall performance of a tournament using fuzzy logic [40]. This approach improves the accuracy of Bridge data collection and analysis by designing fuzzy sets for player pairs and merging them to apply defuzzification of the centre of gravity. These statistical applications not only help assess Bridge players' skills but also provide a more accurate scenario for in-game evaluation. We will discuss the comparison of these two approaches in section 7.1.

1.4.2 A game theory study

Game theory is used in games of incomplete information, such as Bridge, to help make optimal decisions in the presence of asymmetric information. The application of game theory to Bridge usually focuses on strategy formulation, opponent modelling, and the solution of optimal behaviour, and the work of Ian Frank and David Basin provides an insight into the use of search algorithms to compute optimal strategies in games with incomplete information [13].
They proposed an algorithm for calculating equilibrium strategies based on the 'best defence' model. In this game model, players not only need to choose the best action based on the current information but also need to consider the possible strategies adopted by their opponents. By introducing the 'best defence' simulation, a method is proposed that can provide Bridge players with optimal strategy suggestions. This approach is particularly suitable for dealing with situations where information is incomplete, such as when the opponent's hand is not fully known.
Furthermore, Ian Frank and David Basin analyse the limitations of using sampling algorithms in games with incomplete information and point out two major problems in complex games such as Bridge: 'strategy fusion' and 'non-locality' [13]. The term 'strategy fusion' refers to the fact that subtle differences between different strategies cannot be accurately distinguished during game simulation, which may lead to unclear final decisions and thus affect the accuracy of the results. 'Non-locality' means that players often need to consider information beyond the current situation when making decisions, which increases the complexity and computational difficulty of decision-making.

1.4.3 AI and Bridge

There are also notable applications of AI and computer science in the field of contract Bridge. First, Yeh et al. proposed a Bridge calling system based on deep reinforcement learning [47]. The system uses a deep neural network to autonomously learn the calling rules from the raw data and achieves a balance between exploration and exploitation through an upper confidence bound (UCB) algorithm. Experiments show that the model outperforms computer programs that use manually designed calling systems. This result demonstrates the potential for learning effective calling strategies without prior human knowledge.
Secondly, Smith et al. developed a program called Bridge BARON, which uses Hierarchical Task Network (HTN) planning to simulate the process of playing a bright hand of Bridge [39]. This

approach breaks down complex game planning into smaller subtasks and avoids the inefficiencies of traditional game tree search in an incomplete information environment such as Bridge. The success of Bridge BARON demonstrates the feasibility of AI planning techniques in card games and provides an important direction for the subsequent development of computer Bridge programs.
Furthermore, in dealing with uncertainty in decision-making of Bridge AI, recursive Monte Carlo (RMC) searching has been applied to this game with incomplete information [14]. Compared to traditional Monte Carlo methods, RMC performs much better in Bridge scenarios, especially in the no-bid phase. It significantly improves the average results and provides important insights into the development of Bridge AI and computational methods in complex decision-making.

1.4.4 Studies in social sciences

The social study of Bridge focuses on how players make decisions and interact in uncertain environments. Many researchers have studied the behavioural patterns and coping strategies of Bridge players when facing incomplete information. Punch's research analyzed how Bridge players use experience from tournaments to handle uncertainty, emphasizing how players improve their performance under high-pressure conditions through constant practice and feedback [36]. Additionally, interviews with 52 elite Bridge players revealed that these players exhibit a high level of impression management and strategic interaction during matches, demonstrating notable abilities in cooperation, discipline, and dealing with errors [24].
Furthermore, social research on the Bridge also covers the discussion of gender differences. Some scholars have analyzed the impact of women's Bridge tournaments on female players [38]. Women's tournaments provide opportunities for women to participate in international competitions, but they also reinforce gender differences to some extent, leading to perceptions that female players are technically inferior to open-category players. This separation not only brings opportunities but also exacerbates gender inequality in the Bridge community.
These studies show that Bridge is not only a competition of intellect but also a context in which players engage in complex interactions and apply strategies in social situations.

1.5 Conclusion

In section 1.1, we surveyed the historical background and fundamental rules of contract Bridge, highlighting its four-phase structure (dealing, bidding, play, and scoring) and emphasizing the balanced yet competitive nature of duplicate Bridge tournaments.
We explored the Mitchell and Howell movements (section 1.3) to see how structural rotations can fairly match pairs in large tournaments, then outlined each method's mathematical implications. We also reviewed scientific literature that spans statistics, game theory, artificial intelligence, and social sciences (section 1.4), revealing that Bridge constitutes not just an intellectual pastime but also a rich domain for rigorous study.
From this chapter, we have gained an integrated understanding of how Bridge is played, organized, and researched, providing essential context for the modeling and data analyses in later chapters. In the upcoming chapter, we will shift focus to the mathematical underpinnings of our study, covering essential statistical and inferential concepts. These tools will guide the construction of likelihood functions, hypothesis tests, and confidence intervals, all of which are vital for robust Bridge tournament analysis.

2 Mathematics background

Before going into the statistical modeling of Bridge tournaments, we must establish the key mathematical and statistical tools underpinning our analysis. This section provides the basic concepts of probability theory and inference that we will use to assess player performance, construct likelihood-based models, and interpret the resulting estimates and tests.
Section 2.1 begins with an overview of probability distributions and sampling principles, focusing on describing random variables (e.g., through cumulative distribution functions) and how to derive sampling statistics, such as mean and variance, from data. These basic ideas will guide us later in developing more advanced frameworks, such as permutation models and maximum likelihood inference, in section 3.2 and section 3.3.1.

2.1 Notions from statistics

Although Bridge scores may not always follow common distributions, understanding basic distribution shapes and sample variability is essential. Statistics provides methods to describe and infer data properties. We briefly review several essential ideas.
Cumulative Distribution Function (CDF): Denoted as $F(x)$, representing the probability that a random variable X takes a value less than or equal to x, i.e.,

$$F(x) = P(X \leq x)$$

Probability Density Function (PDF): Denoted as $f(x)$, representing the probability density of a random variable around a specific value.
Below are the cumulative distribution function (CDF) and probability density function (PDF) results for the maximum and minimum values.

- Cumulative Distribution Function (CDF) of the Maximum Value $X_{(n)}$

$$F_{X_{(n)}}(x) = [F(x)]^n$$

- Probability Density Function (PDF) of the Maximum Value $X_{(n)}$

$$f_{X_{(n)}}(x) = n \cdot [F(x)]^{n-1} \cdot f(x)$$

- Cumulative Distribution Function (CDF) of the Minimum Value $X_{(1)}$

$$F_{X_{(1)}}(x) = 1 - [1 - F(x)]^n$$

- Probability Density Function (PDF) of the Minimum Value $X_{(1)}$

$$f_{X_{(1)}}(x) = n \cdot [1 - F(x)]^{n-1} \cdot f(x)$$

2.2 Parameter estimation

Building upon the generic statistical concepts above, we now turn to the specific context of Bridge tournaments. In Bridge, pairs of players accumulate scores board by board. We aim to use these observed scores to infer each pair's relative skill level.

2.2.1 Match-Point (MP) scores as a statistic

When multiple pairs play the same deal (or set of deals), their performance can be ranked, and match-point (MP) scoring assigns points based on these rankings. For each board, a pair's MP on that board is computed by comparing its result with all other pairs who played the same board. Summing these MP across all boards yields a single numerical value for each pair's overall performance. Hence, the

total MP is a statistic. It is derived solely from the observed data (the board-by-board results) and encapsulates how well a pair performed relative to others.
We often postulate that each pair has a latent "ability" parameter to analyze and compare skills. In later sections, we will formalize this idea with a permutation model, hypothesizing that the probability of one pair outscoring another depends on their relative abilities. We treat the observed MP outcomes as realizations of random events driven by the pairs' skill parameters. We estimate each pair's skill by fitting a statistical model to these outcomes and comparing it across all participants.
A complete mathematical definition of this permutation model and detailed estimation procedures will be provided in section 3, especially in section 3.2.
Connecting the raw scoreboard data to meaningful skill inferences relies on the fundamental ideas discussed in section 2.1. MP scores serve as a univariate summary (a statistic) from which we estimate parameters. Methods such as likelihood ratio tests or confidence intervals help us evaluate whether observed differences in scores are statistically significant or could be due to random variation in card dealing.
With these elements in place, the following sections will show how to apply standard statistical inference, informed by the concepts of random variables, sampling, estimation, and hypothesis testing to the Bridge tournament data.

2.2.2 Statistics and order statistics

Definition 1: Statistics

Let $(X_1, \ldots, X_n)$ be a sample drawn from a population, where X_i denotes the i-th observation in the sample. The population consists of all individuals or elements under study, whereas a sample is a finite subset of the population selected for analysis.
A statistic T is a numerical value computed from the sample data, represented as:

$$T = T(X_1, X_2, \ldots, X_n)$$

A statistic typically summarizes sample information, serving inferential or descriptive purposes. Common examples of statistics include the sample mean and sample median.
In many cases, a statistic is employed to estimate an unknown parameter of the population model. When used in this way, it is referred to as an estimator. Statistics are a function of the observed data and do not depend on any unknown parameters. However, when a statistic is specifically designed to estimate a population parameter, it is called an estimator.

Sample Statistics and Estimators

Let $X_1, X_2, \ldots, X_n$ be a random sample from a population with unknown mean μ and variance σ^2.

- **Sample mean**:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

  This is a **statistic**, as it is computed solely from observed data and does not involve any unknown parameters. It is commonly used as an **estimator** of the population mean μ.

- **Sample variance**:

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2$$

  This is also a **statistic**, derived entirely from the sample. It serves as an **estimator** of the population variance σ^2.

Definition 2: Order Statistics

Let $X_1, X_2, \cdots, X_n$ be independent and identically distributed random variables, representing observed values from the same population sample. Order statistics are obtained by arranging these random variables in ascending order. Specifically, we arrange the values in the sample from smallest to largest, resulting in the ordered random variables $X_{(1)}, X_{(2)}, \cdots, X_{(n)}$, which satisfy

$$X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$$

where $X_{(i)}$ represents the i-th smallest value in the sample.
Order statistics help describe the distribution characteristics of sample data, such as the minimum value $X_{(1)}$, the maximum value $X_{(n)}$, and the median. These statistics help better understand the overall trend and characteristics of the sample data.

In Bridge tournaments, order statistics, particularly the maximum and minimum values, are used to evaluate player performance. The maximum value $X_{(n)}$ helps assess peak performance, identifying top-performing players, while the minimum value $X_{(1)}$ highlights weaknesses. By combining these values, we can evaluate the stability of player performance. As order statistics, these values also reveal the distribution of player scores, such as the gap between maximum and minimum scores, which reflects the consistency of performance and competitive differences among players. Such analysis supports optimizing tournament rules, ensuring fairness and accuracy in evaluations. These applications provide a comprehensive understanding of player performance characteristics and support the development of more effective tournament strategies.

2.2.3 Likelihood function and maximum likelihood estimation (MLE)

Likelihood function is a fundamental statistical concept used to quantify how likely it is to observe given data for different parameter values of a statistical model. It is defined for a sample of parametrized random variables. It is the probability that the sample occurs, depending on the parameters. For instance, if a sample of ten coin tosses has seven heads, then the likelihood is $p^7(1-p)^3$ since the model is Ber(p). The MLE is the p that maximizes this function. For this particular example, p is equal to 0.7.
The likelihood function is used in Maximum Likelihood Estimation, which aims to find the parameter values that make the observed data most likely. In other words, we want to see the value of θ that maximizes $L(\theta)$.

Definition 3: Likelihood Function

For a set of observed data $X = (x_1, x_2, \ldots, x_n)$ and a set of unknown model parameters θ, the likelihood function $L(\theta)$ represents the probability of observing the data X given the parameters θ. It is usually defined as the joint probability density function or joint probability mass function of the observed data: $L(\theta) = f(X \mid \theta)$

This is one more MLE example of continuous probability. Consider a random variable $X \sim U(0, a)$, where a is unknown. The probability density function (PDF) is:

$$f(x) = \begin{cases} \frac{1}{a}, & 0 < x < a \\ 0, & \text{otherwise} \end{cases}.$$

Given n independent samples $X_1, X_2, \ldots, X_n$, the likelihood function is:

$$L(a) = \left(\frac{1}{a}\right)^n, \quad 0 < x_i < a, \forall i.$$

Maximizing $L(a)$ implies choosing the smallest valid a, which must satisfy $a \geq \max(x_1, \ldots, x_n)$. Thus, the maximum likelihood estimate is:$\hat{a} = \max(x_1, \ldots, x_n)$.

The likelihood function is used to estimate the skill parameters of Bridge players. Maximum Likelihood Estimates (MLE) are obtained by maximizing this likelihood function to quantify each player's skill level. The data from Bridge tournaments are treated as observations of players' relative skills, and the likelihood function evaluates the probability of these observations under a specified parameterized probability distribution, seeking the most suitable parameter estimates.
A parameterized probability distribution is a mathematical function that assigns probabilities to outcomes based on one or more parameters. In our context, these parameters represent the unknown skill levels of bridge pairs (denoted by θ or λ) and in the permutation model, a tie parameter ϕ. By varying these parameters, the distribution adjusts the probabilities of different outcomes, allowing the model to fit the observed data.
This parameterized distribution follows the Bradley–Terry model when ties are absent, and extends to the Davidson model (next section) when ties are considered. These models define the probability of one pair beating, losing to, or tying with another, based on their respective skill parameters.
MLE is widely used for skill quantification because it is asymptotically unbiased and efficient. It leverages all available data to produce reliable estimates of players' skill levels. Specifically, MLE establishes a relationship between model parameters and observed outcomes, identifying the values that make the observed data most likely, and thus providing a sound statistical basis for ranking and comparison.

2.2.4 Sufficient statistic

A sufficient statistic effectively summarizes all the information about the parameter θ present in the sample. Once the value of a sufficient statistic is known, the remaining data provide no additional information about the parameter. This concept is fundamental in statistical estimation as it allows the reduction of data without losing information about the parameter of interest.
We will see that MP scores are a sufficient statistic for Bridge players' skills, meaning they contain all the information necessary for skill inference without additional variables. This allows the model to infer skill parameters using only the total MP scores, simplifying the estimation process and ensuring statistical efficiency. In essence, MP scores enable direct and accurate skill comparisons while reducing the complexity of data processing.

Definition 4: Sufficient Statistic

Consider a random sample $X_1, X_2, \ldots, X_n$ drawn from a distribution that depends on an unknown parameter θ. A statistic $T = T(X_1, X_2, \ldots, X_n)$ is called a **sufficient statistic** for the parameter θ if, given the value of T, the conditional distribution of the sample $(X_1, X_2, \ldots, X_n)$ is independent of the parameter θ. In other words, T is sufficient for θ if

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n \mid T = t)$$

is independent of the parameter θ.

For example, let $X_1, X_2, \ldots, X_n$ be independent and identically distributed random variables from a distribution dependent on a parameter θ. The statistic T is considered sufficient if it contains all the information needed to estimate θ. This means that given T, the joint distribution of the sample no longer depends on θ, indicating that the data have been adequately summarized by T.

Example Under Poisson Distribution

Let $X_1, X_2, \ldots, X_n$ be independent and identically distributed (i.i.d.) random variables following a Poisson distribution $P(\lambda)$, i.e.:

$$p(X_i = x_i) = \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}.$$

Define the statistic $T = \sum_{i=1}^{n} X_i$ and determine if T is a sufficient statistic.

proof:

To show that T is a sufficient statistic, we first note that

$$P\big(T = t \;\big|\; X_1 = x_1, \ldots, X_n = x_n\big) \;=\; 1,$$

because once all X_i are specified, the value of T is completely determined. Hence, knowing T captures all information needed about the sum of the sample.
Next, we compute

$$P(X_1 = x_1, \ldots, X_n = x_n) \;=\; \prod_{i=1}^{n} \frac{\lambda^{x_i}\, e^{-\lambda}}{x_i!} \;=\; \frac{\lambda^{\sum_{i=1}^{n} x_i}\, e^{-n\lambda}}{\prod_{i=1}^{n} x_i!}.$$

Since $T = \sum_{i=1}^{n} X_i$, we know T follows a Poisson distribution with parameter $n\lambda$, so

$$P(T = t) \;=\; \frac{(n\lambda)^t\, e^{-n\lambda}}{t!}.$$

Using Bayes' formula:

$$P\big(X_1 = x_1, \ldots, X_n = x_n \;\big|\; T = t\big) = \frac{P(X_1 = x_1, \ldots, X_n = x_n) \;\cdot\; P(T = t)}{P(T = t)},$$

we substitute the above probabilities and canceling out $e^{-n\lambda}$:

$$P\big(X_1 = x_1, \ldots, X_n = x_n \;\big|\; T = t\big) = \frac{\frac{\lambda^{\sum x_i}\, e^{-n\lambda}}{\prod x_i!}}{\frac{(n\lambda)^t\, e^{-n\lambda}}{t!}} \;=\; \frac{\lambda^{\sum x_i}}{\prod x_i!}\,\frac{t!}{(n\lambda)^t}.$$

This expression no longer depends on λ. Therefore, T is a sufficient statistic.

□

The factorization theorem was first introduced by Fisher in the 1920s [11]. Later, Koopman (1936) [25] and Pitman (1936) [35] provided formal statements and proofs of the theorem.

Theorem 2.1: Factorization Theorem

Let $X_1, X_2, \ldots, X_n$ be a sample from a distribution family with joint probability density function (for continuous random variables) or joint probability mass function (for discrete random variables) $f(X_1, X_2, \ldots, X_n \mid \theta)$, where θ is an unknown parameter. A statistic $T = T(X_1, X_2, \ldots, X_n)$ is called a sufficient statistic for θ if the joint function can be factorized into two parts:

$$f(X_1, X_2, \ldots, X_n \mid \theta) = g(T(X_1, X_2, \ldots, X_n), \theta) \cdot h(X_1, X_2, \ldots, X_n)$$

where g is a non-negative function that depends on the sufficient statistic T and the parameter θ, and h is a non-negative function that does not depend on θ.
This means that all the information about the parameter θ is contained in the sufficient statistic T, thereby ensuring that T is sufficient.

Example of Continuous Function (Exponential Distribution)

Let X follow a generalized exponential distribution with the probability density function:

$$f(x) = m\,x^{m-1}\,\theta^{-m}\,e^{-(x/\theta)^m}, \quad x > 0, \theta > 0, m > 0.$$

Given a sample $X_1, X_2, \ldots, X_n$, we want to verify if the statistic

$$T = \sum_{i=1}^{n} X_i^m$$

Is a sufficient statistic.

proof:

Given a sample $X_1, X_2, \ldots, X_n$, the joint probability density function (or likelihood function) is:

$$L = \prod_{i=1}^{n} f(x_i) = \prod_{i=1}^{n} \left(m\,x_i^{m-1}\,\theta^{-m}\,e^{-(x_i/\theta)^m} \right).$$

This can be simplified as:

$$L = m^n \cdot \prod_{i=1}^{n} x_i^{m-1} \cdot \theta^{-nm} \cdot e^{-\sum_{i=1}^{n}(x_i/\theta)^m}.$$

To further simplify, we decompose it into:

$$L = \left(m^n \prod_{i=1}^{n} x_i^{m-1} \right) \cdot \left(\theta^{-nm}\,e^{-\frac{1}{\theta^m}\sum_{i=1}^{n} x_i^m} \right).$$

Applying the factorization theorem 2.1: To check for sufficiency, we need:

- A function $g(T\,|\,\theta)$ that depends only on T and θ.
- A function $h(x_1, \ldots, x_n)$ that does not involve θ.

Hence, we identify:

$$g(T\,|\,\theta) = \theta^{-nm}\,e^{-T\,/\,\theta^m}, \quad \text{where} \quad T = \sum_{i=1}^{n} x_i^m,$$

$$h(x_1, \ldots, x_n) = m^n \prod_{i=1}^{n} x_i^{m-1}.$$

Because g depends only on T and θ, and h does not involve θ, the factorization theorem 2.1 is satisfied. Therefore, $T = \sum_{i=1}^{n} x_i^m$ is a sufficient statistic for the parameter θ. □

Example of Discrete Function (Poisson Distribution)

In the previous Poisson example 2.2.4, we have a sample $X_1, X_2, \ldots, X_n$, each from a Poisson distribution with parameter λ. Its probability mass function is:

$$p(X_i = x_i \mid \lambda) = \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}, \quad x_i = 0, 1, 2, \ldots.$$

The joint probability mass function is:

$$p(X_1 = x_1, \ldots, X_n = x_n \mid \lambda) = \prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!} = \lambda^{\sum_{i=1}^{n} x_i} e^{-n\lambda} \prod_{i=1}^{n} \frac{1}{x_i!}.$$

Define the statistic $T = \sum_{i=1}^{n} X_i$. We can factorize the joint PMF as:

$$p(X_1, \ldots, X_n \mid \lambda) = \underbrace{\frac{\lambda^t e^{-n\lambda}}{t!}}_{g(T \mid \lambda)} \cdot \underbrace{\frac{t!}{\prod_{i=1}^{n} x_i!}}_{h(X_1, \ldots, X_n)},$$

where $t = T(X_1, \ldots, X_n) = \sum_{i=1}^{n} x_i$.
We see that $g(T \mid \lambda)$ depends only on T and λ, whereas h does not involve λ. Hence, by the factorization theorem 2.1 , $T = \sum_{i=1}^{n} X_i$ is a sufficient statistic for λ. □

2.2.5 Is the sample mean a sufficient statistic?

A further question we can ask is: **Is the sample mean a sufficient statistic?** In other words, does the sample mean contain all the information about the parameter, such that, given the sample mean, the original sample no longer contains additional information? We illustrate this by comparing two different distributions, Uniform and Poisson, to show how the same question can yield opposite answers depending on the model's structure.

Analysis of Uniform Distribution (the sample mean isn't a sufficient statistic)

Consider independent random variables $X_1, X_2, \ldots, X_n$ drawn from a uniform distribution $U(0, \theta)$, where θ is unknown. The probability density function (PDF) of each sample is:

$$f_X(x \mid \theta) = \begin{cases} \frac{1}{\theta}, & 0 \le x \le \theta, \\ 0, & \text{otherwise.} \end{cases}$$

The joint PDF is:

$$f(x_1, \ldots, x_n \mid \theta) = \begin{cases} \frac{1}{\theta^n}, & 0 \le x_i \le \theta \ (\forall i), \\ 0, & \text{otherwise.} \end{cases}$$

Define the sample maximum:

$$X_{(n)} = \max(X_1, \ldots, X_n).$$

Using the Factorization Theorem 2.1 , we write:

$$f(x_1, \ldots, x_n \mid \theta) = \underbrace{\tfrac{1}{\theta^n} I\big(0 \le X_{(n)} \le \theta\big)}_{g(X_{(n)} \mid \theta)} \cdot \underbrace{1}_{h(x_1, \ldots, x_n)},$$

showing that $X_{(n)}$ is a sufficient statistic for θ.

The sample mean is defined as:$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i$. From the perspective of minimal sufficiency, if $\bar{X}$ were sufficient, it would have to include the complete information of $X_{(n)}$. Since different samples with the same $\bar{X}$ can yield different $X_{(n)}$, we conclude that no alternative factorization exists that makes $\bar{X}$ a sufficient statistic.

Conclusion For $U(0, \theta)$, $X_{(n)}$ is a sufficient statistic, whereas $\bar{X}$ is not. Since sufficiency requires capturing all information about θ, and $X_{(n)}$ is minimally sufficient, any statistic failing to determine $X_{(n)}$ is not sufficient. The sample mean does not uniquely determine $X_{(n)}$. Hence, it is not a sufficient statistic.

Analysis of Poisson Distribution $P(\lambda)$ (the sample mean is a sufficient statistic)

Consider $X_1, \ldots, X_n$ i.i.d. from a Poisson(λ) distribution:

$$p(X_i = x_i) = \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}, \quad x_i = 0, 1, \ldots.$$

As shown previously, the statistic

$$T = \sum_{i=1}^{n} X_i$$

is sufficient for λ. For Poisson(λ), we also have

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} X_i = \frac{T}{n}.$$

Since T is sufficient, and $\bar{X}$ is just T multiplied by the constant $\frac{1}{n}$, $\bar{X}$ is likewise sufficient. That is, the sample mean for a Poisson distribution carries the same information about λ as the sum.

2.3 Accuracy of the estimation

2.3.1 Confidence interval

A confidence interval (CI) measures the uncertainty of an estimate through probability. It provides a range of possible values for a parameter at a given level of confidence $1-\alpha$, where α is the significance level. This indicates the likelihood that the confidence interval does not contain the proper overall parameter. In other words, the confidence level $1-\alpha$ suggests the probability that the actual value will fall into the interval in repeated sampling. For example, if the confidence level is 95%, then $\alpha = 0.05$, which means that the probability that the interval contains the overall mean is 95%. Confidence intervals give upper and lower bounds on the estimates and help in understanding the statistic's precision and the uncertainty caused by sampling error.
When estimating the overall variance, the chi-square distribution can describe the relationship between the sample variance and the overall variance. This is particularly useful when assessing overall variability and is applicable when dealing with variance inference problems. In Bridge tournament analyses, the chi-square distribution is used to deal with tournament tie scores to assess whether player skill differences are statistically significant. Using the chi-square distribution can better understand how tie scores affect overall inferences and ensure that conclusions are accurate and reliable.

Definition 5: Chi-Square Distribution and Confidence Interval (Confidence Interval for the Population Variance with Unknown Mean and Variance)

Consider a random sample from a normal distribution:

$$X_1, X_2, \ldots, X_n \sim N(\mu, \sigma^2),$$

where the population mean μ is unknown, and the population variance σ^2 is also unknown. Note that the assumption of normality is sufficient for the chi-square distribution to hold, regardless of the value of μ. That is, the population mean μ does not need to be zero for the chi-square distribution of the sample variance to be valid.
The relationship between the sample variance and the population variance can be described by the chi-square distribution:

$$\frac{(n-1)S^2}{\sigma^2} \sim \chi^2(n-1)$$

where $\chi^2(n-1)$ represents a chi-square distribution with $n-1$ degrees of freedom. The confidence interval for the population variance σ^2 at confidence level $1-\alpha$ is:

$$\left(\frac{(n-1)S^2}{\chi^2_{1-\alpha/2,n-1}}, \frac{(n-1)S^2}{\chi^2_{\alpha/2,n-1}}\right)$$

Where $\chi^2_{\alpha/2,n-1}$ and $\chi^2_{1-\alpha/2,n-1}$ are the quantiles of the chi-square distribution for the left-tail and right-tail cumulative probabilities, respectively.

2.3.2 Hypothesis testing for population mean

Hypothesis testing is used in statistics to determine whether sample data provide sufficient evidence to reject the null hypothesis about the overall parameters. It includes the null hypothesis (H_0) and the alternative hypothesis (H_1). The null hypothesis usually means no significant effect or difference like a population mean equal to a certain value ($\mu = \mu_0$). The alternative hypothesis suggests a significant effect or difference, like the mean not equal to that value ($\mu \neq \mu_0$). The goal is to use the data to decide if the null hypothesis should be rejected.
Hypothesis testing in Bridge tournament data is used to estimate Bridge players' skills. It helps compare players' abilities by clearly defining the null and alternative hypotheses. This allows us to see if there is a real difference in skills. Hypothesis testing and confidence intervals are useful in

statistics but have different goals. Hypothesis testing decides if there is a significant difference from a given value, while a confidence interval provides a range for the parameter. In practice, they work well together. For example, if the value in the null hypothesis is outside the confidence interval, it supports rejecting the null hypothesis. This approach makes the conclusions more reliable and supports better decision-making.

Lemma 1: Chi-Square Test for Population Variance(Testing the Population Variance with Unknown Population Mean)

Consider a hypothesis test for the population variance σ^2, where the population mean is unknown. We use the chi-square test statistic:

$$\chi^2 = \frac{(n-1)S^2}{\sigma_0^2} \sim \chi^2(n-1)$$

where S^2 is the sample variance, σ_0^2 is the hypothesized population variance, and $\chi^2(n-1)$ is the chi-square distribution with $n-1$ degrees of freedom.
For a two-tailed test with significance level α, the rejection region is defined by:

$$\chi^2 < \chi^2_{\alpha/2,n-1} \quad \text{or} \quad \chi^2 > \chi^2_{1-\alpha/2,n-1}$$

where $\chi^2_{\alpha/2,n-1}$ and $\chi^2_{1-\alpha/2,n-1}$ are the critical values from the chi-square distribution. If the calculated χ^2 falls in the rejection region, we reject the null hypothesis H_0.

Example of Chi-Square Test for Population Variance

Suppose we want to test the variance of a component's durability, with a sample size of $n = 9$, sample variance $S^2 = 0.007$, and hypothesized variance $\sigma_0^2 = 0.005^2$.

Solution:

State the Hypotheses:

$$H_0 : \sigma^2 = 0.005^2$$
$$H_1 : \sigma^2 \neq 0.005^2$$

Calculate the test statistic:

$$\chi^2 = \frac{(n-1)S^2}{\sigma_0^2} = \frac{8 \times 0.007^2}{0.005^2} = 15.68$$

Determine the rejection region for $\alpha = 0.05$:

$$\chi^2_{\alpha/2,8} = 17.53, \quad \chi^2_{1-\alpha/2,8} = 2.18 \quad \Rightarrow \quad \chi^2 < 2.18 \quad \text{or} \quad \chi^2 > 17.53$$

Since $2.18 < \chi^2 = 15.68 < 17.53$, we do not reject H_0. Thus, there is insufficient evidence to conclude that the population variance differs from 0.005^2.

2.3.3 Likelihood ratio test

In statistics, the likelihood ratio test (LRT) is a hypothesis testing method used to compare the fit of two competing statistical models [31]. Typically, one model is obtained by maximizing the likelihood function over the entire parameter space, while the other is obtained by imposing certain constraints. The test is based on the likelihood ratio of these two models. If the constrained model (i.e., the null hypothesis) fits the observed data well, the difference between the likelihood values of the two models

should not exceed the sampling error.

Definition 6: Likelihood Ratio Test

Suppose we have a statistical model with parameter space Θ. The null hypothesis typically states that the parameter θ lies within a specified subset Θ_0 of Θ, while the alternative hypothesis assumes that θ lies in the complement of Θ_0, denoted as Θ_0^c.
The likelihood ratio test statistic is defined as follows:

$$\lambda_{LR} = -2 \ln \left[\frac{\sup_{\theta \in \Theta_0} \mathcal{L}(\theta)}{\sup_{\theta \in \Theta} \mathcal{L}(\theta)} \right].$$

Here, the numerator and denominator represent the maximum values of the likelihood function under the null hypothesis and the full parameter space, respectively [45]. Since all likelihoods are positive and the constrained maximum cannot exceed the unconstrained maximum, the likelihood ratio is always between 0 and 1.
Typically, the likelihood ratio test statistic can also be expressed as a difference in log-likelihoods:

$$\lambda_{LR} = -2 \left[\ell(\theta_0) - \ell(\hat{\theta}) \right],$$

where $\ell(\hat{\theta}) = \ln \sup_{\theta \in \Theta} \mathcal{L}(\theta)$ is the log of the maximum likelihood function.

The Likelihood Ratio Test (LRT) is a widely used statistical method for model comparison, particularly in evaluating whether imposing constraints on a more complex model significantly affects its explanatory power. According to Wilks' theorem, under the null hypothesis, the likelihood ratio statistic λ_{LR} asymptotically follows a chi-square distribution, where the degrees of freedom correspond to the difference in the number of parameters between the competing models [45].
For the test to be valid, the models must be nested, meaning that the restricted model is obtained by applying additional constraints to the parameter space of the full model [30]. This ensures that the restricted model is a special case of the full model, allowing LRT to assess whether the imposed constraints significantly reduce the model's explanatory power. If the reduction in explanatory ability is statistically significant, the null hypothesis supporting the restricted model is rejected in favor of the more complex alternative.

Definition 7: Nested Models

A statistical model $\mathcal{M}_1$ is said to be nested within another model $\mathcal{M}_2$ if the parameter space of $\mathcal{M}_1$, denoted as Θ_0, is a subset of the parameter space of $\mathcal{M}_2$, denoted as Θ:

$$\Theta_0 \subseteq \Theta$$

This means that the simpler (restricted) model $\mathcal{M}_1$ can be obtained by applying additional constraints to the parameters of the more complex (full) model $\mathcal{M}_2$.

In the context of Bridge tournament analysis, we apply this concept to assess whether assuming a uniform skill level for all players is statistically justified. Specifically, we compare the following nested models:

- **Full Model:** Each player has an independently estimated skill parameter, allowing for variation across individuals.
- **Restricted Model:** All players share the same skill level, hence only one parameter for skill.

Since the restricted model can be derived from the full model by setting all individual skill parameters equal, it is a nested version of the full model. This property allows us to apply the Likelihood Ratio Test (LRT) to determine whether the assumption of uniform skill level significantly reduces the model's ability to explain match outcomes.

Example: Testing whether a coin is fair using the likelihood ratio test (LRT)

We aim to test if the probability of the coin landing on heads (p) is equal to 0.5 or if it differs from 0.5. A coin is flipped 10 times, and the number of heads observed is $X = 7$.

Under the null hypothesis (H_0), the coin is fair, i.e. the probability of landing on heads is $p = 0.5$. In this case, the data follow a binomial distribution:

$$X \sim \text{Binomial}(n = 10, p = 0.5).$$

Under the alternative hypothesis (H_1), the coin is not fair ($p \neq 0.5$), so the data still follow a binomial distribution but with p as an unknown parameter:

$$X \sim \text{Binomial}(n = 10, p).$$

To compute the maximum likelihood estimates:

- Under H_0, we fix $p = 0.5$.
- Under H_1, the MLE is $\hat{p} = X/n = 7/10 = 0.7$.

Log-likelihood under H_0:

$$\ell(H_0) = \ln\left(\binom{10}{7}\right) + 7\,\ln(0.5) + 3\,\ln(0.5).$$

A more precise calculation yields

$$\ell(H_0) = \ln(120) + 10\,\ln(0.5) \;\approx\; 4.7875 + 10 \times (-0.6931) \;=\; -2.144.$$

Log-likelihood under H_1:

$$\ell(H_1) = \ln\left(\binom{10}{7}\right) + 7\,\ln(0.7) + 3\,\ln(0.3).$$

Numerically, $\ln(0.7) \approx -0.3567$, $\ln(0.3) \approx -1.204$, so

$$\ell(H_1) \;\approx\; 4.7875 \;+\; 7 \times (-0.3567) \;+\; 3 \times (-1.204) \;=\; -1.321.$$

Likelihood Ratio Test statistic:

$$\lambda_{LR} = -2\Big[\ell(H_0) - \ell(H_1)\Big] = -2\Big[(-2.144) - (-1.321)\Big] = -2 \times \big(-0.823\big) = 1.646.$$

Under H_0, λ_{LR} asymptotically follows a χ^2 distribution with 1 degree of freedom. At the 5% significance level, the critical value is

$$\chi^2_{\text{critical}} = 3.841.$$

Since $\lambda_{LR} \approx 1.646 < 3.841$, we fail to reject H_0. Therefore, the observed data (7 heads out of 10) do not provide sufficient evidence to conclude that the coin is biased. □

2.4 Conclusion

This section outlined the essential statistical and mathematical concepts foundational to our subsequent analysis of Bridge tournaments. We began with fundamental notions from statistics (section 2.1), reviewing core concepts such as cumulative distribution functions (CDF) and probability density functions (PDF), emphasizing their applications in describing distributions and deriving relevant statistics. This groundwork is critical, particularly as we introduce advanced statistical methods such as permutation models and maximum likelihood estimation (MLE).
Subsequently, we discussed applying these general statistical concepts within the context of Bridge tournaments (section 2.2). We specifically explored how match-point (MP) scores function as a vital statistic summarizing player performance, providing a quantitative measure from which skill inferences can be made. Understanding MP scores as a statistic is instrumental when formalising the permutation model in subsequent sections.
Moreover, we examined order statistics, highlighting their significance in analyzing tournament player performance. By employing maximum and minimum order statistics, we can assess peak performances and consistency among players, which are critical aspects of designing fair and accurate competitive assessments.
We also provided an in-depth review of the likelihood function and maximum likelihood estimation, demonstrating their effective role in quantifying player skills. This methodology is central to subsequent analyses as it provides a robust statistical framework for accurately estimating individual player abilities based on tournament data.
Furthermore, sufficient statistics were introduced, emphasizing how certain statistics, like total MP scores, encapsulate all necessary information about player skills. Recognizing sufficient statistics simplifies the estimation process and enhances computational efficiency, a valuable property utilized extensively in practical Bridge data analysis.
Finally, we addressed statistical inference techniques, including confidence intervals, hypothesis testing, and the likelihood ratio test (section 2.3), underscoring their utility in evaluating the statistical significance of observed data and differences in player performance. These techniques provide critical tools for rigorously evaluating and comparing Bridge tournament outcomes.
In summary, this section established the theoretical statistical foundation necessary for the robust analysis of Bridge tournaments. Having established the statistical foundation, the following section will present a concrete framework for analyzing duplicate Bridge tournament data. We will define the scoring system, discuss pairwise comparisons, and set the stage for modeling player skills using likelihood-based approaches.

3 Scores, rankings, and strengths of Bridge pairs

3.1 Scoring system in a duplicate Bridge tournament

The scoring system in a duplicate Bridge tournament aims to assess each pair's skills fairly by ensuring that multiple pairs across different tables play each board. Match points (MP) help rank pairs effectively while eliminating the "deal effect" and providing a level playing field. The final score calculation and ranking based on accumulated match points allow an accurate comparison of the pairs' relative skills in the tournament.

3.1.1 Symbols and N/S and E/W pair assignments

- n: Number of participating pairs, ranging from 1 to n.
- B: Number of boards in the tournament, ranging from 1 to B.
- T: Number of tables, from 1 to T.

In a duplicate Bridge tournament, each board is played at multiple tables to ensure fairness by eliminating the luck of the deal. Symbols α and β represent the North/South (N/S) and East/West (E/W) pairs, respectively. These symbols α and β are mappings from $\{1,\ldots,B\} \times \{1,\ldots,T\}$ into $\{1,2,\ldots,n\}$. This means that for each board and table combination, α and β help determine which pairs are playing against each other on any given board.
The concept of 'boards' refers to a set of boards that appear several times in a tournament at different tables. The boards are pre-determined before the start of the tournament, and it is also specified which players will play against each other at each table for each table of boards. This arrangement can be understood as a 'matching' problem in combinatorial mathematics, similar to the concept found in graph theory, where teams are systematically paired to ensure fairness.

- $\alpha(b,t)$: denotes the pairs of south/north orientated players on board b and table t.
- $\beta(b,t)$: denotes the pairs of East/West pair on board b and table t.

Each board b can appear on different tables t in different tournament rounds. It is essential to distinguish between the terms 'table' and 'round'. A 'table' refers to the physical location where the boards are played. At the same time, a round indicates the order in which the boards are played in a tournament, which allows each pair of players to play multiple tables of boards against different opponents.

3.1.2 Scoring system and final score calculation

The ranking method is used to make fair comparisons between pairs of players holding similar boards, thus eliminating the effects of randomly dealt distributions.

- S_{bt}: the raw score obtained by the North/South player pair $\alpha(b,t)$ on board b and table t.
- The corresponding East/West player's score for $\beta(b,t)$ is $-S_{bt}$.
- R_{bt}: the ranking of the score S_{bt} among all the North/South scores of the board b.This value is between 0 and $T-1$. The E/W pair at the same table receives a score of $T-1-R_{bt}$.

Match points are awarded based on the relative performance of each pair compared to others playing the same board. The final score of pair i, denoted by r_i, is the sum of all match points earned by the pair over all boards played. The formula for calculating r_i is:

$$r_i = \sum_{\substack{(b,t) \\ i=\alpha(b,t)}} R_{bt} + \sum_{\substack{(b,t) \\ i=\beta(b,t)}} (T-1-R_{bt}) \tag{1}$$

Where (b, t) are the boards and tables where pair i played, either as an N/S or E/W pair, the two components of this sum represent the match points obtained when playing in the N/S or E/W positions, respectively. For any fixed board b, pair i participates only once, either in the N/S or E/W position. This ensures that each participant competes fairly in the tournament.

Example of Scoring

Consider an example with five N/S pairs at a particular board, with raw scores of $+420$, $+450$, $+420$, -100, and -100. The corresponding match points are 2.5, 4, 2.5, 0.5, and 0.5 respectively, based on their ranking among all N/S pairs. The E/W pairs receive complementary scores such that the total points for each table are $T - 1$.

3.1.3 Determining the winner

The pair with the highest total match-points r_i is declared the tournament winner. The maximum possible match-points for a pair is $m(T-1)$, where m is the number of boards the pair plays. A summarized score can be expressed as a percentage:

$$\left(\frac{r_i}{m(T-1)}\right) \times 100 \tag{2}$$

This scoring system ensures that the final rankings reflect the pairs' relative skills and exclude the influence of luck.

3.2 Permutation model

The permutation model is a statistical model for analysing rankings by comparing the relative performance between multiple groups to estimate their intrinsic ability. The model assumes that each group has a fixed parameter of ability, which is assessed through repeated comparisons. This eliminates the influence of random factors and obtains a more accurate estimate of ability. The permutation model is particularly well-suited to scenarios involving repeated comparisons and can be extended to deal with ties and other special situations.
In duplicate Bridge tournaments, permutation models are used to analyze the match point (MP) scores of Bridge players to assess their intrinsic skill levels. The performance of each pair of players on the same deck is affected by the skill of their opponents and random factors. By comparing the scores of all players on the same deck, the permutation model relates the game's outcome to the skill of each pair, thus eliminating 'dealing effects' and allowing for fair comparisons. This section describes the model's assumptions and methodology and how it can be used to infer the skills and rankings of Bridge players.

3.2.1 Likelihood function and model to link probabilities to player skills

We use the vector $R_b = (R_{b1}, \ldots, R_{bT})$ to denote the MP scores for each pair of players on board b. These scores have been defined in the previous section. The likelihood function for the entire game can be expressed as the product of the scores for each board:

$$P(R_1, R_2, \ldots, R_B) = \prod_{b=1}^{B} P(R_b). \tag{3}$$

Here, we assume all boards' scores are independent, simplifying the model's analysis and statistical treatment. However, this assumption of independence does not always hold true in actual Bridge games. Bridge is a multifaceted intellectual game, and each player's performance is affected not only by intrinsic skills but also by psychological states, strategic decisions, and external disturbances. These factors include mood swings and strategic adjustments. For example, when a player performs poorly on a particular board, they may be affected by negative emotions in subsequent boards,

resulting in more aggressive or conservative strategies to compensate for previous mistakes or to reduce risk. These strategic choices affect players' performance on a single board and trigger a chain reaction throughout the game, leading to score correlations across boards.
Therefore, although we usually assume independence in our modeling, the correlation between different boards cannot be ignored in real tournaments. In the subsequent sections related to the analysis of actual tournament data, we will discuss the impact of this non-independence in more detail, analyzing how players adjusted their strategies based on previous results and how these adjustments affected the overall score distribution and tournament results.
The goal of the permutation model is to link the set of scores $P(R_b)$ of board b to the skill levels of the participating pairs. For each board, pairs are represented by $\alpha(b,1),\ldots,\alpha(b,T)$, with pair $\alpha(b,t)$ sharing the MP scores with pair $\beta(b,t)$. The MP score obtained by pair $\alpha(b,t)$ depends on the difference in skill between $\alpha(b,t)$ and $\beta(b,t)$.
Each pair is assumed to have an intrinsic skill value θ_i, and the score of pair $\alpha(b,t)$ depends on the skill difference between $\alpha(b,t)$ and $\beta(b,t)$, specifically given by:

$$\theta_{\alpha(b,t)} - \theta_{\beta(b,t)}. \tag{4}$$

This assumption implies that a pair with a higher skill value is likelier to outperform those with a lower skill value. It should be noted that if θ is a parameter from a continuous distribution, the probability of a tie is zero, which should be considered when applying the model.

3.2.2 Pairwise comparisons and likelihood function

Notation Clarification:
(1) b denotes the board index (i.e., the deal number).
(2) i, j represent the pair indices, ranging from 1 to T, where T is the total number of pairs.
(3) $R_{b,i}$ is a random variable for the score (or rank) of pair i on board b.
(4) $\alpha(b,i)$ **and** $\beta(b,i)$ stand for the skill parameters associated with pair i on board b, for its two seats (e.g., North-South or East-West). If needed, they can be interpreted as the skill difference between the two players within a pair. Note that we do not explicitly include the table index t here, because we focus on the comparisons within the same board b.

For board b, suppose we observe that pair i scores lower than pair j. We write this event as

$$\alpha(b,i) \text{ loses to } \alpha(b,j),$$

and collect all such pairwise comparisons into the set

$$L_b = \{(i,j) : \alpha(b,i) \text{ loses to } \alpha(b,j)\}. \tag{5}$$

Babington Smith (1950) assumed that the permutation model satisfies [3]:

$$P(R_b) \quad \propto \prod_{(i,j)\in L_b} P\big(R_{b,i} < R_{b,j}\big), \tag{6}$$

meaning that the overall score distribution on board b is determined by the pairwise outcomes among all pairs. In other words, if we know which pair beats which on each board, we can reconstruct the final ranking for that board.
Furthermore, under the Bradley–Terry model [4],

$$P\big(R_{b,i} < R_{b,j}\big) \;=\; F\Big(\big[\theta_{\alpha(b,j)} \;-\; \theta_{\beta(b,j)}\big] \;-\; \big[\theta_{\alpha(b,i)} \;-\; \theta_{\beta(b,i)}\big]\Big), \tag{7}$$

where $F(\cdot)$ is the CDF of a known distribution. Intuitively, if the skill difference in pair j is higher than that in pair i, pair j is more likely to achieve a higher score on that board.

To understand $P(R_{b,i} < R_{b,j})$ more concretely, consider that each pair's performance can be viewed as a random draw (e.g., X for pair i and Y for pair j) from distributions shaped by their skill levels.

The probability $P(X < Y)$ can be expressed using the cumulative distribution function $F(\cdot)$, which incorporates both the skill gap and inherent randomness. Consequently, a larger skill difference for pair j yields a higher probability of j surpassing i in board b.

For further mathematical details and formal derivations of this model, we refer the reader to the references [34].

3.2.3 Bradley-Terry model application

From the Bradley-Terry model [4], $F(x)$ is assumed to follow a logistic distribution, which takes the form:

$$F(x) = \frac{1}{1 + \exp(-x)}, \quad -\infty < x < \infty, \tag{8}$$

Although the functional form of $F(x)$ is fixed, the input x itself depends on unknown skill parameters. Modeling the probability of a pair winning in the interval $[0, 1]$. Suppose that

$$\lambda_{bt} = \exp[\theta_{\alpha(b,t)} - \theta_{\beta(b,t)}], \tag{9}$$

Here, $\theta_{\alpha(b,t)}$ and $\theta_{\beta(b,t)}$ are the parameters to be estimated via maximum likelihood, thereby making the distribution parameterized. Then, the expression for the pairwise comparison probability can be simplified as:

$$P(R_{bi} < R_{bj}) = \frac{\lambda_{bj}}{\lambda_{bi} + \lambda_{bj}}, \tag{10}$$

indicating that the model can effectively describe the skill dynamics between pairs playing the same board. Mathematical details refer to Yu and Lam [34].

Proof of equation (10) simplification for pairwise probability

Given the pairwise probability expression from the Bradley-Terry model equation (7):

$$P(R_{bi} < R_{bj}) = F\left([\theta_{\alpha(b,j)} - \theta_{\beta(b,j)}] - [\theta_{\alpha(b,i)} - \theta_{\beta(b,i)}]\right),$$

where $F(\cdot)$ is the CDF of a logistic distribution, expressed as equation (8):

$$F(x) = \frac{1}{1+\exp(-x)}, \quad -\infty < x < \infty,$$

We know the skill-based strength of a pair on board b and table t as:

$$\lambda_{bt} = \exp[\theta_{\alpha(b,t)} - \theta_{\beta(b,t)}],$$

To compare two pairs i and j on board b, we assume they are both North/South (or both East/West), and we rank their scores relative to other pairs playing the same role on the same board. In section 3.2.4, it will be explained why this formula can be used for Bridge. In this case, their performance depends not only on their own skill but also on the strength of their opponents. Let t_i and t_j denote the tables where pairs i and j played board b, and let $\text{opp}(i) = \beta(b, t_i)$, $\text{opp}(j) = \beta(b, t_j)$ be their respective opponents. We define the skill strength of each pair in its match as:

$$\lambda_{bi} = \exp[\theta_i - \theta_{\text{opp}(i)}], \quad \lambda_{bj} = \exp[\theta_j - \theta_{\text{opp}(j)}],$$

Substituting into the original formula, we have:

$$P(R_{bi} < R_{bj}) = F\left(\theta_j - \theta_{\text{opp}(j)} - (\theta_i - \theta_{\text{opp}(i)})\right) = F\left((\theta_j - \theta_i) - (\theta_{\text{opp}(j)} - \theta_{\text{opp}(i)})\right).$$

This expression still depends on the skill difference between the opponents. In practical applications, to simplify the model and reduce variance from unknown or unbalanced opponent strength, we assume that opponent skill differences average out. leading to:

$$P(R_{bi} < R_{bj}) = F\left(\log(\lambda_{bj}) - \log(\lambda_{bi})\right) = F(\theta_j - \theta_i).$$

Since the logistic cdf function $F(x)$ is used, we have:

$$P(R_{bi} < R_{bj}) = \frac{1}{1+\exp\left(-(\theta_j - \theta_i)\right)}.$$

Simplifying the exponential term:

$$\exp\left(-(\theta_j - \theta_i)\right) = \frac{\exp(\theta_i)}{\exp(\theta_j)} = \frac{\lambda_{bi}}{\lambda_{bj}},$$

we substitute back to get:

$$P(R_{bi} < R_{bj}) = \frac{1}{1+\frac{\lambda_{bi}}{\lambda_{bj}}} = \frac{\lambda_{bj}}{\lambda_{bi}+\lambda_{bj}},$$

which matches the simplified form:

$$P(R_{bi} < R_{bj}) = \frac{\lambda_{bj}}{\lambda_{bi}+\lambda_{bj}}.$$

This derivation shows the connection between the full opponent-aware formulation and the simplified expression commonly used for maximum likelihood estimation.
The simplified form of the pairwise probability (10) is derived by applying the logistic distribution as the cumulative distribution function and substituting the skill differences.

□

3.2.4 Model justification for Bridge data analysis

This section explains why the permutation model is suitable for Bridge tournament data. The model satisfies three conditions: the Reversibility Condition, the Skill Parameter Explanation, and the Sufficiency of Total Match Points.

i. Reversibility condition

The reversibility condition implies that the rank order of the N/S pairs, denoted by R_b, can be reflected in the rank order of the E/W pairs, denoted by R'_b. From Critchlow et al.(1991) [9], we have:

$$R'_b = T - 1 - R_b, \tag{11}$$

where the model for R_b is based on the skill differences $\theta_{\alpha(b,t)} - \theta_{\beta(b,t)}$, while the model for R'_b is based on the skill differences $\theta_{\beta(b,t)} - \theta_{\alpha(b,t)}$.
The reversibility condition means that if we swap South/North player pairs and East/West player pairs, the original scoring order should be reflected symmetrically. For example, if a particular N/S pair is ranked first among all players, the corresponding E/W pair should be ranked last after the exchange. The essence of this condition lies in the symmetry of the Bridge game rules, i.e., the N/S and E/W pairs face the same conditions, so the relative skill differences between the two sides should remain consistent.
To better understand this condition, we can think of it as a 'mirror relationship'. When looking at the entire match, the relative strengths between the N/S and E/W and East/West pairs should remain the same after the exchange. This is the core meaning of the reversibility condition in the model and serves as the basis for deriving the ranking relationship between the N/S and E/W pairs.

ii. Skill parameters explanation

The skill parameter θ indicates the intrinsic skill level of a Bridge pair. When considering rankings in a Bridge tournament, assume that two pairs of players $\alpha(b,i)$ and $\alpha(b,j)$ have similar skill levels.
This comparison relies on the assumption that both pairs have played against opponents of comparable skill levels. Under this assumption, the difference in outcomes between the two pairs can be reasonably attributed to their intrinsic skill parameters.
When comparing these two pairs of players, we can use the probability ratio of their rankings as a measure of skill difference. The ranked probability ratio is expressed as follows:

$$\frac{P(\cdots < R_{bj} < R_{bi} < \cdots)}{P(\cdots < R_{bi} < R_{bj} < \cdots)} = \exp[\theta_{\alpha(b,i)} - \theta_{\alpha(b,j)}]. \tag{12}$$

The formula shows that when the neighbour rankings of two pairs of players are exchanged, the probability ratio of these two rankings is represented by the difference between their θ values. The difference in θ values can be interpreted as the logarithm of the two probability ratio values. In other words, if the θ value is larger, the corresponding Bridge pair is more likely to be ranked ahead of the other pair, and vice versa. This interpretation makes skill differences between Bridge player pairs more intuitive and allows us to quantify relative skill levels when assessing their performance. For a more detailed description, see Yu and Lam(1996) [34].

Lemma 2: Skill Parameters Representation

Assuming that the two pairs face opponents of similar overall skill, the difference in skill parameters between two pairs, $\alpha(b,i)$ and $\alpha(b,j)$, can be expressed as the logarithm of the odds ratio of their ranking probabilities. Specifically, we have:

$$\theta_{\alpha(b,i)} - \theta_{\alpha(b,j)} = \log\left(\frac{P(R_{bj} < R_{bi})}{P(R_{bi} < R_{bj})}\right).$$

Proof of skill parameters explanation

To prove this lemma, let $P(\cdots < R_{bj} < R_{bi} < \cdots)$ denote the probability that pair $\alpha(b,j)$ ranks ahead of pair $\alpha(b,i)$. Suppose that the pairs $\beta(b,i)$ and $\beta(b,j)$ are equally skillful, i.e., $\theta_{\beta(b,i)} = \theta_{\beta(b,j)}$. From equation (9), we define λ_{bt} as follows:

$$\lambda_{bt} = \exp[\theta_{\alpha(b,t)} - \theta_{\beta(b,t)}], \quad t = 1, \ldots, T.$$

Using equation (10), the expression for the pairwise comparison probability can be simplified as:

$$P(R_{bi} < R_{bj}) = \frac{\lambda_{bj}}{\lambda_{bi} + \lambda_{bj}}.$$

Similarly, the probability that pair $\alpha(b,j)$ ranks higher than pair $\alpha(b,i)$ is given by:

$$P(R_{bj} < R_{bi}) = \frac{\lambda_{bi}}{\lambda_{bi} + \lambda_{bj}}.$$

To proceed, we first express the ratio of λ_{bj} to λ_{bi} using their definitions:

$$\lambda_{bi} = \exp(\theta_{\alpha(b,i)} - \theta_{\beta(b,i)}), \quad \lambda_{bj} = \exp(\theta_{\alpha(b,j)} - \theta_{\beta(b,j)}).$$

The ratio can then be written as:

$$\frac{\lambda_{bi}}{\lambda_{bj}} = \exp((\theta_{\alpha(b,i)} - \theta_{\beta(b,i)}) - (\theta_{\alpha(b,j)} - \theta_{\beta(b,j)})),$$

where $\theta_{\beta(b,i)} = \theta_{\beta(b,j)}$.

Now, consider the ratio of the two probabilities:

$$\frac{P(R_{bj} < R_{bi})}{P(R_{bi} < R_{bj})} = \frac{\lambda_{bi}}{\lambda_{bj}} = \exp(\theta_{\alpha(b,i)} - \theta_{\alpha(b,j)}).$$

This shows that the difference in skill parameters between pairs $\alpha(b,i)$ and $\alpha(b,j)$ is represented by the logarithm of the odds ratio of their ranking probabilities.
Thus, the skill difference between pairs $\alpha(b,i)$ and $\alpha(b,j)$ can be interpreted as:

$$\theta_{\alpha(b,i)} - \theta_{\alpha(b,j)} = \log\left(\frac{P(R_{bj} < R_{bi})}{P(R_{bi} < R_{bj})}\right).$$

In other words, the skill difference is represented by the log ratio of the two probabilities, indicating the log odds of one pair ranking ahead of the other. □

iii. Sufficiency of total match Points

The total MP scores obtained by the pairs are sufficient statistics for the skill parameters. This means the skill parameters can be effectively estimated using the total MP scores, summarized as sufficient statistics. Critchlow and Fligner (1993) discuss this property in detail [8]. The proof of this result is provided in Section 3.3.

3.3 Permutation model in the presence of ties

In this section, the permutation model used to analyse Bridge game scores is extended to deal with the case of tied scores. This section discusses the modifications needed to more realistically analyse match point (MP) data, considering that tie scores occur frequently in actual Bridge tournaments. To incorporate tie scores, two sets are defined for pairwise comparisons between players:

- **Set** L_b: denotes the pair in which a player $\alpha(b,i)$ loses to player $\alpha(b,j)$. This set is denoted as:

$$L_b = \{(i,j) : \alpha(b,i) \text{ loses to } \alpha(b,j)\}$$

- **Set** D_b: denotes the pair in which a player $\alpha(b,i)$ draws with player $\alpha(b,j)$. This is defined as:

$$D_b = \{(i,j) : \alpha(b,i) \text{ draws with } \alpha(b,j), i < j\} \tag{13}$$

The probability $P(R_b)$ for a given ranking is calculated by considering both the losses and ties:

$$P(R_b) = P(R_{b1}, \cdots, R_{bT}) \propto \prod_{(i,j)\in L_b} P(R_{bi} < R_{bj}) \prod_{(i,j)\in D_b} P(R_{bi} = R_{bj}) \tag{14}$$

This expression takes into account both the comparison of one player's rankings over another and the occurrence of a tie.

3.3.1 Probability of pairwise comparisons in the case of a specified tie

When modeling pairwise comparison probabilities in Bridge tournaments, the traditional logistic distribution model (such as the Bradley-Terry model) performs well in describing win-loss situations but fails to effectively handle ties. Since logistic distribution is continuous when two teams have equal strength parameters, the probability of a tie is zero. However, in actual Bridge matches, ties are not uncommon, especially when two teams have similar strengths. Therefore, relying solely on logistic distribution is insufficient for accurately reflecting all possible outcomes in real matches.
In the model, we refer to the method proposed by Davidson (1970) and introduce a parameter $\phi > 0$ to handle the possibility of ties [10]. The sum of the probabilities of all outcomes is equal to 1:

$$P(R_{bi} < R_{bj}) + P(R_{bi} > R_{bj}) + P(R_{bi} = R_{bj}) = 1 \tag{15}$$

Specifically, the probabilities of winning, losing, and tying are calculated using the following formulas:

- **Probability of Losing**:

$$P(R_{bi} < R_{bj}) = \frac{\lambda_{bj}}{\lambda_{bi} + \lambda_{bj} + \phi\sqrt{\lambda_{bi}\lambda_{bj}}}$$

- **Probability of Winning**:

$$P(R_{bi} > R_{bj}) = \frac{\lambda_{bi}}{\lambda_{bi} + \lambda_{bj} + \phi\sqrt{\lambda_{bi}\lambda_{bj}}}$$

- **Probability of Tying**:

$$P(R_{bi} = R_{bj}) = \frac{\phi\sqrt{\lambda_{bi}\lambda_{bj}}}{\lambda_{bi} + \lambda_{bj} + \phi\sqrt{\lambda_{bi}\lambda_{bj}}} \tag{16}$$

In these formulas, the introduction of ϕ is key. By adjusting the value of ϕ, we can control the probability of ties:

- When $\phi = 0$, the model degenerates to the traditional logistic distribution model, and the probability of a tie is zero.
- When $\phi > 0$, the probability of a tie becomes positive, allowing the model to capture situations where the two teams are equally matched.

In summary, by introducing discrete components to handle ties, the new model effectively addresses the shortcomings of the logistic distribution model, making it more consistent with the needs of real matches, particularly in contract Bridge, where ties are not uncommon.

3.3.2 Sufficient Statistics analysis for skill parameters

Theorem 3.1 forms a cornerstone in our statistical inference framework. It states that the total match points r_i for each pair (i.e., the sum of MP scores across all boards) and the total number of tied pairs $\sum_{b=1}^{B} d_b$ jointly constitute sufficient statistics for the skill parameters $\{\theta_i\}$ and the tie parameter ϕ. By relying solely on these summaries, we can estimate $(\boldsymbol{\theta}, \phi)$ without retaining the full set of per-board configurations, greatly simplifying likelihood-based inference.

Theorem 3.1: Sufficient Statistics for skill parameters

In duplicate Bridge tournaments, the total match points (MPs) obtained by each pair of players, denoted as $\{r_i\}$, along with the total number of tied pairs across all rounds, denoted as $\sum_{b=1}^{B} d_b$, are sufficient statistics for estimating both the skill parameters $\{\theta_i\}$ and the tie parameter ϕ.

Based on the permutation model with ties proposed by Yu and Lam (1996), using equations (14) and (16), the likelihood function for the observed outcomes across all B boards is given by:

$$P(R_1, \ldots, R_B) = \prod_{b=1}^{B} \frac{\phi^{d_b} \lambda_{b1}^{R_{b1}} \cdots \lambda_{bT}^{R_{bT}}}{C_b(\theta, \phi)} \tag{17}$$

- ϕ^{d_b}: The impact of ties on board b, where d_b is the number of tied pairs. The more ties observed, the greater the weight of the tie parameter ϕ in the likelihood.
- $\lambda_{bt}^{R_{bt}}$: The contribution of pair t on board b. Here, $\lambda_{bt} = \exp(\theta_{\alpha(b,t)} - \theta_{\beta(b,t)})$ captures the relative skill between the North/South and East/West pairs at table t. The exponent R_{bt} denotes how many other pairs the pair at table t beat on board b (i.e., their number of pairwise victories).
- $C_b(\boldsymbol{\theta}, \phi)$ is a normalizing constant ensuring the probabilities over all configurations of board b sum to 1.

Proof of theorem 3.1

We have B boards and assume equation (17) holds for the observed data $\mathbf{R} = (R_1, \ldots, R_B)$. Thus,

$$P(\mathbf{R} \mid \boldsymbol{\theta}, \phi) = \prod_{b=1}^{B} \frac{\phi^{d_b} \prod_{t=1}^{T} \lambda_{bt}^{R_{bt}}}{C_b(\boldsymbol{\theta}, \phi)},$$

with $\lambda_{bt} = \exp(\theta_{\alpha(b,t)} - \theta_{\beta(b,t)})$. Rewriting:

$$P(\mathbf{R} \mid \boldsymbol{\theta}, \phi) = \prod_{b=1}^{B} \frac{\phi^{d_b}}{C_b(\boldsymbol{\theta}, \phi)} \exp\Big[\sum_{t=1}^{T} \big(\theta_{\alpha(b,t)} - \theta_{\beta(b,t)}\big) R_{bt}\Big].$$

Expanding the exponent:

$$P(\mathbf{R} \mid \boldsymbol{\theta}, \phi) = \prod_{b=1}^{B} \frac{\phi^{d_b}}{C_b(\boldsymbol{\theta}, \phi)} \exp\left(\sum_{t=1}^{T} \left[\theta_{\alpha(b,t)} R_{bt} + \theta_{\beta(b,t)} (T - 1 - R_{bt})\right]\right)$$

Further combining the exponential terms:

$$P(\mathbf{R} \mid \boldsymbol{\theta}, \phi) = \phi^{\sum_{b=1}^{B} d_b} \exp\left(\sum_{t=1}^{T} \sum_{b=1}^{B} \left[\theta_{\alpha(b,t)} R_{bt} + \theta_{\beta(b,t)} (T - 1 - R_{bt})\right]\right) \prod_{b=1}^{B} C_b(\boldsymbol{\theta}, \phi)^{-1}$$

To simplify $\exp\left(\sum_{t=1}^{T} \sum_{b=1}^{B} \left[\theta_{\alpha(b,t)} R_{bt} + \theta_{\beta(b,t)} (T - 1 - R_{bt})\right]\right)$, we define r_i as the total accumulated contribution(or total MP scores) for player i across all boards and tables, like

equation (1). After changing, add each table and each board to sum up the total contribution for each player. From Yu and Lam, the result is a single summation of all players:

$$\exp\left(\sum_{i=1}^{n} \theta_i r_i\right)$$

.
Define $C(\boldsymbol{\theta}, \phi) = \prod_{b=1}^{B} C_b(\boldsymbol{\theta}, \phi)$. Thus,

$$P(\mathbf{R} \mid \boldsymbol{\theta}, \phi) = \frac{\phi^{\sum_{b=1}^{B} d_b} \exp\left(\sum_{i=1}^{n} \theta_i r_i\right)}{C(\boldsymbol{\theta}, \phi)}.$$

To identify sufficient statistics for $(\boldsymbol{\theta}, \phi)$, we invoke the Factorization Theorem 2.1, which states

$$P(\mathbf{R} \mid \boldsymbol{\theta}, \phi) = g\big(\mathbf{S}(\mathbf{R}), \boldsymbol{\theta}, \phi\big) \times h(\mathbf{R}),$$

where g depends on $\mathbf{R}$ only through some statistic $\mathbf{S}(\mathbf{R})$, and $h(\mathbf{R})$ does not depend on $(\boldsymbol{\theta}, \phi)$. In our case,

$$P(\mathbf{R} \mid \boldsymbol{\theta}, \phi) = \underbrace{\phi^{\sum_{b=1}^{B} d_b} \exp\Big(\sum_{i=1}^{n} \theta_i r_i\Big)}_{g(\mathbf{S}(\mathbf{R}), \boldsymbol{\theta}, \phi)} \times \underbrace{\frac{1}{C(\boldsymbol{\theta}, \phi)}}_{h(\mathbf{R})},$$

and the data enter $g(\cdot)$ only through

$$\mathbf{S}(\mathbf{R}) = \Big(r_1, r_2, \ldots, r_n, \sum_{b=1}^{B} d_b\Big).$$

Since $C(\boldsymbol{\theta}, \phi)$ is a normalizing constant that does not depend on the specific realization $\mathbf{R}$ beyond these sufficient statistics, it follows that $\{r_i\}$ and $\sum_{b=1}^{B} d_b$ capture all the information about $(\boldsymbol{\theta}, \phi)$. Hence, by the Factorization Theorem,

$$\Big(r_1, \ldots, r_n, \sum_{b=1}^{B} d_b\Big)$$

is a sufficient statistic for $(\boldsymbol{\theta}, \phi)$.

□

3.4 Selecting the best pair: confidence subset approach

In competitive bridge tournaments, it is often of interest not only to estimate each pair's skill parameter θ_i, but also to identify the best-performing pair—the one with the highest true skill level. A natural approach is to rank the maximum likelihood estimates $\hat{\theta}_i$ and declare the top-ranked pair as the best. When the datas involve sufficient comparisons across all pairs, this approach can be reliable due to the high connectivity among players and dense match structure. It is often the case in small-scale tournaments using Howell movements.
However, in large-scale tournaments or in designs such as the Mitchell movement, structural limitations make accurate comparison more difficult. In these formats, not all pairs compete against each other directly, and different pairs may play different subsets of boards. As a result, the available information for pairwise comparisons is incomplete and unbalanced, making it difficult to fairly assess relative skill levels across the entire pool of participants. In such settings, relying solely on point estimates $\hat{\theta}_i$ may lead to overconfident or misleading conclusions.

To solve this problem, Yu and Lam (1996) proposed a *confidence subset* method under a permutation model framework. Their approach constructs a subset $S \subseteq \{1, \ldots, n\}$ such that, with high confidence,

the truly best pair lies within S:

$$P\big((n) \in S\big) \geq 1 - \alpha,$$

where (n) denotes the index of the pair with the highest skill parameter.
The method is derived under large-sample asymptotics, using simultaneous confidence bounds on pairwise differences $\theta_j - \theta_i$ based on estimated variances from the MLEs. It provides a conservative alternative to selecting a single "winner," especially useful when skill estimates are close and comparisons are incomplete. The approach adjusts for statistical uncertainty via the estimated variance-covariance structure of $\hat{\theta}$ and is most suitable when the number of boards is large and the tournament design is symmetric.

To further improve the discrimination capability of this method, we propose a weighted extension that adjusts the confidence thresholds based on the magnitude of each estimated skill level. Specifically, pairs with higher $\hat{\theta}_j$ values are assigned smaller correction terms via a scaling factor ω_j, reflecting their stronger performance and raising the bar for others to surpass them.
This modification preserves the original coverage guarantee while producing smaller, more selective confidence subsets. In empirical evaluations across two datasets, the weighted procedure consistently narrowed the candidate set without excluding the true best pair, achieving better balance between statistical confidence and practical interpretability.

The full development of this method, including the mathematical formulation, proof of validity, and detailed comparisons with the unweighted version, is presented in Appendix A.3. A comparison of results across two datasets demonstrates the empirical advantage of our weighted-correction approach over the original formulation by Yu and Lam. *Since this family of methods is more relevant to large-scale or Mitchell-style tournaments and less applicable to our dataset, it is not included in the main analysis.*

3.5 Conclusion

In this section, we first reviewed the details of the scoring system in duplicate Bridge tournaments (section 3.1), covering the assignment of pairs, tables, and boards, as well as the methodology for scoring and calculating final rankings. Through these discussions, we understood that the scoring system significantly reduces the luck factor from randomly dealt boards by allowing each pair to compete under identical conditions. The provided example clarified the scoring process and how winners are identified. This knowledge is fundamental for conducting statistical analyses of player skills in subsequent sections.
Next, we examined the permutation Model (section 3.2), a core statistical framework employed in this study. The model assumes a fixed intrinsic skill parameter for each pair, assessed through repeated comparisons. We explored how the likelihood function connects pairwise match outcomes to skill levels, utilizing the Bradley-Terry Model Application. The section also offered rigorous mathematical justification for using this model in Bridge data analysis, highlighting three critical criteria: the Reversibility Condition, Skill Parameter Explanation, and the Sufficiency of Total Match Points. These criteria underpin the robustness of the model when applied to practical data analysis.
Moreover, the permutation model was extended to handle tie situations frequently occurring in actual Bridge tournaments (section 3.3). The revised model effectively reflects realistic outcomes by introducing the tie parameter ϕ, incorporating tie scenarios explicitly. The section specifically addressed how probabilities of pairwise comparisons are adjusted to include ties and provided a formal proof demonstrating that total match points and the number of ties constitute sufficient statistics for estimating both skill and tie parameters.
Finally, methods for estimating skill parameters were discussed (section 3.4), particularly addressing the issue of over-parameterization. We learned how simplifying parameter estimation improves stability. Additionally, the section proposed a statistically robust method for selecting the best-performing pairs based on estimated skills, ensuring high confidence through constructing an appropriate random subset.

This section established a comprehensive theoretical framework essential for the subsequent practical analysis of Bridge tournament data. Building on the overview of duplicate tournament data, the next section applies the permutation model to a real Bridge dataset. We will demonstrate how maximum likelihood estimation and Bootstrap methods validate our theoretical assumptions and highlight any shortcomings that prompt further refinements.

4 Verification of an analysis by Yu and Lam

4.1 Data preparation and representation

The data analysed in this section are derived from [34], which examined a four-table Howell rotating duplicate Bridge tournament. This dataset is particularly suitable for validating the permutation model, as it ensures balanced competition and diverse interactions among player pairs. The following analysis is conducted to verify the findings of Yu and Lam, using their dataset to test the applicability of the permutation model. This includes a re-examination of the derived skill parameters and tie-breaking probabilities to confirm their consistency with the original results while incorporating additional insights. The tournament setup included:

- $B = 28$: Number of boards;
- $n = 8$: Number of player pairs;
- $T = 4$: Number of North-South (NS) and East-West (EW) pairs per board.

The key data components include:

1. **NS and EW Pair Assignments:** The assignment of player pairs to North–South ($\alpha(b,t)$) and East–West ($\beta(b,t)$) positions follows a 28-board Howell rotation pattern. This arrangement guarantees that each pair competes against a wide range of opponents, thereby reducing systematic bias. The table below illustrates this rotation scheme.

	$\alpha(b,t)$				$\beta(b,t)$			
t / b	1	2	3	4	1	2	3	4
1–4	2	3	5	8	4	7	6	1
5–8	3	4	6	8	5	1	7	2
9–12	4	5	7	8	6	2	1	3
13–16	5	6	1	8	7	3	2	4
17–20	6	7	2	8	1	4	3	5
21–24	7	1	3	8	2	5	4	6
25–28	1	2	4	8	3	6	5	7

Figure 4: 4-Table Howell movements with 28 boards

2. **MP Scores:** A 28×8 matrix containing Match Points for all player pairs.

Board	1	2	3	4	5	6	7	8
1	2 (50)	2.5 (460)	0 (−100)	0.5(−460)	2.5 (460)	0.5 (−460)	3 (100)	1 (−50)
2	1 (140)	1 (−230)	0 (−450)	2 (230)	3 (100)	0 (−100)	3 (450)	2 (−140)
3	1 (230)	1 (−620)	0 (−1430)	2 (620)	3 (130)	0 (−130)	3 (1430)	2 (−230)
4	2 (170)	2 (80)	0 (−300)	1 (−80)	3 (130)	0 (−130)	3 (300)	1 (−170)
5	1.5 (−50)	0 (−150)	0 (−90)	1.5 (50)	3 (90)	1.5 (50)	1.5 (−50)	3 (150)
6	3 (−150)	1.5(−200)	3 (300)	0 (150)	0 (−300)	1.5 (200)	1.5(−200)	1.5 (200)
7	0 (−620)	1 (−600)	1 (100)	3 (620)	2 (−100)	0 (−140)	3 (140)	2 (600)
8	2 (170)	1 (−50)	3 (100)	1 (−170)	0 (−100)	0 (−180)	3 (180)	2 (50)
9	2 (100)	1 (−100)	3 (140)	3 (110)	2 (100)	0 (−110)	1 (−100)	0 (−140)
10	1 (−120)	3 (200)	2 (−100)	3 (800)	0 (−200)	0 (−800)	2 (120)	1 (100)
11	1.5 (480)	3 (490)	0 (450)	1.5(−480)	0 (−490)	1.5 (480)	1.5(−480)	3 (−450)
12	2 (−500)	1 (−600)	3 (−200)	3 (680)	2 (600)	0 (−680)	1 (500)	0 (200)
13	3 (200)	0 (−200)	2 (180)	1 (100)	0 (−300)	1 (−180)	3 (300)	2 (−100)
14	1 (400)	0.5(−400)	0.5 (−420)	0.5(−420)	0 (250)	2.5 (420)	3 (−250)	2.5 (420)
15	1 (−130)	0.5 (130)	0.5 (−50)	0.5 (−50)	0 (−500)	2.5 (50)	3 (500)	2.5 (50)
16	2 (−130)	2 (130)	2 (600)	3 (620)	3 (100)	1 (−600)	0 (−100)	0 (−620)
17	2 (460)	3 (−400)	0 (400)	2 (460)	2 (460)	1 (−460)	1 (−460)	1 (−460)
18	0.5(−150)	2.5 (150)	0.5 (−150)	2 (−130)	3 (200)	2.5 (150)	1 (130)	0 (−200)
19	3 (90)	3 (400)	0 (−400)	2 (50)	1 (−130)	0 (−90)	1 (−50)	2 (130)
20	2.5 (110)	3 (100)	0 (−100)	2.5 (110)	1 (90)	0.5 (−110)	0.5(−110)	2 (−90)
21	1 (−100)	0.5(−130)	2.5 (130)	0.5(−130)	2 (100)	3 (400)	2.5 (130)	0 (−400)
22	0.5(−630)	0.5 (600)	0.5 (−630)	2.5 (630)	2.5 (630)	0.5 (600)	2.5(−600)	2.5 (−600)
23	2 (110)	2 (110)	0 (−140)	3 (140)	1 (−110)	0 (−120)	1 (−110)	3 (120)
24	0.5 (−50)	0 (−140)	0.5 (−50)	2.5 (50)	2.5 (50)	1 (−110)	3 (140)	2 (110)
25	0 (−50)	3 (180)	3 (50)	2 (150)	1 (−150)	0 (−180)	2 (−80)	1 (80)
26	2 (−100)	0 (−150)	1 (100)	2 (−100)	1 (100)	3 (150)	1 (100)	2 (−100)
27	3 (500)	2 (170)	0 (−500)	0 (−100)	3 (100)	1 (−170)	2 (50)	1 (−50)
28	1.5(−100)	3 (630)	1.5 (100)	1.5(−100)	1.5 (100)	0 (−630)	3 (170)	0 (−170)
Total	44.5 (330)	45.5(−140)	29.5(−2460)	49 (3350)	45 (1410)	24.5(−2880)	56 (2150)	42 (−1760)

Figure 5: Match-point scores and raw scores from Yu and Lam

- Each row represents one board (Board 1 to Board 28).
- Each column corresponds to a player pair (1 to 8).
- Each cell contains:
  - The raw score achieved;
  - The corresponding MP score.
- The last row aggregates each pair's total raw scores and total MP scores.

4.1.1 Illustration of match point scoring

To demonstrate how match point (MP) scores are calculated in a duplicate bridge tournament, we take Board 16 as an example. According to the Howell movement design from figure 4, the board was played at four different tables with the following pairings:

Table 3: Table Assignments for Board 16

Table	N/S Pair	E/W Pair
1	5	7
2	6	3
3	1	2
4	8	4

The raw scores and corresponding MP scores achieved by the N/S pairs are shown below:

Table 4: Raw Scores and MP Scores for N/S Pairs on Board 16

N/S Pair	Raw Score	MP Score
5	100	3
6	-600	1
1	-130	2
8	-620	0

The E/W pairs receive the complementary MP scores, calculated as 3 − N/S MP. Their raw scores are the negatives of the corresponding N/S scores:

Table 5: Raw Scores and MP Scores for E/W Pairs on Board 16

E/W Pair	MP Score	Raw Score
7	0	-100
3	2	600
2	1	130
4	3	620

The MP score for pair 2 in the original dataset 5 board 16 was incorrectly recorded as 2. The correct MP score should be 1. The table below corrects for this discrepancy and ensures that the MP scores are consistent with the corresponding raw results. Additional examples of data inconsistencies and their corrections can be found in the appendix A.1.

4.1.2 Corrected match point scores

To ensure consistency with duplicate bridge tournament rules, each board should yield a total of 12 match points (MP) across the four tables (since there are 4 North–South and 4 East–West pairs, and each comparison awards a total of 3 points). However, in the original dataset, some rows do not satisfy this condition. For example, the total MP for Board 14 was only 10.5, indicating an inconsistency in the scoring.
We reviewed and recalculated all MP scores based on the raw results and tie-handling rules. The table below presents the corrected MP scores for all 28 boards. Each row now correctly sums to 12.0, ensuring internal consistency and validity for subsequent modeling and analysis.

Table 6: Corrected Match Point Scores Per Board (Row Sums Equal to 12)

	Pair 1	Pair 2	Pair 3	Pair 4	Pair 5	Pair 6	Pair 7	Pair 8	Row Sum
Board 1	2.0	2.5	0.0	0.5	2.5	0.5	3.0	1.0	12.0
Board 2	1.0	1.0	0.0	2.0	3.0	0.0	3.0	2.0	12.0
Board 3	1.0	1.0	0.0	2.0	3.0	0.0	3.0	2.0	12.0
Board 4	2.0	2.0	0.0	1.0	3.0	0.0	3.0	1.0	12.0
Board 5	1.5	0.0	0.0	1.5	3.0	1.5	1.5	3.0	12.0
Board 6	3.0	1.5	3.0	0.0	0.0	1.5	1.5	1.5	12.0
Board 7	0.0	1.0	1.0	3.0	2.0	0.0	3.0	2.0	12.0
Board 8	2.0	1.0	3.0	1.0	0.0	0.0	3.0	2.0	12.0
Board 9	2.0	1.0	3.0	3.0	2.0	0.0	1.0	0.0	12.0
Board 10	1.0	3.0	2.0	3.0	0.0	0.0	2.0	1.0	12.0
Board 11	1.5	3.0	0.0	1.5	0.0	1.5	1.5	3.0	12.0
Board 12	2.0	1.0	3.0	3.0	2.0	0.0	1.0	0.0	12.0
Board 13	3.0	0.0	2.0	1.0	0.0	1.0	3.0	2.0	12.0
Board 14	1.0	2.0	0.5	0.5	0.0	2.5	3.0	2.5	12.0
Board 15	1.0	2.0	0.5	0.5	0.0	2.5	3.0	2.5	12.0
Board 16	2.0	1.0	2.0	3.0	3.0	1.0	0.0	0.0	12.0
Board 17	2.0	3.0	0.0	2.0	2.0	1.0	1.0	1.0	12.0
Board 18	0.5	2.5	0.5	2.0	3.0	2.5	1.0	0.0	12.0
Board 19	3.0	3.0	0.0	2.0	1.0	0.0	1.0	2.0	12.0
Board 20	2.5	3.0	0.0	2.5	1.0	0.5	0.5	2.0	12.0
Board 21	1.0	0.5	2.5	0.5	2.0	3.0	2.5	0.0	12.0
Board 22	0.5	0.5	0.5	2.5	2.5	0.5	2.5	2.5	12.0
Board 23	2.0	2.0	0.0	3.0	1.0	0.0	1.0	3.0	12.0
Board 24	0.5	0.0	0.5	2.5	2.5	1.0	3.0	2.0	12.0
Board 25	0.0	3.0	3.0	2.0	1.0	0.0	2.0	1.0	12.0
Board 26	2.0	0.0	1.0	2.0	1.0	3.0	1.0	2.0	12.0
Board 27	3.0	2.0	0.0	0.0	3.0	1.0	2.0	1.0	12.0
Board 28	1.5	3.0	1.5	1.5	1.5	0.0	3.0	0.0	12.0

4.2 Maximum likelihood estimation pseudocode

The objective is to estimate Player skill parameters θ and tie probability parameter ϕ. This is achieved using Maximum Likelihood Estimation (MLE), which shows in section 4.2.4. The complete implementation in Python is provided in Appendix B.1.

4.2.1 Step 1: Building `alpha` and `beta`

We first define two lists of patterns, $\alpha\_patterns$ for NS pair assignments and $\beta\_patterns$ for EW pair assignments. Each pattern is repeated 4 times to produce a total of 28 boards. We then convert them into NumPy arrays, each of shape $(28, 4)$. Additionally, we define $B = 28$ (boards), $n = 8$ (teams), and $T = 4$ (tables per board).

Algorithm 1 Constructing `alpha` and `beta`

procedure BUILDALPHABETA
  $\alpha\_patterns \leftarrow [(2,3,5,8),\ (3,4,6,8),\ (4,5,7,8),\ \dots]$
  $\beta\_patterns \leftarrow [(4,7,6,1),\ (5,1,7,2),\ (6,2,1,3),\ \dots]$
  $\alpha \leftarrow [\,],\ \beta \leftarrow [\,]$
  ▷ Repeat each pattern 4 times to form 28 boards in total
  for $i \leftarrow 0$ to 6 **do** ▷ since we have 7 patterns
    $a\_pat \leftarrow \alpha\_patterns[i]$
    $b\_pat \leftarrow \beta\_patterns[j]$
    for $r \leftarrow 1$ to 4 **do**
      α.append(a_pat)
      β.append(b_pat)
  ▷ Convert to NumPy arrays of shape $(28, 4)$
  $\alpha \leftarrow$ np.array(α)
  $\beta \leftarrow$ np.array(β)
  ▷ Set dimensions: $B = 28,\ n = 8,\ T = 4$
  return $(\alpha,\ \beta,\ B,\ n,\ T)$

4.2.2 Step 2: Constructing L_b and D_b

For each board b, we retrieve which teams sit at NS ($\alpha[b,:]$) and EW ($\beta[b,:]$) positions. We then obtain the scores for the 4 NS pairs from `mp_scores`. Next, we generate pairs (i, j) among these 4 tables and classify them into:

- D_b: sets of table pairs that result in a draw (equal scores),
- L_b: sets of table pairs with winners and losers.

We store these results in a dictionary appended to `boards_data`.

Algorithm 2 Building `boards_data` with Win/Loss (L_b) and Draw (D_b) Sets

procedure BUILDBOARDSDATA(α, β, mp_scores, B, n, T)
  `boards_data` $\leftarrow [\,]$
  for $b \leftarrow 0$ to $B-1$ **do**
    `ns_pairs` $\leftarrow \alpha[b,:]$
    `ew_pairs` $\leftarrow \beta[b,:]$ ▷ Extract scores for the NS teams
    `scores` $\leftarrow$ mp_scores[b, `ns_pairs` -1] ▷ -1 because team indices start from 1
    $L_b \leftarrow [\,],\ D_b \leftarrow [\,]$
    for $i \leftarrow 0$ to $T-1$ **do**
      for $j \leftarrow i+1$ to $T-1$ **do**
        if isClose(`scores`$[i]$, `scores`$[j]$) **then**
          D_b.append$((i, j))$
        else if `scores`$[i] <$ `scores`$[j]$ **then**
          L_b.append$((i, j))$ ▷ i loses to j
        else
          L_b.append$((j, i))$ ▷ j loses to i
  return `boards_data`

4.2.3 Step3: Unpacking parameters

In the permutation model equation (16), ϕ only needs to be greater than zero. However, if ϕ is allowed to take any positive value without constraint, it may lead to numerical issues during optimization. For

example, when ϕ becomes very large, the denominator in the likelihood formula can become too small, causing instability such as division by zero or $\log(0)$.
To improve optimization stability and efficiency, we restrict ϕ to the interval $(0, 1)$ using a logistic transformation. This prevents ϕ from growing too large and keeps the optimization within a well-behaved range. Although this limits the maximum value of ϕ, it does not affect the structure or validity of the model itself.
Yu and Lam do not specify how the tie parameter ϕ is estimated or constrained in practice. Therefore, we adopt a practical approach that balances numerical stability with model flexibility.

We modeled using a logit transformation:

$$\phi = \frac{\exp(\psi)}{1 + \exp(\psi)}.$$

We interpret the last element of x as ψ, which is mapped into $\phi \in (0, 1)$ via a logistic function. The first $n - 1$ entries of x are taken to be $\theta_1, \ldots, \theta_{n-1}$, while θ_n is forced to 0 to serve as a reference baseline. $\theta_n = 0$ eliminates redundant degrees of freedom in the model, preventing meaningless inflation of parameter values.

Algorithm 3 UNPACK_PARAMS

function UNPACKPARAMS(x, n)
  $\psi \leftarrow x[n-1]$
  $\phi \leftarrow \frac{\exp(\psi)}{1+\exp(\psi)}$ ▷ Logistic transform of ψ
  $\theta \leftarrow \mathbf{0}_n$
  for $k \leftarrow 0$ to $n-2$ **do**
    $\theta[k] \leftarrow x[k]$
  $\theta[n-1] \leftarrow 0$ ▷ Fix baseline for identifiability
  return (θ, ϕ)

4.2.4 Step 4: Log-likelihood and gradient

This step focuses on computing the log-likelihood function $\ell(\theta, \phi)$ and its gradients with respect to both θ and ϕ. These quantities are essential for parameter estimation in Step 5, where a BFGS optimization algorithm minimizes the negative log-likelihood.

MLE Formulation The pairwise outcome probabilities defined in equation (16) include both win/loss and tie cases. Let L_b and D_b denote the sets of decisive and tie outcomes on board b, respectively. Assuming conditional independence across pairwise comparisons within and across boards, with equation (3), the total likelihood becomes:

$$L(\theta, \phi) = \prod_{b=1}^{B} \left[\prod_{(i,j)\in L_b} \frac{\lambda_j}{\lambda_i + \lambda_j + \phi\sqrt{\lambda_i \lambda_j}} \cdot \prod_{(i,j)\in D_b} \frac{\phi\sqrt{\lambda_i \lambda_j}}{\lambda_i + \lambda_j + \phi\sqrt{\lambda_i \lambda_j}} \right] \tag{18}$$

Taking logarithms and summing over all boards yields the log-likelihood function:

$$\ell(\theta, \phi) = \sum_{b=1}^{B} \left[\sum_{(i,j)\in L_b} \ln\left(\frac{\lambda_j}{\lambda_i + \lambda_j + \phi\sqrt{\lambda_i \lambda_j}} \right) + \sum_{(i,j)\in D_b} \ln\left(\frac{\phi\sqrt{\lambda_i \lambda_j}}{\lambda_i + \lambda_j + \phi\sqrt{\lambda_i \lambda_j}} \right) \right] \tag{19}$$

Maximizing this function yields the MLEs of θ and ϕ. In practice, we minimize the negative log-likelihood using gradient-based optimization. Our implementation uses the BFGS algorithm for efficient convergence.

The Denominator (denom) The common denominator appearing in both win/loss and tie probabilities is defined as:

$$\text{denom} \;=\; \lambda_i + \lambda_j + \phi \cdot \sqrt{\lambda_i \, \lambda_j}.$$

Gradient for θ_k The total gradient for θ_k can be expressed as:

$$\frac{\partial \ell}{\partial \theta_k} \;=\; \text{win/loss contribution} \;+\; \text{draw contribution}.$$

For the skill parameter θ_k, with equation (9), the gradient contributions from ℓ_{L_b} (win/loss pairs) and ℓ_{D_b} (draw pairs) are computed as follows:

- **Win/loss contribution:** For win/loss pairs L_b, the gradient is:

$$\frac{\partial \ell_{L_b}}{\partial \theta_k} \;=\; \sum_{(i,j)\in L_b} \Big[\frac{\partial \log(\lambda_j)}{\partial \theta_k} \;-\; \frac{\partial \log\big(\lambda_i + \lambda_j + \phi \sqrt{\lambda_i \, \lambda_j}\big)}{\partial \theta_k} \Big].$$

- **Draw contribution:** For draw pairs D_b, the gradient is:

$$\frac{\partial \ell_{D_b}}{\partial \theta_k} \;=\; \sum_{(i,j)\in D_b} \Big[\frac{\partial \log(\phi)}{\partial \theta_k} \;+\; \frac{\partial \big(0.5 \, \log(\lambda_i) \;+\; 0.5 \, \log(\lambda_j)\big)}{\partial \theta_k} \;-\; \frac{\partial \log\big(\lambda_i + \lambda_j + \phi \sqrt{\lambda_i \, \lambda_j}\big)}{\partial \theta_k} \Big].$$

Gradient for ψ via logistic link For the tie probability parameter ψ, we use the logistic map $\phi = \frac{1}{1+e^{-\psi}}$. Hence, the chain rule applies:

$$\frac{\partial \ell}{\partial \psi} \;=\; \frac{\partial \ell}{\partial \phi} \cdot \frac{\partial \phi}{\partial \psi}, \quad \text{where} \quad \frac{\partial \phi}{\partial \psi} \;=\; \phi \, (1 - \phi).$$

By combining the expressions for $\frac{\partial \ell}{\partial \phi}$ and $\frac{\partial \phi}{\partial \psi}$, we obtain the complete gradient with respect to ψ.

Purpose of the Gradient The gradient vector $\nabla \ell = (\partial \ell / \partial \theta, \partial \ell / \partial \psi)$ is returned along with the log-likelihood value and passed to Step 5. Since standard optimization libraries minimize by default, we return $(-\ell, -\nabla \ell)$. This ensures an efficient search for the minimum of the negative log-likelihood, enabling us to estimate the optimal parameters $\hat{\theta}$ and $\hat{\phi}$.

Algorithm 4 LOGLIKELIHOODANDGRADIENT

function LOGLIKELIHOODANDGRADIENT(x, boards_data, n, T)
  $(\theta, \phi) \leftarrow$ UNPACKPARAMS(x, n) ▷ $\phi = 1/(1 + e^{-\psi})$
  $\ell \leftarrow 0$, grad_theta $\leftarrow \mathbf{0}_n$, grad_psi $\leftarrow 0$
  for each bd **in** boards_data **do**
    Extract L_b, D_b, ns_pairs, ew_pairs from bd
    for $t \leftarrow 0$ to $T - 1$ **do**
      $i_{ns}, i_{ew} \leftarrow$ ns_pairs$[t]$, ew_pairs$[t]$
      $\lambda \leftarrow \exp(\theta[i_{ns}] - \theta[i_{ew}])$
      Store λ and its partial derivatives
    for each (i, j) in L_b **do**
      Compute λ_i, λ_j, denom
      Update ℓ with $\log(\lambda_j)$ and $\log(\text{denom})$
      Update grad_theta and grad_psi
    for each (i, j) in D_b **do**
      Compute λ_i, λ_j, denom
      Update ℓ using $\log(\phi)$, $\log(\lambda_i)$, $\log(\lambda_j)$, and $\log(\text{denom})$
      Update grad_theta and grad_psi
  Combine grad_theta and grad_psi into grad
  return $(-\ell, -\text{grad})$

4.2.5 Step 5: Full model estimation via BFGS

We optimise the model parameters in this step using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm. BFGS is a quasi-Newton method widely recognized for its efficiency in solving unconstrained nonlinear optimization problems. It dynamically approximates the Hessian matrix using gradient evaluations, making it computationally efficient while retaining the fast convergence properties of Newton's method [28].

The optimization of the negative log-likelihood function $-\mathcal{L}(\theta, \phi)$ requires a robust and efficient algorithm. BFGS offers the following advantages:

- **Rapid Convergence:** Compared to first-order methods such as gradient descent, BFGS achieves superlinear convergence near the optimal solution, significantly reducing the number of iterations required [26].
- **Explicit Hessian Computation:** Instead of computing and inverting the Hessian matrix (which can be computationally expensive for high-dimensional problems), BFGS approximates it using gradient information. This reduces computational complexity while maintaining accuracy [27].
- **Ease of Implementation:** BFGS is implemented in many popular optimization libraries, such as SciPy, making it straightforward to use in practice. For our problem, we use a library implementation to minimize the negative log-likelihood.

In the model, the optimisation process consists of the following steps:

1. **Initialization:** We initialize the parameter vector $x = (\theta_1, \ldots, \theta_{n-1}, \psi)$ to zeros, implying $\theta_i = 0$ for all i and $\psi = 0$, which corresponds to a draw probability $\phi = 0.5$.
2. **Optimization:** Using a BFGS-based routine (`MinimizeBFGS`), we minimize the negative log-likelihood function. This routine requires the function value and gradient, defined in Step 4.
3. **Output:** Upon convergence, we extract the optimal parameters $\hat{\theta}$ and $\hat{\phi}$, compute the final log-likelihood value ℓ_{full}, and optionally rank the teams based on their skill parameters $\hat{\theta}$.

Algorithm 5 Optimizing the Full Model and Outputting Results

procedure FULLMODELESTIMATION(n, `boards_data`)
  $x\_init \leftarrow \mathbf{0}_n$ ▷ $\theta_1..\theta_{n-1} = 0,\ \psi = 0 \implies \phi = 0.5$
  res $\leftarrow$ MinimizeBFGS$\Big($LogLikelihoodAndGradient, x_init, `boards_data`, n, $T\Big)$
  $\hat{x} \leftarrow$ res.x
  $\ell_{\text{full}} \leftarrow -$ res.fun
  $(\hat{\theta}, \hat{\phi}) \leftarrow$ UNPACKPARAMS($\hat{x}$, n)
  print "Converged = ", res.success
  print "Full model log-likelihood = ", ℓ_{full}
  ▷ Display each θ_i
  for $i \leftarrow 1$ to n **do**
    print "theta_i = ", $\hat{\theta}[i-1]$
  print "phi = ", $\hat{\phi}$
  ▷ Sort $\hat{\theta}$ in descending order to rank teams
  `theta_indices` $\leftarrow$ argsort($\hat{\theta}$) (descending)
  print "Ranking by skill:"
  for $k \leftarrow 1$ to n **do**
    $idx \leftarrow$ `theta_indices`$[k]$
    print "Rank k: theta_(idx) = ", $\hat{\theta}[idx-1]$

4.2.6 Permutation model result

From the figure 6, Optimisation results using the BFGS algorithm show successful convergence. The skill parameters (θ) reflect the relative abilities of the players, with $\theta_n = 0$ fixed as the baseline for identifiability. Rankings based on these parameters show that player 7 has the highest skill $\theta_7 = 0.7906$ and player 6 has the lowest skill $\theta_6 = -0.9986$. In addition, the negative log-likelihood value of the final optimisation (-143.6938) indicates that the model fits the observed data well.

```
Optimization terminated successfully.
         Current function value: 143.693841
         Iterations: 14
         Function evaluations: 21
         Gradient evaluations: 21
Converged: True
Log-likelihood: -143.6938411916855
θ_1 = 0.1336
θ_2 = 0.1899
θ_3 = -0.7132
θ_4 = 0.3866
θ_5 = 0.1838
θ_6 = -0.9896
θ_7 = 0.7906
θ_8 = 0.0000
ϕ = 0.2938
Ranking of θ by skill level (from highest to lowest):
Rank 1: θ_7 = 0.7906
Rank 2: θ_4 = 0.3866
Rank 3: θ_2 = 0.1899
Rank 4: θ_5 = 0.1838
Rank 5: θ_1 = 0.1336
Rank 6: θ_8 = 0.0000
Rank 7: θ_3 = -0.7132
Rank 8: θ_6 = -0.9896
```

Figure 6: Permutation Model result

The optimization results highlight the need for convergence to ensure meaningful and reliable outputs. Without convergence, the model is not guaranteed to reach the optimum point (minimum or maximum) of the objective function, resulting in unstable or incorrect parameter estimates ($\hat{\theta}$ and $\hat{\phi}$). The convergence of the BFGS algorithm ensures computational efficiency, stability and valid interpretation, as evidenced by the negative log-likelihood of successful optimisation and well-defined results.

4.3 Bootstrap-Based Inference under the Davidson Model

Due to the limited number of observations in this study (only 28 boards), traditional inferential methods that rely on large-sample asymptotic theory—such as standard error formulas or normality assumptions—are often unreliable or inapplicable. Furthermore, the underlying model used in this thesis is an extension of the Bradley–Terry model with an additional tie parameter ϕ, which makes analytical derivation of uncertainty measures especially difficult.
To overcome these challenges, we adopt the Parametric Bootstrap, a resampling-based method that allows for data-driven estimation of uncertainty without relying on analytical approximations. The Bootstrap method is a general statistical technique used to estimate the distribution of sample statistics, such as the mean, variance, or confidence intervals. In this thesis, we apply the parametric version, which assumes a specific probabilistic model to simulate new data and re-estimate parameters. In our case, we use Davidson's model equation (16).

4.3.1 Uncertainty estimation via parametric bootstrap

To estimate the uncertainty in the model parameters θ and ϕ, we adopt a parametric Bootstrap procedure based on the fitted Davidson model. Since the Davidson model characterizes the

probability distribution over win–loss–tie outcomes, the resampling is performed at the level of pairwise comparison results, rather than using raw scores or match-point totals. The steps are as follows:

1. Fit the Davidson model to the observed data to obtain the MLE results $\hat{\theta}$ and $\hat{\phi}$.
2. For each observed comparison between players i and j, simulate a new match outcome based on the model's probability structure:

$$P(R_{bi} < R_{bj}) = \frac{\lambda_{bj}}{\lambda_{bi} + \lambda_{bj} + \phi\sqrt{\lambda_{bi}\lambda_{bj}}}, \quad P(R_{bi} > R_{bj}) = \frac{\lambda_{bi}}{\lambda_{bi} + \lambda_{bj} + \phi\sqrt{\lambda_{bi}\lambda_{bj}}}$$

$$P(R_{bi} = R_{bj}) = \frac{\phi\sqrt{\lambda_{bi}\lambda_{bj}}}{\lambda_{bi} + \lambda_{bj} + \phi\sqrt{\lambda_{bi}\lambda_{bj}}}$$

   These probabilities are then used to simulate new pairwise comparison outcomes: for example, if the original result between Pair 1 and Pair 2 was a tie, the simulated outcome might instead be a win for Pair 1, a win for Pair 2, or remain a tie—each sampled according to the estimated Davidson probabilities.
3. Construct a synthetic dataset consisting of these simulated pairwise outcomes. This dataset retains the same board structure and matchups as the original tournament (i.e., which pair played against which, and on which board), but replaces the actual results with simulated ones. Note that the synthetic data contains only outcome labels (win/loss/tie), and not ranks or MP scores.
4. Refit the Davidson model to the synthetic dataset to obtain a new set of parameter estimates $\theta^{(b)}$ and $\phi^{(b)}$.
5. Repeat Steps 2–4 for $B = 1000$ iterations to generate empirical sampling distributions for all parameters.

Let $\{\theta^{(b)}\}_{b=1}^{B}$ denote the set of bootstrap replicates. A 95% confidence interval for each skill parameter θ_i is then constructed using the 2.5th and 97.5th percentiles of its empirical distribution:

$$\theta_i^{\text{lower}} = \text{percentile}(\{\theta_i^{(b)}\}, 2.5), \quad \theta_i^{\text{upper}} = \text{percentile}(\{\theta_i^{(b)}\}, 97.5)$$

This simulation-based method provides a flexible and robust way to assess parameter uncertainty while preserving the structural dependencies inherent in tournament matchups. It is particularly advantageous for small datasets with complex tie behavior, where analytical variance estimation is either intractable or unreliable.
The complete implementation of this procedure is provided in the Appendix B.2.

4.3.2 Parametric Bootstrap results

We first focus on the results of the parametric Bootstrap method with a repetition number of $B = 1000$. For each parameter θ_i, figure 7 presents the point estimate, along with its approximate 95% confidence interval.

```
=== Starting Parametric Bootstrap Analysis (B=1000) ===
In 1000 simulations, the 95% confidence interval for each θ_i is:
θ_1 = 0.1336  95% CI = [-0.5643, 0.8639]
θ_2 = 0.1899  95% CI = [-0.5017, 0.8947]
θ_3 = -0.7132  95% CI = [-1.4697, -0.0153]
θ_4 = 0.3866  95% CI = [-0.2474, 1.0674]
θ_5 = 0.1838  95% CI = [-0.4785, 0.8868]
θ_6 = -0.9896  95% CI = [-1.8629, -0.3386]
θ_7 = 0.7906  95% CI = [0.1286, 1.4896]
θ_8 = 0.0000  95% CI = [0.0000, 0.0000]
```

Figure 7: Parametric Bootstrap 95% Confidence Intervals for each θ_i. Pair 8 is the benchmark pair with skill parameter fixed at 0.

The results can be categorized into distinct cases based on the confidence intervals:

- Intervals crossing zero: For example, the confidence intervals of $\theta_1, \theta_2, \theta_4, \theta_5$ include zero, indicating that their relative advantage or disadvantage remains statistically inconclusive at this sample size. This suggests that further data may be needed to draw definitive conclusions about their performance.
- Entirely positive or negative: In contrast, θ_3 and θ_6 have confidence intervals entirely below zero, suggesting a consistent weakness in their performance. Meanwhile, θ_7 remains distinctly above zero, indicating a stable and significantly stronger skill level. These cases provide more definitive evidence of the relative strengths of the players.

Compared to the permutation model results in figure 6, both methods identify the same players at the extremes: θ_7 consistently shows superior performance, while θ_3 and θ_6 are notably weaker. However, the results do not support statistically significant distinctions among the remaining players, especially those with confidence intervals crossing zero. This suggests that point-based rankings may be misleading in such cases, and it is more appropriate to interpret the results in terms of performance tiers or uncertainty bands.
In this context, the parametric bootstrap serves not only to estimate parameters but also to quantify the confidence we can place in comparative skill assessments. Rather than producing strict rankings, it reveals which differences are statistically credible and which remain uncertain.

4.4 Comparison of results

The following table compares the replicated results with the original values:

Table 7: Comparison of Current Results and Original Results by Yu and Lam

Parameter	Current Result	Original Result	Rank (Current)	Rank (Original)
θ_1	0.1336	0.0702	5	5
θ_2	0.1899	0.1113	3	3
θ_3	-0.7132	-0.4266	7	7
θ_4	0.3866	0.2271	2	2
θ_5	0.1838	0.1030	4	4
θ_6	-0.9896	-0.5934	8	8
θ_7	0.7906	0.4579	1	1
θ_8	0.0000	0.0000	6	6
ϕ	0.2938	0.6845	-	-

From table 7, the rankings of the skill parameters θ are consistent across both sets of results, ensuring the relative order of player skill levels remains valid. While the numerical values differ

slightly, for example, θ_7 has the highest value, and θ_6 has the lowest in both results. The replicated result ($\phi = 0.2938$) is significantly lower than the original value ($\phi = 0.6845$).

4.4.1 Scatter plot

A scatter plot was used to compare the estimated parameters obtained using the two methods $\theta_1, \ldots, \theta_8$ to assess the consistency of the results. To analyse the results, we chose a scatterplot rather than a histogram because the The dataset contains only 8 parameter values (θ_1 to θ_8), which is not enough to construct a meaningful histogram. A scatterplot provides a direct comparison of the two sets of results, with each point corresponding to a specific θ, making the difference in values visually apparent and easy to interpret.

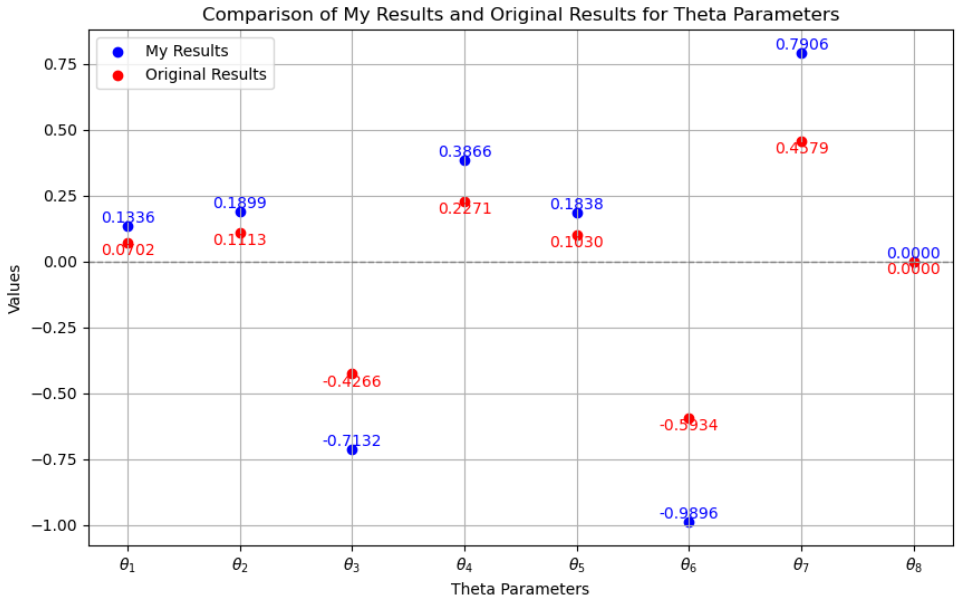


Figure 8: Comparison of My Results and Original Results for Theta Parameters

The scatter plot demonstrates that the overall ranking of parameters is consistent between our results and the original results, validating the correctness of ourimplementation. The scatterplot shows that the overall ordering of the parameters is consistent between ourresults and the original results, which verifies that our implementation is correct. There are numerical differences in some of the parameters, and we will analyse why these differences occur.

4.4.2 Correlation coefficient

We can plot the scatter diagram of “Current Result” versus “Original Result”.

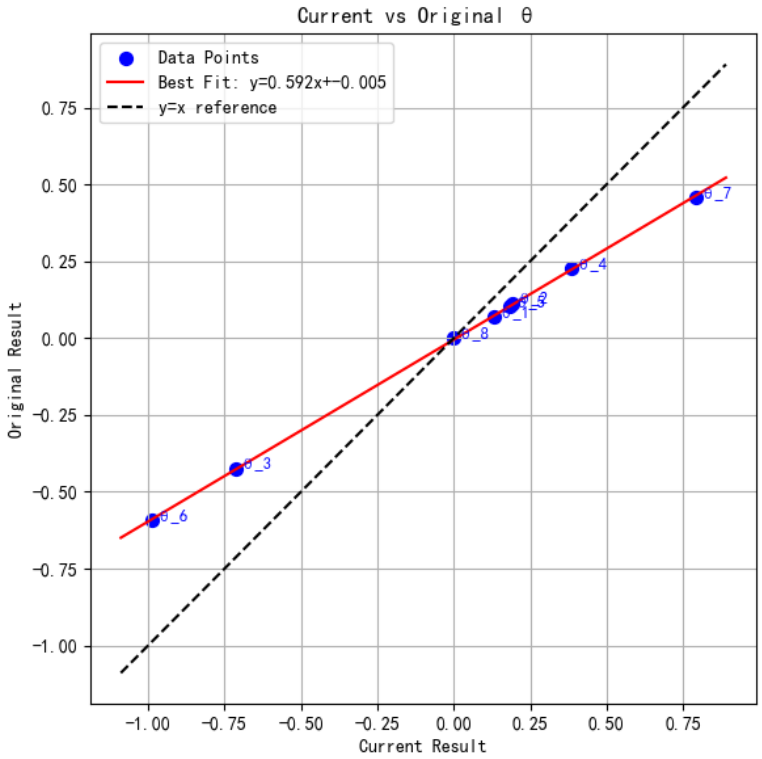


Figure 9: scatter diagram of “Current Result” versus “Original Result”

After that, we can use the Pearson correlation coefficient R to quantitatively measure the linear relationship between the two sets of results.

Pearson correlation coefficient

It is computed by:

$$R = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}},$$

where $\bar{x}$ and $\bar{y}$ are the means of the two datasets, respectively. When R is close to 1 (or -1), it indicates a strong positive (or negative) linear correlation, whereas a value close to 0 means almost no linear relationship.
This can be easily obtained in Python by calling `np.corrcoef(x, y)[0, 1]`.

The results show that the Pearson correlation coefficient $R \approx 0.99994$, which is extremely close to 1, indicates a strong positive linear relationship. In other words, these θ parameters in the ‘current version’ and the ‘original version’ are almost ‘in step’. The slope and intercept of the red fitted line are about 0.59 and 0.01, respectively, showing only small differences from the diagonal $y = x$. Although the deviation is more pronounced at some points (e.g., θ_6), the overall agreement is very high, indicating that the differences between the two estimation methods are minimal.

4.4.3 Reasons for the discrepancy

We attempted various optimization methods, including the BFGS algorithm, to replicate the results; however, some discrepancies remained. These differences are likely due to variations in parameter initialization, optimization settings, and numerical precision. Yu and Lam do not provide sufficient implementation details, such as initial values, specific optimisation methods, stopping criteria, or how tie probabilities were estimated, which makes exact reproduction difficult.
A key factor is how the tie parameter ϕ is modeled. In our implementation, ϕ is restricted to the interval $(0, 1)$ using a logistic transformation to improve optimization stability. Since Yu and Lam do not explain how ϕ was estimated or constrained, it is possible that different parameterizations led to

substantially different values, especially given the tie component's sensitivity.
Another important reason is a data error identified in the original paper by Yu and Lam. For example, the MP score for Pair 2 in Board 16 appears to be incorrect and was corrected in our dataset. Even small inconsistencies like this can affect pairwise comparisons and influence parameter estimation, particularly for tie-related components.
While different optimization methods may also influence results, a more in-depth comparison of three algorithms is provided in Appendix A.2. The outcomes were nearly identical under these methods, but this does not fully rule out the effect of algorithmic settings, such as stopping conditions or search directions.

4.4.4 Impact of the discrepancy

Although the optimization results show numerical differences, some aspects of the model behavior remain consistent. In particular, the relative rankings of the skill parameters θ are unchanged, preserving the overall ordering of player strength.
However, the actual values of both θ and ϕ differ noticeably from those reported by Yu and Lam. These differences affect the model's output, especially in the predicted probability of ties and the interpretation of performance gaps between players. As a result, the model may yield different assessments of how evenly matched two players are or how likely a tie is to occur.
The parameter estimates in my thesis—although numerically different from those by Yu and Lam—are internally consistent. The values of θ and ϕ work together coherently within the permutation model to produce valid probabilistic predictions. This indicates that the model remains self-consistent, even if the parameters themselves differ.
Overall, both implementations reflect a coherent mapping between skill levels and tie behavior. This variation underscores that even when rankings are preserved, the model's quantitative predictions can shift, highlighting its sensitivity to implementation choices rather than indicating a problem with the model itself.

4.5 Likelihood ratio test

This section compares the full model (where all skill parameters are freely estimated) with the restricted model (where all skill parameters are assumed to be equal). A likelihood ratio (LR) test is used to determine if there is a statistically significant improvement in the full model compared to the restricted model.

4.5.1 Compared likelihood ratio test and one-way ANOVA

In the task of detecting whether eight pairs of players in a Bridge tournament have the same skill level, we chose the Likelihood Ratio Test (LRT) over the traditional one-way ANOVA. This is because there are significant differences in the assumptions and applicability of the two methods, and the complex nature of the Bridge data lends itself better to the modelled inference of the LRT.
One-way ANOVA assumes that the data are independent, normally distributed, and have equal variances, and determines whether there is a significant difference between groups by comparing the means of the different groups [29]. However, this method is based on a simple linear model requiring errors to obey a normal distribution. In Bridge tournaments, the data are derived from two-by-two comparisons between pairs of players, which presents a nonlinear probability structure (nonlinear ranking model) and fails to satisfy the underlying assumptions of ANOVA.
In contrast, the Likelihood Ratio Test (LRT) is based on a nonlinear probability model and is more suitable for the data structure of a Bridge tournament. In LRT, the null hypothesis assumes that all players have the same skill level and the model can be reduced to a common skill parameter; the alternative hypothesis allows each pair to have a different skill parameter. By maximising the likelihood values under the null and alternative hypotheses and calculating the difference between the two, LRT provides a natural way of modelling inference [30]. According to Wilkes' theorem, the

LRT statistic obeys a chi-square distribution over large samples with degrees of freedom equal to the difference in the number of model parameters [45].
Thus, LRT provides a more flexible and precise analysis that captures differences in players' skill levels in a way that one-way ANOVA cannot.

4.5.2 Methodology

In the full model, each skill parameter θ_i $(i = 1, 2, \ldots, n)$ is allowed to vary freely, with a fixed condition $\theta_n = 0$.
The log-likelihood function for the full model is defined as:

$$LL_{\text{full}} = \sum_b \left[\sum_{(i,j)\in L_b} \log\left(\frac{\lambda_j}{\lambda_i + \lambda_j + \phi\sqrt{\lambda_i\lambda_j}}\right) + \sum_{(i,j)\in D_b} \left(\log(\phi) + 0.5\log(\lambda_i\lambda_j) - \log(\lambda_i + \lambda_j + \phi\sqrt{\lambda_i\lambda_j})\right)\right],$$

The restricted model assumes all skill parameters are equal, i.e., $\theta_1 = \theta_2 = \cdots = \theta_8 = \theta_{\text{common}}$. The tie probability ϕ remains the same as in the full model. The simplified log-likelihood function for each board is:

$$LL_{\text{restricted}} = m \cdot \log\left(\frac{1}{2+\phi}\right) + d \cdot \log\left(\frac{\phi}{2+\phi}\right),$$

where: m is the number of win-loss pairs (L_b), and d is the number of tied pairs (D_b).
The total log-likelihood is the sum across all boards:

$$LL_{\text{restricted}} = \sum_b \left[m \cdot \log\left(\frac{1}{2+\phi}\right) + d \cdot \log\left(\frac{\phi}{2+\phi}\right)\right].$$

The Likelihood Ratio (LR) statistic is defined as:

$$LR = -2\left(LL_{\text{restricted}} - LL_{\text{full}}\right).$$

The LR statistic measures the difference in fit between the two models. If LR is large, it indicates that the restricted model significantly underfits the data compared to the full model, rejecting the assumption that all skill parameters are equal.

4.5.3 Implementation

- In the restricted model, we assume *all teams have the same skill* (θ_{common}). Therefore, the probability of winning or losing is $1/(2+\phi)$, while the probability of a draw is $\phi/(2+\phi)$.
- The function above simply counts, for each board, how many win/loss pairs (m) and how many draw pairs (d) exist, then computes the corresponding log-likelihood contribution under this restricted assumption.
- We return $-\ell$ because most optimization routines (like BFGS) *minimize* the target function.

Algorithm 6 Restricted Model and LR Computation

function LOGLIKELIHOODRESTRICTED(x, $boardsData$, n, T)
  $\theta_{\text{common}} \leftarrow x[0]$
  $\psi \leftarrow x[1]$
  $\phi \leftarrow \exp(\psi)/(1+\exp(\psi))$
  $\ell \leftarrow 0$
  for all bd **in** $boardsData$ **do**
    $L_b \leftarrow bd["L_b"]$, $D_b \leftarrow bd["D_b"]$
    $m \leftarrow \text{len}(L_b)$, $d \leftarrow \text{len}(D_b)$
    $\ell_{\text{board}} \leftarrow m \cdot [-\ln(2+\phi)] + d \cdot [\ln(\phi) - \ln(2+\phi)]$
    $\ell \leftarrow \ell + \ell_{\text{board}}$
  return $-\ell$ ▷ Negative log-likelihood for minimization

- We initialize the restricted-model parameters to zero: the first element $x[0]$ is θ_{common} (not used in the probability formula) and $x[1]$ is ψ, which determines ϕ.
- After fitting the restricted model via BFGS, we get $\ell_{\text{res}} = -$res_res.fun.
- Finally, the Likelihood Ratio (LR) is computed as

$$-2\left(\ell_{\text{res}} - \ell_{\text{full}}\right).$$

  If LR is sufficiently large, the restricted model is significantly worse than the full model.
- In a formal hypothesis test, one can compare LR to a χ^2 distribution with degrees of freedom equal to the difference in parameter counts.

Algorithm 7 Fitting the Restricted Model and Computing the LR Statistic

$x\_init\_res \leftarrow [0.0, 0.0]$ ▷ $\theta_{\text{common}} = 0$, $\psi = 0 \implies \phi = 0.5$ initially
res_res $\leftarrow$ MinimizeBFGS(LogLikelihoodRestricted, x_init_res)
$\ell_{\text{res}} \leftarrow -$ res_res.fun
print "Restricted model log-likelihood: ", ℓ_{res}
print "Restricted model parameters: ", res_res.x
▷ Recall that ℓ_{full} was computed earlier from the full model.
LR $\leftarrow -2\left(\ell_{\text{res}} - \ell_{\text{full}}\right)$
print "Likelihood Ratio: ", LR

4.5.4 Results and interpretation

Using a likelihood ratio test, the full model (in which the skill parameters θ are freely estimated) and the restricted model (in which all skill parameters are assumed to be equal) were compared. The likelihood ratio statistics obtained from our results are $LR = 37.76$.

Under the null hypothesis, the likelihood ratio statistic LR follows a chi-squared distribution with 7 degrees of freedom. At a significance level of $\alpha = 0.05$, the critical value is: $\chi^2_{0.05,7} = 14.07$. The corresponding P-value for $LR = 37.76$ is: $P \ll 0.001$. Since the P-value is significantly smaller than 0.05, we reject the null hypothesis that all skill parameters are equal.

In the original study, the likelihood ratio statistic was reported as $LR = 21.85$. Although our result ($LR = 37.76$) is higher, the conclusion remains the same. The null hypothesis of equal skill parameters is rejected, and significant differences exist among the skill parameters θ of the eight Bridge pairs.

4.5.5 Comparison between MP total score ranking and θ ranking

The table below presents the total MP scores, their corresponding ranks, and the ranks derived from the estimated skill parameters θ:

Pair	MP Score	MP Rank	θ Value	θ Rank
1	44.5	5	0.1336	5
2	45.5	3	0.1899	3
3	29.5	7	-0.7132	7
4	49.0	2	0.3866	2
5	45.0	4	0.1838	4
6	24.5	8	-0.9896	8
7	56.0	1	0.7906	1
8	42.0	6	0.0000	6

The rankings derived from the total MP scores are identical to those obtained from the estimated skill parameters θ. This consistency confirms that the MP scores are a valid and reliable measure for comparing the skills of the Bridge pairs.

4.6 Analysis of Tie Frequencies in the Permutation Model

The main objective of this section is to compute the expected frequency of tied results under the permutation model, based on the estimated parameters θ and ϕ. When $T = 4$ tables play the same board, a tie is defined as the event that two or more tables obtain identical match-point scores for both the North–South and East–West pairs on that board. Whenever this occurs, the corresponding tables are said to form a tie group.
To analyze tie patterns systematically, we enumerate all possible ways of grouping the four tables according to score equality. That is, we list all the partitions of 4 tables where one or more subsets contain equal scores. For each such grouping, we compute the number of tie pairs—unordered pairs of tables that received the same score—using combinatorics. The total number of tie pairs on board b is denoted by $|\mathcal{D}_b|$.

4.6.1 Tie groupings and their combinatorial counting

Lemma 3: Tied Pair Counting

Let a tournament round consist of T tables. If a subset of k tables have identical scores, the number of tied pairs $|\mathcal{D}|$ within this subset is given by:

$$|\mathcal{D}| = \binom{k}{2} = \frac{k(k-1)}{2}$$

The total number of tied pairs for a board is the sum of these values across all tie groups.

The following table lists all possible ways to partition the four tables into tie groups, along with the corresponding number of tied pairs:

Table 8: Possible Tie Groupings and Their Tied Pair Counts for $T = 4$

Grouping	Description	Number of Tied Pairs $\lvert\mathcal{D}_b\rvert$	Combinatorial Formula
(4)	All four tables have the same score.	6	$\binom{4}{2} = 6$
$(3, 1)$	Three tables share the same score, and the fourth has a different score.	3	$\binom{3}{2} = 3$
$(2, 2)$	Two pairs of tables, each with identical scores.	2	$\binom{2}{2} + \binom{2}{2} = 2$
$(2, 1, 1)$	One pair of tied tables, while the other two have distinct scores.	1	$\binom{2}{2} = 1$
$(1, 1, 1, 1)$	All four tables have distinct scores (no ties).	0	–

A clear instance of a $(3, 1)$ tie structure can be found in Board 17. The North–South raw and match-point (MP) scores for the four tables are summarized in Table 9.

Table 9: Raw Scores and MP Scores for N/S Pairs on Board 17

N/S Pair	Raw Score	MP Score
Pair 6	-460	1
Pair 7	-460	1
Pair 2	-400	3
Pair 8	-460	1

As shown in the table, Pairs 6, 7, and 8 all received identical raw scores of -460, which translated into the same MP score of 1. This implies that these three tables were tied on this board. In contrast, Pair 2 achieved a distinct raw score of -400, resulting in a higher MP score of 3.
To determine the number of tied pairs, we list all unordered combinations $(6, 7), \quad (6, 8), \quad (7, 8)$. Each of these constitutes a tie pair. Hence, the total number of tied pairs in this board is $\lvert\mathcal{D}_b\rvert = \binom{3}{2} = 3$

4.6.2 Results

Table 10: Comparison of Observed and Expected Frequencies of Tied Pairs

Number of tied pairs	Observed frequency (Original)	Expected frequency (Original)	Expected frequency (Original parameters)	Expected frequency (My parameters)
0	14	12.83	12.86	19.73
1	11	12.61	12.60	7.82
2	1	1.40	1.37	0.34
3	2	1.13	1.14	0.11
6	0	0.03	0.03	0.00

- **Number of tied pairs:** This column indicates the number of tied pairs observed on each board, occurring when the players score the same. Values include 0, 1, 2, 3 and 6 ties.
- **Observed frequency (Original):** This is the frequency of tied pairs per deck as observed from actual tournament data.
- **Expected frequency (Original):** This column shows the theoretically expected frequencies calculated using the model and parameters θ and ϕ) provided in the original paper. These values were taken directly from the original conclusions in the paper.

- **Expected frequency (Original parameters):** I used the parameters provided by Yu and Lam's paper (θ and ϕ) and implemented the same model with our code. The expected frequencies obtained were very close to the original expected values, verifying that our implementation was correct.
- **Expected frequency (My parameters):** Using our independently calculated Maximum Likelihood Estimates (MLE) for θ and ϕ, I recalculated the expected frequencies. These values differ from the original results, suggesting that the choice of parameters affects the predictive distribution of ties.

By using the parameters θ and ϕ) by Yu and Lam, and recalculating the expected frequencies, we obtained results consistent with those reported in the paper by Yu and Lam. This confirms the correctness of our model implementation and validates our approach.

4.7 Conclusion

In sections 4.1 and 4.2, we presented the four-table Howell rotating duplicate Bridge dataset from [34], focusing on how each board was organized and setting up the maximum likelihood estimation (MLE) pseudo code step by step. These preparatory steps clearly mapped the raw data (including pair assignments and match-point scores) and the statistical model.
Next, sections 4.3 and 4.4 described how we estimated skill parameters θ and the tie probability ϕ using a quasi-Newton (BFGS) method, and how we applied a parametric Bootstrap to quantify the uncertainty of those estimates. In comparing our replicated parameters to the original work, we found that the overall skill ranking was consistent despite some numerical differences. This close alignment confirms that the permutation model remains robust under slightly varied conditions.
In section 4.5, we performed a likelihood ratio (LR) test, comparing the full model (distinct skill parameters) to a restricted model (common skill parameter). The LR statistic rejected the null hypothesis, indicating significant skill disparities among the eight player pairs. This result matched the primary conclusion from [34] and reinforced the model's ability to detect meaningful performance gaps.
Finally, section 4.6 examined tie frequencies by enumerating the possible scoring patterns and contrasting the observed counts with those predicted by the estimated parameters. Although our tie probability estimate diverged somewhat from Yu and Lam's result, both results captured the essential properties of ties in a small, competitive Bridge environment.
Overall, this chapter demonstrates that the permutation model, equipped with MLE, bootstrap-based inference, and a formal LR test, can effectively uncover and validate differences in player abilities. With the permutation model's strengths and limitations clarified, we will examine data from the 2024 China National Bridge Championships. We will apply the same modeling framework to a much larger, high-stakes event, aiming to confirm or refine the insights gleaned so far.

5 Analysis of the 2024 China National Bridge Championship

5.1 Data preparation and representation

Since its first participation in the Asian Games in 1974, Chinese Bridge has rapidly developed into an essential intellectual sport in China. 1982 saw the establishment of the Chinese Bridge Association and its first participation in the World Bridge Championships, which gradually established its dominant position in the Asian region. The Chinese team has performed well in all the Asia-Pacific Bridge Championships, with the men's team winning 8 Asia-Pacific Championships and 1 Asian Cup Championship. In contrast, the women's team is even more outstanding, winning 16 Asia-Pacific Championships and 1 Asian Cup Championship [5]. These achievements show the intense competitiveness of the Chinese Bridge.
Bridge's popularity in China grows daily, from professional competition to universal participation, covering youth, adult, and senior groups. The China Bridge Championships, as the highest level of national tournaments, contains several subdivisions, such as Open Pairs, Mixed Pairs, Teams, and Youth competitions, etc., to satisfy the needs of different Bridge enthusiasts. The schedules and structures for all events at the 2024 National Bridge Championships are taken from the official scoring platform, GemBridge [19].

5.1.1 Dataset: 2024 National Bridge Championship Open Pairs Final

The dataset used in this study is from the final stage of the Open Pairs event of the 2024 National Bridge Championships held in Leshan City, Sichuan Province. The Open Pairs tournament consists of three stages [19]:

Stage 1: Open pairs preliminary round

- **Participants:** 161 pairs (322 players).
- **Tables:** There are 80 tables, with a pair of players taking a turn in each round.
- **Boards:** A total of 44 boards, with 4 boards played per round. Each board has been played by 80 NS pairs and 80 EW pairs.
- **Pairing Structure:** Each pair played against 11 other pairs but did not face all pairs. The pairs that had their turn played 10 rounds (40 boards), and the other pairs played 11 rounds (44 boards).
- **Scoring:** Match Points (MP) were calculated for each pair based on their performance on each board relative to other pairs, and the MP scores were summed up.
- **Progression:** The top 35 pairs based on cumulative MP scores advanced to the semifinals.

Stage 2: Open pairs semifinal round

- **Participants:** 35 pairs (70 players).
- **Tables:** There are 17 tables, with a pair of players taking a turn in each round.
- **Boards:** A total of 44 boards, with 4 boards played per round. Each board has been played by 17 NS pairs and 17 EW pairs.
- **Pairing Structure:** Each pair played against 11 other pairs but did not face all pairs. The pairs that had their turn played 10 rounds (40 boards), and the other pairs played 11 rounds (44 boards).
- **Progression:** The top 12 pairs based on cumulative MP scores advanced to the final stage.

Stage 3: Open pairs final round

- **Participants:** 12 pairs (24 players).
- **Tables:** 6 tables.
- **Boards:** A total of 44 boards.
- **Pairing Structure:** Each pair played against all other 11 pairs, ensuring direct competition with every other pair.
- **Results:**The winner is ranked first based on the cumulative MP score.

The final stage of the Open Pairs event was chosen for the following reasons:

- The Howell rotation system used in the final stage aligns well with the assumptions of the permutation model and ensures balanced competition.
- The two-by-two pairwise structure between all pairs provided minimised bias and is consistent with the assumptions of the permutation model.
- The focus on the final stage helps to capture the skill level of the top performers. This data selection reduces the impact of outliers and facilitates the validation of our model.

5.1.2 The time schedule

The data is extracted from the official website of the Chinese Contract Bridge Association [6]. The table below outlines the schedule for the final stage of the 2024 National Bridge Championships Open Pairs. The 11 rounds held on April 15 2024, were evenly distributed throughout the day, with each round lasting approximately 32 minutes, with a break scheduled after the sixth round to ensure that the players were focused [19].

Date	Time	Stage of Competition
2024-04-15	09:00–09:32	Doubles Final Round 1
2024-04-15	09:35–10:07	Doubles Final Round 2
2024-04-15	10:10–10:42	Doubles Final Round 3
2024-04-15	10:45–11:17	Doubles Final Round 4
2024-04-15	11:20–11:52	Doubles Final Round 5
2024-04-15	11:55–12:27	Doubles Final Round 6
2024-04-15	14:30–15:02	Doubles Final Round 7
2024-04-15	15:05–15:37	Doubles Final Round 8
2024-04-15	15:40–16:12	Doubles Final Round 9
2024-04-15	16:15–16:47	Doubles Final Round 10
2024-04-15	16:50–17:22	Doubles Final Round 11

Table 11: Schedule for Doubles Final on April 15, 2024

5.1.3 Howell Movement

Seating arrangements and participant details for the 2024 National Bridge Championships Open Pairs Finals have been extracted from the official scoring website GemBridge [18].

Table 12: NS and EW Pair Assignments by Round and Board

Round	Boards	NS Pairs ($\alpha(b,t)$)	EW Pairs ($\beta(b,t)$)
1	1–4	12, 11, 3, 9, 8, 7	1, 2, 10, 4, 5, 6
2	5–8	12, 1, 4, 10, 9, 8	2, 3, 11, 5, 6, 7
3	9–12	12, 2, 5, 11, 10, 9	3, 4, 1, 6, 7, 8
4	13–16	12, 3, 6, 1, 11, 10	4, 5, 2, 7, 8, 9
5	17–20	12, 4, 7, 2, 1, 11	5, 6, 3, 8, 9, 10
6	21–24	12, 5, 8, 3, 2, 1	6, 7, 4, 9, 10, 11
7	25–28	12, 6, 9, 4, 3, 2	7, 8, 5, 10, 11, 1
8	29–32	12, 7, 10, 5, 4, 3	8, 9, 6, 11, 1, 2
9	33–36	12, 8, 11, 6, 5, 4	9, 10, 7, 1, 2, 3
10	37–40	12, 9, 1, 7, 6, 5	10, 11, 8, 2, 3, 4
11	41–44	12, 10, 2, 8, 7, 6	11, 1, 9, 3, 4, 5

5.1.4 MP scores by board and pair

The dataset table 24 has been carefully compiled by collating the results of each game played during the tournament. There were 44 boards and 6 tables in each round, and the dataset contains 528 entries detailing the Total Points (TP) of the North-South (NS) and East-West (EW) pairings, as well as their respective pairing assignments. This dataset provides the basis for MP scores analysis and ranking modelling of the results of the championship.
After obtaining the detailed total scores for the 528 pairing points, the raw total scores are converted to comparative scores to obtain Match Point (MP) scores. The MP scores provide the relative rankings of each pair in each deck, a method which ensures that the assessments are standardised and reflect the overall level of each pair in the tournament. The following table lists the 528 MP scores obtained based on the total scores for the 44 boards and 12 pairs of players participating in the tournament. Each row corresponds to a board, and each column represents the score of a pair of players [16].

Table 13: MP Scores by Board and Pair from the 2024 Open Pairs Final

Board 1-44	**Pair 1**	**Pair 2**	**Pair 3**	**Pair 4**	**Pair 5**	**Pair 6**	**Pair 7**	**Pair 8**	**Pair 9**	**Pair 10**	**Pair 11**	**Pair 12**
1	4	0	8	6	8	10	0	2	4	2	10	6
2	10	5	5	8	1	1	9	9	2	5	5	0
3	2	2	0	7	7	2	8	3	3	10	8	8
4	2	7	8	10	2	7	3	8	0	2	3	8
5	0	4	10	6	0	4	8	2	6	10	4	6
6	8	10	2	8	7	2	7	3	8	3	2	0
7	4	1	6	9	6	10	6	4	0	4	1	9
8	0	4	10	2	4	4	0	10	6	6	8	6
9	7	8	7	2	3	7	7	0	10	3	3	3
10	10	4	2	6	0	6	0	6	4	10	4	8
11	4	2	8	8	6	0	2	8	2	8	10	2
12	8	0	3	10	2	0	6	3	7	4	10	7
13	9	4	2	10	8	6	1	6	1	9	4	0
14	7	9	7	3	3	1	3	3	9	1	7	7
15	8	6	1	9	9	4	2	4	0	10	6	1
16	3	7	0	2	10	3	7	0	4	6	10	8
17	3	8	7	10	7	0	3	2	7	7	3	3
18	3	8	10	8	2	2	0	2	7	7	3	8
19	2	9	1	4	10	6	9	1	8	4	6	0
20	9	2	1	2	8	8	9	8	1	4	6	2
21	3	0	8	7	10	4	0	3	2	10	7	6
22	9	3	9	7	3	7	7	3	1	7	1	3
23	9	1	9	6	1	4	9	4	1	9	1	6
24	5	5	0	10	5	5	5	5	5	0	10	5
25	6	4	4	10	6	4	6	6	4	0	6	4
26	7	3	10	6	10	3	2	7	0	4	0	8
27	7	3	9	9	7	3	7	7	3	1	1	3
28	8	0	10	2	7	3	2	8	8	7	3	2
29	6	10	0	4	4	0	4	2	6	10	6	8
30	2	8	2	8	8	8	2	2	8	2	2	8
31	7	7	3	3	9	1	0	4	10	9	1	6
32	1	8	8	2	2	9	4	2	1	8	6	9
33	7	3	7	3	7	3	10	10	3	0	0	7
34	0	4	6	4	6	10	2	1	9	9	8	1
35	4	8	4	6	2	6	4	0	0	10	6	10
36	3	7	10	0	10	0	3	7	8	4	2	6
37	1	1	1	6	4	9	9	9	1	4	9	6
38	0	1	1	4	6	9	9	10	2	6	8	4
39	8	7	7	0	10	3	3	2	3	7	7	3
40	7	8	4	0	7	3	10	6	2	3	10	0
41	9	5	5	9	2	8	1	5	5	1	0	10
42	10	6	0	8	4	6	2	10	4	0	4	6
43	1	2	1	4	10	0	6	9	8	9	6	4
44	8	6	9	4	6	10	10	8	2	6	6	10

5.1.5 Final ranking

Final rankings for the 2024 National Bridge Championships Open Pairs Final from the official scoring platform GemBridge [17]. The numbers in the first column are the rankings; the names in the third column are the names of the two players in a pair; the MPs in the fourth column are the scores converted by the common method of Bridge; and the last column is the group number and combination name for each pair.

排名	%	姓名	MPs	带分	#	
1	60.272%	王国强-张立雄	253	2.772%	4#	中国企业体协(陕西茅台1935)
2	59.139%	贾新春-宁军	248	2.775%	5#	创远设备1
3	57.687%	孙明皓-梁於河	240	3.142%	10#	四月春风
4	54.61%	刘书平-沈青峰	228	2.792%	1#	江苏佳宏
5	53.717%	张扩-邬雪男	222	3.262%	12#	江苏佳宏1
6	53.667%	曹大男-潘利平	222	3.212%	11#	苏州友通
7	52.969%	邹崇松-马国伟	221	2.742%	3#	攀枝花桥友
8	50.98%	何勇-杨天奎	211	3.025%	8#	君逸数码
9	50.222%	刘钧-石峰	209	2.722%	2#	筑油队
10	48.72%	张崇斌-周清华	202	2.811%	7#	步润植思
11	47.334%	郑伟-李爱民	196	2.789%	6#	三星堆2
12	45.78%	曾翔-姚涛	188	3.053%	9#	吉林鼎元

Figure 10: Final rankings for the 2024 National Bridge Championships Open Pairs

In the final ranking of the 2024 National Bridge Championship Open Pairs, the top three pairs were highly competitive. Wang Guoqiang and Zhang Lixiong (pair 4) secured first place with **253 MP**, followed closely by Jia Xinchun and Ning Jun (pair 5) (**248 MP**) and Sun Minghao and Liang Yuhe (pair 10)(**240 MP**).
Specifically, the 5th place (pair 12) and the 6th place (pair 11) both scored **222 MP**. Because they have the same total score, they should generally be ranked side by side. If additional criteria for determining the rankings were introduced, their rankings would be different. For example, this difference can be determined by the IMP score or other tie-breaker rules, such as head-to-head results or performance in key rounds.

The actual rank order is thus:

$$\text{pair}\,4 > \text{pair}\,5 > \text{pair}\,10 > \text{pair}\,1 > \text{pair}\,12 > \text{pair}\,11 > \text{pair}\,3 > \text{pair}\,8 > \text{pair}\,2 > \text{pair}\,7 > \text{pair}\,6 > \text{pair}\,9.$$

5.2 Results interpretation of the permutation model

```
Converged: True
Log-likelihood: -696.4726315141663
θ_1 = 0.0667
θ_2 = -0.1456
θ_3 = -0.0110
θ_4 = 0.3478
θ_5 = 0.2907
θ_6 = -0.2908
θ_7 = -0.2240
θ_8 = -0.1232
θ_9 = -0.3803
θ_10 = 0.2017
θ_11 = -0.0001
θ_12 = 0.0000
Φ = 0.6187

Ranking of θ by skill level (from highest to lowest):
Rank 1: θ_4 = 0.3478
Rank 2: θ_5 = 0.2907
Rank 3: θ_10 = 0.2017
Rank 4: θ_1 = 0.0667
Rank 5: θ_12 = 0.0000
Rank 6: θ_11 = -0.0001
Rank 7: θ_3 = -0.0110
Rank 8: θ_8 = -0.1232
Rank 9: θ_2 = -0.1456
Rank 10: θ_7 = -0.2240
Rank 11: θ_6 = -0.2908
Rank 12: θ_9 = -0.3803
```

Figure 11: Ranking of the Permutation Model with 44 boards

From figure 11, we see that the BFGS optimization has successfully converged, reaching a final negative log-likelihood of -696.473. Each skill parameter θ_i reflects a pair's relative strength, with $\theta_{12} = 0$ taken as the baseline for identifiability. The estimates show that θ_4 is the highest (0.3478) whereas θ_9 is the lowest (-0.3803), This implies a moderate skill gap across the 12 teams.
Meanwhile, the negative log-likelihood of -696.47 confirms a reasonable fit to the observed data, and the efficient convergence under BFGS ensures that both $\{\theta_i\}$ and ϕ are reliably estimated. This highlights once again the importance of reaching a converged solution: Without convergence, the model might produce unstable or biased parameter estimates. In contrast, the final θ values and $\phi \approx 0.6187$ here show a stable interpretation consistent with the observed competition levels.
The permutation model is validated by its high consistency with the actual match results. The estimated skill parameters are in perfect agreement with the actual match point (MP) rankings, proving the accuracy of the model in assessing the skill level of the players. This confirms the viability of the model as a ranking method.
In addition, the dataset used in this study is more extensive than those studied in previous papers, further proving the reliability of the findings. The high level of agreement between the calculated rankings and the actual results highlights the usefulness of the model and its potential as an improved alternative for ranking competitive Bridge players.

5.3 Bootstrap analysis

Figure 12 lists the estimates of θ and γ obtained by fitting a permutation model (allowing for ties) to our Bridge data, along with the 95% confidence intervals obtained by the parameter Bootstrap ($B = 1000$ replicates). For identification purposes, the baseline parameter θ_{12} is set to zero so that its reported point estimate is always 0.0000.

```
=== Starting Parametric Bootstrap Analysis (B=1000) ===

Parametric Bootstrap done with B=1000.
95% confidence interval for each θ_i:
θ_1 = 0.0667  95% CI = [-0.3484, 0.4855]
θ_2 = -0.1456  95% CI = [-0.5656, 0.2627]
θ_3 = -0.0110  95% CI = [-0.4301, 0.4069]
θ_4 = 0.3478  95% CI = [-0.0701, 0.7855]
θ_5 = 0.2907  95% CI = [-0.1513, 0.7447]
θ_6 = -0.2908  95% CI = [-0.7253, 0.1480]
θ_7 = -0.2240  95% CI = [-0.6601, 0.2070]
θ_8 = -0.1232  95% CI = [-0.5469, 0.2694]
θ_9 = -0.3803  95% CI = [-0.8146, 0.0339]
θ_10 = 0.2017  95% CI = [-0.2038, 0.6442]
θ_11 = -0.0001  95% CI = [-0.4117, 0.4126]
θ_12 = 0.0000  95% CI = [0.0000, 0.0000]
```

Figure 12: Point estimates and 95% Bootstrap confidence intervals for team parameters θ_i

From figure 12, all intervals include zero, indicating that none of the teams appear statistically superior or inferior to the reference team ($\theta_{12} = 0$) based on these data alone. Notably, while the point estimates for θ_4 and θ_5 are slightly above zero—suggesting a potentially better performance—and that of θ_9 is around -0.3803, these intervals cross zero, so we cannot rule out random variation as the cause of these differences.

5.4 Likelihood ratio test (LRT)

Using a likelihood ratio test, we compared the *full model*—in which the skill parameters $\theta_1, \ldots, \theta_{12}$ (with one baseline fixed) are freely estimated—with the restricted model, wherein all skill parameters are assumed equal. The resulting likelihood ratio statistic from our data is $\mathrm{LR} = 24.638$.

Under the null hypothesis that all skill parameters are equal, the statistic LR approximately follows a chi-squared distribution with 11 degrees of freedom (i.e. $\Delta df = 11$ if the restricted model removes $(12 - 1) = 11$ skill parameters). At a significance level of $\alpha = 0.05$, the critical value is $\chi^2_{0.05,\,11} \approx 19.68$. Since $\mathrm{LR} = 24.638 > 19.68$, the p-value is therefore < 0.05.

Hence, we reject the null hypothesis that all skill parameters are equal. In other words, allowing each θ_i to vary independently yields a significantly better fit to the observed data, indicating genuine differences in skill among these 12 players (or pairs).

5.5 Tie groupings and their combinatorial enumeration for $T = 6$

In order to analyze the frequency of ties when $T = 6$ tables play the same board, we enumerate all possible partitions of the six tables into groups with identical scores. Each configuration corresponds to a tie grouping, and for each grouping, we compute the number of tied pairs based on Lemma 3, which states that a group of k tied tables contributes $\binom{k}{2}$ pairs to the total. The sum of all such contributions across tie groups gives the total number of tied pairs $|\mathcal{D}_b|$ for a particular board.
The table below summarizes all admissible tie groupings, their interpretations, the number of tied pairs for each case, and the associated combinatorial expressions:

Table 14: Tie Groupings and Corresponding Tied Pair Counts for $T = 6$

Grouping	Interpretation	Tied Pairs $\lvert\mathcal{D}_b\rvert$	Combinatorial Expression
(6)	All six tables have the same score.	15	$\binom{6}{2} = 15$
$(5, 1)$	Five tables tie; one table has a distinct score.	10	$\binom{5}{2} = 10$
$(4, 2)$	Four tables tie; the other two form a separate tie.	7	$\binom{4}{2} + \binom{2}{2} = 6 + 1 = 7$
$(4, 1, 1)$	Four tables tie; the remaining two each differ.	6	$\binom{4}{2} = 6$
$(3, 3)$	Two disjoint groups of three tied tables.	6	$\binom{3}{2} + \binom{3}{2} = 3 + 3 = 6$
$(3, 2, 1)$	Three tables tie; two tie separately; one differs.	4	$\binom{3}{2} + \binom{2}{2} = 3 + 1 = 4$
$(3, 1, 1, 1)$	Three tables tie; three others are distinct.	3	$\binom{3}{2} = 3$
$(2, 2, 2)$	Three separate groups of two tied tables.	3	$\binom{2}{2} + \binom{2}{2} + \binom{2}{2} = 1 + 1 + 1 = 3$
$(2, 2, 1, 1)$	Two tie groups of size two; two distinct tables.	2	$\binom{2}{2} + \binom{2}{2} = 1 + 1 = 2$
$(2, 1, 1, 1, 1)$	One tie pair; all others have unique scores.	1	$\binom{2}{2} = 1$
$(1, 1, 1, 1, 1, 1)$	All six tables have distinct scores (no ties).	0	–

Table 13 compares the observed frequency (Observed) of different numbers of tie pairs k in 44 deals with the theoretical expectations (Expected). Here, k represents how many pairs of tables (out of 15 possible pairwise comparisons among 6 tables) ended in a tie within a single deal.
The model substantially overestimates the probability of having extremely few ties, such as $k = 0$ or $k = 1$. Specifically, it predicts around 13.27 deals with no ties ($k = 0$), but only 1 was observed, and 20.12 deals with a single tie ($k = 1$), whereas 12 were actually observed. In contrast, the model underestimates the probability of observing moderate or higher numbers of ties (e.g., $k = 3$, $k = 4$, $k = 6$), where the observed frequencies exceed the model's predictions several times. For instance, $k = 3$ has a predicted value of only 2.30 yet an observed count of 7; $k = 4$ has an expected value of 0.88 but occurred 8 times; and even for rarer events such as 6 or more ties, the model assigns nearly zero probability, although such scenarios did happen in the actual data.

=== Distribution of ties (Observed vs. Expected) ===

k	Observed	Expected
0	1	13.271
1	12	20.117
2	6	7.324
3	7	2.299
4	8	0.879
6	5	0.093
7	3	0.015
10	1	0.001
15	1	0.000

Figure 13: Comparison of Observed and Expected Frequencies of Tied Pairs

Such discrepancies could arise from several factors. Firstly, real Bridge games may be affected by tactical adjustments, fluctuations in player performance, and other random factors that deviate from the model's independence assumptions (similar reasons are analysed in detail in the section 6.4). Second, the permutation model employs a single global tie parameter, ϕ, which may not capture changes in tie trends between pairs of teams. The changes in the. Third, our data are derived from assumptions about the scores of MPs already in the tournament. Some of the existing scoring rules or constraints in tournaments may introduce structural biases that the model does not explain.

5.6 Conclusion

In section 5.1, we introduced the 2024 National Bridge Championships Open Pairs Final dataset, covering its multi-stage structure (preliminary, semifinal, and final), daily scheduling, Howell movement details, board-by-board MP scores, and the official final ranking. This material established a robust empirical context in which each pair competed against all others, ensuring balanced comparisons suitable for statistical modeling.

In section 5.2, we interpreted the permutation model's estimation results for this championship dataset. The estimated skill parameters θ and the tie probability ϕ underscored a moderate skill gap among the 12 competing pairs, and a final negative log-likelihood around -696 indicated a reasonable fit. Bootstrap analysis in section 5.3 revealed broad confidence intervals for most θ_i, suggesting that only a few pairs' estimated skill levels diverged strongly from the chosen baseline.

Section 5.4 presented a likelihood ratio test (LRT) to compare the full, distinct-θ model against a restricted common-skill model. The significant LRT statistic confirmed that different skill parameters provide a substantially better fit, indicating genuine performance disparities across pairs. Finally, in section 5.5, we examined tie groupings for $T = 6$ tables and contrasted the observed frequency of tied pairs with those predicted by the permutation model. Despite the model's solid overall performance, the underestimation of higher tie frequencies pointed to complex dynamics.

Overall, the 2024 China National Bridge Championships dataset validates the permutation model's applicability to large-scale, elite tournaments. In the next chapter, we will introduce a new threshold-based model armed with lessons from both smaller-scale and major tournament data. This alternative formulation seeks to improve interpretability and tie-handling by anchoring outcomes on a skill-difference threshold, offering a fresh perspective on how ties arise.

6 Extended analysis of the threshold model

In earlier chapters, we examined the Davidson model as an extension of the Bradley–Terry framework to incorporate tie outcomes, using a tie parameter ϕ to adjust win probabilities. While effective in accounting for ties probabilistically, this approach offers limited interpretability, as ϕ indirectly affects the outcome probabilities without clearly distinguishing a tie as a third category. In this chapter, we introduce a modified approach that replaces the ϕ parameter with an explicit threshold γ, allowing ties to emerge deterministically when the skill difference between two players falls within a specified range. This new formulation provides clearer insight into how skill proximity leads to draws and offers a more transparent mechanism for modelling ties in competitive settings.

At the same time, in many actual Bridge or other tournament datasets, we have observed that if two pairs are close enough in skill, a draw is much more likely. This observation suggests an explicit modelling of ties as an independent outcome category. Hence, we propose a threshold-based approach in which a single parameter γ defines a tie region of width γ: whenever $|\theta_i - \theta_j| \leq \frac{\gamma}{2}$, those two sides are deemed to tie, otherwise, they have a decisive outcome. It is important to note that this tie decision is not based on directly observed draws between players i and j, but rather on their estimated skill parameters θ_i and θ_j, which are inferred from results against other opponents. Therefore, the outcome is indirectly affected by the strength and composition of each player's matchup history.

A significant benefit is interpretability: γ directly represents the half-width of the tie interval so that any difference in skill within $\pm(\gamma/2)$ corresponds to a tie. This approach provides a direct and clear mechanism linking tie outcomes explicitly to the difference in players' skills: within a certain threshold, ties become naturally likely, while beyond this threshold, the probability of decisive outcomes rapidly increases. Thus, the threshold model establishes an intuitive connection between the "tie" event and skill proximity, whereas the permutation model obscures this relationship implicitly.

Threshold model's practical implementation is more straightforward than Davidson's $\sqrt{\lambda_i \lambda_j}$ denominator or permutation enumeration; in Davidson-type models, the λ_i parameters may implicitly include information about a player's interaction with their opponents, such as participation rates or comparative strength, adding complexity to estimation and interpretation. In contrast, the threshold model relies only on the estimated skill parameters θ_i, which are treated independently and do not explicitly encode opponent-specific characteristics. In the threshold model, the log-likelihood depends simply on whether or not $|\theta_i - \theta_j|$ is within the tie boundaries, and so can be encoded directly. This flexible thresholding approach allows for multiple or category-specific thresholds $(\gamma_1, \gamma_2, \dots)$, making the model more adaptable. It improves the likelihood fit and provides more intuitive results, especially when minor skill differences naturally lead to a tie. Furthermore, the threshold framework extends naturally from the classical Bradley–Terry paradigm, thus demonstrating broader generality and flexibility for use in other competitive domains such as sports tournaments or rating systems, especially where tie occurrences are explicitly frequent.

6.1 Threshold model

6.1.1 Model setup

Suppose there are B boards indexed by $b \in \{1, \ldots, B\}$. On board b, we compare NS pair i versus EW pair j. Let $\theta_{\alpha(b,i)}$ be the skill parameter of the NS player i on board b, and similarly $\theta_{\beta(b,j)}$ for EW pair j. We introduce a threshold parameter $\gamma \geq 0$, which determines the "width" of the tie interval. Define the skill difference between pairs (b, i) and (b, j):

$$X_{b,i,j} \;=\; \theta_{\alpha(b,i)} \;-\; \theta_{\beta(b,j)} + \epsilon_{b,i,j} \tag{20}$$

where $\epsilon_{b,i,j} \sim \text{Logistic}(0, 1)$ is an independent random noise term. It accounts for randomness in observed outcomes not explained by skill differences, and enables a probabilistic formulation of match results. We adopt the logistic distribution for mathematical convenience, as it yields closed-form expressions for all relevant probabilities.

In this threshold model:

- If $X_{b,i,j} > \frac{\gamma}{2}$, we say pair i is better than pair j, denoted $R_{b,i} > R_{b,j}$.
- If $-\frac{\gamma}{2} \leq X_{b,i,j} \leq \frac{\gamma}{2}$, then pair i and pair j are tied on board b ($R_{b,i} = R_{b,j}$).
- If $X_{b,i,j} < -\frac{\gamma}{2}$, then pair i is considered worse than pair j, i.e. $R_{b,i} < R_{b,j}$.

6.1.2 Probability formulas

Define the logistic CDF as:

$$F(z) = \frac{1}{1 + e^{-z}} \tag{21}$$

Then the probabilities of losing, tying, and winning on board b, between pairs i and j, are given by:

$$P(R_{b,i} < R_{b,j}) = F\left(-\frac{\gamma}{2} - \left[\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}\right]\right) \tag{22}$$

$$P(R_{b,i} = R_{b,j}) = F\left(\frac{\gamma}{2} - \left[\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}\right]\right) - F\left(-\frac{\gamma}{2} - \left[\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}\right]\right) \tag{23}$$

$$P(R_{b,i} > R_{b,j}) = 1 - F\left(\frac{\gamma}{2} - \left(\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}\right)\right) \tag{24}$$

Remarks

- When $\theta_{\alpha(b,i)} \ll \theta_{\beta(b,j)}$, the difference $\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}$ is large negative, making $-\frac{\gamma}{2} - (\theta_{\alpha(b,i)} - \theta_{\beta(b,j)})$ large and positive. Hence $P(R_{b,i} < R_{b,j}) \approx 1$, indicating pair i is highly likely to lose.
- Ties occur whenever $(\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}) \in [-\frac{\gamma}{2}, \frac{\gamma}{2}]$, whose length is γ. Thus, $\gamma > 0$ ensures a nonzero probability of ties.
- If $\gamma = 0$, then $P(R_{b,i} = R_{b,j}) = 0$ for all (b, i, j), reducing the model to a no-tie Bradley–Terry scenario.

Proofs of Formulas (22), (23), (24)

Proof of formula (22)

From the model definition and (20), the condition $R_{b,i} < R_{b,j}$ implies:

$$X_{b,i,j} = (\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}) + \epsilon_{b,i,j} < -\frac{\gamma}{2},$$

where $\epsilon_{b,i,j} \sim \text{Logistic}(0,1)$. Then,

$$P(R_{b,i} < R_{b,j}) = P\left(\epsilon_{b,i,j} < -\frac{\gamma}{2} - (\theta_{\alpha(b,i)} - \theta_{\beta(b,j)})\right).$$

Using the CDF of the Logistic distribution:

$$P(R_{b,i} < R_{b,j}) = F\left(-\frac{\gamma}{2} - (\theta_{\alpha(b,i)} - \theta_{\beta(b,j)})\right).$$

Derivation of z:

$$P\left(\epsilon_{b,i,j} < z\right) = F(z), \quad z = -\frac{\gamma}{2} - \left(\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}\right). \tag{25}$$

□

Proof of formula (23)

From the model definition, $R_{b,i} = R_{b,j}$ implies:

$$-\frac{\gamma}{2} \leq X_{b,i,j} \leq \frac{\gamma}{2}.$$

Then,

$$P(R_{b,i} = R_{b,j}) = P\left(-\frac{\gamma}{2} \leq X_{b,ij} \leq \frac{\gamma}{2}\right) = P\left(-\frac{\gamma}{2} \leq \epsilon_{b,i,j} \leq \frac{\gamma}{2}\right).$$

Using the logistic CDF:

$$P(R_{b,i} = R_{b,j}) = P\left(-\frac{\gamma}{2} \leq \epsilon_{b,ij} \leq \frac{\gamma}{2}\right) = F\left(\frac{\gamma}{2} - \left(\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}\right)\right) - F\left(-\frac{\gamma}{2} - \left(\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}\right)\right)$$

□

Proof of formula (24)

The probability is given by:

$$P(R_{b,i} > R_{b,j}) = P\left(X_{b,ij} > \frac{\gamma}{2}\right) = 1 - P\left(X_{b,ij} \leq \frac{\gamma}{2}\right).$$

Thus:

$$P\left(X_{b,ij} \leq \frac{\gamma}{2}\right) = F\left(\frac{\gamma}{2} - \left(\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}\right)\right).$$

Substituting back, we get:

$$P(R_{b,i} > R_{b,j}) = 1 - F\left(\frac{\gamma}{2} - \left(\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}\right)\right).$$

Derivation of z:

$$P(\epsilon_{b,i,j} > z) = 1 - F(z), \quad z = \frac{\gamma}{2} - \left(\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}\right).$$

□

6.1.3 Theorem: Sufficient statistics

Theorem 6.1: Sufficient Statistics

Consider a tournament with B boards. On each board, pairwise comparisons yield events of $R_{b,i} < R_{b,j}$ (loss), $R_{b,i} > R_{b,j}$ (win), or $R_{b,i} = R_{b,j}$ (tie). Let r_i denote the total score for player i, and $\sum_{b=1}^{B} d_b$ represent the total number of tie comparisons across all boards.
Under the interval γ-model, if the pairwise comparisons fully represent each player's participation, the likelihood function $\ell(\{\theta_i\}, \gamma)$ can be factorized as:

$$\ell(\{\theta_i\}, \gamma) \propto \exp\left(\sum_i r_i \theta_i\right) \cdot \gamma^{\sum_{b=1}^{B} d_b}.$$

Thus, the sufficient statistics for $\{\theta_i\}$ and γ are $\{r_i\} \cup \left\{\sum_{b=1}^{B} d_b\right\}$.

Proof of theorem 6.1

The likelihood function across all B boards is:

$$\ell(\{\theta_i\}, \gamma) = \prod_{b=1}^{B} P(R_b),$$

where from (13) and (14):

$$P(R_b) \propto \prod_{(i,j)\in L_b} P(R_{b,i} < R_{b,j}) \prod_{(i,j)\in D_b} P(R_{b,i} = R_{b,j}).$$

Taking the logarithm:

$$\ln \ell(\{\theta_i\}, \gamma) = \sum_{b=1}^{B} \ln P(R_b).$$

Substituting $P(R_b)$:

$$\ln P(R_b) = \sum_{(i,j)\in L_b} \ln P(R_{b,i} < R_{b,j}) + \sum_{(i,j)\in D_b} \ln P(R_{b,i} = R_{b,j}).$$

See Remark i. Contribution of tie events for details :

$$\sum_{(i,j)\in D_b} \ln P(R_{b,i} = R_{b,j}) \propto d_b \ln \gamma.$$

See Remark ii. Contribution of lose events for details.

$$\sum_{(i,j)\in L_b} \ln P(R_{b,i} < R_{b,j}) \propto \sum_i r_i \theta_i.$$

Combining all contributions:

$$\ln \ell(\{\theta_i\}, \gamma) \propto \sum_i r_i \theta_i + \left(\sum_{b=1}^{B} d_b\right) \ln \gamma.$$

Exponentiating:

$$\ell(\{\theta_i\}, \gamma) \propto \exp\left(\sum_i r_i \theta_i\right) \cdot \gamma^{\sum_{b=1}^{B} d_b}.$$

□

Remarks

i. Contribution of tie events

Assuming $\theta_{\alpha(b,i)} \approx \theta_{\beta(b,j)}$, the probability (23) simplifies to:

$$P(R_{b,i} = R_{b,j}) \approx \frac{\gamma}{1 + \exp(\theta_{\alpha(b,i)} - \theta_{\beta(b,j)})}.$$

Taking the natural logarithm:

$$\ln P(R_{b,i} = R_{b,j}) \approx \ln \gamma + (\text{other constant terms}).$$

Summing over all tie events:

$$\sum_{(i,j)\in D_b} \ln P(R_{b,i} = R_{b,j}) \approx d_b \ln \gamma + (\text{other constant terms}).$$

Therefore,

$$\sum_{(i,j)\in D_b} \ln P(R_{b,i} = R_{b,j}) \propto d_b \ln \gamma.$$

ii. Contribution of lose events

From (24):

$$\sum_{(i,j)\in L_b} \ln P(R_{b,i} < R_{b,j}) = \sum_{(i,j)\in L_b} \ln F\left(-\frac{\gamma}{2} - (\theta_{\alpha(b,i)} - \theta_{\beta(b,j)})\right),$$

Using (21), we rewrite:

$$\ln F\left(-\frac{\gamma}{2} - (\theta_{\alpha(b,i)} - \theta_{\beta(b,j)})\right) = -\ln\left(1 + e^{\frac{\gamma}{2} + (\theta_{\alpha(b,i)} - \theta_{\beta(b,j)})}\right).$$

By linear approximation, for small z, the logistic function can be approximated as:

$$F(z) \approx e^z.$$

Thus:

$$\ln F(z) \approx z.$$

Applying this and from formula (25) :

$$\ln F\left(-\frac{\gamma}{2} - (\theta_{\alpha(b,i)} - \theta_{\beta(b,j)})\right) \approx -\frac{\gamma}{2} - (\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}).$$

$$\sum_{(i,j)\in L_b} \ln P(R_{b,i} < R_{b,j}) \approx \sum_{(i,j)\in L_b} -\frac{\gamma}{2} - \sum_{(i,j)\in L_b} (\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}).$$

The first term is a constant:

$$\sum_{(i,j)\in L_b} -\frac{\gamma}{2}.$$

The second term is reorganized based on player contributions:

$$\sum_{(i,j)\in L_b} (\theta_{\alpha(b,i)} - \theta_{\beta(b,j)}) \propto \sum_i r_i \theta_i,$$

where r_i represents the total score for player i. □

6.2 Validation of the threshold model through Yu–Lam data

The following sections will present the construction of the log-likelihood and gradient formulations of the threshold model, show how the model can be solved by numerical optimisation (e.g., BFGS), and compare its performance with existing methods (e.g., Davidson's method or passivation methods). This evidence will demonstrate the feasibility and advantages of threshold-based models for representing draws in competitive environments.
We start by validating our method against our method through code using the data from section 4.1.

6.2.1 The threshold model estimation

```
Converged: True
Final log-likelihood value: -144.29592643485478
Estimated gamma = 0.5557661903756365 (tie threshold width)
Theta_1 = 0.0432
Theta_2 = 0.1951
Theta_3 = -0.5991
Theta_4 = 0.3312
Theta_5 = 0.1669
Theta_6 = -0.8395
Theta_7 = 0.6588
Theta_8 = 0.0000

==== Ranking of player pairs from highest to lowest skill ====
  Rank 1: Pair 7,  θ =0.6588
  Rank 2: Pair 4,  θ =0.3312
  Rank 3: Pair 2,  θ =0.1951
  Rank 4: Pair 5,  θ =0.1669
  Rank 5: Pair 1,  θ =0.0432
  Rank 6: Pair 8,  θ =0.0000
  Rank 7: Pair 3,  θ =-0.5991
  Rank 8: Pair 6,  θ =-0.8395
```

Figure 14: Results of the threshold model estimation on Dataset from section 4.1

Figure 14 illustrates the outcome of fitting our threshold model to the data using BFGS optimization. The method attained *successful convergence*, as evidenced by the final negative log-likelihood value of -144.29592643. An important parameter in this model is γ, the half-width of the tie interval; here it is estimated as $\hat{\gamma} = 0.5558$, indicating that any two players/pairs whose skill difference is within ± 0.2779 are predicted to tie with significant probability.

Predicted to tie

In the threshold model, γ represents the "total width of the tie interval," meaning that $|\theta_i - \theta_j|$, is less than $\frac{\gamma}{2}$, they are judged to have tied. Therefore, if the estimated $\hat{\gamma} = 0.5558$, then the half-width of the interval is:

$$\frac{\hat{\gamma}}{2} = \frac{0.5558}{2} \approx 0.2779$$

This means that if the difference between any two players does not exceed ± 0.2779, the model considers them very likely to tie.

We fix $\theta_8 = 0$ for identifiability, so each θ_i reflects the relative advantage or disadvantage compared to pair 8. The resulting estimates show that pair 7 achieves the highest skill ($\theta_7 \approx 0.6588$), while pair 6 is the lowest ($\theta_6 \approx -0.8395$). The rank order is thus:

$$\text{pair 7} > \text{pair 4} > \text{pair 2} > \text{pair 5} > \text{pair 1} > \text{pair 8} > \text{pair 3} > \text{pair 6}.$$

This ordering matches intuitively with the performance patterns in the dataset and aligns well with alternative methods reported in previous literature.

The moderate value of γ around 0.556 suggests that ties are neither overly frequent nor vanishingly rare, consistent with the moderate level of competition in this dataset. Additionally, the relatively large spread in θ values ($\theta_7 - \theta_6 \approx 1.4983$) indicates a substantial skill gap from top to bottom. Overall, the final log-likelihood of -144.296 highlights a good fit to the observed outcomes, and the convergence of BFGS guarantees stable parameter inference.

6.2.2 Parametric Bootstrap analysis under the threshold model

Figure 15 lists the estimates of θ and γ obtained by fitting a threshold model (allowing for ties) to our Bridge data, along with the 95% confidence intervals obtained by the parameter Bootstrap ($B = 1000$ replicates). For identification purposes, the baseline parameter θ_8 is set to zero, so its reported point estimate is always 0.0000.

```
=== Starting Parametric Bootstrap (B=1000) ===

95% CI for each θ_i:
  θ_1 = 0.0432  CI= [-0.5810, 0.6731]
  θ_2 = 0.1951  CI= [-0.4499, 0.8209]
  θ_3 = -0.5991  CI= [-1.2805, -0.0296]
  θ_4 = 0.3312  CI= [-0.2446, 0.9472]
  θ_5 = 0.1669  CI= [-0.4452, 0.7905]
  θ_6 = -0.8395  CI= [-1.6303, -0.2679]
  θ_7 = 0.6588  CI= [0.0630, 1.3251]
  θ_8 = 0.0000  CI= [0.0000, 0.0000]

95% CI for gamma: [0.3602, 0.8501]
```

Figure 15: Parametric Bootstrap estimates for the threshold mode with 8 pairs

From Figure 15, the estimate $\theta_7 \approx 0.66$, with a confidence interval mainly in the positive range, suggests that pairing 7 has a higher level of competence than pairing 8 (the baseline), which is statistically unlikely to be judged as negative. In contrast, $\theta_3 \approx -0.60$ and $\theta_6 \approx -0.84$ lean toward negative values, with confidence intervals covering a large proportion of the negative values, suggesting that the probability of being evaluated as below baseline is higher under this model.
Several of the estimated θ_i values have wide confidence intervals that cross zero, suggesting that our data (which includes only 28 boards) does not provide strong statistical evidence to distinguish the ability of these board pairs from the baseline. This is expected given the relatively small sample size and the additional complexity associated with allowing ties.
The estimated tie breaker parameter is $\gamma \approx 0.56$, with a 95% confidence interval of approximately $[0.36, 0.85]$. Interpreted in terms of a threshold model, this suggests that when the skill difference $|\theta_i - \theta_j|$ is within approximately 0.18 to 0.43, a tie result is relatively likely. Since the data contain a certain number of tied pairs, an insignificant γ estimate is plausible. In summary, while the model provides a useful structure for analyzing ability when ties are possible, the wide confidence intervals reflect limited information from several boards. More data are needed to obtain tighter confidence intervals.

6.2.3 Comparison of tied-pair frequencies

```
Observed vs. Expected frequency of Number of tied pairs
k   Observed   Expected
0     14         13.57
1     11         10.43
2      1          3.36
3      2          0.58
4      0          0.06
5      0          0.00
6      0          0.00
```

Figure 16: Comparison of observed vs. expected tie distributions under threshold model with 8 pairs

Table 15: Observed vs. Expected Frequency of Tied Pairs

Number of tied pairs	Observed frequency	Expected(Original)	Expected(New)
0	14	12.83	13.57
1	11	12.61	10.43
2	1	1.40	3.36
3	2	1.13	0.58
4	0	0	0.06
5	0	0	0.00
6	0	0.03	0.00

Figure 16 compares the threshold model's results to the original table. In both approaches, most boards have 0–1 tied pairs (with close observed vs. expected counts), whereas $k \geq 2$ are comparatively rare. Our method overestimates $k = 2, 3$, just as the original model also deviates somewhat for moderate ties. Such differences stem from the single-γ plus independent-pairwise assumption, which can under- or overpredict multi-pair ties. Further enhancements (e.g. multiple γ or random board effects) may reduce these gaps.

6.3 Validation of the threshold model versus 2024 China Championship data

Then, we validate our method against our method through code using the data from section 5.1.4.

6.3.1 The threshold model estimation

In this threshold model figure17 , one pair (pair 12) is fixed to $\theta_{12} = 0$ for identifiability, and the BFGS optimization converges at a log-likelihood of approximately -696.5473. The estimated tie-threshold is $\gamma \approx 0.9744$, meaning that if two pairs differ by no more than $\gamma/2 \approx 0.4872$ in skill, the model deems them likely to tie.

```
Converged: True
Final log-likelihood: -696.5472856221662
Estimated gamma = 0.9744
Theta_1 = 0.0622
Theta_2 = -0.1055
Theta_3 = -0.0007
Theta_4 = 0.2977
Theta_5 = 0.2372
Theta_6 = -0.2263
Theta_7 = -0.1577
Theta_8 = -0.0784
Theta_9 = -0.2962
Theta_10 = 0.1750
Theta_11 = 0.0109
Theta_12 = 0.0000

=== Ranking by skill, descending order ===
  Rank 1: Pair 4,  θ = 0.2977
  Rank 2: Pair 5,  θ = 0.2372
  Rank 3: Pair 10,  θ = 0.1750
  Rank 4: Pair 1,  θ = 0.0622
  Rank 5: Pair 11,  θ = 0.0109
  Rank 6: Pair 12,  θ = 0.0000
  Rank 7: Pair 3,  θ = -0.0007
  Rank 8: Pair 8,  θ = -0.0784
  Rank 9: Pair 2,  θ = -0.1055
  Rank 10: Pair 7,  θ = -0.1577
  Rank 11: Pair 6,  θ = -0.2263
  Rank 12: Pair 9,  θ = -0.2962
```

Figure 17: Ranking under the threshold model with 12 pairs

Examining the final θ values reveals that pair 4 has the highest skill at $\theta_4 = 0.2977$, while pair 9 is the lowest at $\theta_9 = -0.2962$. Between these extremes, the rank order (from top to bottom) is:

$$\text{pair}\,4 > \text{pair}\,5 > \text{pair}\,10 > \text{pair}\,1 > \text{pair}\,11 > \text{pair}\,12 > \text{pair}\,3 > \text{pair}\,8 > \text{pair}\,2 > \text{pair}\,7 > \text{pair}\,6 > \text{pair}\,9.$$

Comparing this with the real ranking results 5.1.5, we find that the rankings are exactly the same, except for 5th and 6th place. But as we mentioned before, the scores of the fifth and sixth places are exactly the same, so their rankings can be switched. So it can be concluded that the Threshold model performs well in ranking and achieves the same results as the real results. The negative log-likelihood of -696.5473 further indicates a reasonable fit to observed data, and the convergence of BFGS ensures stable parameter estimates for $\{\theta_i\}$ and γ.

6.3.2 Parametric Bootstrap analysis under the threshold model

Figure 18 lists the estimates of θ and γ obtained by fitting a threshold model (allowing for ties) to our Bridge data, along with the 95% confidence intervals obtained by the parameter Bootstrap ($B = 1000$ replicates). For identification purposes, the baseline parameter θ_{12} is set to zero so that its reported point estimate is always 0.0000.

```
95% CI for each θ_i:
  θ_1 = 0.0622  CI= [-0.2879, 0.4197]
  θ_2 = -0.1055  CI= [-0.4418, 0.2451]
  θ_3 = -0.0007  CI= [-0.3373, 0.3308]
  θ_4 = 0.2977  CI= [-0.0444, 0.6321]
  θ_5 = 0.2372  CI= [-0.0836, 0.5683]
  θ_6 = -0.2263  CI= [-0.5564, 0.1425]
  θ_7 = -0.1577  CI= [-0.4843, 0.1717]
  θ_8 = -0.0784  CI= [-0.3909, 0.2406]
  θ_9 = -0.2962  CI= [-0.6331, 0.0426]
  θ_10 = 0.1750  CI= [-0.1527, 0.5135]
  θ_11 = 0.0109  CI= [-0.3528, 0.3576]
  θ_12 = 0.0000  CI= [0.0000, 0.0000]

95% CI for gamma: [0.8458, 1.1415]
```

Figure 18: Parametric Bootstrap estimates for the threshold mode with 8 pairs

From 18, most parameters θ_i have confidence intervals that include zero. Statistically, we cannot identify who is significantly stronger or weaker than the baseline. Possible reasons include:

- Limited sample size or insufficient board data leads to poor discrimination between different players;
- The true skill levels may be close, producing statistically indistinguishable results;
- A high tie rate or scoring mechanism that reduces observed win/loss differences.

The tie threshold parameter γ is estimated to be in the range $[0.85, 1.14]$. An approximate threshold of 0.5 for the skill difference is thus consistent with the large proportion of ties observed.

6.3.3 Analysis of tie frequencies in the threshold model

Figure 19 compares the observed number of tied pairs k per board to the expected frequency predicted by a single-threshold γ model under a pairwise-independence assumption with $T = 6$ pairs.

```
=== Distribution of #ties (Observed vs. Expected) ===
 k   Observed   Expected
 0        1      0.842
 1       12      3.811
 2        6      8.050
 3        7     10.527
 4        8      9.533
 5        0      6.332
 6        5      3.187
 7        3      1.237
 8        0      0.374
 9        0      0.088
10        1      0.016
11        0      0.002
12        0      0.000
13        0      0.000
14        0      0.000
15        1      0.000
```

Figure 19: Comparison of observed vs. expected tie distributions under the threshold model.

As shown in Figure 19, the model generally aligns well with the observed data for low tie counts ($k = 0$ or $k = 2$ to 4). However, the model significantly underestimates the occurrence of single ties ($k = 1$),

predicting only 3.811 occurrences compared to the observed 12. Furthermore, it struggles to capture extreme outcomes: while the model predicts no chance of $k = 15$, one such instance was observed in reality.
This discrepancy suggests that while effective in capturing moderate tie frequencies, the single-threshold model may be too simplistic to fully describe the distribution of pairwise ties, especially in cases where score clustering is more pronounced.

6.3.4 Comparison with the permutation model method

k	Observed	Expected (original)	Expected (New method)
0	1	13.2714	0.842
1	12	20.1172	3.811
2	6	7.3239	8.05
3	7	2.2991	10.527
4	0	0.8789	9.533
5	0	0	6.332
6	5	0.093	3.187
7	3	0.0153	1.237
8	0	0	0.374
9	0	0	0.088
10	1	0.0012	0.016
11	0	0	0.002
12	0	0	0
13	0	0	0
14	0	0	0
15	1	0	0

Table 16: Comparison of Observed and Expected Values

Table 16 contrasts the tie-count distributions predicted by our threshold-based approach (right column) with those from the original method (left column). We note the following:

- **Our method** yields closer alignment with the observed counts for most values of k, especially $k = 0$ and $k = 1$, which dominate the distribution.
- **The original method** exhibits larger discrepancies for moderate or higher k (e.g., $k = 2, 3, 5$), indicating difficulty in capturing multiple-pair ties or rare tie patterns.

By examining the log-likelihood and distributional fit, our threshold model more accurately represents how often $k = 0, 1, 2$ ties occur per board. Its improvement is evident in smaller numerical errors at crucial tie counts and a better match to real data. Therefore, our approach offers superior predictive accuracy for tie frequencies under this dataset.

6.3.5 Limitations of the threshold model assumptions

The threshold model introduced in this thesis assumes that all pairwise tie outcomes on a given board are mutually independent. Specifically, for a board involving $T = 6$ players, the $\binom{6}{2} = 15$ pairwise tie indicators are modeled as independent Bernoulli(p_{ij}) random variables. Although this assumption facilitates computation and enables the analytical derivation of tie count distributions, it fails to capture structural dependencies frequently arising in practical bridge settings.

The primary issue with this independence assumption is that it neglects board-specific correlations between tie outcomes. In practice, all players on a given board share the same deal and often reach similar contracts due to common hand characteristics. If several players adopt analogous bidding and play strategies, their resulting scores tend to cluster, increasing the likelihood of correlated ties. Consequently, treating these ties as independent underestimates the probability of multiple simultaneous ties and may correspondingly overestimate isolated tie scenarios (e.g., $k = 0$ or $k = 1$).

A Concrete Example

Consider a board where players S_1, S_2, S_3 adopt the same bidding and play strategies, leading them to reach the same contract and achieve nearly identical scores. Meanwhile, other players exhibit greater variability in their strategies, resulting in a more dispersed score distribution. In this scenario, S_1, S_2, S_3 form multiple ties.

Under the threshold model's independence assumption, the probability of ties is computed separately for S_1 vs. S_2, S_2 vs. S_3, and S_1 vs. S_3, assuming these events occur independently. This implies that to obtain $k = 3$ or more ties, all these independent events must simultaneously occur. However, in practice, S_1, S_2, S_3 are highly correlated due to their shared strategies, making these ties a systemic outcome rather than independent occurrences.

More specifically, bridge scores depend on individual choices, opponents' actions, and the overall deal context. If S_1, S_2, S_3 all decide to play 3NT (No Trump contract), they are also likely to make similar decisions during play, such as choosing identical opening leads or managing suits similarly. As a result, their final scores become nearly identical—not by chance, but because of strategic alignment. These interdependent decisions imply that the outcome of one pair inherently influences the others, resulting in correlated tie outcomes rather than independently generated events.

In contrast, the threshold model treats each pairwise tie event as an independent Bernoulli variable, leading to an inherent underestimation of the likelihood of multiple simultaneous ties, coupled with a potential overestimation of isolated tie probabilities.

The second limitation arises from the implicit assumption of opponent homogeneity. In the threshold model, multiple North–South (NS) pairs on the same board are compared based on their scores to infer relative skill levels. This process implicitly assumes that all NS pairs face East–West (EW) opponents of similar skill levels, allowing observed differences in NS scores to be attributed solely to differences among the NS pairs themselves. However, this assumption may not always hold in practice.

Although the standard Howell movement ensures each pair eventually plays against all others, thereby balancing opponent distribution across the tournament, the model conducts pairwise comparisons on a per-board basis. In doing so, it does not explicitly model individual opponent effects, instead treating all opponents as averaged. For example, when comparing the scores of two NS pairs on a particular board, the model ignores potential differences in the strength of the EW pairs they faced, implicitly assuming such variations average out across the tournament. However, substantial opponent variability on specific boards, especially involving EW pairs with distinct strategies or skill levels, may meaningfully influence observed outcomes.

As a result, even though all pairs encounter comparable opponents across the tournament, the model's local treatment of individual boards may overlook significant variation in opponent structure. This oversight can introduce bias in assessing skill differences among NS pairs and potentially reduce the explanatory power and fairness of rankings derived from the model, particularly in cases

demanding high precision.

In conclusion, the model faces two critical limitations:

- The independence assumption neglects strategy-induced correlations among players sharing the same deal;
- The equal-opponent assumption ignores per-board opponent strength variations influencing observed scores.

While the threshold model provides a coherent and computationally tractable framework for analyzing ties and ranking players, its simplifying assumptions restrict alignment with real-world observations. Potential extensions include incorporating board-level random effects to capture deal-specific clustering or employing hierarchical models to explicitly account for opponent strength. Additionally, methods that jointly model all scores on a given board, rather than decomposing them into pairwise outcomes, may yield more realistic and robust insights into tournament dynamics. These refinements and their implications are discussed further in the concluding section.

6.4 Conclusion

In section 6.1, we introduced the threshold-based approach to modeling ties. We defined a half-width parameter γ that creates a "tie interval" whenever two competitors' skill parameters lie within $\pm(\gamma/2)$. Section 6.1.1 outlined the model setup, while section 6.1.2 provided explicit probability formulas using the logistic distribution. Section 6.1.3 established a theoretical foundation by characterizing sufficient statistics for $\{\theta_i\}$ and γ, and section A.3.3 proposed a weighted-correction selection method to improve confidence-set construction for identifying the best player.
In section 6.2, we tested this threshold model on the smaller dataset from section 4.1. BFGS optimization produced stable estimates of $\{\theta_i\}$ and γ, while a parametric Bootstrap procedure underscored the considerable uncertainty inherent in limited board samples. Tie-frequency comparisons showed that a single-threshold model largely aligns with observed data for small to moderate tie counts, although multi-tie scenarios sometimes exceed the model's predictions.
Next, section 6.3 extended the threshold model to the larger 2024 China National Bridge Championships dataset from section 5.1.4. Here, the final skill rankings nearly matched the official results, demonstrating that the threshold model can handle real-world tournaments effectively. We again noted, however, that a single γ may not fully capture instances where multiple tables simultaneously tie. In particular, subsection stressed how the pairwise-independence assumption can overlook board-level correlations—if players adopt similar bidding and play strategies on a particular deal, ties can cluster far more than an independent Bernoulli framework suggests.
Overall, this chapter shows that threshold-based tie modeling offers both interpretability and competitive performance relative to earlier Davidson- or permutation-based models. By letting γ define a straightforward tie interval, we capture the notion that small skill differences often lead to drawn outcomes. Having explored the threshold model in detail, the upcoming section will consolidate all key findings. We will summarize each method's performance, discuss broader implications for Bridge tournament analytics, and highlight avenues for future research and model enhancements.

7 Discussions and additional results

This section presents a multifaceted examination of Bridge tournament evaluations, encompassing a fuzzy logic performance assessment method, an analysis of final rankings in the 2024 National Bridge Championships Open Pairs event, and a critique of the World Bridge Federation's Master Points (MP) system. By contrasting the fuzzy logic framework with threshold-based modeling and discussing how randomness and multi-stage structures affect rankings, we aim to highlight the strengths and limitations of different approaches while also proposing viable alternatives to traditional MP accumulation.

7.1 A fuzzy logic approach to Bridge performance

This section introduces and applies a fuzzy logic method [40] for evaluating Bridge pairs' performance. Unlike the permutation or threshold-based approaches, fuzzy logic allows us to classify and combine performance categories into a single "defuzzified" score, providing an alternative view of each pair's relative strength. We compare our fuzzy logic results with the threshold model results presented earlier (Figure 17) to see how different methods can yield distinct rankings and interpretations. This section compares these two methods regarding their principles, calculation processes, strengths, and weaknesses.

7.1.1 Introduction to the fuzzy logic approach

Fuzzy logic provides a framework to handle uncertainty and partial truth by defining "membership" in linguistic categories. Here, we assign each pair's score percentage (or average Match Point, MP) to a performance category, such as *Excellent*, *Very Good*, *Good*, *Mediocre*, and *Unsatisfactory*, denoted by $\{A, B, C, D, F\}$. Unlike crisp classification—where a pair's score is either in or out of a category—fuzzy membership can capture gradations in performance. We then aggregate membership degrees across all pairs to understand how the field is distributed among these categories.

Definition 8: Membership Function

Suppose there are n pairs, and let n_X be the number of pairs whose performance (in percentage) falls into category $X \in \{A, B, C, D, F\}$. We define the membership function as:

$$m(X) = \frac{n_X}{n}, \qquad \sum_{X \in \{A,B,C,D,F\}} m(X) = 1.$$

This function represents the relative frequency of each performance category in the total population. It allows us to evaluate how the entire group of players is distributed among categories and serves as the foundation for computing a single "defuzzified" score.

The centroid method is then used to convert these fuzzy categories into a single numerical evaluation so that more substantial categories have a more significant influence on the final score.
Instead of providing a detailed proof, we note that the resulting centroid coordinates follow directly from [40], where each category is mapped to a specific numeric center. We assign $F \mapsto 1,\ D \mapsto 3,\ C \mapsto 5,\ B \mapsto 7,\ A \mapsto 9.$, and we place each membership rectangle over width 2 with height $\frac{m(X_i)}{2}$, ensuring the total area is 1.

Corollary 1: Centroid Coordinates

Hence, from [40], the final (defuzzified) score is given by:

$$x_c = \frac{1}{2}\Big[\, m(F) + 3\,m(D) + 5\,m(C) + 7\,m(B) + 9\,m(A)\Big],$$

$$y_c = \frac{1}{2}\Big[\, m(F)^2 + m(D)^2 + m(C)^2 + m(B)^2 + m(A)^2\Big].$$

Remark:

- **Consistency of area:** With each category assigned width 2 and height $\frac{m(X_i)}{2}$, the total area $\iint_F dx\,dy$ becomes $\sum_{i=1}^{5} 2\times\frac{m(X_i)}{2} = \sum_{i=1}^{5} m(X_i) = 1$. Hence, dividing the first moment integrals by 1 yields the stated centroid.
- **Interpretation of x_c:** A higher x_c value means the membership mass is shifted toward stronger categories (A or B). Thus, x_c is a weighted overall performance indicator.
- **Interpretation of y_c:** The coordinate y_c gauges how spread out or concentrated the memberships are. If one category dominates, $\sum m(X_i)^2$ is larger, which can raise or lower y_c depending on sign conventions. In effect, a higher y_c can indicate more disparity (if many players are in the same category or extremes), whereas a lower y_c may indicate a smoother distribution across categories.

7.1.2 Applying the fuzzy logic approach to 2024 China Championship data

In Bridge tournaments, a simple overall average of Match-Point (MP) scores can hide the fact that a pair may perform exceptionally well on specific boards but poorly on others. By dividing the percentage scores for each board into five categories (A/B/C/D/F), we preserved the distribution of high and low results. The final fuzzy membership of each pair in these categories then reflects the frequency with which they achieved the highest and lowest results.

Let $\mathrm{MP}_{i,j}$ be the MP score of pair j on board i, with maximum 10. We define the percentage:

$$p_{i,j} = \frac{\mathrm{MP}_{i,j}}{10}\times 100\%.$$

From [40], we classify $p_{i,j}$ as

$$\text{A if } p > 65\%,\quad \text{B if } 55\% < p \le 65\%,\quad \text{C if } 48\% < p \le 55\%,\quad \text{D if } 40\% \le p \le 48\%,\quad \text{F if } p < 40\%.$$

We grouped the percentage scores of the 12 pairs of players on each board into categories (A/B/C/D/F) and calculated the fuzzy centroid coordinates (x_c, y_c) by calculating the frequency of each pair's entry into these categories on the 44 boards. For pair j, let $\mathrm{countA}_j, \ldots, \mathrm{countF}_j$ be the number of boards on which pair j is assigned to A/B/C/D/F, respectively. Dividing by the total number of boards (44) yields

$$m(A)_j = \frac{\mathrm{countA}_j}{44},\quad \ldots,\quad m(F)_j = \frac{\mathrm{countF}_j}{44},$$

so that $m(A) + m(B) + m(C) + m(D) + m(F) = 1$.

To find Fuzzy centroid (x_c, y_c), we assign numeric centers

$$A \mapsto 9,\quad B \mapsto 7,\quad C \mapsto 5,\quad D \mapsto 3,\quad F \mapsto 1,$$

and define

$$x_c = \tfrac{1}{2}\big[1\,m(F) + 3\,m(D) + 5\,m(C) + 7\,m(B) + 9\,m(A)\big],$$

$$y_c = \tfrac{1}{2}\left[m(F)^2 + m(D)^2 + m(C)^2 + m(B)^2 + m(A)^2\right].$$

A larger x_c indicates more frequent higher-category (A/B) boards, while y_c shows how concentrated or spread out these categories are.

7.1.3 Detailed analysis of fuzzy results

We extend the fuzzy classification analysis by examining each pair's fuzzy centroid (x_c, y_c), ranking based on x_c, and also interpreting $\mathbf{y_c}$ as a measure of performance stability. Figure 20 displays each pair's membership distribution $m(A)$, $m(B)$, $m(C)$, $m(D)$, $m(F)$ and the resulting centroid.

```
=== Fuzzy Classification Results Per Board ===
Pair  1: m(A)=0.455, m(B)=0.045, m(C)=0.045, m(D)=0.091, m(F)=0.364   => x_c=2.636, y_c=0.176
Pair  2: m(A)=0.364, m(B)=0.045, m(C)=0.091, m(D)=0.136, m(F)=0.364   => x_c=2.409, y_c=0.147
Pair  3: m(A)=0.432, m(B)=0.045, m(C)=0.068, m(D)=0.068, m(F)=0.386   => x_c=2.568, y_c=0.174
Pair  4: m(A)=0.432, m(B)=0.159, m(C)=0.023, m(D)=0.114, m(F)=0.273   => x_c=2.864, y_c=0.150
Pair  5: m(A)=0.455, m(B)=0.114, m(C)=0.045, m(D)=0.091, m(F)=0.295   => x_c=2.841, y_c=0.159
Pair  6: m(A)=0.273, m(B)=0.114, m(C)=0.045, m(D)=0.136, m(F)=0.432   => x_c=2.159, y_c=0.147
Pair  7: m(A)=0.341, m(B)=0.091, m(C)=0.045, m(D)=0.068, m(F)=0.455   => x_c=2.295, y_c=0.169
Pair  8: m(A)=0.318, m(B)=0.091, m(C)=0.068, m(D)=0.091, m(F)=0.432   => x_c=2.273, y_c=0.154
Pair  9: m(A)=0.295, m(B)=0.068, m(C)=0.068, m(D)=0.114, m(F)=0.455   => x_c=2.136, y_c=0.158
Pair 10: m(A)=0.432, m(B)=0.068, m(C)=0.045, m(D)=0.159, m(F)=0.295   => x_c=2.682, y_c=0.153
Pair 11: m(A)=0.318, m(B)=0.182, m(C)=0.045, m(D)=0.091, m(F)=0.364   => x_c=2.500, y_c=0.138
Pair 12: m(A)=0.341, m(B)=0.205, m(C)=0.045, m(D)=0.068, m(F)=0.341   => x_c=2.636, y_c=0.140

=== Ranking Based on x_c (Descending Order) ===
Rank 1: Pair 4, x_c=2.864, y_c=0.150
Rank 2: Pair 5, x_c=2.841, y_c=0.159
Rank 3: Pair 10, x_c=2.682, y_c=0.153
Rank 4: Pair 1, x_c=2.636, y_c=0.176
Rank 5: Pair 12, x_c=2.636, y_c=0.140
Rank 6: Pair 3, x_c=2.568, y_c=0.174
Rank 7: Pair 11, x_c=2.500, y_c=0.138
Rank 8: Pair 2, x_c=2.409, y_c=0.147
Rank 9: Pair 7, x_c=2.295, y_c=0.169
Rank 10: Pair 8, x_c=2.273, y_c=0.154
Rank 11: Pair 6, x_c=2.159, y_c=0.147
Rank 12: Pair 9, x_c=2.136, y_c=0.158
```

Figure 20: Fuzzy classification outcomes and x_c ranking for each pair.

We further examine x_c to highlight the overall ranking. From Figure 20, Pair 4 and Pair 5 have $x_c \approx 2.864$ and 2.841, placing them in the top two spots. This indicates they have significantly more boards in A or B, whereas pairs like Pair 9 or Pair 6 end up around $x_c < 2.2$, reflecting more F or D boards. For instance, Pair 4 has $(m(A), m(B)) = (0.432, 0.159)$, while Pair 9 has $m(F) = 0.455$, lowering its x_c.

Meanwhile, we interpret y_c as an indicator of how concentrated or spread out each pair's distribution is. If a pair is heavily concentrated in a small subset of categories, one or two membership values will be large, raising $\sum m(X)^2$ and hence y_c. If the pair is more evenly distributed across multiple categories, y_c remains moderate or smaller. For instance, Pair 1 has $(m(A), m(F)) = (0.455, 0.364)$, showing a high proportion of extremes (A and F), thus $y_c \approx 0.176$, suggesting notable variability.

Notably, Pair 12 and Pair 1 share the same $x_c = 2.636$, but Pair 12's $y_c = 0.140$ whereas Pair 1's $y_c = 0.176$. This implies Pair 1 experiences more drastic swings between top and bottom boards, while Pair 12 avoids such extremes. The difference in y_c reflects their stability discrepancies and can also serve as an additional reference for coaches or competitors when evaluating overall match performance.

Moreover, with 44 boards in this event, luck and variability still have some impact. If the tournament expanded to, say, 60 or 80 boards, we would expect slight fluctuations in x_c and y_c to average out more strongly, reducing random effects and making the final rankings more robust.

7.1.4 A consolidated analysis: threshold vs. fuzzy logic Methods

In this section, we compare the findings from both the Threshold Model (Section 6.3) and the fuzzy logic Model (Section 7.1) on the 2024 Bridge tournament data. The Threshold Model emphasizes

pairwise skill gaps via parameters θ_i and a tie threshold γ. It sorts pairs based on their estimated latent skill level, focusing on how likely one pair is to outscore another in head-to-head scenarios.
Fuzzy Logic Model classifies each board outcome into categories (A, B, C, D, F) depending on the MP percentage, then computes a "defuzzified" centroid (x_c, y_c) per pair. Here, x_c is larger for pairs that accumulate more high-category boards (A or B), whereas y_c provides a sense of how concentrated or volatile those board-level results are.

(a) Threshold Model Ranking:

Pair4 > Pair5 > Pair10 > Pair1 > Pair11 > Pair12 > Pair3 > Pair8 > Pair2 > Pair7 > Pair6 > Pair9.

(b) Fuzzy Logic Ranking:

Pair4 > Pair5 > Pair10 > Pair1 > Pair12 > Pair3 > Pair11 > Pair2 > Pair7 > Pair8 > Pair6 > Pair9.

Both models strongly agree that Top pairs (e.g., Pairs 4, 5, 10) dominate in winning or high-percentage boards, leading to consistently higher θ estimates or more frequent A/B categories. Bottom pairs (e.g., Pairs 9, 6) struggle both in direct comparisons and in generating top-board outcomes, thus staying near the lower end. This alignment suggests that either method is robust enough to identify strong enough or weak enough performances.

The middle rankings, such as Pairs 1, 11, 12, 3, 2, and 8, exhibit slight variations across different evaluation models. In the threshold model, the ranking follows a structured order: Pair1 > Pair11 > Pair12 > Pair3 > $\cdots$. However, the fuzzy logic approach results in a slight reordering:Pair1 > Pair12 > Pair3 > Pair11 > $\cdots$. These discrepancies stem from fundamental differences in how each method processes board-level data. The threshold model aggregates incremental gains and losses into a single skill metric, θ_i, giving more weight to consistently medium-to-high finishes. Conversely, the fuzzy logic model categorizes board scores into discrete performance levels. For example, a board result in the 55–65% range is classified as category B, meaning that repeated moderate results contribute less to x_c than a few exceptionally high (A) performances.
Each ranking model provides distinct insights into performance evaluation. The threshold model is particularly useful for pairwise comparisons, answering questions like "If Pair i meets Pair j, who is favored?" It infers a stable underlying skill difference and incorporates a tie threshold γ to refine comparisons. On the other hand, fuzzy logic highlights performance variability, emphasizing the frequency of extreme outcomes. The secondary metric y_c allows analysts to determine whether a pair exhibits steady (mostly B/C) or volatile (frequent A/F) performance patterns, offering an alternative perspective for coaching decisions.

7.1.5 Refinements to the ranking system

To enhance the robustness and fairness of the ranking system, two key refinements can be considered: adjusting category boundaries and refining the weighting system.

Adjusting Category Boundaries The current classification model assigns performance into fixed categories based on predefined thresholds (e.g., A: 65%+, B: 55–65%). However, these thresholds may introduce ranking biases, as small fluctuations in scores near a boundary can shift a board's classification.
A simple illustration: if we raise the A-boundary from 65% to 70%, a pair that barely crossed the old threshold on 4 boards may now only count 2 of those as A. Then $m(A)$ drops from $4/44 \approx 0.091$ to $2/44 \approx 0.045$, reducing the pair's x_c and potentially altering its rank.
A sensitivity analysis can be conducted by slightly adjusting these thresholds (e.g., raising A from 65% to 70%, or lowering B from 55% to 50%) to observe how rankings respond to such changes. If rankings remain largely stable, the classification system is robust. However, if minor adjustments lead to significant ranking shifts, recalibrating the boundaries could improve fairness and ensure consistent rankings.

Refining the Weighting System The current weighting approach assigns fixed values (1, 3, 5, 7, 9) to performance categories, assuming equal spacing between them. However, in competitive settings, the difference between a top-tier and mid-tier performance is often greater than the difference between two adjacent mid-tier performances.
A nonlinear weighting scheme can be introduced to amplify the influence of top-tier results, ensuring that exceptional performances contribute more significantly to rankings. For instance, assigning weights like {1, 3, 6, 10, 15} would increase the gap between higher tiers and mid-tiers, so a few A-level boards have more pronounced impact on x_c.Additionally, not all boards in a tournament are of equal importance—some may be more difficult or strategically decisive. Board-specific weighting can be applied to give higher importance to such boards, ensuring that rankings reflect not just overall consistency but also performance in critical situations.

7.1.6 Effects on fairness, variability, and ranking consistency

Many people believe that contract Bridge is a fair game, but there is still an element of luck because the hands players receive and how the boards play out can vary. Two ranking models, one based on performance thresholds and another using fuzzy logic help analyze player performance, but they treat mid-level players differently, affecting fairness and ranking stability.
One key finding is that both models consistently agree on the strongest and weakest players, confirming that skill is the biggest factor in large tournaments. No matter which model is used, the best and worst pairs tend to stay in similar positions. This suggests that in long tournaments, the impact of luck decreases, and true skill becomes more evident in the rankings.
However, the differences become more noticeable for mid-tier players. Small variations in board results or how the models classify performance can lead to significant ranking shifts. The threshold model smooths out minor wins and losses, focusing on overall consistency, while the fuzzy logic model groups results in a way that may overlook steady, small improvements. This means that factors like lucky card distributions or slight differences in how the models work can influence mid-range rankings more than those at the top or bottom.
To make rankings as fair as possible, a combination of both models could be useful. The threshold model is great for identifying consistent players and making fair head-to-head comparisons, while the fuzzy logic model helps detect variability and highlight players who perform inconsistently. By using both approaches together, tournament organizers can create a more balanced ranking system that accounts for both stability and performance fluctuations.

7.2 Analysis of final rankings in the Open Pairs event

This section provides a comprehensive examination of the final standings in the 2024 National Bridge Championships Open Pairs event. We begin with an overview of the final ranking table and notable observations regarding each pair's progression from the preliminary and semifinal stages. We also highlight how these results align with statistical findings from other parts of our study, including the parametric bootstrap approach, thus reinforcing the robustness of the final standings reported.

7.2.1 Observations and trends in final rankings

The final rankings for the 2024 National Bridge Championships Open Pairs show the dynamic nature of competitive performance at different stages of the tournament. Some pairs maintained a stable ranking throughout the tournament, while others showed significant fluctuations between the preliminary round [15], semifinal [20], and final [17].

Final Stages		Names Pinyin	Previous Stages	
Final Rank	Pair Number		Semifinal Rank	Preliminary Rank
1	4#	Wang Guoqiang - Zhang Lixiong	10	24
2	5#	Jia Xinchun - Ning Jun	9	29
3	10#	Sun Minghao - Liang Yuhe	3	17
4	1#	Liu Shuping - Shen Qingfeng	7	14
5	12#	Zhang Kuo - Wu Xuenan	1	20
6	11#	Cao Danan - Pan Liping	2	9
7	3#	Zou Chongsong - Ma Guowei	11	6
8	8#	He Yong - Yang Tiankui	5	10
9	2#	Liu Jun - Shi Feng	12	5
10	7#	Zhang Chongbin - Zhou Qinghua	6	19
11	6#	Zheng Wei - Li Aimin	8	30
12	9#	Zeng Xiang - Yao Tao	4	36

Table 17: Final Rank Comparison with Pinyin and Preliminary Rank

Notably, the championship-winning pair, Wang Guoqiang - Zhang Lixiong (#4), ranked 24th in the preliminaries stage but steadily rose to 10th place in the semi-finals before eventually taking first place. A similar upward trajectory can be observed for the second-place pair, Jia Xinchun - Ning Jun (#5), who advanced from 29th in the preliminary to 9th in the semifinals, highlighting the importance of performance adaptability in the later stages.
Conversely, some pairs that had performed well in the previous rounds slipped in the final stage. Zhang Kuo - Wu Xuenan (#12), who led the semifinals, ended up in 5th place, illustrating the increased intensity of the competition and potential changes in results when direct pairwise comparisons are made. Similarly, Zeng Xiang - Yao Tao (#9) achieved an impressive comeback from 36th in the preliminary round to 4th in the semifinals but ended up 12th in the final, suggesting that peaks in performance may not always be sustained at all stages of the competition.

7.2.2 Statistical uncertainty in final rankings

While the final rankings provide a clear order, statistical analysis reveals considerable uncertainty, particularly for mid-tier players whose placements are susceptible to short-term fluctuations in luck and performance.
Parametric bootstrap analysis confirms this variability. From figure 18, many estimated skill parameters have 95% confidence intervals overlapping zero, meaning there is no statistically significant distinction between these players and the baseline level. Additionally, the estimated tie threshold (γ) falls within $[0.85, 1.14]$, indicating that minor skill differences often result in near-equal outcomes, making it difficult to distinguish actual performance gaps from random variation.
Mid-tier rankings are especially volatile. The wide confidence intervals suggest that these players' rankings are unstable, likely influenced by short-term factors such as favorable card distributions or temporary shifts in form rather than consistent skill superiority.
Moreover, the three-stage competition structure further amplifies this effect. Since rankings are determined across preliminary, semifinal, and final rounds, even brief streaks of good luck (e.g., several advantageous hands) or momentary surges in performance can lead to substantial shifts in placement.
Ultimately, while the final standings decide the winners, they may not fully reflect actual skill levels. The statistical uncertainty, particularly for mid-tier players, suggests that some rankings are shaped more by short-term variability than by long-term ability. The multi-stage format intensifies these fluctuations, emphasizing the role of randomness in tournament outcomes. Therefore, while the final results determine the champions, they may not precisely capture each player's true strength. This leads us to further examine the sources of randomness in the competition structure and the strategic considerations behind it, as discussed in the following subsections.

7.2.3 Scoring method and randomness in Bridge competitions

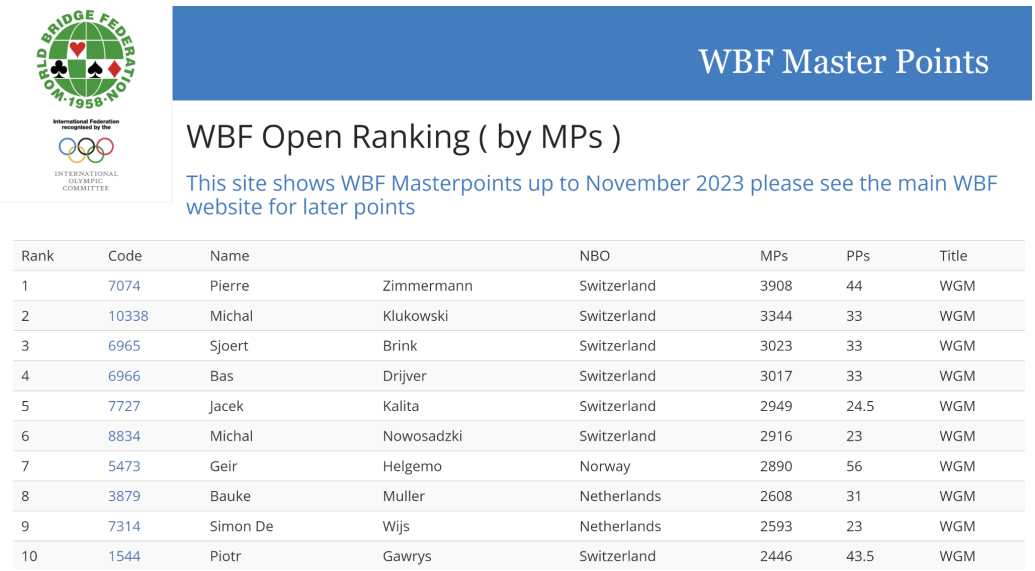
From the results, it is evident that the ranking of the same team can fluctuate significantly across the three stages, indicating a strong element of randomness in Bridge competitions. This randomness mainly arises from the following factors.
The scoring method in Bridge introduces an element of luck. Although the competition rules strive to be fair—such as having the same board played by different NS/EW pairs and comparing scores under the same conditions—the varying opponents each pair faces at different stages still introduce considerable variability in results.
Bridge employs a relative scoring system rather than an absolute one. For example, in the IMP (International Match Point) scoring system, a pair's score depends on their performance relative to other pairs rather than their individual score. This means that even if a pair performs well in a given match, if other pairs perform even better, their score might still be low. This relative scoring system amplifies the randomness of the competition, leading to greater ranking fluctuations.

7.2.4 Comparison of Bridge with other competitive games

From a game theory perspective, contract Bridge is an incomplete information game because players do not have full knowledge of their opponents' hands and must make decisions under uncertainty. This is in contrast to complete information games such as chess or Go. In the bidding phase, each pair of players must not only convey information to their peers, but also control the risk of revealing too much information to their opponents. Defensive strategy is affected by the opponent's bidding; deceptive or aggressive preemptive bidding can force the opponent into a sub-optimal position. Contract Bridge also exemplifies the dynamics of Bayesian games, in which pairs update and refine their strategies based on bidding and partial information gained during the game. In addition, Bridge is a dynamic game: as more is known about the opponent's style (aggressive or conservative), the strategy changes as the board changes.
Bridge is very different from complete information games such as Chess or Go. In chess and Go, all players know the state of the game perfectly well, and strategy is based primarily on calculating optimal paths. In contrast, Bridge involves hidden information and is a non-zero-sum incomplete information game, which means that in addition to calculating the optimal moves, both players must also evaluate risks, predict their opponents' actions, and dynamically adjust their strategies.
There are also similarities between Bridge and card games such as poker. For example, Texas Hold'em also involves hidden messages and signalling. In poker, players send mixed signals through bets, folds, and other actions in an attempt to influence their opponents' decisions. Similarly, the bidding and defence strategies of Bridge involve the strategic handling of incomplete information in order to outwit opponents.
In summary, while the final rankings reflect the ultimate outcomes of the 2024 Open Pairs event, the relatively large bootstrap confidence intervals demonstrate that these results can hinge heavily on stochastic swings and short-term performance factors. The pronounced shifts in pair rankings between stages further underscore how skill, randomness, and adaptive strategy collectively shape Bridge competitions. Consequently, Bridge stands apart from perfect information games like chess and Go, presenting a complex environment where partial information, psychological insight, and probability all intersect in determining winners and losers.

7.3 Limitations of the MP system and an alternative approach

The World Bridge Federation (WBF) ranks players using the Master Points (MP) system, which accumulates points based on tournament performance, with higher values assigned to prestigious events. While straightforward, this approach inherently favors experienced players with extensive participation, often failing to reflect true skill.



WBF Master Points

WBF Open Ranking (by MPs)

This site shows WBF Masterpoints up to November 2023 please see the main WBF website for later points

Rank	Code	Name		NBO	MPs	PPs	Title
1	7074	Pierre	Zimmermann	Switzerland	3908	44	WGM
2	10338	Michal	Klukowski	Switzerland	3344	33	WGM
3	6965	Sjoert	Brink	Switzerland	3023	33	WGM
4	6966	Bas	Drijver	Switzerland	3017	33	WGM
5	7727	Jacek	Kalita	Switzerland	2949	24.5	WGM
6	8834	Michal	Nowosadzki	Switzerland	2916	23	WGM
7	5473	Geir	Helgemo	Norway	2890	56	WGM
8	3879	Bauke	Muller	Netherlands	2608	31	WGM
9	7314	Simon De	Wijs	Netherlands	2593	23	WGM
10	1544	Piotr	Gawrys	Switzerland	2446	43.5	WGM

Figure 21: WBF Open Ranking by MPs, as of November 2023 [46].

For example, Pierre Zimmermann, the current top-ranked WBF Open player with 3,908 MPs, has a long history of high-level finishes in global championships. Since 2021, he has also captained the Swiss national team and organized major tournaments, further cementing his influence in the Bridge community [44]. Similarly, Bauke Muller, the Netherlands' leading player since 1962, has competed internationally for over four decades and played a key role in Dutch victories in the Bermuda Bowl and European Open Teams Championship. Known as "the professor" for his analytical skills, Muller exemplifies how longevity and frequent participation shape rankings under the MP system [42].
However, this method of accumulation puts novice players at a distinct disadvantage, especially those who entered the competitive scene late or have limited opportunities to participate in tournaments. Even if they show exceptional skill and consistently score well, their low MP totals make it difficult for them to compete with veteran players who have accumulated points over the years. As a result, these talented newcomers face an uphill battle as they climb the rankings, which can lead to frustration and a loss of motivation. Not only does this system limit their recognition, it also prevents newcomers from fully participating in high-level competition, ultimately slowing the infusion of new competitive energy into the Bridge community.
In practice, high-prestige events such as the Bermuda Bowl or European Open Teams often award large MP bonuses, so those with frequent appearances and long careers in such events tend to hold top positions for extended periods. This skew can mask recent outstanding performances by new or less frequent participants.
To address these limitations, rankings should emphasize relative skill and recent performance rather than total points. Our threshold model refines this approach by focusing on board-level match results rather than long-term point accumulation. Unlike the MP system, which may mask rapid progress or undervalue new competitors, our approach takes into account both the total match point results and the probability of a draw estimated through the maximum likelihood method.
There are two major benefits to this model. One is that it reduces the bias against newer or lesser-played players, as head-to-head performances are assessed equally on each board. The other benefit is that it captures consistency more accurately, ensuring that players who regularly outplay their opponents are ranked highly, and that isolated strong performances do not artificially inflate scores.
Furthermore, one might implement such a system in real tournaments as either a parallel ranking metric or as a staged replacement for the MP standings. This would help Bridge the gap between veterans with high accumulated totals and newer but strong-performing players, giving both cohorts a fair path to recognition.

7.4 Conclusion

In this discussion, we introduced a fuzzy logic approach to evaluate Bridge pair performance, offering a fresh viewpoint alongside the threshold-based method. While both methods consistently identified top and bottom pairs, their mid-tier rankings occasionally diverged, reflecting distinct ways of quantifying board-level outcomes. Such differences underscore the influence of random factors, multi-stage formats, and categorization choices on competition results.

Analysis of the 2024 National Bridge Championships Open Pairs event further demonstrated how randomness can amplify ranking fluctuations, especially for middle-ranked pairs. Yet, consistent strong performance across multiple boards remains a reliable indicator of high skill, as shown by final placements and bootstrap confidence intervals. These observations lend support to adopting flexible models—like the fuzzy logic or threshold-based approaches—that better accommodate short-term variability.

At the same time, we highlighted the limitations of the World Bridge Federation's traditional Master Points system, which tends to favor long-standing competitors. By prioritizing recent performance, head-to-head outcomes, and variable board weighting, alternative ranking schemes can offer a more equitable route for emerging talents to gain recognition. In future tournaments, implementing such combined methods may offer a more dynamic ranking approach, thereby creating a fairer and more accurate reflection of player skill. Overall, the interplay of fuzzy logic, threshold estimation, and parametric analysis provides a richer, more nuanced perspective on Bridge tournaments. Future work might involve combining these methodologies, or refining them further, to deliver fairer, more accurate rankings for all participants.

8 Conclusion

In this thesis, we have developed and applied a comprehensive framework for analyzing duplicate contract Bridge tournaments, combining new statistical modelling techniques with novel performance evaluation methods. We also study the strength of Bridge players, building on the work of Yu and Lam [34]. We have reproduced their results and extended that approach by introducing a new statistical analysis, thereby enriching the existing methodology for pairwise skill assessment. Our research resulted in several key findings and contributions, both theoretical and practical.

8.1 Key findings and contributions

We introduced a threshold-based extension of the Bradley-Terry model [4] for ranking players (pairs) in duplicate Bridge, a context in which tied results present a statistical difficulty. In this ranking framework, bias can arise if ties are disregarded or treated as negligible. Therefore, we address this bias by incorporating a dedicated tie parameter, enabling us to capture the nontrivial likelihood of equally matched outcomes. This new model addresses a limitation of classical paired-comparison approaches, which assume every comparison produces a winner. We derived the model's likelihood function and showed analytically that the total match points and number of ties are sufficient statistics for estimating the skill parameters $\{\theta_i\}$ and the tie parameter. Furthermore, we proposed a statistically rigorous selection procedure that identifies the top-performing pair with a pre-specified confidence level, backed by theoretical guarantees (as presented in theorem A.1 and theorem A.2). These mathematical contributions provide a strong foundation for the model's validity and utility.

We validated our model, which extends Yu and Lam's permutation model by incorporating a tie parameter, on both a published dataset and a new championship dataset. Using data from a four-table Howell movement [34], we demonstrated that our threshold model replicates and confirms previous results, successfully ranking the pairs in line with the original analysis and detecting significant skill differences via likelihood ratio tests. We then applied the model to the 2024 China National Bridge Championship (Open Pairs) data, which included 12 finalist pairs and 44 boards in the final round. The model provided a clear ranking of the pairs, identifying the eventual champions and top finishers as those with the highest θ estimates. We also quantified the uncertainty in the rankings by constructing confidence intervals for each θ_i and found that the top and bottom ranks were separated by statistically significant margins. The tie parameter φ estimated from this data indicated a meaningful probability of ties, validating the need for its inclusion. Compared to permutation-based methods that treat ties as statistical artifacts, the threshold model offers an explicit and interpretable structure: when the skill gap is below a certain threshold γ, a tie is the expected outcome. This not only improves model fit but also aligns better with the empirical behavior of close matches. As a result, our method is both theoretically sound and practically more transparent. Overall, the model's performance on real-world data confirms its effectiveness and practical relevance for tournament analysis.

To complement our ranking model, we applied a fuzzy logic approach from Voskoglou(2014) [40] to evaluate players' performance from a different perspective. Each pair's board results were mapped into qualitative categories (Excellent, Very Good, Good, Mediocre, Unsatisfactory), and we computed fuzzy centroid coordinates (x_c, y_c) that summarize their performance level and consistency. This analysis yielded additional insights, such as identifying which pairs had very stable performances and which were prone to high variability. When comparing the fuzzy logic outcomes with the threshold model rankings, we found them largely consistent on the extremes (the best and worst pairs), while providing nuanced contrasts for mid-ranked pairs. This dual analysis demonstrated that our fuzzy metric x_c can serve as an alternative performance score, and y_c as an indicator of consistency or volatility, thereby offering valuable information that a single-number ranking cannot capture.

Our results shed light on the dynamics of high-level Bridge competitions. We observed that, over

the course of a full tournament, skill is the predominant factor in final rankings. An expectation of the duplicate format that our analysis confirmed with empirical evidence. At the same time, we documented significant ranking shifts between the preliminary, semifinal, and final stages of the championship, attributing these to the inherent randomness in short-term match-point results and the varying field of opponents. We discussed how the relative scoring system and the need for strategic adjustments introduce variance that can benefit or penalize pairs in shorter spans. These findings highlight the importance of having multiple boards and stages to achieve fair outcomes. They also suggest that our methods (particularly the fuzzy y_c metric and the statistical confidence intervals for θ) can help discern whether a given ranking difference was likely due to skill or merely due to chance. Such understanding is important for players and organizers alike. For instance, it can inform decisions on tournament format or the interpretation of results from any single event.

8.2 Practical significance

The outcomes of this research have practical implications for Bridge analytics and tournament organization. The threshold model we developed can be used as a tool for improving ranking fairness and accuracy. For example, tournament directors could use our model alongside traditional scoring to identify discrepancies or to provide players with an objective skill rating over a series of events. Coaches and players, on the other hand, can benefit from the fuzzy logic analysis: by understanding whether a pair's performance was consistently above average or a mix of brilliant successes and costly errors, training can be tailored accordingly. More broadly, our approach illustrates how combining rigorous statistical models with interpretable fuzzy assessments can yield a comprehensive evaluation system, a concept that could be applied not only in Bridge but in other competitive domains (such as other mind sports or team competitions). Ultimately, our work contributes to the goal of making competitive Bridge results more analyzable, explainable, and justifiable, thereby enhancing the transparency and educational value of the game's outcomes.

8.3 Future research directions

Building on this work, we identify several avenues for future research and improvement. One direction is to extend the threshold model to account for potential variability in player performance. For instance, a hierarchical or dynamic model could allow each pair's skill to evolve across the phases of a tournament, capturing momentum or fatigue. Additionally, incorporating covariates (like players' experience levels or system complexity) might explain some of the performance variance and could be included in an extended model.

Further testing of our methods on different tournament formats and scoring systems is important. Applying the model to IMP scoring games or team-of-four events would examine its generality. We also suggest conducting simulation studies to see how the model performs under various known skill distributions and tie probabilities – this would help quantify the conditions under which the model is most reliable.

The fuzzy logic approach could be refined by exploring alternative definitions of performance categories or using a larger set of linguistic terms to capture gradations of performance. One could also investigate a continuous fuzzy scoring system that avoids hard cut-offs (for example, using membership functions that assign degrees to all categories for a given board result). This might smooth out some of the sensitivity we identified regarding category thresholds.

Finally, an intriguing direction would be to integrate the statistical and fuzzy approaches into a unified framework. This could involve using the fuzzy scores as features in a predictive model of match outcomes or conversely, using statistical model outputs to inform adaptive fuzzy sets. With the growing availability of detailed Bridge data (including bidding and play records), machine learning techniques, possibly combining our ranking insights with features extracted from play-by-play data, could further enhance performance assessment. Such interdisciplinary efforts at the interface of statistics, artificial intelligence, and game theory would continue to advance the analytical depth in understanding games like Bridge.

In closing, this thesis has advanced the quantitative analysis of Bridge by providing a new modeling approach for rankings and by demonstrating the benefits of blending statistical and fuzzy logic methods. We have shown that by carefully modeling the structure of duplicate competitions and by examining performance from multiple viewpoints, one can extract meaningful conclusions about player skill and the nature of competition even in a game as complex as Bridge. The mathematical contributions, including the extension of the Bradley–Terry model to incorporate ties with proven sufficiency and selection results form a foundation for future analytical studies of competitive rankings. The practical insights, such as distinguishing consistent excellence from volatile performance, offer valuable guidance for players and organizers aiming to foster fairness and excellence in tournament play. We hope that our work spurs further research at the intersection of statistics, data science, and competitive game analysis, and that it ultimately contributes to more informed and equitable competition in the timeless game of Bridge.

References

[1] American Contract Bridge League. *Laws of Duplicate Bridge*. American Contract Bridge League, Horn Lake, MS, 2014.

[2] American Contract Bridge League. *Laws of Rubber Bridge*. American Contract Bridge League, Horn Lake, MS, 2014. Archived (PDF) from the original on 6 April 2016.

[3] B. Babington Smith. Discussion on professor ross's paper. *Journal of the Royal Statistical Society: Series B (Methodological)*, 12(1):41–59, 1950.

[4] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[5] Chinese Contract Bridge Association. 2024 national bridge championship results. https://www.ccba.org.cn/express/expressinfo.aspx?expressid=16473, 2024. [Accessed 12 Jan. 2025].

[6] Chinese Contract Bridge Association. History of national bridge tournaments. https://www.ccba.org.cn/Tour/HistoryList.aspx, 2024. [Accessed 12 Jan. 2025].

[7] Wikipedia contributors. Contract bridge. https://en.wikipedia.org/wiki/Contract_bridge, 2025. [Accessed 26 Feb. 2025].

[8] Douglas E Critchlow and Michael A Fligner. Ranking models with item covariates. In *Probability models and statistical analyses for ranking data*, pages 1–19. Springer, 1993.

[9] Douglas E Critchlow, Michael A Fligner, and Joseph S Verducci. Probability models on rankings. *Journal of mathematical psychology*, 35(3):294–318, 1991.

[10] Roger R Davidson. On extending the bradley-terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65(329):317–328, 1970.

[11] Ronald A Fisher. On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 222(594-604):309–368, 1922.

[12] Henry G. Francis, Alan F. Truscott, and Dorothy A. Francis. *The Official Encyclopedia of Bridge*. American Contract Bridge League, Memphis, TN, 6th edition, 2001. Archived (PDF) from the original on 9 October 2022.

[13] Ian Frank and David Basin. Search in games with incomplete information: A case study using bridge card play. *Artificial Intelligence*, 100(1-2):87–123, 1998.

[14] Timothy Furtak and Michael Buro. Recursive monte carlo search for imperfect information games. In *2013 IEEE Conference on Computational Inteligence in Games (CIG)*, pages 1–8. IEEE, 2013.

[15] GEM Bridge. Preliminary round results. http://www.gembridge.cn/score/PairSectionRanks?tourStart=2024-04-13&tour=27910&event=b824bd9f-59bf-45f0-a691-263f87229ada&section=306cda36-00c9-413f-bd56-1d8e81d1cc22&flight=&from=ccba, 2024. [Accessed 15 Mar. 2025].

[16] GemBridge. Board-by-board results for the 2024 national bridge championship open pairs final. http://www.gembridge.cn/score/PairBoards?tourStart=2024-04-13&tour=27910&event=b824bd9f-59bf-45f0-a691-263f87229ada&section=aa403fea-8ca0-4fab-9bf1-12c170ec4b8d&board=1&from=ccba, 2024. [Accessed 12 Jan. 2025].

[17] GemBridge. Final rankings for the 2024 national bridge championship open pairs final. http://www.gembridge.cn/score/PairSectionRanks?tourStart=2024-04-13&tour=27910&event=b824bd9f-59bf-45f0-a691-263f87229ada&section=aa403fea-8ca0-4fab-9bf1-12c170ec4b8d&flight=&from=ccba, 2024. [Accessed 12 Jan. 2025].

[18] GemBridge. Pair start list for the 2024 national bridge championship open pairs final. http://www.gembridge.cn/score/PairStartList?tourStart=2024-04-13&tour=27910&event=b824bd9f-59bf-45f0-a691-263f87229ada&section=aa403fea-8ca0-4fab-9bf1-12c170ec4b8d&letter=All&from=ccba, 2024. [Accessed 12 Jan. 2025].

[19] GemBridge. Schedule for the 2024 national bridge championship. http://www.gembridge.cn/score/Results?tourStart=2024-04-11&tour=27910&from=ccba, 2024. [Accessed 12 Jan. 2025].

[20] GemBridge. Semifinal round results. http://www.gembridge.cn/score/PairSectionRanks?tourStart=2024-04-13&tour=27910&event=b824bd9f-59bf-45f0-a691-263f87229ada&section=1a087081-d85d-4fb5-bca0-c0affc2456c5&flight=&from=ccba, 2024. [Accessed 15 Mar. 2025].

[21] Walter B Gibson. *Hoyle's Modern Encyclopedia of Card Games: Rules of All the Basic Games and Popular Variations*. Crown, 1974.

[22] Shanti S Gupta. On some multiple decision (selection and ranking) rules. *Technometrics*, 7(2):225–245, 1965.

[23] Eddie Kantar. *Bridge For Dummies*. Wiley Publishing, Inc., Hoboken, NJ, 2nd edition, 2006.

[24] Gideon Keren. Facing uncertainty in the game of bridge: A calibration study. *Organizational Behavior and Human Decision Processes*, 39(1):98–114, 1987.

[25] Bernard Osgood Koopman. On distributions admitting a sufficient statistic. *Transactions of the American Mathematical society*, 39(3):399–409, 1936.

[26] Dmitry Kovalev, Robert M Gower, Peter Richtárik, and Alexander Rogozin. Fast linear convergence of randomized bfgs. *arXiv preprint arXiv:2002.11337*, 2020.

[27] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[28] Aryan Mokhtari and Alejandro Ribeiro. Global convergence of online limited memory bfgs. *The Journal of Machine Learning Research*, 16(1):3151–3181, 2015.

[29] Douglas C Montgomery. *Design and analysis of experiments*. John wiley & sons, 2017.

[30] Alexander McFarlane Mood. Introduction to the theory of statistics. 1950.

[31] Jerzy Neyman and Egon Sharpe Pearson. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

[32] David Owen. Turning tricks: The rise and fall of contract bridge. *The New Yorker*, 10, 2007.

[33] People's Daily Online. Deng xiaoping and bridge. http://cpc.people.com.cn/n1/2017/1213/c69113-29703199.html, 2017. [Accessed 25 Feb. 2025].

[34] LH Philip and K Lam. Analysis of duplicate bridge tournament data. *Scandinavian journal of statistics*, pages 621–633, 1996.

[35] Edwin James George Pitman. Sufficient statistics and intrinsic accuracy. In *Mathematical Proceedings of the cambridge Philosophical society*, volume 32, pages 567–579. Cambridge University Press, 1936.

[36] Samantha Punch and Miriam Snellgrove. Playing your life: Developing strategies and managing impressions in the game of bridge. *Sociological Research Online*, 26(3):601–619, 2021.

[37] Terence Reese. *Teach Yourself Bridge*. Teach Yourself Books, Hodder and Stoughton, London, 1990.

[38] Ashley Rogers, Miriam Snellgrove, and Samantha Punch. Between equality and discrimination: The paradox of the women's game in the mind-sport bridge. *World Leisure Journal*, 64(4):342–360, 2022.

[39] Stephen JJ Smith, Dana S Nau, and Thomas A Throop. Total-order multi-agent task-network planning for contract bridge. In *AAAI/IAAI, Vol. 1*, pages 108–113. Citeseer, 1996.

[40] Michael Gr Voskoglou. Assessing the players' performance in the game of bridge: A fuzzy logic approach. *arXiv preprint arXiv:1404.7279*, 2014.

[41] Wikipedia contributors. Auction bridge. https://en.wikipedia.org/wiki/Auction_bridge, 2025. [Accessed 26 Feb. 2025].

[42] Wikipedia contributors. Bauke muller. https://en.wikipedia.org/wiki/Bauke_Muller, 2025. [Accessed 16 Feb. 2025].

[43] Wikipedia contributors. Latin square — wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Latin_square, 2025. [Accessed 26 Feb. 2025].

[44] Wikipedia contributors. Pierre zimmermann (bridge). https://en.wikipedia.org/wiki/Pierre_Zimmermann_(bridge), 2025. [Accessed 16 Feb. 2025].

[45] Samuel S Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The annals of mathematical statistics*, 9(1):60–62, 1938.

[46] World Bridge Federation. Wbf open ranking (by mps). http://www.wbfmasterpoints.com/rankingOpen.asp, 2023. [Accessed 16 Feb. 2025].

[47] Chih-Kuan Yeh, Cheng-Yu Hsieh, and Hsuan-Tien Lin. Automatic bridge bidding using deep reinforcement learning. *IEEE Transactions on Games*, 10(4):365–377, 2018.

A Supplementary analyses by Yu and Lam

A.1 Additional modifications of match point scoring by Yu and Lam

A.1.1 Board 14

Table 18: Table Assignments for Board 14

Table	N/S Pair	E/W Pair
1	5	7
2	6	3
3	1	2
4	8	4

Table 19: Raw Scores and MP Scores for N/S Pairs on Board 14

N/S Pair	Raw Score	MP Score
5	250	0
6	420	2.5
1	400	1
8	420	2.5

Table 20: Raw Scores and MP Scores for E/W Pairs on Board 14

E/W Pair	MP Score	Raw Score
7	3	−250
3	0.5	−420
2	2	−400
4	0.5	−420

The MP score for pair 2 in the original dataset figure 5 board 14 was incorrectly recorded as 0.5. The correct MP score should be 2.

A.1.2 Board 15

Table 21: Table Assignments for Board 15

Table	N/S Pair	E/W Pair
1	5	7
2	6	3
3	1	2
4	8	4

Table 22: Raw Scores and MP Scores for N/S Pairs on Board 15

N/S Pair	Raw Score	MP Score
5	−500	0
6	+50	2.5
1	−130	1
8	+50	2.5

Table 23: Raw Scores and MP Scores for E/W Pairs on Board 15

E/W Pair	MP Score	Raw Score
7	3	+500
3	0.5	−50
2	2	+130
4	0.5	−50

The MP score for pair 2 in the original dataset figure 5 board 15 was incorrectly recorded as 0.5. The correct MP score should be 2.

A.2 Comparison of optimization methods

To further examine whether the choice of optimization algorithm contributes to the observed discrepancies, I applied three different methods: BFGS, L-BFGS-B, and Newton-CG. The main results presented in the paper are based on BFGS, while L-BFGS-B and Newton-CG were tested for comparison.
L-BFGS-B is a quasi-Newton method that approximates the Hessian matrix using limited memory, making it suitable for large-scale problems with or without bound constraints. Newton-CG is a second-order method that uses conjugate gradient techniques to compute approximate Newton steps and is often more efficient near the solution due to explicit curvature information.

```
Converged: True
Log-likelihood: -143.69384120048267
θ_1 = 0.1336
θ_2 = 0.1899
θ_3 = -0.7131
θ_4 = 0.3866
θ_5 = 0.1839
θ_6 = -0.9896
θ_7 = 0.7906
θ_8 = 0.0000
ϕ = 0.2938
Ranking of θ by skill level (from highest to lowest):
Rank 1: θ_7 = 0.7906
Rank 2: θ_4 = 0.3866
Rank 3: θ_2 = 0.1899
Rank 4: θ_5 = 0.1839
Rank 5: θ_1 = 0.1336
Rank 6: θ_8 = 0.0000
Rank 7: θ_3 = -0.7131
Rank 8: θ_6 = -0.9896
```

(a) 'L-BFGS-B' Optimized Method

```
Converged: True
Full model log-likelihood: -143.69384119168296
θ_1 = 0.1336
θ_2 = 0.1899
θ_3 = -0.7132
θ_4 = 0.3866
θ_5 = 0.1838
θ_6 = -0.9896
θ_7 = 0.7906
θ_8 = 0.0000
ϕ = 0.2938
Ranking of θ by skill level (from highest to lowest):
Rank 1: θ_7 = 0.7906
Rank 2: θ_4 = 0.3866
Rank 3: θ_2 = 0.1899
Rank 4: θ_5 = 0.1838
Rank 5: θ_1 = 0.1336
Rank 6: θ_8 = 0.0000
Rank 7: θ_3 = -0.7132
Rank 8: θ_6 = -0.9896
```

(b) Newton-CG Optimized Method

Figure 22: Comparison of Optimization Methods

The optimization results presented in Figure 22a and Figure 22b show that both methods converged to nearly identical outcomes. The log-likelihood values for L-BFGS-B and Newton-CG are -143.69384120048267 and -143.69384119168296, respectively, indicating negligible differences. Similarly, the estimated skill parameters (θ) and rankings are consistent across both methods, with Player 7 having the highest skill ($\theta_7 \approx 0.7906$) and Player 6 the lowest ($\theta_6 \approx -0.9896$). The estimated tie probability ($\phi \approx 0.2938$) is also the same across all methods.
These results suggest that, under the current implementation, different optimization methods do not materially affect the estimation outcome. However, this does not entirely rule out the influence of optimizer-specific settings—such as initialization points, line search strategy, or stopping criteria—which may still lead to different solutions in more sensitive scenarios.

A.3 Supplementary methods

This section provides theoretical extensions and refinements to the ranking framework introduced in the main text. We begin with a confidence-based subset selection method for identifying the top pair, and further propose a weighted correction to improve discrimination.

A.3.1 Parameter estimation and confidence-based ranking

This appendix presents an extension of the main model by introducing a framework to select the most skillful pair in a bridge tournament while accounting for uncertainty in parameter estimation. It begins with a discussion of parameter identifiability, followed by the construction of a confidence subset that contains the best pair with high probability.

Estimating the skill level parameters $\theta_1, \ldots, \theta_n$ and the tie parameter ϕ poses an over-parameterization problem. To resolve this, one parameter must be fixed. For instance, in a Howell movement, we typically set $\theta_n = 0$, allowing the remaining θ_i to be identified.

In Mitchell movements, where the $n = 2k$ pairs are divided into two disjoint groups $G_1 = \{1, \ldots, k\}$ and $G_2 = \{k+1, \ldots, 2k\}$, only within-group comparisons are made. Hence, one may set $\theta_k = \theta_{2k} = 0$ to ensure identifiability in both groups. Maximum likelihood estimates (MLEs) are then obtained by maximizing the log-likelihood using numerical methods such as the quasi-Newton algorithm.

In Mitchell tournaments, the function $C(\theta, \phi)$ used in the likelihood is symmetric within each group. As a result, the MLE rankings for θ_i typically align with the total match point (MP) scores, justifying the use of MP rankings as skill proxies. However, in designs such as Howell, where the symmetry assumption breaks down and comparisons are more comprehensive but less structured, MLE-based rankings are generally more reliable.

A.3.2 Constructing a confidence subset to contain the best pair

To address the uncertainty associated with selecting the best-performing pair from a group of closely matched competitors, we apply a confidence-based multiple comparison method adapted from Yu and Lam (1996). Beyond point estimation, a common objective is to identify the best-performing pair—the one with the highest true skill parameter:

$$\theta_{(n)} = \max\{\theta_1, \ldots, \theta_n\}.$$

Since the true ranking is unknown and sampling variability may blur differences between skill estimates, we aim to construct a *confidence subset* $S \subseteq \{1, \ldots, n\}$ such that:

$$P[(n) \in S] \geq 1 - \alpha.$$

This idea, originating from Gupta (1965) [22], has been adapted by Yu and Lam (1996) into the context of duplicate bridge under a permutation model. Instead of selecting a single pair with the largest estimate $\hat{\theta}_i$, we identify a set of candidates whose performances are statistically indistinguishable from the top.

This method is particularly useful when many MLEs $\hat{\theta}_i$ are close, and the observed differences may be attributed to noise. It provides a conservative, multiple-comparison-based approach that balances confidence with selectivity.

The following theorem, adapted from Yu and Lam (1996), offers a concrete way to construct such a subset using Bonferroni-adjusted confidence intervals on pairwise differences.

Theorem A.1: Constructing a random subset that contains the best pair

Let W_{ij} be the estimate for the variance of $\hat{\theta}_i - \hat{\theta}_j$. Consider constructing a random subset S, which contains pairs that satisfy the following condition:

$$S = \{i : \hat{\theta}_i \geq \max(\hat{\theta}_j - z_{\alpha/(n-1)}\sqrt{W_{ji}}),\ j \neq i\}$$

where z_α is the upper quantile point of a standard normal distribution. The theorem claims that if the number of boards B is sufficiently large, the following holds:

$$P[(n) \in S] \geq 1 - \alpha$$

That is, the probability that the best pair is included in the random subset S is at least $1 - \alpha$.

Proof of theorem A.1

We aim to ensure that the best pair (n) appears in S with probability at least $1 - \alpha$.
Under a normal or asymptotic approximation, for each difference $\hat{\theta}_j - \hat{\theta}_c$, we have variance $W_{j,c}$.
Using a Bonferroni correction at $\alpha/(n-1)$ for n parameters, it follows that

$$P\Big[\theta_j - \theta_c \ \geq\ \hat{\theta}_j - \hat{\theta}_c \ -\ z\big(\tfrac{\alpha}{n-1}\big)\sqrt{W_{j,c}},\ \forall j \neq c\Big] \ \geq\ 1 - \alpha.$$

Let $c = (n)$. If $\theta_{(n)}$ is truly largest, then $\theta_j - \theta_{(n)} \leq 0$ for all $j \neq (n)$. Hence, on the above high-probability event,

$$0 \ \geq\ \theta_j - \theta_{(n)} \ \geq\ \hat{\theta}_j - \hat{\theta}_{(n)} \ -\ z\big(\tfrac{\alpha}{n-1}\big)\sqrt{W_{j,(n)}},$$

which rearranges to

$$\hat{\theta}_{(n)} \ \geq\ \hat{\theta}_j \ -\ z\big(\tfrac{\alpha}{n-1}\big)\sqrt{W_{j,(n)}} \quad (\forall j \neq (n)).$$

We define Construct S

$$S \ =\ \Big\{i : \hat{\theta}_i \ \geq\ \max_{j \neq i}\big[\hat{\theta}_j - z\big(\tfrac{\alpha}{n-1}\big)\sqrt{W_{j,i}}\big]\Big\}.$$

Since (n) satisfies this inequality against all $j \neq (n)$, it follows that $(n) \in S$. Hence $P((n) \in S) \geq 1 - \alpha$.

□

In section A.3.3, we build upon this method by introducing a weighted correction, which adjusts the comparison thresholds using the magnitude of $\hat{\theta}_j$ to improve discrimination and reduce the size of S. A full comparison of the original and weighted methods is presented in section A.3.4. The full Python implementation, including Hessian computation and pairwise variance extraction, is provided in Appendix D.

A.3.3 Weighted-correction selection method

While the original subset selection method provides valid coverage, it may yield overly large candidate sets when many skill estimates are similar. To solve this, we introduce a weighted correction that sharpens discrimination by scaling the confidence threshold based on the strength of the compared player.
This approach introduces a weighting factor ω_j, which scales the variance correction term based on the magnitude of the estimated skill level $\hat{\theta}_j$.

Theorem A.2: Weighted-Correction Selection Method

Let $\hat{\theta}_i \quad (i = 1, \ldots, n)$ denotes the estimated skill levels of the n competing pairs (players). Define$\widehat{W}_{ij} \approx \mathrm{Var}(\hat{\theta}_i - \hat{\theta}_j)$ as the approximate variance of the difference in skill estimates between pairs i and j. Further, let $z_{\alpha/(n-1)}$ be the upper $\alpha/(n-1)$ critical value of the standard normal distribution at significance level $\alpha/(n-1)$. In addition, define a *weighting factor*

$$\omega_j = \frac{1}{1 + |\hat{\theta}_j|}.$$

Under this weighting correction, we construct the subset.

$$S = \left\{ i : \hat{\theta}_i \geq \max_{j \neq i} \left[\hat{\theta}_j - \omega_j \cdot z_{\alpha/(n-1)} \sqrt{\widehat{W}_{ij}} \right] \right\}.$$

Then, if the "true best" pair is denoted by (n^*) (i.e. the one with the largest true skill parameter $\theta_{(n^*)}$), we have

$$P[(n^*) \in S] \geq 1 - \alpha \tag{26}$$

In other words, with probability at least $(1 - \alpha)$, the set S contains the truly best pair.

Proof of Theorem A.2

We build upon the argument of Theorem A.1. In large-sample settings, one can construct a $(1 - \alpha)$ simultaneous confidence region for all differences $\theta_j - \theta_c \ (j \neq c)$ such that

$$\theta_j - \theta_c \geq \hat{\theta}_j - \hat{\theta}_c - z_{\alpha/(n-1)} \sqrt{\widehat{W}_{j,c}}, \quad \forall j \neq c,$$

Holds jointly with probability at least $1 - \alpha$.
Next, define the weighting factor

$$\omega_j = \frac{1}{1 + |\hat{\theta}_j|}.$$

Replacing $z_{\alpha/(n-1)} \sqrt{\widehat{W}_{j,c}}$ by $\omega_j \, z_{\alpha/(n-1)} \sqrt{\widehat{W}_{j,c}}$ still preserves the joint coverage, because $\omega_j \leq 1$ for all j. Hence, with probability at least $1 - \alpha$,

$$\theta_j - \theta_c \geq \hat{\theta}_j - \hat{\theta}_c - \omega_j \, z_{\alpha/(n-1)} \sqrt{\widehat{W}_{j,c}}, \quad \forall j \neq c.$$

Now, suppose the true best pair is n^*, so $\theta_{(n^*)} \geq \theta_j$ for all $j \neq n^*$. Thus

$$\theta_j - \theta_{(n^*)} \leq 0,$$

and under the above event,

$$0 \geq \theta_j - \theta_{(n^*)} \geq \hat{\theta}_j - \hat{\theta}_{(n^*)} - \omega_j \, z_{\alpha/(n-1)} \sqrt{\widehat{W}_{j,(n^*)}}, \quad \forall j \neq n^*.$$

Rewriting yields

$$\hat{\theta}_{(n^*)} \geq \hat{\theta}_j - \omega_j \, z_{\alpha/(n-1)} \sqrt{\widehat{W}_{j,(n^*)}}, \quad \forall j \neq n^*.$$

By the definition of S,

$$(n^*) \in \left\{ i : \hat{\theta}_i \geq \max_{j \neq i} \left[\hat{\theta}_j - \omega_j \, z_{\alpha/(n-1)} \sqrt{\widehat{W}_{j,i}} \right] \right\}.$$

Hence $(n^*) \in S$. Because all these comparisons hold simultaneously with probability at least $1 - \alpha$, it follows that

$$P\big[(n^*) \in S\big] \;\geq\; 1 - \alpha,$$

As claimed. □

Remarks.

- The factor $\omega_j = 1/(1 + |\hat{\theta}_j|)$ provides a small or large offset depending on the magnitude of $\hat{\theta}_j$. If $\hat{\theta}_j$ is large in magnitude, then ω_j is smaller, effectively raising the bar for other players i to surpass j.
- When $\omega_j = 1$ identically, we recover the original unweighted statement of Theorem A.1 (i.e. with no weighting factor).
- In large-sample (asymptotic) contexts, these weighted bounds still retain the same simultaneous confidence guarantee: $P[(n^*) \in S] \geq 1 - \alpha$

A.3.4 Comparisons across two methods for selecting the best pair

To assess the empirical performance of the weighted-correction method introduced in Theorem A.2, we compare it against the original unweighted procedure from theorem A.1on two independent datasets.

Comparison using data from section 4.1 Applying the original (unweighted) selection rule yields a confidence subset

$$S_{\text{unweighted}} = \{1,\, 2,\, 4,\, 5,\, 7,\, 8\}.$$

In contrast, the weighted-correction procedure produces a smaller subset

$$S_{\text{weighted}} = \{4,\, 7\},$$

achieving the same 95% confidence level while excluding more borderline cases.

Comparison using data from section 5.1.4 The original method identifies the following subset:

$$S_{\text{unweighted}} = \{1,\, 2,\, 3,\, 4,\, 5,\, 8,\, 10,\, 11,\, 12\},$$

indicating that nine players remain statistically indistinguishable from the top. In comparison, the weighted approach produces a more concise subset:

$$S_{\text{weighted}} = \{1,\, 3,\, 4,\, 5,\, 10,\, 11,\, 12\},$$

which remains consistent with previous findings, while providing sharper exclusion.
Across both datasets, the new weighted approach retains the same $(1 - \alpha)$ coverage condition but achieves:

- **Stronger discrimination:** borderline players are more confidently excluded.
- **Reasonable set size:** S avoids being unnecessarily large.
- **No coverage loss:** $\Pr(\text{true best} \in S) \geq 1 - \alpha$ stays valid.

Hence, the new method consistently outperforms the original scheme in multi-comparison selection scenarios.

B Python code for permutation model(use data from Yu and Lam)

B.1 MLE

```
import numpy as np
from scipy.optimize import minimize

alpha_patterns = [
    (2,3,5,8),
    (3,4,6,8),
    (4,5,7,8),
    (5,6,1,8),
    (6,7,2,8),
    (7,1,3,8),
    (1,2,4,8)
]

beta_patterns = [
    (4,7,6,1),
    (5,1,7,2),
    (6,2,1,3),
    (7,3,2,4),
    (1,4,3,5),
    (2,5,4,6),
    (3,6,5,7)
]

alpha = []
beta = []
for i in range(7):
    a_pat = alpha_patterns[i]
    b_pat = beta_patterns[i]
    for _ in range(4):
        alpha.append(a_pat)
        beta.append(b_pat)

alpha = np.array(alpha)
beta = np.array(beta)

B = 28
n = 8
T = 4

mp_scores = np.array([
    [2,   2.5,  0,    0.5,  2.5,  0.5,  3,    1   ],
    [1,   1,    0,    2,    3,    0,    3,    2   ],
    [1,   1,    0,    2,    3,    0,    3,    2   ],
    [2,   2,    0,    1,    3,    0,    3,    1   ],
    [1.5, 0,    0,    1.5,  3,    1.5,  1.5,  3   ],
    [3,   1.5,  3,    0,    0,    1.5,  1.5,  1.5 ],
    [0,   1,    1,    3,    2,    0,    3,    2   ],
    [2,   1,    3,    1,    0,    0,    3,    2   ],
    [2,   1,    3,    3,    2,    0,    1,    0   ],
    [1,   3,    2,    3,    0,    0,    2,    1   ],
    [1.5, 3,    0,    1.5,  0,    1.5,  1.5,  3   ],
    [2,   1,    3,    3,    2,    0,    1,    0   ],
    [3,   0,    2,    1,    0,    1,    3,    2   ],
    [1,   2,    2,  0.5,  0,    2.5,  3,    2.5
    ],
    [1,   2,    2,  0.5,  0,    2.5,  3,    2.5
```

```
    ],
    [2,   1,    2,    3,    3,    1,    0,    0   ],
    [2,   3,    0,    2,    2,    1,    1,    1   ],
    [0.5, 2.5,  0.5,  2,    3,    2.5,  1,    0   ],
    [3,   3,    0,    2,    1,    0,    1,    2   ],
    [2.5, 3,    0,    2.5,  1,    0.5,  0.5,  2   ],
    [1,   0.5,  2.5,  0.5,  2,    3,    2.5,  0   ],
    [0.5, 0.5,  0.5,  2.5,  2.5,  0.5,  2.5,  2.5 ],
    [2,   2,    0,    3,    1,    0,    1,    3   ],
    [0.5, 0,    0.5,  2.5,  2.5,  1,    3,    2   ],
    [0,   3,    3,    2,    1,    0,    2,    1   ],
    [2,   0,    1,    2,    1,    3,    1,    2   ],
    [3,   2,    0,    0,    3,    1,    2,    1   ],
    [1.5, 3,    1.5,  1.5,  1.5,  0,    3,    0   ]
])

assert alpha.shape == (28,4)
assert beta.shape == (28,4)
assert mp_scores.shape == (28,8)

boards_data = []
for b in range(B):
    ns_pairs = alpha[b,:]
    scores = mp_scores[b, ns_pairs-1]
    L_b = []
    D_b = []
    for i in range(T):
        for j in range(i+1,T):
            if np.isclose(scores[i], scores[j]):
                D_b.append((i,j))
            elif scores[i] < scores[j]:
                L_b.append((i,j))
            else:
                L_b.append((j,i))
    boards_data.append({
        'alpha': ns_pairs,
        'beta': beta[b,:],
        'L_b': L_b,
        'D_b': D_b
    })

def unpack_params(x, n):
    psi = x[-1]
    phi = np.exp(psi)/(1+np.exp(psi))
    theta = np.zeros(n)
    theta[:n-1] = x[:-1]
    theta[n-1] = 0.0
    return theta, phi

def log_likelihood_and_gradient(x, boards_data, n, T):
    theta, phi = unpack_params(x, n)
    ll = 0.0
    grad_theta = np.zeros(n)
    grad_psi = 0.0

    for bd in boards_data:
        ns_pairs = bd['alpha']
        ew_pairs = bd['beta']
        L_b = bd['L_b']
        D_b = bd['D_b']

```

```
        lambda_b = np.zeros(T)
        dlam_dtheta = np.zeros((T,n))
        for t in range(T):
            i_ns = ns_pairs[t]
            i_ew = ew_pairs[t]
            val_ns = theta[i_ns-1]
            val_ew = theta[i_ew-1]
            lam = np.exp(val_ns - val_ew)
            lambda_b[t] = lam
            dlam = np.zeros(n)
            dlam[i_ns-1] = lam
            dlam[i_ew-1] = -lam
            dlam_dtheta[t,:] = dlam

        # L_b part
        for (i,j) in L_b:
            lam_i = lambda_b[i]
            lam_j = lambda_b[j]
            sqrt_term = np.sqrt(lam_i*lam_j)
            denom = lam_i + lam_j + phi*sqrt_term
            ll += np.log(lam_j) - np.log(denom)
            dlam_i_theta = dlam_dtheta[i,:]
            dlam_j_theta = dlam_dtheta[j,:]
            d_sqrt_term_theta = 0.5/sqrt_term*(lam_j*dlam_i_theta + lam_i*
    dlam_j_theta)
            grad_theta += (dlam_j_theta/lam_j) - (1/denom)*(dlam_i_theta +
    dlam_j_theta + phi*d_sqrt_term_theta)
            dldphi = - sqrt_term/denom
            dldpsi = dldphi * phi*(1-phi)
            grad_psi += dldpsi

        # D_b part
        for (i,j) in D_b:
            lam_i = lambda_b[i]
            lam_j = lambda_b[j]
            sqrt_term = np.sqrt(lam_i*lam_j)
            denom = lam_i + lam_j + phi*sqrt_term
            ll += np.log(phi) + 0.5*(np.log(lam_i)+np.log(lam_j)) - np.log(denom
    )
            dlam_i_theta = dlam_dtheta[i,:]
            dlam_j_theta = dlam_dtheta[j,:]
            d_sqrt_term_theta = 0.5/sqrt_term*(lam_j*dlam_i_theta + lam_i*
    dlam_j_theta)
            grad_theta += 0.5*(dlam_i_theta/lam_i + dlam_j_theta/lam_j) - (1/
    denom)*(dlam_i_theta + dlam_j_theta + phi*d_sqrt_term_theta)
            dldphi = 1/phi - sqrt_term/denom
            dldpsi = dldphi*phi*(1-phi)
            grad_psi += dldpsi

    grad = np.zeros(n)
    grad[:n-1] = grad_theta[:n-1]
    grad[-1] = grad_psi
    return -ll, -grad

# MLE for the full model
x_init = np.zeros(n)
res = minimize(lambda x: log_likelihood_and_gradient(x, boards_data, n, T),
               x_init, jac=True, method='BFGS', options={'disp': True})

hat_params = res.x
hat_theta, hat_phi = unpack_params(hat_params, n)
```

```

# Store the log-likelihood value of the full model in ll_full for later use
ll_full = -res.fun

print("Converged:", res.success)
print("Full model log-likelihood:", ll_full)
for i in range(n):
    print(f"theta_{i+1} = {hat_theta[i]:.4f}")
print(f"phi = {hat_phi:.4f}")

theta_with_index = list(enumerate(hat_theta, start=1))
theta_sorted = sorted(theta_with_index, key=lambda x: x[1], reverse=True)
print("Ranking of theta by skill level (from highest to lowest):")
for rank, (idx, val) in enumerate(theta_sorted, start=1):
    print(f"Rank {rank}: theta_{idx} = {val:.4f}")
```

B.2 Parametric bootstrap 95% confidence intervals for each θ_i

```

def prob_outcomes(lam_i, lam_j, phi):
    sqrt_term = np.sqrt(lam_i * lam_j)
    denom = lam_i + lam_j + phi * sqrt_term
    p_i_win = lam_i / denom
    p_j_win = lam_j / denom
    p_tie = (phi * sqrt_term) / denom
    return p_i_win, p_tie, p_j_win

def simulate_data(theta, phi, boards_data):
    new_boards = []
    for bd in boards_data:
        ns_pairs = bd['alpha']
        ew_pairs = bd['beta']
        T = len(ns_pairs)
        lambda_b = []
        for t in range(T):
            i_ns = ns_pairs[t] - 1
            i_ew = ew_pairs[t] - 1
            lam_t = np.exp(theta[i_ns] - theta[i_ew])
            lambda_b.append(lam_t)

        L_b = []
        D_b = []
        for i in range(T):
            for j in range(i + 1, T):
                lam_i = lambda_b[i]
                lam_j = lambda_b[j]
                p_i, p_tie, p_j = prob_outcomes(lam_i, lam_j, phi)
                rnd = np.random.rand()
                if rnd < p_i:
                    L_b.append((j, i))
                elif rnd < p_i + p_tie:
                    D_b.append((i, j))
                else:
                    L_b.append((i, j))

        new_boards.append({
            'alpha': bd['alpha'],
            'beta': bd['beta'],
            'L_b': L_b,
```

```
            'D_b': D_b
        })
    return new_boards

def parametric_bootstrap(boards_data, B=1000):
    theta_hat, phi_hat, _ = fit_davidson_model(boards_data, n)
    boot_thetas = []
    boot_phis = []

    for b in range(B):
        sim_data = simulate_data(theta_hat, phi_hat, boards_data)
        th_b, ph_b, _ = fit_davidson_model(sim_data, n)
        boot_thetas.append(th_b)
        boot_phis.append(ph_b)

    boot_thetas = np.array(boot_thetas)
    boot_phis = np.array(boot_phis)
    ci_lower = np.percentile(boot_thetas, 2.5, axis=0)
    ci_upper = np.percentile(boot_thetas, 97.5, axis=0)

    return {
        'theta_mle': theta_hat,
        'phi_mle': phi_hat,
        'theta_samples': boot_thetas,
        'phi_samples': boot_phis,
        'theta_ci_lower': ci_lower,
        'theta_ci_upper': ci_upper
    }

print("\n=== Starting Parametric Bootstrap Analysis (B=1000) ===")
B = 1000
boot_result = parametric_bootstrap(boards_data, B=B)
theta_ci_lo = boot_result['theta_ci_lower']
theta_ci_hi = boot_result['theta_ci_upper']

print(f"In {B} simulations, the 95% confidence interval for each theta_i is:")
for i in range(n):
    print(f"theta_{i+1} = {theta_mle[i]:.4f}  95% CI = [{theta_ci_lo[i]:.4f}, {
   theta_ci_hi[i]:.4f}]")
```

B.3 Likelihood ratio test

```
def log_likelihood_restricted(x, boards_data, n, T):
    theta_common = x[0]
    psi = x[1]
    phi = np.exp(psi)/(1+np.exp(psi))

    ll = 0.0
    for bd in boards_data:
        L_b = bd['L_b']
        D_b = bd['D_b']
        m = len(L_b)
        d = len(D_b)
        # Likelihood under the restricted model
        ll_board = m*(-np.log(2+phi)) + d*(np.log(phi)-np.log(2+phi))
        ll += ll_board
    return -ll

x_init_res = [0.0, 0.0]
```

```
res_res = minimize(lambda x: log_likelihood_restricted(x, boards_data, n, T),
                   x_init_res, method='BFGS', options={'disp': True})
ll_res = -res_res.fun

print("Restricted model log-likelihood:", ll_res)
print("Restricted model parameters:", res_res.x)

# Calculate LR statistic
LR = -2*(ll_res - ll_full)
print("Likelihood Ratio:", LR)
```

B.4 Tied frequency

```
import math
import itertools


def compute_lambda(ns_pairs, ew_pairs, theta):
    lambda_array = []
    for t in range(len(ns_pairs)):
        i_ns = ns_pairs[t]
        i_ew = ew_pairs[t]
        lam = math.exp(theta[i_ns-1] - theta[i_ew-1])
        lambda_array.append(lam)
    return lambda_array

def P_less(lambda_array, i, j, phi):
    lam_i = lambda_array[i]
    lam_j = lambda_array[j]
    denom = lam_i + lam_j + phi*math.sqrt(lam_i*lam_j)
    return lam_j / denom

def P_equal(lambda_array, i, j, phi):
    lam_i = lambda_array[i]
    lam_j = lambda_array[j]
    denom = lam_i + lam_j + phi*math.sqrt(lam_i*lam_j)
    return (phi*math.sqrt(lam_i*lam_j)) / denom

def generate_partitions(set_elements):
    if not set_elements:
        yield []
        return
    first = set_elements[0]
    for rest_partition in generate_partitions(set_elements[1:]):
        for i in range(len(rest_partition)):
            new_part = rest_partition[:i] + [rest_partition[i] + [first]] +
   rest_partition[i+1:]
            yield new_part
        yield [[first]] + rest_partition

def generate_all_tie_rankings(T):
    elements = list(range(T))
    all_rankings = []
    for partition in generate_partitions(elements):
        for perm in itertools.permutations(partition):
            ordered_part = [tuple(sorted(p)) for p in perm]
            all_rankings.append(tuple(ordered_part))
    # Remove duplicates
    all_rankings = list(set(all_rankings))
```

```
    return all_rankings

def extract_Lb_Db(ranking):
    L_b = []
    D_b = []
    # Equal pairs within each tie group
    for group in ranking:
        if len(group) > 1:
            for pair in itertools.combinations(group, 2):
                i, j = pair
                if i<j:
                    D_b.append((i,j))
                else:
                    D_b.append((j,i))
    # Ordering between different groups
    for a in range(len(ranking)):
        for b in range(a+1, len(ranking)):
            group_a = ranking[a]
            group_b = ranking[b]
            # group_a beats group_b
            for i_idx in group_a:
                for j_idx in group_b:
                    # j_idx loses to i_idx => (j_idx,i_idx)
                    L_b.append((j_idx,i_idx))
    return L_b, D_b

def count_ties(ranking):
    tie_count = 0
    for group in ranking:
        m = len(group)
        if m > 1:
            tie_count += m*(m-1)//2
    return tie_count

def expected_tie_frequency(boards_data, theta, phi, T):
    tie_counts_of_interest = [0,1,2,3,6]
    expected_count = {k:0.0 for k in tie_counts_of_interest}

    all_rankings = generate_all_tie_rankings(T)

    B = len(boards_data)
    for b in range(B):
        ns_pairs = boards_data[b]['alpha']
        ew_pairs = boards_data[b]['beta']
        lambda_array = compute_lambda(ns_pairs, ew_pairs, theta)

        Q_values = []
        tie_of_each = []
        for R in all_rankings:
            L_b, D_b = extract_Lb_Db(R)
            Q_R = 1.0
            for (i,j) in L_b:
                Q_R *= P_less(lambda_array, i, j, phi)
            for (i,j) in D_b:
                Q_R *= P_equal(lambda_array, i, j, phi)

            Q_values.append(Q_R)
            tie_of_each.append(count_ties(R))

        sum_Q = sum(Q_values)
        for idx, qv in enumerate(Q_values):
```

```
            p = qv/sum_Q if sum_Q>0 else 0.0
            k = tie_of_each[idx]
            if k in expected_count:
                expected_count[k] += p

    return expected_count

# Use previously obtained MLE parameters
hat_params = res.x
theta = np.zeros(n)
theta[:n-1] = hat_params[:n-1]
theta[n-1] = 0.0

psi = hat_params[-1]
phi = math.exp(psi)/(1+math.exp(psi))

expected_freq = expected_tie_frequency(boards_data, theta, phi, T)
print("Expected frequency:", expected_freq)
```

B.5 Theta from the paper

```
# Use the MLE parameters given in the paper (results already known and
   independent from previous parts)
theta = [0.0702, 0.1113, -0.4266, 0.2271, 0.1030, -0.5934, 0.4579, 0.0]
phi = 0.6845

expected_freq = expected_tie_frequency(boards_data, theta, phi, T)
print("Expected frequency:", expected_freq)
```

C Python code for Threshold Model(use data from the Chinese Contract Bridge Association)

C.1 Threshold model for ranking

```
import numpy as np
from scipy.optimize import minimize

ns_patterns = [
    (12, 11, 3,  9,  8,  7),
    (12, 1,  4,  10, 9,  8),
    (12, 2,  5,  11, 10, 9),
    (12, 3,  6,  1,  11, 10),
    (12, 4,  7,  2,  1,  11),
    (12, 5,  8,  3,  2,  1),
    (12, 6,  9,  4,  3,  2),
    (12, 7,  10, 5,  4,  3),
    (12, 8,  11, 6,  5,  4),
    (12, 9,  1,  7,  6,  5),
    (12, 10, 2,  8,  7,  6),
]

ew_patterns = [
    (1,  2,  10, 4,  5,  6),
    (2,  3,  11, 5,  6,  7),
    (3,  4,  1,  6,  7,  8),
    (4,  5,  2,  7,  8,  9),
    (5,  6,  3,  8,  9,  10),
    (6,  7,  4,  9,  10, 11),
    (7,  8,  5,  10, 11, 1),
    (8,  9,  6,  11, 1,  2),
    (9,  10, 7,  1,  2,  3),
    (10, 11, 8,  2,  3,  4),
    (11, 1,  9,  3,  4,  5),
]

alpha_list = []
beta_list  = []
for i in range(len(ns_patterns)):
    a_pat = ns_patterns[i]
    b_pat = ew_patterns[i]
    for _ in range(4):
        alpha_list.append(a_pat)
        beta_list.append(b_pat)

alpha = np.array(alpha_list)
beta  = np.array(beta_list)

B = alpha.shape[0]
T = alpha.shape[1]
n = 12

mp_scores = np.array([
    [4, 0, 8, 6, 8, 10, 0, 2, 4, 2, 10, 6],
    [10,5, 5, 8, 1, 1,  9, 9, 2, 5, 5,  0],
    [2, 2, 0, 7, 7, 2,  8, 3, 3, 10,8,  8],
    [2, 7, 8,10, 2, 7,  3, 8, 0, 2, 3,  8],
    [0, 4,10, 6, 0, 4,  8, 2, 6,10, 4,  6],
    [8,10, 2, 8, 7, 2,  7, 3, 8, 3, 2,  0],
    [4, 1, 6, 9, 6,10,  6, 4, 0, 4, 1,  9],
```

```
    [0, 4,10, 2, 4, 4,  0,10, 6, 6, 8,  6],
    [7, 8, 7, 2, 3, 7,  7, 0,10, 3, 3,  3],
    [10,4, 2, 6, 0, 6,  0, 6, 4,10, 4,  8],
    [4, 2, 8, 8, 6, 0,  2, 8, 2, 8,10,  2],
    [8, 0, 3,10, 2, 0,  6, 3, 7, 4,10,  7],
    [9, 4, 2,10, 8, 6,  1, 6, 1, 9, 4,  0],
    [7, 9, 7, 3, 3, 1,  3, 3, 9, 1, 7,  7],
    [8, 6, 1, 9, 9, 4,  2, 4, 0,10, 6,  1],
    [3, 7, 0, 2,10, 3,  7, 0, 4, 6,10,  8],
    [3, 8, 7,10, 7, 0,  3, 2, 7, 7, 3,  3],
    [3, 8,10, 8, 2, 2,  0, 2, 7, 7, 3,  8],
    [2, 9, 1, 4,10, 6,  9, 1, 8, 4, 6,  0],
    [9, 2, 1, 2, 8, 8,  9, 8, 1, 4, 6,  2],
    [3, 0, 8, 7,10, 4,  0, 3, 2,10, 7,  6],
    [9, 3, 9, 7, 3, 7,  7, 3, 1, 7, 1,  3],
    [9, 1, 9, 6, 1, 4,  9, 4, 1, 9, 1,  6],
    [5, 5, 5, 5, 5, 5,  5, 5, 5, 5, 5,  5],
    [5, 5, 0,10, 5, 5,  5, 5, 5, 0,10,  5],
    [6, 4, 4,10, 6, 4,  6, 6, 4, 0, 6,  4],
    [7, 3,10, 6,10, 3,  2, 7, 0, 4, 0,  8],
    [7, 3, 9, 9, 7, 3,  7, 7, 3, 1, 1,  3],
    [8, 0,10, 2, 7, 3,  2, 8, 8, 7, 3,  2],
    [6,10, 0, 4, 4, 0,  4, 2, 6,10, 6,  8],
    [2, 8, 2, 8, 8, 8,  2, 2, 8, 2, 2,  8],
    [7, 7, 3, 3, 9, 1,  0, 4,10, 9, 1,  6],
    [1, 8, 8, 2, 2, 9,  4, 2, 1, 8, 6,  9],
    [7, 3, 7, 3, 7, 3, 10,10, 3, 0, 0,  7],
    [0, 4, 6, 4, 6,10,  2, 1, 9, 9, 8,  1],
    [4, 8, 4, 6, 2, 6,  4, 0, 0,10, 6, 10],
    [3, 7,10, 0,10, 0,  3, 7, 8, 4, 2,  6],
    [1, 1, 1, 6, 4, 9,  9, 9, 1, 4, 9,  6],
    [0, 1, 1, 4, 6, 9,  9,10, 2, 6, 8,  4],
    [8, 7, 7, 0,10, 3,  3, 2, 3, 7, 7,  3],
    [7, 8, 4, 0, 7, 3, 10, 6, 2, 3,10,  0],
    [9, 5, 5, 9, 2, 8,  1, 5, 5, 1, 0, 10],
    [10,6, 0, 8, 4, 6,  2,10, 4, 0, 4,  6],
    [1, 2, 1, 4,10, 0,  6, 9, 8, 9, 6,  4],
])

assert mp_scores.shape == (44, 12), "mp_scores must be (44,12) to match B=44, n
    =12"


boards_data = []
for b in range(B):
    ns_pairs = alpha[b, :]
    ew_pairs = beta[b, :]
    scores_ns = mp_scores[b, ns_pairs - 1]

    L_b = []
    D_b = []
    for i in range(T):
        for j in range(i + 1, T):
            si = scores_ns[i]
            sj = scores_ns[j]
            if np.isclose(si, sj):
                D_b.append((i, j))
            elif si < sj:
                L_b.append((i, j))
            else:
                L_b.append((j, i))
```

```
116
117     boards_data.append({
118         'alpha': ns_pairs,
119         'beta':  ew_pairs,
120         'L_b':   L_b,
121         'D_b':   D_b,
122     })
123
124
125 def logistic_cdf(x):
126     return 1 / (1 + np.exp(-x))
127
128 def logistic_pdf(x):
129     e = np.exp(x)
130     return e / (1 + e)**2
131
132 def threshold_loglik_grad(params, boards_data, n, T):
133     """
134     params: length n, first (n-1) are theta_i, last is psi=log(gamma)
135     """
136     psi = params[-1]
137     gamma = np.exp(psi)
138     theta = np.zeros(n)
139     theta[:n-1] = params[:-1]
140
141     ll = 0.0
142     grad_theta = np.zeros(n)
143     grad_psi = 0.0
144
145     for bd in boards_data:
146         ns_pairs = bd['alpha']
147         ew_pairs = bd['beta']
148         L_b = bd['L_b']
149         D_b = bd['D_b']
150
151         skill = np.zeros(T)
152         dskill_dtheta = np.zeros((T, n))
153         for t in range(T):
154             i_ns = ns_pairs[t] - 1
155             i_ew = ew_pairs[t] - 1
156             val = theta[i_ns] - theta[i_ew]
157             skill[t] = val
158
159             gtemp = np.zeros(n)
160             gtemp[i_ns] = 1.0
161             gtemp[i_ew] = -1.0
162             dskill_dtheta[t, :] = gtemp
163
164         # Process wins/losses
165         for (i, j) in L_b:
166             d_ij = skill[i] - skill[j]
167             z = -(gamma/2 + d_ij)
168             p_ij = logistic_cdf(z)
169             p_ij = max(p_ij, 1e-15)
170             ll += np.log(p_ij)
171
172             fz = logistic_pdf(z)
173             coeff = (fz / p_ij) * (-1.0)
174             grad_theta += coeff * (dskill_dtheta[i] - dskill_dtheta[j])
175             grad_psi += (fz / p_ij) * (-0.5) * gamma
176
```

```
        # Process ties
        for (i, j) in D_b:
            d_ij = skill[i] - skill[j]
            left  = -(gamma/2 + d_ij)
            right =  (gamma/2) - d_ij
            p_tie = logistic_cdf(right) - logistic_cdf(left)
            p_tie = max(p_tie, 1e-15)
            ll += np.log(p_tie)

            f_left  = logistic_pdf(left)
            f_right = logistic_pdf(right)
            coeff_tie = (-f_right + f_left) / p_tie
            grad_theta += coeff_tie * (dskill_dtheta[i] - dskill_dtheta[j])
            grad_psi += ( (0.5*f_right) + (0.5*f_left) ) / p_tie * gamma

    # assemble gradient
    g = np.zeros_like(params)
    g[:n-1] = grad_theta[:n-1]
    g[-1] = grad_psi
    return -ll, -g


def fit_threshold_model(boards_data, n, T):
    x_init = np.zeros(n)
    x_init[-1] = np.log(1.0)

    res = minimize(
        fun=lambda x: threshold_loglik_grad(x, boards_data, n, T),
        x0=x_init, jac=True, method='BFGS'
    )
    hat_params = res.x
    ll = -res.fun

    psi = hat_params[-1]
    gamma = np.exp(psi)
    hat_theta = np.zeros(n)
    hat_theta[:n-1] = hat_params[:-1]
    return res, hat_theta, gamma, ll


res, hat_theta, hat_gamma, ll_val = fit_threshold_model(boards_data, n, T)

print("Converged:", res.success)
print("Final log-likelihood:", ll_val)
print(f"Estimated gamma = {hat_gamma:.4f}")
for i in range(n):
    print(f"Theta_{i+1} = {hat_theta[i]:.4f}")

theta_indexed = list(enumerate(hat_theta, start=1))
theta_sorted = sorted(theta_indexed, key=lambda x: x[1], reverse=True)
print("\n=== Ranking by skill, descending order ===")
for rank, (pid, val) in enumerate(theta_sorted, start=1):
    print(f"  Rank {rank}: Pair {pid}, theta = {val:.4f}")
```

C.2 Threshold model for tie probability

```
def calc_expected_ties_distribution(boards_data, theta, gamma):

    import numpy as np
```

```
    from itertools import product

    freq_k = np.zeros(16)
    B = len(boards_data)

    for bd in boards_data:
        ns = bd['alpha']
        ew = bd['beta']

        T = len(ns)
        skill = []
        for t in range(T):
            i_ns = ns[t] - 1
            i_ew = ew[t] - 1
            skill.append(theta[i_ns] - theta[i_ew])

        pairs = []
        for i in range(T):
            for j in range(i+1, T):
                pairs.append((i, j))

        p_ties = []
        for (i, j) in pairs:
            d_ij = skill[i] - skill[j]
            left  = -(gamma/2 + d_ij)
            right =  (gamma/2) - d_ij
            p_tie = logistic_cdf(right) - logistic_cdf(left)

            p_tie = np.clip(p_tie, 1e-15, 1-1e-15)
            p_ties.append(p_tie)

        pr_k = np.zeros(16)
        for bits in product([0,1], repeat=15):
            kk = sum(bits)
            prob = 1.0
            for idx, bval in enumerate(bits):
                if bval == 1:
                    prob *= p_ties[idx]
                else:
                    prob *= (1 - p_ties[idx])
            pr_k[kk] += prob

        freq_k += pr_k

    return freq_k


def print_observed_vs_expected_ties(boards_data, theta, gamma):


    # (1) Predicted distribution
    expected_k = calc_expected_ties_distribution(boards_data, theta, gamma)

    # (2) Observed distribution
    import numpy as np
    observed_k = np.zeros(16)
    for bd in boards_data:
        k = len(bd['D_b'])
        observed_k[k] += 1

    print("\n=== Distribution of #ties (Observed vs. Expected) ===")
```

```
    print(" k   Observed   Expected")
    for k in range(16):
        if observed_k[k] == 0 and expected_k[k] < 1e-8:
            continue
        print(f"{k:2d}   {observed_k[k]:3.0f}       {expected_k[k]:7.3f}")

print_observed_vs_expected_ties(boards_data, hat_theta, hat_gamma)
```

D Python code for selecting the best pair

This section provides the Python implementation for computing the pairwise covariances and variances of the estimated skill parameters. These are essential for constructing the confidence subset S (theorem A.1) and for the improved weighted-correction method (theorem A.2).

D.1 Estimating $\hat{W}_{ij}$ via the numerical Hessian

To apply this method, we require estimates of the variance $\hat{W}_{ij} = \mathrm{Var}(\hat{\theta}_i - \hat{\theta}_j)$. These are obtained from the asymptotic covariance matrix of the MLE, denoted $\hat{\Sigma}$, which is approximated using a finite-difference estimate of the Hessian matrix of the log-likelihood function.

The Hessian matrix is defined as:

$$H = \left[\frac{\partial^2 \ell(\theta)}{\partial \theta_i \partial \theta_j} \right]_{i,j=1}^{d},$$

where $\ell(\theta)$ is the negative log-likelihood function. Since analytic expressions are unavailable, we use a nested finite-difference approximation.
First, we approximate the gradient using forward differences:

$$\frac{\partial \ell(\theta)}{\partial \theta_j} \approx \frac{\ell(\theta + h e_j) - \ell(\theta)}{h},$$

where h is a small step size and e_j is the j-th unit vector.
After that, we approximate the second derivatives by differencing partial gradients:

$$\frac{\partial^2 \ell(\theta)}{\partial \theta_i \partial \theta_j} \approx \frac{1}{h} \left[\frac{\partial \ell(\theta + h e_j)}{\partial \theta_i} - \frac{\partial \ell(\theta)}{\partial \theta_i} \right].$$

This is equivalent to computing:

$$H_{ij} \approx \frac{\nabla_i \ell(\theta + h e_j) - \nabla_i \ell(\theta)}{h},$$

where ∇_i denotes the i-th component of the gradient.
Once H is computed, the asymptotic covariance matrix is approximated as:

$$\hat{\Sigma} = (-H)^{-1},$$

and the pairwise variance of differences is calculated as:

$$\hat{W}_{ij} = \hat{\Sigma}_{ii} + \hat{\Sigma}_{jj} - 2\hat{\Sigma}_{ij}.$$

This method avoids symbolic derivation, is computationally efficient, and respects the correlation structure among parameters. It provides theoretically grounded and practically tractable estimates of uncertainty in pairwise comparisons.

D.2 Calculation of covariance and variance

```python
def compute_hessian(x, boards_data, n, T, epsilon=1e-6):
    def func(xx):
        return negative_log_likelihood(xx, boards_data, n, T)
    d = len(x)
    hessian = np.zeros((d, d))

    for i in range(d):
        def grad_i(xx):
        #Forward Difference
            g = approx_fprime(xx, func, epsilon)
```

```
            return g[i]
        hessian[i, :] = approx_fprime(x, grad_i, epsilon)
    return hessian

final_x = res.x
hessian_matrix = compute_hessian(final_x, boards_data, n, T)
cov_matrix = inv(hessian_matrix)

def extract_pairwise_covariances(cov_matrix, n):
    pairwise_cov = []
    for i in range(n):
        for j in range(i + 1, n):
            pairwise_cov.append((i + 1, j + 1, cov_matrix[i, j]))
    return pairwise_cov

pairwise_covariances = extract_pairwise_covariances(cov_matrix, n)
print("\nPairwise Covariances:")
for i, j, cov in pairwise_covariances:
    print(f"Cov(theta_{i}, theta_{j}) = {cov:.6f}")

def compute_pairwise_variances(cov_matrix, n):
    pairwise_variances = []
    for i in range(n):
        for j in range(i + 1, n):
            var_i = cov_matrix[i, i]
            var_j = cov_matrix[j, j]
            cov_ij = cov_matrix[i, j]
            w_ij = var_i + var_j - 2 * cov_ij
            pairwise_variances.append((i + 1, j + 1, w_ij))
    return pairwise_variances

pairwise_variances = compute_pairwise_variances(cov_matrix, n)
print("\nPairwise Variances (W_ij):")
for i, j, var_ in pairwise_variances:
    print(f"Var(theta_{i} - theta_{j}) = {var_:.6f}")
```

```
Optimization terminated successfully.
         Current function value: 143.693841
         Iterations: 14
         Function evaluations: 21
         Gradient evaluations: 21
Optimization success: True
Final negative log-likelihood: 143.6938411916855
Estimated phi: 0.2937590836269098
θ_1 = 0.1336
θ_2 = 0.1899
θ_3 = -0.7132
θ_4 = 0.3866
θ_5 = 0.1838
θ_6 = -0.9896
θ_7 = 0.7906
θ_8 = 0.0000

Ranking of θ by skill level (from highest to lowest):
Rank 1: θ_7 = 0.7906
Rank 2: θ_4 = 0.3866
Rank 3: θ_2 = 0.1899
Rank 4: θ_5 = 0.1838
Rank 5: θ_1 = 0.1336
Rank 6: θ_8 = 0.0000
Rank 7: θ_3 = -0.7132
Rank 8: θ_6 = -0.9896

Pairwise Covariances:
Cov(θ_1, θ_2) = 0.075997
Cov(θ_1, θ_3) = 0.072661
Cov(θ_1, θ_4) = 0.068692
Cov(θ_1, θ_5) = 0.070781
Cov(θ_1, θ_6) = 0.072427
Cov(θ_1, θ_7) = 0.069229
Cov(θ_1, θ_8) = -0.001194
Cov(θ_2, θ_3) = 0.074599
Cov(θ_2, θ_4) = 0.071533
Cov(θ_2, θ_5) = 0.072435
Cov(θ_2, θ_6) = 0.074296
Cov(θ_2, θ_7) = 0.072397
Cov(θ_2, θ_8) = -0.000095
Cov(θ_3, θ_4) = 0.069002
Cov(θ_3, θ_5) = 0.069321
Cov(θ_3, θ_6) = 0.082684
Cov(θ_3, θ_7) = 0.064137
Cov(θ_3, θ_8) = -0.010603
Cov(θ_4, θ_5) = 0.069249
Cov(θ_4, θ_6) = 0.068005
Cov(θ_4, θ_7) = 0.069926
Cov(θ_4, θ_8) = 0.001434
Cov(θ_5, θ_6) = 0.069561
Cov(θ_5, θ_7) = 0.070575
Cov(θ_5, θ_8) = -0.000820
Cov(θ_6, θ_7) = 0.062425
Cov(θ_6, θ_8) = -0.013440
Cov(θ_7, θ_8) = 0.006361

Pairwise Variances (W_ij):
Var(θ_1 - θ_2) = 0.111360
Var(θ_1 - θ_3) = 0.120127
Var(θ_1 - θ_4) = 0.118973
Var(θ_1 - θ_5) = 0.115320
Var(θ_1 - θ_6) = 0.124971
Var(θ_1 - θ_7) = 0.119153
```

```
Var(θ_1 - θ_8) = 0.254223
Var(θ_2 - θ_3) = 0.120986
Var(θ_2 - θ_4) = 0.118026
Var(θ_2 - θ_5) = 0.116747
Var(θ_2 - θ_6) = 0.125968
Var(θ_2 - θ_7) = 0.117553
Var(θ_2 - θ_8) = 0.256762
Var(θ_3 - θ_4) = 0.125184
Var(θ_3 - θ_5) = 0.125071
Var(θ_3 - θ_6) = 0.111287
Var(θ_3 - θ_7) = 0.136167
Var(θ_3 - θ_8) = 0.279872
Var(θ_4 - θ_5) = 0.116124
Var(θ_4 - θ_6) = 0.131555
Var(θ_4 - θ_7) = 0.115497
Var(θ_4 - θ_8) = 0.246705
Var(θ_5 - θ_6) = 0.128968
Var(θ_5 - θ_7) = 0.114726
Var(θ_5 - θ_8) = 0.251740
Var(θ_6 - θ_7) = 0.143969
Var(θ_6 - θ_8) = 0.289923
Var(θ_7 - θ_8) = 0.238106
```

In [ ]:

D.3 For Theorem 2 from Yu and Lam

```
\begin{lstlisting}[language=Python]
# W-matrix
W = np.zeros((n, n))
for i in range(n):
    for j in range(n):
        W[i, j] = cov_matrix[i, i] + cov_matrix[j, j] - 2 * cov_matrix[i, j]

alpha_sig = 0.05  # significance level
z_val = norm.ppf(1 - alpha_sig / (n - 1))
print(f"\nSignificance level alpha={alpha_sig}, multi-comparison z={z_val:.4f}\n
    ")

S = []
for i in range(n):
    LBs = []
    print(f"** Checking candidate i = {i+1} (theta_{i+1} = {hat_theta[i]:.4f})
    **")
    for j in range(n):
        if j == i:
            continue
        LB_j = hat_theta[j] - z_val * np.sqrt(W[j, i])
        LBs.append(LB_j)
        relation = ">=" if hat_theta[i] >= LB_j else "<"
        print(f"   - Compare vs j={j+1}: LB_j={LB_j:.4f}, so theta_{i+1}({
    hat_theta[i]:.4f}) {relation} LB_j")
    max_LB = max(LBs)
    print(f"   => max_LB = {max_LB:.4f}. Compare with theta_{i+1}={hat_theta[i
    ]:.4f}")
    if hat_theta[i] >= max_LB:
        S.append(i+1)
        print(f"   => i={i+1} passes, included in S.\n")
    else:
        print(f"   => i={i+1} fails.\n")

print("Final subset S =", S)
print(f"Interpretation: S contains the truly best pair with probability >= {1 -
    alpha_sig:.2f}.")
```

Significance level α=0.05, multi-comparison z=2.4500

** Checking candidate i = 1 (θ_1 = 0.1336) **
- Compare vs j=2: LB_j=-0.6277, so θ_1(0.1336) >= LB_j
- Compare vs j=3: LB_j=-1.5623, so θ_1(0.1336) >= LB_j
- Compare vs j=4: LB_j=-0.4584, so θ_1(0.1336) >= LB_j
- Compare vs j=5: LB_j=-0.6481, so θ_1(0.1336) >= LB_j
- Compare vs j=6: LB_j=-1.8557, so θ_1(0.1336) >= LB_j
- Compare vs j=7: LB_j=-0.0551, so θ_1(0.1336) >= LB_j
- Compare vs j=8: LB_j=-1.2353, so θ_1(0.1336) >= LB_j
=> max_LB = -0.0551. Compare with θ_1=0.1336
=> i=1 passes, included in S.

** Checking candidate i = 2 (θ_2 = 0.1899) **
- Compare vs j=1: LB_j=-0.6840, so θ_2(0.1899) >= LB_j
- Compare vs j=3: LB_j=-1.5653, so θ_2(0.1899) >= LB_j
- Compare vs j=4: LB_j=-0.4551, so θ_2(0.1899) >= LB_j
- Compare vs j=5: LB_j=-0.6533, so θ_2(0.1899) >= LB_j
- Compare vs j=6: LB_j=-1.8592, so θ_2(0.1899) >= LB_j
- Compare vs j=7: LB_j=-0.0494, so θ_2(0.1899) >= LB_j
- Compare vs j=8: LB_j=-1.2415, so θ_2(0.1899) >= LB_j
=> max_LB = -0.0494. Compare with θ_2=0.1899
=> i=2 passes, included in S.

** Checking candidate i = 3 (θ_3 = -0.7132) **
- Compare vs j=1: LB_j=-0.7156, so θ_3(-0.7132) >= LB_j
- Compare vs j=2: LB_j=-0.6623, so θ_3(-0.7132) < LB_j
- Compare vs j=4: LB_j=-0.4802, so θ_3(-0.7132) < LB_j
- Compare vs j=5: LB_j=-0.6826, so θ_3(-0.7132) < LB_j
- Compare vs j=6: LB_j=-1.8069, so θ_3(-0.7132) >= LB_j
- Compare vs j=7: LB_j=-0.1135, so θ_3(-0.7132) < LB_j
- Compare vs j=8: LB_j=-1.2961, so θ_3(-0.7132) >= LB_j
=> max_LB = -0.1135. Compare with θ_3=-0.7132
=> i=3 fails.

** Checking candidate i = 4 (θ_4 = 0.3866) **
- Compare vs j=1: LB_j=-0.7115, so θ_4(0.3866) >= LB_j
- Compare vs j=2: LB_j=-0.6518, so θ_4(0.3866) >= LB_j
- Compare vs j=3: LB_j=-1.5800, so θ_4(0.3866) >= LB_j
- Compare vs j=5: LB_j=-0.6510, so θ_4(0.3866) >= LB_j
- Compare vs j=6: LB_j=-1.8782, so θ_4(0.3866) >= LB_j
- Compare vs j=7: LB_j=-0.0420, so θ_4(0.3866) >= LB_j
- Compare vs j=8: LB_j=-1.2169, so θ_4(0.3866) >= LB_j
=> max_LB = -0.0420. Compare with θ_4=0.3866
=> i=4 passes, included in S.

** Checking candidate i = 5 (θ_5 = 0.1838) **
- Compare vs j=1: LB_j=-0.6984, so θ_5(0.1838) >= LB_j
- Compare vs j=2: LB_j=-0.6472, so θ_5(0.1838) >= LB_j
- Compare vs j=3: LB_j=-1.5796, so θ_5(0.1838) >= LB_j
- Compare vs j=4: LB_j=-0.4482, so θ_5(0.1838) >= LB_j
- Compare vs j=6: LB_j=-1.8694, so θ_5(0.1838) >= LB_j
- Compare vs j=7: LB_j=-0.0392, so θ_5(0.1838) >= LB_j
- Compare vs j=8: LB_j=-1.2293, so θ_5(0.1838) >= LB_j
=> max_LB = -0.0392. Compare with θ_5=0.1838
=> i=5 passes, included in S.

** Checking candidate i = 6 (θ_6 = -0.9896) **
- Compare vs j=1: LB_j=-0.7325, so θ_6(-0.9896) < LB_j
- Compare vs j=2: LB_j=-0.6797, so θ_6(-0.9896) < LB_j
- Compare vs j=3: LB_j=-1.5305, so θ_6(-0.9896) >= LB_j
- Compare vs j=4: LB_j=-0.5020, so θ_6(-0.9896) < LB_j
- Compare vs j=5: LB_j=-0.6960, so θ_6(-0.9896) < LB_j
- Compare vs j=7: LB_j=-0.1390, so θ_6(-0.9896) < LB_j

```
   - Compare vs j=8: LB_j=-1.3192, so θ_6(-0.9896) >= LB_j
   => max_LB = -0.1390. Compare with θ_6=-0.9896
   => i=6 fails.

** Checking candidate i = 7 (θ_7 = 0.7906) **
   - Compare vs j=1: LB_j=-0.7121, so θ_7(0.7906) >= LB_j
   - Compare vs j=2: LB_j=-0.6501, so θ_7(0.7906) >= LB_j
   - Compare vs j=3: LB_j=-1.6172, so θ_7(0.7906) >= LB_j
   - Compare vs j=4: LB_j=-0.4460, so θ_7(0.7906) >= LB_j
   - Compare vs j=5: LB_j=-0.6460, so θ_7(0.7906) >= LB_j
   - Compare vs j=6: LB_j=-1.9192, so θ_7(0.7906) >= LB_j
   - Compare vs j=8: LB_j=-1.1955, so θ_7(0.7906) >= LB_j
   => max_LB = -0.4460. Compare with θ_7=0.7906
   => i=7 passes, included in S.

** Checking candidate i = 8 (θ_8 = 0.0000) **
   - Compare vs j=1: LB_j=-1.1017, so θ_8(0.0000) >= LB_j
   - Compare vs j=2: LB_j=-1.0516, so θ_8(0.0000) >= LB_j
   - Compare vs j=3: LB_j=-2.0093, so θ_8(0.0000) >= LB_j
   - Compare vs j=4: LB_j=-0.8303, so θ_8(0.0000) >= LB_j
   - Compare vs j=5: LB_j=-1.0454, so θ_8(0.0000) >= LB_j
   - Compare vs j=6: LB_j=-2.3088, so θ_8(0.0000) >= LB_j
   - Compare vs j=7: LB_j=-0.4049, so θ_8(0.0000) >= LB_j
   => max_LB = -0.4049. Compare with θ_8=0.0000
   => i=8 passes, included in S.

Final subset S = [1, 2, 4, 5, 7, 8]
Interpretation: S contains the truly best pair with probability ⩾ 0.95.
```

In [ ]:

D.4 For weighted-correction selection method

```
def compute_hessian(x, boards_data, n, T, epsilon=1e-6):
    def f(xx):
        val, _ = negative_log_likelihood_and_grad(xx, boards_data, n, T)
        return val
    d = len(x)
    hess = np.zeros((d, d))
    for i in range(d):
        def grad_i(xx):
            g = approx_fprime(xx, f, epsilon)
            return g[i]
        hess[i, :] = approx_fprime(x, grad_i, epsilon)
    return hess

hessian = compute_hessian(res.x, boards_data, n, T)
cov_matrix = inv(hessian)

W = np.zeros((n, n))
for i in range(n):
    for j in range(n):
        W[i, j] = cov_matrix[i, i] + cov_matrix[j, j] - 2 * cov_matrix[i, j]

print("\n--- Constructing subset S with Weighted Bound ---")

alpha = 0.05
z_val = norm.ppf(1 - alpha / (n - 1))
print(f"Significance level alpha={alpha}, z={z_val:.4f}\n")

omega = [1 / (1 + abs(hat_theta[j])) for j in range(n)]

S = []
for i in range(n):
    print(f"** Checking candidate i={i+1}, theta_{i+1}={hat_theta[i]:.4f} **")
    LB_values = []
    for j in range(n):
        if j == i:
            continue
        LB_j = hat_theta[j] - omega[j] * z_val * np.sqrt(W[j, i])
        LB_values.append(LB_j)
        relation = ">=" if hat_theta[i] >= LB_j else "<"
        print(f"  Compare vs j={j+1}: LB_j={LB_j:.4f}, => theta_{i+1}({hat_theta
    [i]:.4f}) {relation} LB_j({LB_j:.4f})")

    max_LB = max(LB_values)
    print(f"  => max_LB among j != i is {max_LB:.4f}, theta_{i+1}={hat_theta[i
    ]:.4f}")

    if hat_theta[i] >= max_LB:
        S.append(i + 1)
        print(f"  => i={i+1} PASSES. Included in S.\n")
    else:
        print(f"  => i={i+1} FAILS.\n")

print("Final subset S with weight factors =", S)
print(f"Interpretation: with probability > {1 - alpha:.2f}, the true best is in
    S.")
```

```
--- Constructing subset S with Weighted Bound ---
Significance level alpha=0.05, z=2.4500

** Checking candidate i=1, theta_1=0.1336 **
  Compare vs j=2: LB_j=-0.4972, => theta_1(0.1336) >= LB_j(-0.4972)
  Compare vs j=3: LB_j=-1.2088, => theta_1(0.1336) >= LB_j(-1.2088)
  Compare vs j=4: LB_j=-0.2228, => theta_1(0.1336) >= LB_j(-0.2228)
  Compare vs j=5: LB_j=-0.5189, => theta_1(0.1336) >= LB_j(-0.5189)
  Compare vs j=6: LB_j=-1.4249, => theta_1(0.1336) >= LB_j(-1.4249)
  Compare vs j=7: LB_j=0.3183, => theta_1(0.1336) <  LB_j(0.3183)
  Compare vs j=8: LB_j=-1.2353, => theta_1(0.1336) >= LB_j(-1.2353)
  => max_LB among j != i is 0.3183, theta_1=0.1336
  => i=1 FAILS.

** Checking candidate i=2, theta_2=0.1899 **
  Compare vs j=1: LB_j=-0.5877, => theta_2(0.1899) >= LB_j(-0.5877)
  Compare vs j=3: LB_j=-1.2106, => theta_2(0.1899) >= LB_j(-1.2106)
  Compare vs j=4: LB_j=-0.2204, => theta_2(0.1899) >= LB_j(-0.2204)
  Compare vs j=5: LB_j=-0.5233, => theta_2(0.1899) >= LB_j(-0.5233)
  Compare vs j=6: LB_j=-1.4267, => theta_2(0.1899) >= LB_j(-1.4267)
  Compare vs j=7: LB_j=0.3215, => theta_2(0.1899) <  LB_j(0.3215)
  Compare vs j=8: LB_j=-1.2415, => theta_2(0.1899) >= LB_j(-1.2415)
  => max_LB among j != i is 0.3215, theta_2=0.1899
  => i=2 FAILS.

** Checking candidate i=3, theta_3=-0.7132 **
  Compare vs j=1: LB_j=-0.6155, => theta_3(-0.7132) <  LB_j(-0.6155)
  Compare vs j=2: LB_j=-0.5263, => theta_3(-0.7132) <  LB_j(-0.5263)
  Compare vs j=4: LB_j=-0.2385, => theta_3(-0.7132) <  LB_j(-0.2385)
  Compare vs j=5: LB_j=-0.5481, => theta_3(-0.7132) <  LB_j(-0.5481)
  Compare vs j=6: LB_j=-1.4004, => theta_3(-0.7132) >= LB_j(-1.4004)
  Compare vs j=7: LB_j=0.2857, => theta_3(-0.7132) <  LB_j(0.2857)
  Compare vs j=8: LB_j=-1.2961, => theta_3(-0.7132) >= LB_j(-1.2961)
  => max_LB among j != i is 0.2857, theta_3=-0.7132
  => i=3 FAILS.

** Checking candidate i=4, theta_4=0.3866 **
  Compare vs j=1: LB_j=-0.6119, => theta_4(0.3866) >= LB_j(-0.6119)
  Compare vs j=2: LB_j=-0.5175, => theta_4(0.3866) >= LB_j(-0.5175)
  Compare vs j=3: LB_j=-1.2191, => theta_4(0.3866) >= LB_j(-1.2191)
  Compare vs j=5: LB_j=-0.5214, => theta_4(0.3866) >= LB_j(-0.5214)
  Compare vs j=6: LB_j=-1.4362, => theta_4(0.3866) >= LB_j(-1.4362)
  Compare vs j=7: LB_j=0.3256, => theta_4(0.3866) >= LB_j(0.3256)
  Compare vs j=8: LB_j=-1.2169, => theta_4(0.3866) >= LB_j(-1.2169)
  => max_LB among j != i is 0.3256, theta_4=0.3866
  => i=4 PASSES. Included in S.

** Checking candidate i=5, theta_5=0.1838 **
  Compare vs j=1: LB_j=-0.6004, => theta_5(0.1838) >= LB_j(-0.6004)
  Compare vs j=2: LB_j=-0.5137, => theta_5(0.1838) >= LB_j(-0.5137)
  Compare vs j=3: LB_j=-1.2189, => theta_5(0.1838) >= LB_j(-1.2189)
  Compare vs j=4: LB_j=-0.2154, => theta_5(0.1838) >= LB_j(-0.2154)
  Compare vs j=6: LB_j=-1.4318, => theta_5(0.1838) >= LB_j(-1.4318)
  Compare vs j=7: LB_j=0.3272, => theta_5(0.1838) <  LB_j(0.3272)
  Compare vs j=8: LB_j=-1.2293, => theta_5(0.1838) >= LB_j(-1.2293)
  => max_LB among j != i is 0.3272, theta_5=0.1838
  => i=5 FAILS.

** Checking candidate i=6, theta_6=-0.9896 **
  Compare vs j=1: LB_j=-0.6305, => theta_6(-0.9896) <  LB_j(-0.6305)
  Compare vs j=2: LB_j=-0.5409, => theta_6(-0.9896) <  LB_j(-0.5409)
  Compare vs j=3: LB_j=-1.1902, => theta_6(-0.9896) >= LB_j(-1.1902)
  Compare vs j=4: LB_j=-0.2542, => theta_6(-0.9896) <  LB_j(-0.2542)
  Compare vs j=5: LB_j=-0.5594, => theta_6(-0.9896) <  LB_j(-0.5594)
```

```
  Compare vs j=7: LB_j=0.2715, => theta_6(-0.9896) <  LB_j(0.2715)
  Compare vs j=8: LB_j=-1.3192, => theta_6(-0.9896) >= LB_j(-1.3192)
  => max_LB among j != i is 0.2715, theta_6=-0.9896
  => i=6 FAILS.

** Checking candidate i=7, theta_7=0.7906 **
  Compare vs j=1: LB_j=-0.6125, => theta_7(0.7906) >= LB_j(-0.6125)
  Compare vs j=2: LB_j=-0.5161, => theta_7(0.7906) >= LB_j(-0.5161)
  Compare vs j=3: LB_j=-1.2409, => theta_7(0.7906) >= LB_j(-1.2409)
  Compare vs j=4: LB_j=-0.2138, => theta_7(0.7906) >= LB_j(-0.2138)
  Compare vs j=5: LB_j=-0.5171, => theta_7(0.7906) >= LB_j(-0.5171)
  Compare vs j=6: LB_j=-1.4568, => theta_7(0.7906) >= LB_j(-1.4568)
  Compare vs j=8: LB_j=-1.1955, => theta_7(0.7906) >= LB_j(-1.1955)
  => max_LB among j != i is -0.2138, theta_7=0.7906
  => i=7 PASSES. Included in S.

** Checking candidate i=8, theta_8=0.0000 **
  Compare vs j=1: LB_j=-0.9562, => theta_8(0.0000) >= LB_j(-0.9562)
  Compare vs j=2: LB_j=-0.8535, => theta_8(0.0000) >= LB_j(-0.8535)
  Compare vs j=3: LB_j=-1.4697, => theta_8(0.0000) >= LB_j(-1.4697)
  Compare vs j=4: LB_j=-0.4909, => theta_8(0.0000) >= LB_j(-0.4909)
  Compare vs j=5: LB_j=-0.8545, => theta_8(0.0000) >= LB_j(-0.8545)
  Compare vs j=6: LB_j=-1.6526, => theta_8(0.0000) >= LB_j(-1.6526)
  Compare vs j=7: LB_j=0.1230, => theta_8(0.0000) <  LB_j(0.1230)
  => max_LB among j != i is 0.1230, theta_8=0.0000
  => i=8 FAILS.

Final subset S with weight factors = [4, 7]
Interpretation: with probability > 0.95, the true best is in S.
```

In [ ]:

E Python code for Fuzzy Logic by Yu-Lam data)

```python
def classify_percentage(p):
    if p > 65:
        return 'A'
    elif p > 55:
        return 'B'
    elif p > 48:
        return 'C'
    elif p >= 40:
        return 'D'
    else:
        return 'F'

cat_list = ['A', 'B', 'C', 'D', 'F']
countABCDF = np.zeros((num_pairs, 5), dtype=int)

for i in range(num_boards):
    for j in range(num_pairs):
        mp_val = mp_scores[i, j]
        p_ij = mp_val / 10.0 * 100.0
        cat = classify_percentage(p_ij)
        idx = cat_list.index(cat)
        countABCDF[j, idx] += 1

m_fuzzy = countABCDF / float(num_boards)

def calc_xc_yc_for_one(mA, mB, mC, mD, mF):
    x_c = 0.5 * (1*mF + 3*mD + 5*mC + 7*mB + 9*mA)
    y_c = 0.5 * (mF**2 + mD**2 + mC**2 + mB**2 + mA**2)
    return x_c, y_c

xc_yc_array = np.zeros((num_pairs, 2))

for j in range(num_pairs):
    mA = m_fuzzy[j, 0]
    mB = m_fuzzy[j, 1]
    mC = m_fuzzy[j, 2]
    mD = m_fuzzy[j, 3]
    mF = m_fuzzy[j, 4]
    x_c, y_c = calc_xc_yc_for_one(mA, mB, mC, mD, mF)
    xc_yc_array[j, 0] = x_c
    xc_yc_array[j, 1] = y_c

print("=== Fuzzy Classification Results Per Board ===")
for j in range(num_pairs):
    print(f"Pair {j+1:2d}: "
          f"m(A)={m_fuzzy[j,0]:.3f}, m(B)={m_fuzzy[j,1]:.3f}, "
          f"m(C)={m_fuzzy[j,2]:.3f}, m(D)={m_fuzzy[j,3]:.3f}, m(F)={m_fuzzy[j,4]:.3f}  "
          f" => x_c={xc_yc_array[j,0]:.3f}, y_c={xc_yc_array[j,1]:.3f}")

ranking = sorted(range(num_pairs), key=lambda idx: xc_yc_array[idx, 0], reverse=True)
print("\n=== Ranking Based on x_c (Descending Order) ===")
for rank_i, pair_idx in enumerate(ranking, start=1):
    x_c = xc_yc_array[pair_idx, 0]
    y_c = xc_yc_array[pair_idx, 1]
    print(f"Rank {rank_i}: Pair {pair_idx+1}, x_c={x_c:.3f}, y_c={y_c:.3f}")
```

F Data from the Chinese Contract Bridge Association

F.1 Total Scores by Round and Board

This table contains data on total scores from the 2024 National Bridge Championships Open Pairs Final [16].

Round	Board	Table	TP-NS	TP-EW	Pair (NS)	Pair (EW)
1	1	B2	990	-990	11#	2#
1	1	B3	920	-920	3#	10#
1	1	B1	480	-480	12#	1#
1	1	B4	450	-450	9#	4#
1	1	B5	300	-300	8#	5#
1	1	B6	-50	50	7#	6#
1	2	B5	-140	140	8#	5#
1	2	B6	-140	140	7#	6#
1	2	B2	-170	170	11#	2#
1	2	B3	-170	170	3#	10#
1	2	B4	-180	180	9#	4#
1	2	B1	-420	420	12#	1#
1	3	B1	420	-420	12#	1#
1	3	B2	420	-420	11#	2#
1	3	B6	420	-420	7#	6#
1	3	B4	170	-170	9#	4#
1	3	B5	170	-170	8#	5#
1	3	B3	140	-140	3#	10#
1	4	B1	660	-660	12#	1#
1	4	B3	660	-660	3#	10#
1	4	B5	660	-660	8#	5#
1	4	B2	630	-630	11#	2#
1	4	B6	630	-630	7#	6#
1	4	B4	-100	100	9#	4#
2	5	B4	170	-170	10#	5#
2	5	B1	100	-100	12#	2#
2	5	B5	100	-100	9#	6#
2	5	B3	100	-100	4#	11#
2	5	B6	50	-50	8#	7#
2	5	B2	-130	130	1#	3#
2	6	B2	490	-490	1#	3#
2	6	B3	490	-490	4#	11#
2	6	B5	490	-490	9#	6#
2	6	B4	460	-460	10#	5#
2	6	B6	460	-460	8#	7#
2	6	B1	400	-400	12#	2#
2	7	B1	1430	-1430	12#	2#
2	7	B3	1430	-1430	4#	11#
2	7	B2	680	-680	1#	3#
2	7	B4	680	-680	10#	5#
2	7	B6	680	-680	8#	7#
2	7	B5	650	-650	9#	6#
2	8	B6	150	-150	8#	7#
2	8	B1	140	-140	12#	2#

Round	Board	Table	TP-NS	TP-EW	Pair (NS)	Pair (EW)
2	8	B4	140	-140	10#	5#
2	8	B5	140	-140	9#	6#
2	8	B3	130	-130	4#	11#
2	8	B2	-50	50	1#	3#
3	9	B6	180	-180	9#	8#
3	9	B2	170	-170	2#	4#
3	9	B3	120	-120	5#	1#
3	9	B5	120	-120	10#	7#
3	9	B1	120	-120	12#	3#
3	9	B4	120	-120	11#	6#
3	10	B5	300	-300	10#	7#
3	10	B1	200	-200	12#	3#
3	10	B2	100	-100	2#	4#
3	10	B4	100	-100	11#	6#
3	10	B6	100	-100	9#	8#
3	10	B3	-620	620	5#	1#
3	11	B4	100	-100	11#	6#
3	11	B5	-50	50	10#	7#
3	11	B3	-100	100	5#	1#
3	11	B1	-140	140	12#	3#
3	11	B2	-140	140	2#	4#
3	11	B6	-140	140	9#	8#
3	12	B4	1430	-1430	11#	6#
3	12	B1	710	-710	12#	3#
3	12	B6	710	-710	9#	8#
3	12	B5	680	-680	10#	7#
3	12	B3	650	-650	5#	1#
3	12	B2	0	0	2#	4#
4	13	B4	200	-200	1#	7#
4	13	B6	200	-200	10#	9#
4	13	B3	100	-100	6#	2#
4	13	B5	-90	90	11#	8#
4	13	B2	-120	120	3#	5#
4	13	B1	-200	200	12#	4#
4	14	B1	430	-430	12#	4#
4	14	B2	430	-430	3#	5#
4	14	B4	430	-430	1#	7#
4	14	B5	430	-430	11#	8#
4	14	B3	150	-150	6#	2#
4	14	B6	150	-150	10#	9#
4	15	B6	200	-200	10#	9#
4	15	B4	170	-170	1#	7#
4	15	B5	140	-140	11#	8#
4	15	B3	110	-110	6#	2#
4	15	B1	100	-100	12#	4#
4	15	B2	100	-100	3#	5#
4	16	B5	500	-500	11#	8#
4	16	B1	400	-400	12#	4#
4	16	B6	-50	50	10#	9#
4	16	B3	-100	100	6#	2#
4	16	B4	-100	100	1#	7#

Round	Board	Table	TP-NS	TP-EW	Pair (NS)	Pair (EW)
4	16	B2	-150	150	3#	5#
5	17	B2	-510	510	4#	6#
5	17	B4	-1010	1010	2#	8#
5	17	B1	-1510	1510	12#	5#
5	17	B3	-1510	1510	7#	3#
5	17	B5	-1510	1510	1#	9#
5	17	B6	-1510	1510	11#	10#
5	18	B1	-450	450	12#	5#
5	18	B2	-450	450	4#	6#
5	18	B4	-450	450	2#	8#
5	18	B5	-480	480	1#	9#
5	18	B6	-480	480	11#	10#
5	18	B3	-980	980	7#	3#
5	19	B3	170	-170	7#	3#
5	19	B4	170	-170	2#	8#
5	19	B6	140	-140	11#	10#
5	19	B2	-50	50	4#	6#
5	19	B5	-110	110	1#	9#
5	19	B1	-670	670	12#	5#
5	20	B3	620	-620	7#	3#
5	20	B5	620	-620	1#	9#
5	20	B6	140	-140	11#	10#
5	20	B1	-100	100	12#	5#
5	20	B2	-100	100	4#	6#
5	20	B4	-100	100	2#	8#
6	21	B2	150	-150	5#	7#
6	21	B4	110	-110	3#	9#
6	21	B1	100	-100	12#	6#
6	21	B3	50	-50	8#	4#
6	21	B6	50	-50	1#	11#
6	21	B5	-100	100	2#	10#
6	22	B4	-630	630	3#	9#
6	22	B6	-630	630	1#	11#
6	22	B1	-660	660	12#	6#
6	22	B2	-660	660	5#	7#
6	22	B5	-660	660	2#	10#
6	22	B3	-660	660	8#	4#
6	23	B4	660	-660	3#	9#
6	23	B6	660	-660	1#	11#
6	23	B1	630	-630	12#	6#
6	23	B3	600	-600	8#	4#
6	23	B2	-100	100	5#	7#
6	23	B5	-100	100	2#	10#
6	24	B1	50	-50	12#	6#
6	24	B2	50	-50	5#	7#
6	24	B3	50	-50	8#	4#
6	24	B4	50	-50	3#	9#
6	24	B5	50	-50	2#	10#
6	24	B6	50	-50	1#	11#
7	25	B4	-100	100	4#	10#
7	25	B6	-130	130	2#	1#

Round	Board	Table	TP-NS	TP-EW	Pair (NS)	Pair (EW)
7	25	B1	-130	130	12#	7#
7	25	B2	-130	130	6#	8#
7	25	B3	-130	130	9#	5#
7	25	B5	800	-800	3#	11#
7	26	B4	-150	150	4#	10#
7	26	B1	-620	620	12#	7#
7	26	B2	-620	620	6#	8#
7	26	B3	-620	620	9#	5#
7	26	B5	-620	620	3#	11#
7	26	B6	-620	620	2#	1#
7	27	B5	790	-790	3#	11#
7	27	B1	690	-690	12#	7#
7	27	B4	480	-480	4#	10#
7	27	B2	450	-450	6#	8#
7	27	B6	450	-450	2#	1#
7	27	B3	-50	50	9#	5#
7	28	B4	50	-50	4#	10#
7	28	B5	50	-50	3#	11#
7	28	B1	-100	100	12#	7#
7	28	B2	-100	100	6#	8#
7	28	B3	-100	100	9#	5#
7	28	B6	-100	100	2#	1#
8	29	B6	100	-100	3#	2#
8	29	B3	-690	690	10#	6#
8	29	B4	-690	690	5#	11#
8	29	B1	-720	720	12#	8#
8	29	B2	-720	720	7#	9#
8	29	B5	-720	720	4#	1#
8	30	B3	550	-550	10#	6#
8	30	B1	450	-450	12#	8#
8	30	B2	420	-420	7#	9#
8	30	B4	420	-420	5#	11#
8	30	B5	420	-420	4#	1#
8	30	B6	400	-400	3#	2#
8	31	B1	-450	450	12#	8#
8	31	B4	-450	450	5#	11#
8	31	B5	-450	450	4#	1#
8	31	B2	-480	480	7#	9#
8	31	B3	-480	480	10#	6#
8	31	B6	-480	480	3#	2#
8	32	B4	200	-200	5#	11#
8	32	B3	200	-200	10#	6#
8	32	B1	170	-170	12#	8#
8	32	B5	110	-110	4#	1#
8	32	B6	110	-110	3#	2#
8	32	B2	-150	150	7#	9#
9	33	B1	50	-50	12#	9#
9	33	B4	50	-50	6#	1#
9	33	B3	-430	430	11#	7#
9	33	B2	-460	460	8#	10#
9	33	B5	-460	460	5#	2#

Round	Board	Table	TP-NS	TP-EW	Pair (NS)	Pair (EW)
9	33	B6	-460	460	4#	3#
9	34	B2	750	-750	8#	10#
9	34	B1	660	-660	12#	9#
9	34	B5	660	-660	5#	2#
9	34	B4	630	-630	6#	1#
9	34	B6	630	-630	4#	3#
9	34	B3	-100	100	11#	7#
9	35	B4	400	-400	6#	1#
9	35	B3	200	-200	11#	7#
9	35	B5	150	-150	5#	2#
9	35	B6	-50	50	4#	3#
9	35	B2	-100	100	8#	10#
9	35	B1	-100	100	12#	9#
9	36	B1	870	-870	12#	9#
9	36	B3	140	-140	11#	7#
9	36	B4	140	-140	6#	1#
9	36	B6	140	-140	4#	3#
9	36	B5	-100	100	5#	2#
9	36	B2	-300	300	8#	10#
10	37	B6	150	-150	5#	4#
10	37	B2	120	-120	9#	11#
10	37	B1	-100	100	12#	10#
10	37	B3	-200	200	1#	8#
10	37	B4	-200	200	7#	2#
10	37	B5	-300	300	6#	3#
10	38	B4	-130	130	7#	2#
10	38	B5	-130	130	6#	3#
10	38	B1	-180	180	12#	10#
10	38	B6	-300	300	5#	4#
10	38	B2	-630	630	9#	11#
10	38	B3	-630	630	1#	8#
10	39	B4	690	-690	7#	2#
10	39	B5	690	-690	6#	3#
10	39	B6	630	-630	5#	4#
10	39	B1	620	-620	12#	10#
10	39	B2	170	-170	9#	11#
10	39	B3	-100	100	1#	8#
10	40	B6	-400	400	5#	4#
10	40	B3	-420	420	1#	8#
10	40	B1	-450	450	12#	10#
10	40	B2	-450	450	9#	11#
10	40	B4	-450	450	7#	2#
10	40	B5	-450	450	6#	3#
11	41	B5	200	-200	7#	4#
11	41	B3	100	-100	2#	9#
11	41	B4	-660	660	8#	3#
11	41	B2	-1370	1370	10#	1#
11	41	B6	-1370	1370	6#	5#
11	41	B1	1440	-1440	12#	11#
11	42	B1	180	-180	12#	11#
11	42	B6	120	-120	6#	5#

Round	Board	Table	TP-NS	TP-EW	Pair (NS)	Pair (EW)
11	42	B3	90	-90	2#	9#
11	42	B4	90	-90	8#	3#
11	42	B2	-100	100	10#	1#
11	42	B5	-100	100	7#	4#
11	43	B4	300	-300	8#	3#
11	43	B1	-50	50	12#	11#
11	43	B6	-50	50	6#	5#
11	43	B3	-50	50	2#	9#
11	43	B5	-100	100	7#	4#
11	43	B2	-140	140	10#	1#
11	44	B2	300	-300	10#	1#
11	44	B4	300	-300	8#	3#
11	44	B5	160	-160	7#	4#
11	44	B1	150	-150	12#	11#
11	44	B3	-110	110	2#	9#
11	44	B6	-200	200	6#	5#

Table 24: total scores from the 2024 National Bridge Championships Open Pairs Final