A Mathematical Analysis and Simulation-Based Evaluation of Local Decision Rules in Skyjo

Felix Reichel *

Posted Date: 16 April 2026

doi: 10.20944/preprints202601.0112.v2

Keywords: Skyjo; stochastic games; partial information; expected value; Monte Carlo simulation; heuristic strategies; card games

 doi:10.20944/preprints202601.0112.v2

Article

A Mathematical Analysis and Simulation-Based Evaluation of Local Decision Rules in Skyjo

Felix Reichel

Department of Economics, Johannes Kepler University of Linz, Austria; felix.reichel@jku.at

Abstract

Skyjo is a simple stochastic card game with partial information, local replacement decisions, and score-reducing column removal events. This paper develops a formal mathematical model of the game, derives expected-score rules for turn-level actions, proves several dominance and threshold results, and evaluates a family of heuristic strategies through Monte Carlo simulation. The focus here lies on local optimality under explicit belief assumptions rather than a full equilibrium solution of the multiplayer game. Finally, a simulation code is provided for reproducibility.

MSC (2020): Primary: 91A60; Secondary: 60G40, 68T20

JEL Codes: C63, C72, D81

Keywords: Skyjo; stochastic games; partial information; expected value; Monte Carlo simulation; heuristic strategies; card games

1. Introduction

Games with partial information and stochastic draws present a recurring challenge in applied game theory. Skyjo is a representative example in which players repeatedly compare known outcomes against uncertain alternatives under score minimization. The game combines features of replacement games, stopping-time effects, and pattern-based removal incentives.

The contribution of this work is threefold:

a formal expected-score model for single-turn decisions under the official rule set,
one-step comparison results under an exchangeable unseen-state model,
empirical evaluation of threshold policies through Monte Carlo simulation with percentile bootstrap intervals.

The analysis does not claim a globally optimal multiplayer strategy. It isolates local decisions that admit exact immediate-score comparisons and then studies their aggregate effect in self-play.

2. Basic Rules and Game Structure of Skyjo

Skyjo is a finite-horizon stochastic card game with partial information and score minimization. Although the official rules allow for multiple players, the strategic structure of the game can be analyzed at the level of a single representative player interacting with a shared deck and discard pile.

2.1. Card Set and Distribution

The game is played with a fixed multiset of numeric cards. Each card carries an integer value and contributes additively to the final score if not removed.

Definition 1 (Skyjo Card Multiset). The Skyjo deck consists of 150 cards with the following value distribution:

$$\#(-2) = 5, \#(-1) = 10, \#(0) = 15, \#(v) = 10 \quad \text{for } v = 1, \ldots, 12$$

 doi:10.20944/preprints202601.0112.v2

Negative-valued cards are scarce and therefore highly desirable, while large positive values create strong incentives for early replacement or column removal.

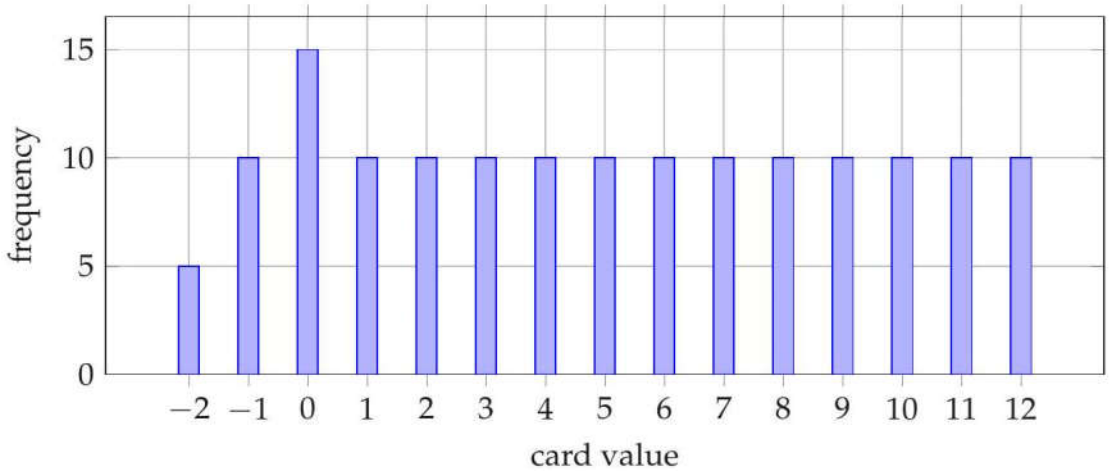

Figure 1. Card value distribution in the Skyjo deck.

2.2. Player Tableau

Each player maintains a private tableau consisting of a 3×4 grid of cards. Cards may be either face-down (hidden) or face-up (revealed).

At the start of the round, all grid positions are filled with cards drawn uniformly from the deck, after which exactly two randomly chosen positions are revealed.

Definition 2 (Tableau State). A tableau position is in one of three states:

- Hidden: card value unknown to the player,
- Revealed: card value known and fixed,
- Removed: position cleared due to column completion.

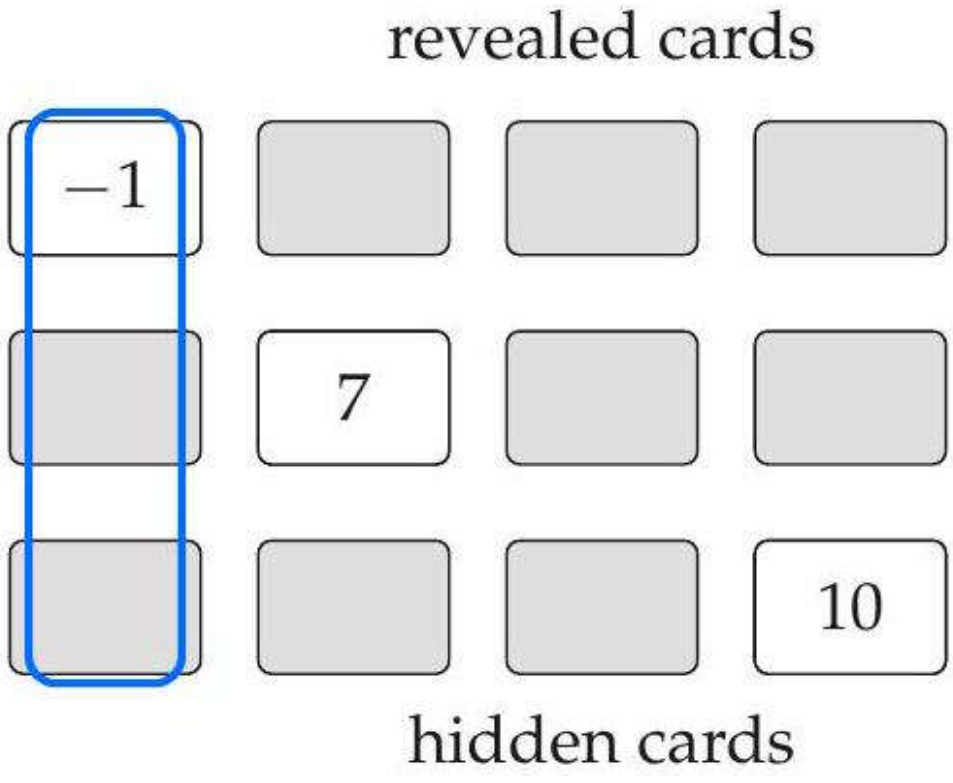

Figure 2. Example player tableau with revealed and hidden cards.

 doi:10.20944/preprints202601.0112.v2

2.3. Player's Turn Structure

A player's turn has three stages.

(i) Draw decision: the player chooses either the top card of the discard pile (known value) or the top card of the draw pile (unknown value).

If the discard pile is chosen, the drawn card must be used to replace exactly one tableau card.

If the draw pile is chosen and the drawn card is kept, it replaces exactly one tableau card. If the draw-pile card is discarded instead, the player must reveal one previously hidden tableau card. That reveal is mandatory under the official rules.

Any tableau card replaced by the drawn card is moved to the discard pile.

A replaced hidden card becomes revealed immediately.

After a replacement or mandatory reveal, if all three cards in a column are revealed and equal, the entire column is removed and contributes zero to the final score.

2.4. Round Termination and Scoring

A round ends when one player has revealed all remaining (non-removed) tableau positions. All other players receive exactly one additional turn.

Definition 3 (Final Score). The final score of a player equals the sum of all remaining card values in the tableau. Lower scores are strictly preferred.

2.5. Strategic Implications

The Skyjo rules induce three interacting incentives:

- replacement of large revealed values,
- risk-reward trade-offs for hidden cards,
- strong nonlinear payoffs from column completion.

These features make Skyjo a natural candidate for expected-value analysis under partial information, while remaining computationally tractable for Monte Carlo evaluation.

3. Game-Theoretic Model

We model Skyjo as a stochastic, imperfect-information, finite-horizon game. The analysis focuses on local score comparisons for a representative player in a two-player self-play environment.

3.1. State Space

Let s denote the current state.

$$s_t = (G_t, R_t, D_t)$$

Write T_s for the player's tableau, R_s for the reveal indicator, P_s for the opponent tableau together with its reveal pattern, and U_s for the multiset of unseen cards.

The unseen multiset U_s consists of the current draw pile together with every face-down tableau card in both players' grids.

Removed cards are denoted by None and contribute zero to the score.

This is the quantity relevant for one-step expectation calculations under exchangeability of unseen cards.

Removed cards are denoted by $\emptyset$ and contribute zero to the score.

3.2. Action Space

At each nonterminal state, the player chooses a composite action

$$a_t = \left(a_t^{\text{draw}}, a_t^{\text{replace}}\right)$$

 doi:10.20944/preprints202601.0112.v2

where:

- $a_t^{\text{draw}} \in \{$ draw pile, discard pile $\}$,
- $a_t^{\text{replace}} \in \{$ discard $\} \cup \{1, \ldots, 12\}$.

Replacement actions specify the tableau index to be replaced.

3.3. Transition Kernel

Transitions are governed by:

- uniform draws from D_t,
- deterministic tableau updates,
- deterministic column-removal rules.

The resulting transition kernel $\mathbb{P}(s_{t+1} \mid s_t, a_t)$ is fully specified by the deck composition and replacement rules.

3.4. Payoff Function

Terminal payoff is defined as

$$u(s_T) = -\sum_{i=1}^{12} G_T(i),$$

where lower tableau sums correspond to higher utility.
The negative sign converts Skyjo into a utility maximization problem.

4. Mapping Rules to Simulation Code

This section records the correspondence between the formal rules and the Python implementation in the Appendix.

4.1. Deck and Belief Model

Card multiset: create_draw_pile()
Belief mean over the full unseen set:

```
def unseen_mean(draw_pile, own_tableau, own_revealed, opp_tableau, opp_revealed):
    unseen = list(draw_pile) + hidden_values(own_tableau, own_revealed) + hidden_values(opp_tableau, opp_revealed)
    return mean(unseen)
```

This implements the unseen-state mean rather than the earlier draw-pile-only average.

4.2. Turn Logic

The formal turn sequence distinguishes three cases: taking the discard pile and replacing, drawing from the draw pile and replacing, and drawing from the draw pile then discarding with a mandatory reveal.

draw → replace/discard → column removal

The central routine is:

```
def play_turn(draw_pile, discard_pile, tableau, revealed, opp_tableau, opp_revealed, threshold):
```

Rule element	Code component
Discard threshold policy	if top_discard <= threshold

 doi:10.20944/preprints202601.0112.v2

Replace revealed card	worst_revealed = max(...)
Replace hidden card	random.choice(hidden)
Column completion	remove_completed_columns()

4.3. Round Termination

Round termination is unchanged: the round ends when one player has revealed every non-removed tableau position, and the other player receives one final turn.

$$\forall i: R(i) = 1 \text{ or } G(i) = \emptyset$$

This is implemented by:

```
def tableau_complete(revealed, tableau):
    return all(revealed[i] or tableau[i] is None for i in range(TABLEAU_SIZE))
```

5. Skyjo Rule Variants and Extensions

Several Skyjo variants can be incorporated without altering the core model.

5.1. Skyjo Action Variant

The Skyjo Action deck introduces special cards allowing:

- additional draws,
- card swaps,
- forced reveals.

Formally, these correspond to temporary action-set expansions:

$$\mathcal{A}_t^{\text{Action}} \supset \mathcal{A}_t.$$

Expected-value analysis remains valid, but transition kernels become history-dependent.

5.2. Alternative Column Rules

Some house rules remove columns only after explicit confirmation. This replaces deterministic removal with a voluntary action, introducing a stopping-time component similar to optimal stopping problems.

6. Strategy Taxonomy

We classify Skyjo strategies by informational sophistication.

6.1. Naive Strategies

- Always draw from the draw pile.
- Replace a random hidden card.

These ignore both revealed information and deck composition.

Parameterized by t:

Parameterized by $t \in \mathbb{R}$:

take the discard card if its value is at most t and use it in a replacement,
replace the largest revealed card exceeding the drawn value whenever such a card exists,
otherwise replace a hidden card if the drawn value is below the current unseen-state mean; if a draw-pile card is discarded, reveal one hidden card.

This is the family studied in Sections 14-16.

 doi:10.20944/preprints202601.0112.v2

6.2. Column-Seeking Strategies

Augment threshold rules with:

- priority for completing 2-of-a-kind columns,
- preference ordering by value v.

These strategies exploit Proposition 6.1 directly.

6.3. Dominance Relations

Proposition 1. Under the official rules, drawing from the draw pile and discarding the drawn card is not weakly dominated by replacement in general, because the mandatory reveal changes the information state even when the immediate tableau sum is unchanged.

The analysis therefore compares one-step score changes directly and treats the reveal as part of the transition law rather than as a null action.

7. Model Scope and Generalization

The framework extends naturally to:

- larger tableaux,
- asymmetric deck compositions,
- replacement games with pattern-based removal.

Skyjo thus serves as a canonical example of local expected-value optimization in games with partial information and non-linear payoffs.

8. Deck Belief State

Let U_s denote the multiset of unseen cards from the perspective of a fixed player at state s. This includes the draw pile and every face-down tableau card in both players' grids.

Definition 4. The unseen-state mean is
mu(s) = (1 / |U_s|) * sum_{u in U_s} u.

$$\mu(\mathcal{U}) = \frac{1}{|\mathcal{U}|}\sum_{v\in\mathcal{U}} v.$$

9. Expected Value of Hidden Cards

Lemma 1. Under the exchangeability assumption on U_s, the expected value of a hidden tableau card equals mu(s).

Proof. Conditional on U_s, each unseen location is a draw without value bias from the multiset U_s. Hence the conditional expectation of a hidden tableau entry is the arithmetic mean of U_s, namely mu(s).

10. Replacement Decisions

Proposition 2. Replacing a hidden tableau card with a known card of value y changes expected score by y - mu(s). In particular, the replacement lowers expected score if and only if y < mu(s).

$$y < \mu(\mathcal{U}).$$

Proof. Before replacement, the hidden position contributes expected value mu(s). After replacement, it contributes y. The difference is y - mu(s).

 doi:10.20944/preprints202601.0112.v2

Proposition 3. If a revealed tableau entry with value z is replaced by a known card y, the immediate score change is y - z. Among revealed entries, the best immediate replacement is therefore any position with maximal z subject to z > y.

Proof. The replacement subtracts z and adds y at the chosen position. Minimizing y - z over revealed positions is equivalent to maximizing z among the admissible entries.

Proposition 4. If a player draws from the draw pile and discards that card, the immediate tableau sum is unchanged, but one previously hidden card must be revealed whenever such a card exists.

Proof. The drawn card never enters the tableau in this branch. The only state change on the tableau is the mandatory reveal required by the official rules.

11. Incentives of Column Removal

Proposition 5. Suppose a column contains two revealed cards of value x and one hidden card. If the player places a further x into the hidden position, the column is removed. Relative to leaving the hidden position unrevealed, the expected immediate score change is -(2x + mu(s)).

$$\Delta(v) = -3 * v + \mu(\mathcal{U}).$$

Proof. Before removal, the expected contribution of the column is 2x + mu(s). After removal, the contribution is 0. The difference is therefore -(2x + mu(s)).

Corollary 1. For fixed mu(s), the gain from such a completion is strictly larger for smaller x.

12. Player's Grid

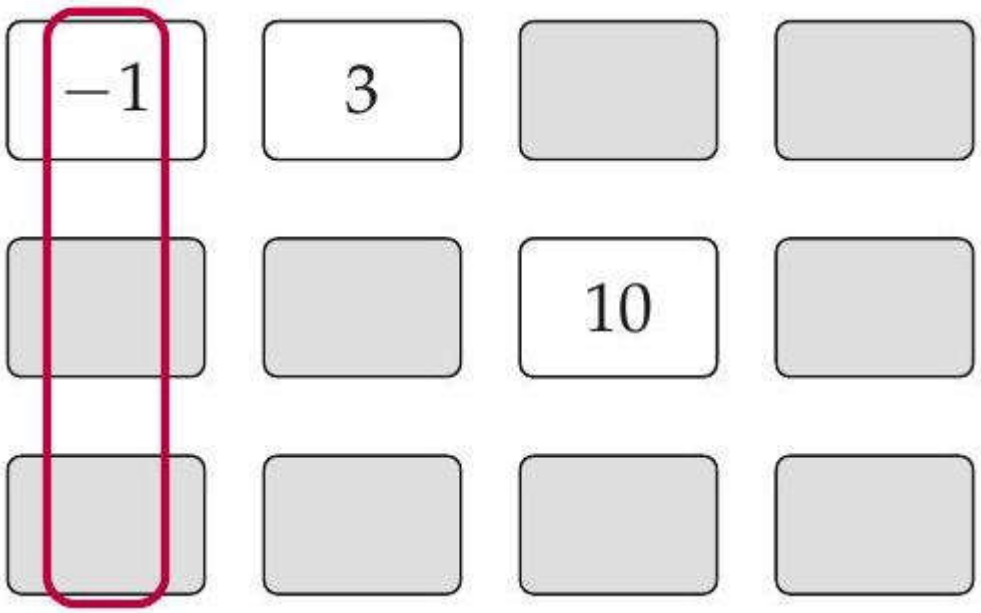

Figure 3. Example grid with one highlighted column.

13. Turn Decision Structure

 doi:10.20944/preprints202601.0112.v2

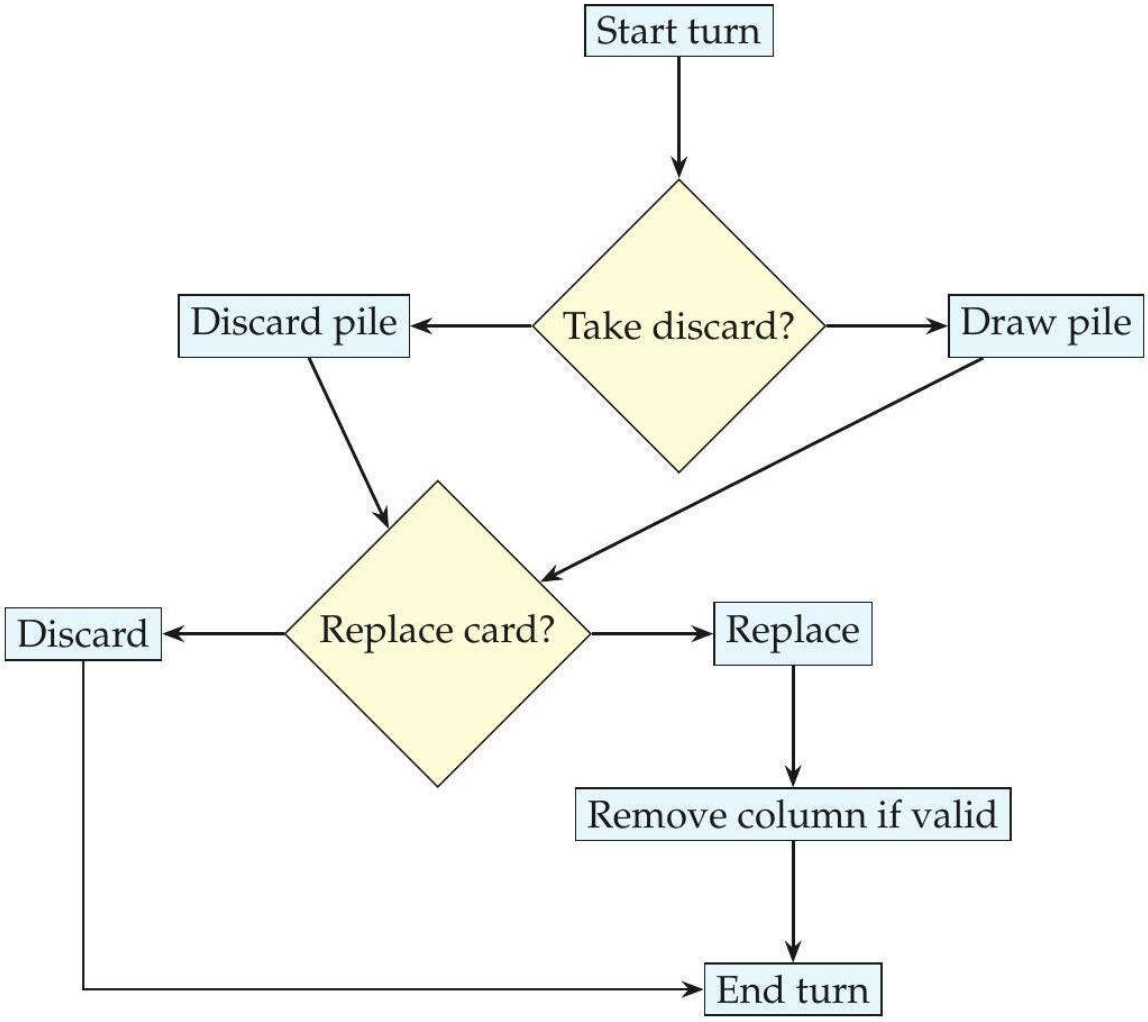

Figure 4. Action flow for a single turn.

14. Simulation Design

Monte Carlo simulations were performed for a two-player self-play setting. Each player used the same heuristic strategy parameterized by a discard threshold t.

Strategy rules:

Take the discard card if its value is at most t.

If a revealed card larger than the drawn value exists, replace the largest such revealed card.

Otherwise replace a hidden card if the drawn value is below the current unseen-state mean.

If a draw-pile card is discarded, reveal one hidden card; column removal is applied immediately after any replacement or reveal.

Each threshold t in {-2,-1,0,...,12} was evaluated in 3000 games, yielding 6000 player scores per threshold. Table 1 reports the sample mean, sample standard deviation, and a 95% percentile bootstrap interval for the mean based on 500 bootstrap resamples.

Table 1. Self-play outcomes.

Threshold t	Mean score	Std. dev.	95% bootstrap CI	n
-2	16.95	10.01	[16.70, 17.20]	6000
-1	16.48	9.93	[16.26, 16.75]	6000
0	15.95	9.94	[15.70, 16.22]	6000
1	15.21	9.77	[14.95, 15.44]	6000

 doi:10.20944/preprints202601.0112.v2

2	14.92	9.54	[14.68, 15.16]	6000
3	15.31	9.05	[15.09, 15.55]	6000
4	17.86	8.83	[17.65, 18.10]	6000
5	21.48	8.85	[21.25, 21.70]	6000
6	26.05	9.29	[25.84, 26.29]	6000
7	30.56	10.08	[30.29, 30.81]	6000
8	35.50	11.08	[35.21, 35.78]	6000
9	40.71	12.20	[40.40, 41.00]	6000
10	45.98	13.36	[45.65, 46.29]	6000
11	51.51	15.05	[51.14, 51.89]	6000
12	57.18	16.08	[56.75, 57.61]	6000

15. Simulation Results

The score curve is nearly flat on the low-threshold range and attains its smallest observed mean at t = 2.

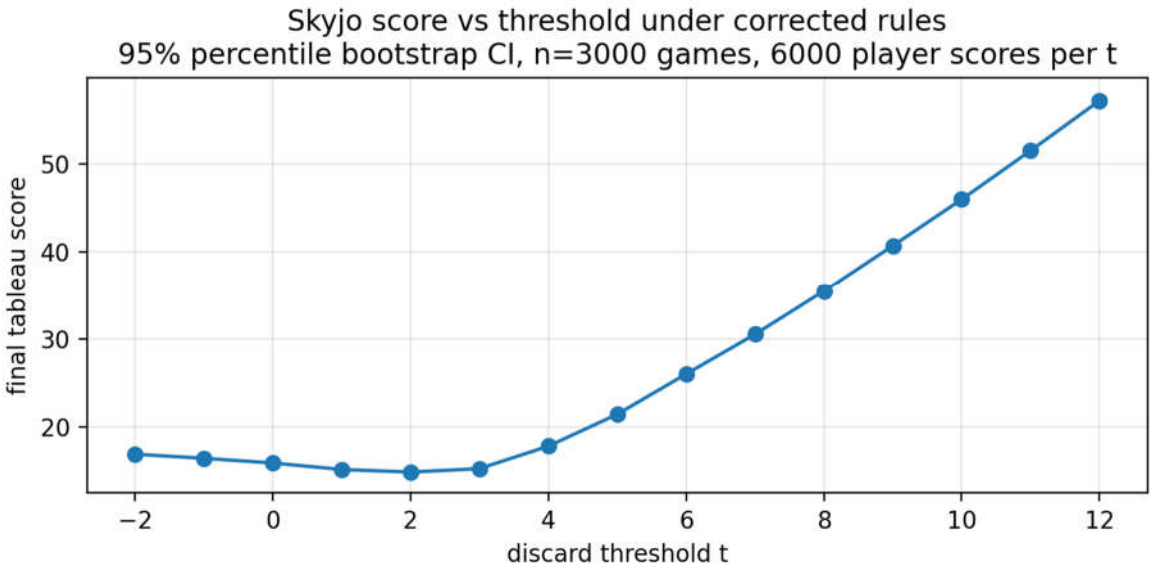

Figure 5. Mean score by discard threshold with 95% percentile bootstrap confidence intervals.

16. Discussion

The threshold curve has a shallow interior minimum rather than the monotone pattern suggested by the earlier table restricted to t >= 0. In these simulations, the smallest observed mean occurs at t = 2, while the values at t = -2, -1, 0, 1, 2, and 3 differ only modestly.

Once t exceeds 3, performance deteriorates steadily. The increase is pronounced from t = 4 onward, which is consistent with the fact that high thresholds commit the player to too many discard-pile cards whose values are no longer favorable relative to the unseen-state mean.

17. Limitations

The model still simplifies the game. The policy is local and does not optimize the choice of which hidden card to reveal after a draw-pile discard, nor does it solve the multiplayer timing problem

 doi:10.20944/preprints202601.0112.v2

induced by round termination. The exchangeability assumption is appropriate for one-step comparisons but not for a full dynamic program on the complete information state.

18. Concluding Remarks

With the state law and the turn rule, the threshold study leads to a more restrained conclusion. Low thresholds remain competitive, but the earlier monotonic argument for $t = 0$ as a distinguished optimum does not survive the official reveal rule or the extension to negative thresholds. Under the current implementation, the low-threshold region is essentially flat, with the best observed performance near $t = 2$.

Funding: Supported by Johannes Kepler Open Access Publishing Fund and the federal state Upper Austria.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflicts of interest.

Appendix A. Python Simulation Code

```
import json
import random
from statistics import mean
import numpy as np


ROWS = 3
COLUMNS = 4
TABLEAU_SIZE = ROWS * COLUMNS
INITIAL_REVEALS = 2
MAX_TURNS = 500
COLS = ((0, 4, 8), (1, 5, 9), (2, 6, 10), (3, 7, 11))
BASE_DECK = tuple([-2] * 5 + [-1] * 10 + [0] * 15 + [v for v in range(1, 13) for _ in range(10)])


def create_draw_pile():
    return list(BASE_DECK)


def remove_completed_columns(tableau, revealed):
    for a, b, c in COLS:
        va, vb, vc = tableau[a], tableau[b], tableau[c]
        if va is None or vb is None or vc is None:
            continue
        if revealed[a] and revealed[b] and revealed[c] and va == vb == vc:
            tableau[a] = tableau[b] = tableau[c] = None
            revealed[a] = revealed[b] = revealed[c] = True


def tableau_score(tableau):
    return sum(x for x in tableau if x is not None)
```

 doi:10.20944/preprints202601.0112.v2


```
def deal_tableau(draw_pile):
    tableau = [draw_pile.pop() for _ in range(TABLEAU_SIZE)]
    revealed = [False] * TABLEAU_SIZE
    for i in random.sample(range(TABLEAU_SIZE), INITIAL_REVEALS):
        revealed[i] = True
    remove_completed_columns(tableau, revealed)
    return tableau, revealed

def draw_card(draw_pile, discard_pile):
    if draw_pile:
        return draw_pile.pop()
    top = discard_pile.pop()
    draw_pile.extend(discard_pile)
    random.shuffle(draw_pile)
    discard_pile[:] = [top]
    return draw_pile.pop()

def hidden_indices(tableau, revealed):
    return [i for i in range(TABLEAU_SIZE) if tableau[i] is not None and not revealed[i]]

def revealed_indices(tableau, revealed):
    return [i for i in range(TABLEAU_SIZE) if tableau[i] is not None and revealed[i]]

def unseen_mean(draw_pile, tableau_a, revealed_a, tableau_b, revealed_b):
    unseen = list(draw_pile)
    unseen.extend(tableau_a[i] for i in hidden_indices(tableau_a, revealed_a))
    unseen.extend(tableau_b[i] for i in hidden_indices(tableau_b, revealed_b))
    return mean(unseen)

def best_revealed_replacement(card, tableau, revealed):
    candidates = [i for i in revealed_indices(tableau, revealed) if tableau[i] > card]
    if not candidates:
        return None
    return max(candidates, key=lambda i: tableau[i])

def choose_hidden_replacement(card, mu, tableau, revealed):
    hidden = hidden_indices(tableau, revealed)
    if hidden and card < mu:
        return random.choice(hidden)
    return None

def reveal_hidden_after_discard(tableau, revealed):
```

 doi:10.20944/preprints202601.0112.v2


```
    hidden = hidden_indices(tableau, revealed)
    if not hidden:
        return
    revealed[random.choice(hidden)] = True
    remove_completed_columns(tableau, revealed)

def play_turn(draw_pile, discard_pile, tableau, revealed, opp_tableau, opp_revealed, threshold):
    mu = unseen_mean(draw_pile, tableau, revealed, opp_tableau, opp_revealed)
    top_discard = discard_pile[-1]

    if top_discard <= threshold:
        card = discard_pile.pop()
        idx = best_revealed_replacement(card, tableau, revealed)
        if idx is None:
            idx = choose_hidden_replacement(card, mu, tableau, revealed)
        if idx is None:
            idx = random.choice(hidden_indices(tableau, revealed))
        discard_pile.append(tableau[idx])
        tableau[idx] = card
        revealed[idx] = True
        remove_completed_columns(tableau, revealed)
        return

    card = draw_card(draw_pile, discard_pile)
    idx = best_revealed_replacement(card, tableau, revealed)
    if idx is None:
        idx = choose_hidden_replacement(card, mu, tableau, revealed)
    if idx is None:
        discard_pile.append(card)
        reveal_hidden_after_discard(tableau, revealed)
        return
    discard_pile.append(tableau[idx])
    tableau[idx] = card
    revealed[idx] = True
    remove_completed_columns(tableau, revealed)

def tableau_complete(revealed, tableau):
    return all(revealed[i] or tableau[i] is None for i in range(TABLEAU_SIZE))

def play_game(threshold):
    draw_pile = create_draw_pile()
    random.shuffle(draw_pile)
```

 doi:10.20944/preprints202601.0112.v2


```
    discard_pile = [draw_pile.pop()]
    tableau_a, revealed_a = deal_tableau(draw_pile)
    tableau_b, revealed_b = deal_tableau(draw_pile)

    end_player = None
    for turn in range(MAX_TURNS):
        if turn % 2 == 0:
            play_turn(draw_pile, discard_pile, tableau_a, revealed_a, tableau_b, revealed_b, threshold)
            if tableau_complete(revealed_a, tableau_a):
                end_player = 0
                break
        else:
            play_turn(draw_pile, discard_pile, tableau_b, revealed_b, tableau_a, revealed_a, threshold)
            if tableau_complete(revealed_b, tableau_b):
                end_player = 1
                break

    if end_player == 0:
        play_turn(draw_pile, discard_pile, tableau_b, revealed_b, tableau_a, revealed_a, threshold)
    elif end_player == 1:
        play_turn(draw_pile, discard_pile, tableau_a, revealed_a, tableau_b, revealed_b, threshold)

    return tableau_score(tableau_a), tableau_score(tableau_b)

def bootstrap_ci_mean(arr, reps=500, alpha=0.05, seed=0):
    rng = np.random.default_rng(seed)
    n = len(arr)
    means = np.empty(reps)
    for b in range(reps):
        idx = rng.integers(0, n, n)
        means[b] = arr[idx].mean()
    lo, hi = np.quantile(means, [alpha / 2, 1 - alpha / 2])
    return float(lo), float(hi)

def simulate_all(n_games=3000):
    results = []
    for t in range(-2, 13):
        random.seed(10000 + t)
        arr = np.empty(n_games * 2, dtype=np.int16)
        for g in range(n_games):
            a, b = play_game(t)
            arr[2 * g] = a
```

 doi:10.20944/preprints202601.0112.v2


```
                arr[2 * g + 1] = b
        lo, hi = bootstrap_ci_mean(arr, reps=500, seed=42 + t)
        results.append({
            "threshold": t,
            "mean": float(arr.mean()),
            "stdev": float(arr.std(ddof=1)),
            "ci_low": lo,
            "ci_high": hi,
            "n_scores": int(arr.size),
        })
    return results


if __name__ == "__main__":
    results = simulate_all(3000)
    with open("skyjo_results.json", "w", encoding="utf-8") as fh:
        json.dump(results, fh, indent=2)
    for row in results:
        print(row)
```

Appendix B. Technical Proofs for the Model

All results in this appendix concern one-step expected-score comparisons under the unseen-state definition and the official turn rule.

Appendix B.1. Exchangeability and Linearity

Lemma 2 (Additivity of Expected Score). Conditional expected tableau score is additive across positions.

Proof. Tableau score is the sum of the contributions of non-removed positions. Conditional expectation is linear, so the expected score equals the sum of the conditional expectations of the individual contributions.

This observation is used throughout the appendix.

Appendix B.2. Hidden-Card Expectation

Lemma 3 (Expected Value of a Hidden Card). Under the exchangeability assumption on U_s, a hidden tableau card has conditional expectation $\mu(s)$.

Proof. Given U_s, the value at any hidden tableau location is sampled without value bias from the multiset U_s. The conditional expectation is therefore the arithmetic mean of U_s.

Appendix B.3. Hidden Replacement

Proposition 6 (Criterion for Hidden Replacement). Replacing a hidden card by a known card y changes conditional expected score by $y - \mu(s)$. Hence the move is immediately favorable exactly when $y < \mu(s)$.

Proof. Before replacement, the hidden position contributes $\mu(s)$ in conditional expectation. After replacement, it contributes y. Subtracting yields $y - \mu(s)$.

 doi:10.20944/preprints202601.0112.v2

Appendix B.4. Revealed Replacement

Proposition 7 (Best Revealed Replacement). If a revealed card z is replaced by y, the immediate score change is y - z. Among revealed cards, the smallest resulting score is obtained by choosing a position with maximal z subject to z > y.

Proof. Only the selected position changes, so the tableau sum changes by y - z. Minimization over admissible positions is equivalent to maximizing z.

Appendix B.5. Draw-Pile Discard with Mandatory Reveal

Proposition 8 (Immediate Effect of Draw-and-Discard). If a player draws from the draw pile and discards the drawn card, the current tableau sum is unchanged, but one hidden tableau card must be revealed whenever such a card exists.

Proof. The drawn card never enters the tableau. The rule requires a reveal, so the tableau values remain the same while the information state changes.

Appendix B.6. Column Completion

Proposition 9 (Immediate Gain from Completing a Column). Suppose two revealed cards in a column both equal x and the third position is hidden. Replacing that hidden position by x and thereby removing the column changes expected score by -(2x + mu(s)).

$$\Delta(v) = -3 * v + \mu(\mathcal{U}).$$

Proof. Before removal, the expected contribution of the column is 2x + mu(s). After removal, the contribution is 0. The difference is therefore -(2x + mu(s)).

Corollary 2 (Ordering by Repeated Value). For fixed mu(s), the magnitude of the gain in Proposition 9 is strictly decreasing in x.

Proof. The expression 2x + mu(s) is strictly increasing in x, so its negative is strictly decreasing.

Appendix B.7. Absence of Threshold Monotonicity on the Domain

Proposition 10. No monotonicity theorem in t follows from the local one-step arguments above once the official reveal rule is enforced and thresholds are extended below 0.

Proof. The local comparisons determine whether a specific known card improves the current tableau relative to hidden or revealed alternatives. They do not impose a total order on long-run self-play outcomes across thresholds, because the frequency of draw-pile discards, the timing of forced reveals, and round-ending effects all vary with t.

Appendix B.8. Simulation Fidelity

Proposition 11 (Implementation Fidelity for the Model). The simulation code in Appendix A implements the unseen-state mean, the mandatory reveal after draw-pile discard, deterministic replacement of the best revealed card when available, and immediate column removal.

Proof. Each of these rule elements has a direct code counterpart in Appendix A: unseen_mean, reveal_hidden_after_discard, best_revealed_replacement, and remove_completed_columns.

Appendix B.9. Scope

The appendix establishes only local score comparisons and code-to-model fidelity. It does not solve the full multiplayer game or derive an optimal global strategy.

 doi:10.20944/preprints202601.0112.v2

Appendix B.10. Scope of Validity

All proofs rely on:
the unseen-state definition,
linearity of conditional expectation,
and local one-step comparisons.

No claim is made regarding equilibrium behavior or globally optimal play.

References

1. Magilano GmbH. Skyjo Official Rules. Magilano, Germany, 2015. Available online: https://www.magilano.com.
2. Magilano GmbH. Skyjo Action Rulebook. Magilano, Germany, 2023.
3. Ross, S.M. Introduction to Probability Models, 11th ed.; Academic Press: Amsterdam, The Netherlands, 2014.
4. Puterman, M.L. Markov Decision Processes: Discrete Stochastic Dynamic Programming; Wiley: Hoboken, NJ, USA, 1994.
5. Fishman, G.S. Monte Carlo: Concepts, Algorithms, and Applications; Springer: New York, NY, USA, 1996.
6. Osborne, M.J.; Rubinstein, A. A Course in Game Theory; MIT Press: Cambridge, MA, USA, 1994.
7. Ferguson, T.S. Optimal stopping and applications. Electronic Journal of Probability, 2008.