Article

Not peer-reviewed version

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

Felix Reichel *

Posted Date: 4 January 2026

doi: 10.20944/preprints202601.0112.v1

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

Article

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

Felix Reichel

Department of Economics, Johannes Kepler University 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.

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

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

JEL Codes: C63, C72, D81

---

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,
- provable dominance and threshold results under a specific deck-belief assumption,
- empirical validation through large-scale Monte Carlo simulation.

The analysis does not claim a globally optimal multiplayer strategy. Instead, it isolates local decisions for which simple and rigorous comparison is then possible.

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:*

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2026 doi:10.20944/preprints202601.0112.v1

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

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

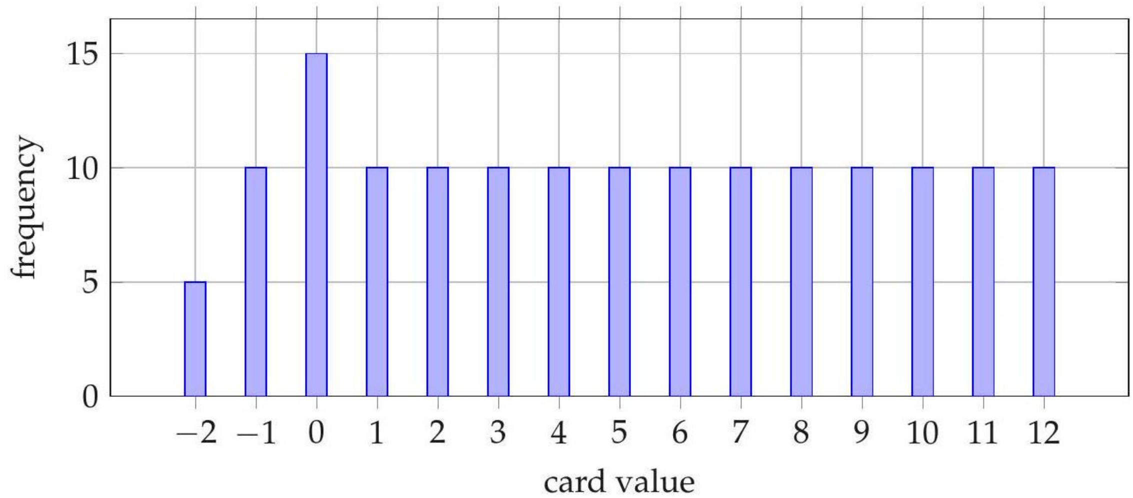

Figure 1. Card value distribution in the Skyjo deck.

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.

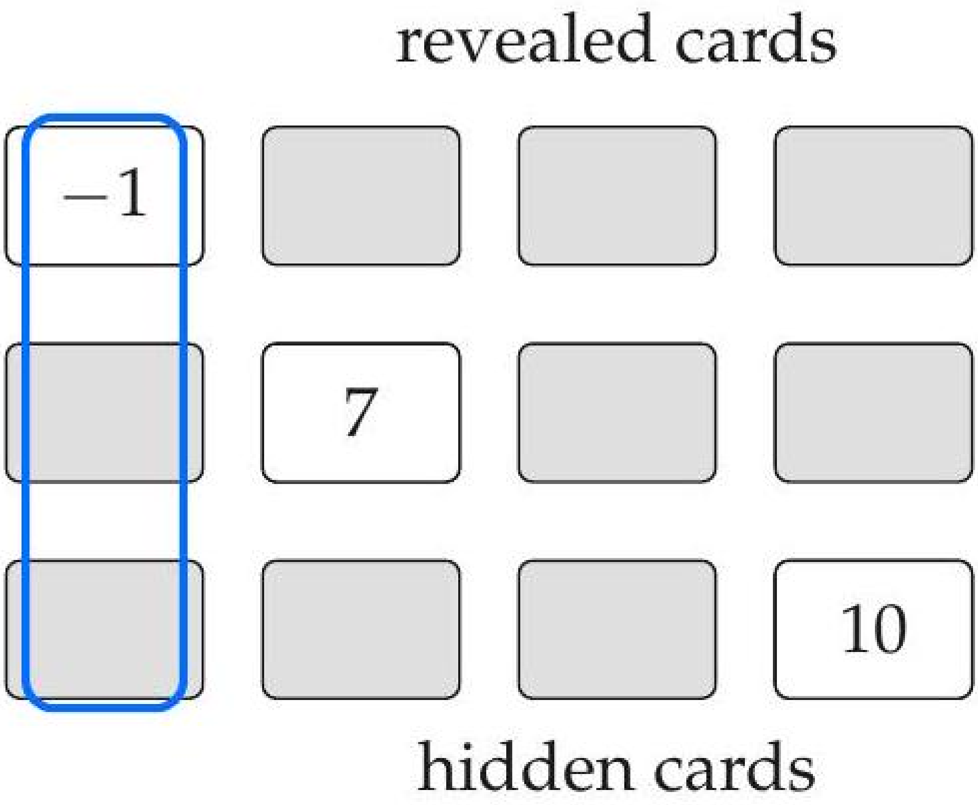

Figure 2. Example player tableau with revealed and hidden cards.

2.3. Player's Turn Structure

A player's turn consists of a sequence of deterministic decisions following a stochastic draw.

(i) Draw decision: The player chooses between

- drawing the top card of the discard pile (known value), or
- drawing the top card of the draw pile (unknown value).

(ii) Replacement decision: After observing the drawn card, the player must either

- discard it immediately, or
- replace exactly one card in the tableau with the drawn card.

If a hidden card is replaced, it becomes revealed.

(iii) Column resolution: If, after replacement, all three cards in a column are revealed and equal in value, the entire column is removed from the tableau 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 a representative player interacting with a shared environment, abstracting from opponent-specific signaling.

3.1. State Space

Let the game state at time t be

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

where:

- $G_t \in (\mathbb{Z} \cup \{\emptyset\})^{12}$ is the player's tableau,
- $R_t \in \{0,1\}^{12}$ is the reveal indicator,
- D_t is the multiset of unseen cards (draw pile + hidden tableau 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

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

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 establishes a precise correspondence between formal rules and the Python implementation provided in the Appendix.

4.1. Deck and Belief Model

- Card multiset: create_draw_pile()
- Deck-belief mean $\mu(\mathcal{U})$:

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2026 doi:10.20944/preprints202601.0112.v1


```
def expected_value_of_draw_pile(draw_pile):
 return mean(draw_pile)
```

This directly implements Definition 3.1.

4.2. Turn Logic

The formal turn sequence:

$$\text{draw} \rightarrow \text{replace/discard} \rightarrow \text{column removal}$$

corresponds to the function:

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

Rule element	Code component
Discard threshold policy	if top_discard <= threshold
Replace revealed card	worst_revealed = max(...)
Replace hidden card	random.choice(hidden)
Column completion	remove_completed_columns()

4.3. Round Termination

The formal round-ending condition

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

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.

doi:10.20944/preprints202601.0112.v1

6.1. Naive Strategies

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

These ignore both revealed information and deck composition.

6.2. Threshold Strategies

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

- take discard if $x \leq t$,
- replace revealed card if dominated,
- otherwise replace hidden card if $y < \mu(\mathcal{U})$.

This class includes the strategy family studied in Section 9.

6.3. 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.

6.4. Dominance Relations

Proposition 1. *Within the class of belief-consistent strategies, discarding a card* $y < \mu(\mathcal{U})$ *without replacement is weakly dominated.*

This explains the empirical failure of conservative discard policies.

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 $\mathcal{U}$ denote the multiset of unseen cards from the perspective of a fixed player. This includes the draw pile and all unrevealed grid cards.

Definition 4. *The deck-belief mean is*

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

9. Expected value of hidden cards

Lemma 1. *Under the deck-belief assumption, the expected value of a hidden grid card equals* $\mu(\mathcal{U})$.

Proof. Each hidden card was drawn uniformly from $\mathcal{U}$. The expectation equals the arithmetic mean of the multiset. □

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2026 doi:10.20944/preprints202601.0112.v1

10. Replacement decisions

Proposition 2. *Replacing a hidden card with a known card of value* y *reduces expected score if and only if*

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

Proof. The expected score change equals $y - \mu(\mathcal{U})$. Negativity gives the condition. □

Proposition 3. *If the discard pile shows a card* $x < \mu(\mathcal{U})$ *and at least one hidden card remains, taking the discard card and replacing a hidden card yields lower expected score than drawing from the unknown pile and discarding.*

Proof. Discard replacement produces expected change $x - \mu(\mathcal{U}) < 0$. Drawing and discarding leaves the grid unchanged. □

Proposition 4. *Given a drawn card of value* y*, replacing the revealed card with maximal value greater than* y *minimizes the immediate resulting score.*

Proof. Replacing position i changes the score by $y - g_i$. Minimization is achieved by choosing the largest admissible g_i. □

11. Incentives of Column removal

Proposition 5. *Suppose a column contains two revealed cards of value* v *and one hidden card. Completing the column yields expected score change*

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

Proof. Before completion, the expected contribution equals $2 * v + \mu(\mathcal{U})$. After completion, the column contributes zero. □

Corollary 1. *Completion attempts for smaller* v *give larger expected score reduction.*

12. Player's Grid

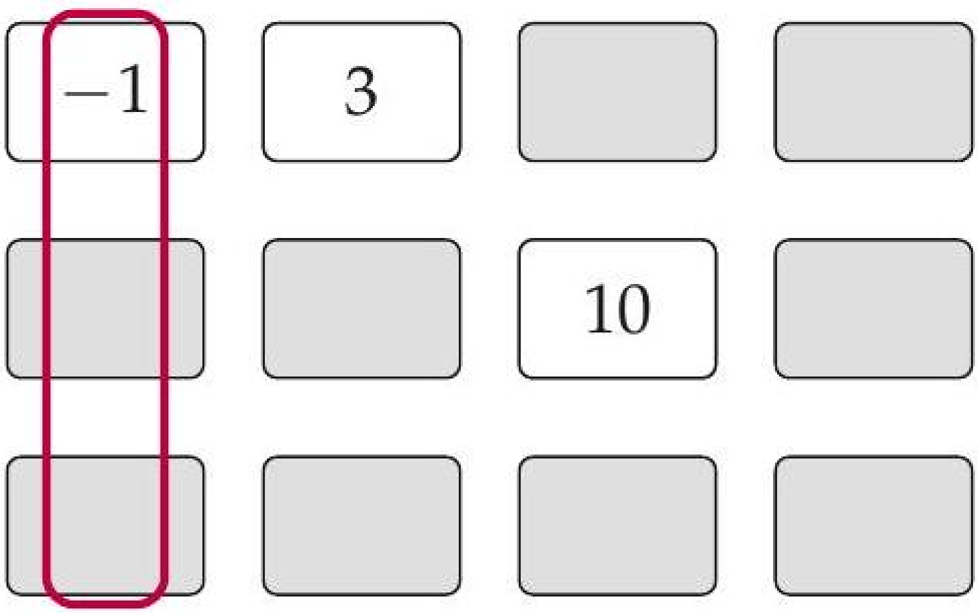

Figure 3. Example grid with one highlighted column.

13. Turn decision structure

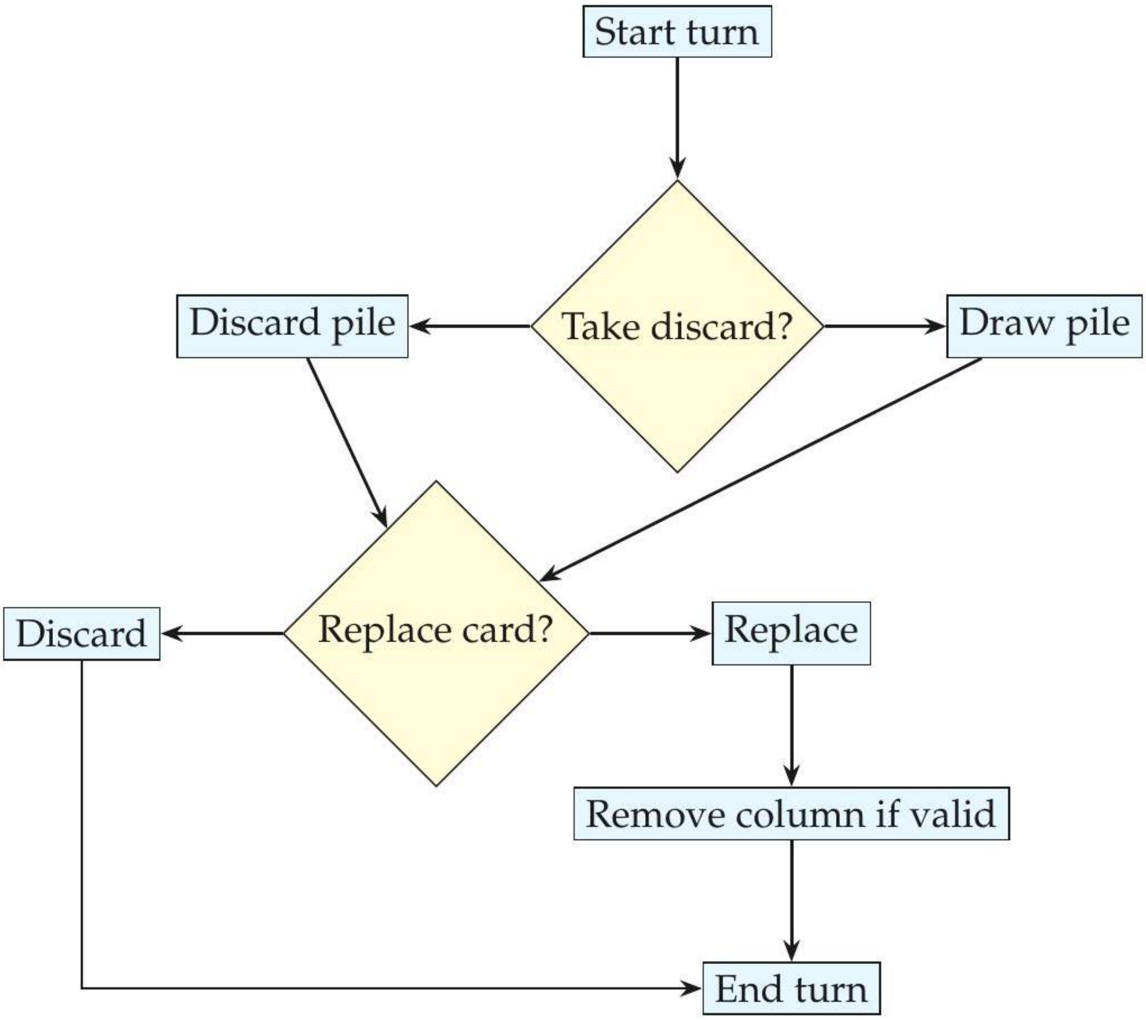

Figure 4. Action flow for a single turn.

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 $x \leq t$.
- Replace the largest revealed card exceeding the drawn value.
- Otherwise, replace a hidden card if the drawn value is below $\mu(\mathcal{U})$.
- If possible, target completion of a two-of-a-kind column.

Each data point consists of 3000 games, counting both players.

15. Simulation results

Table 1. Self-play outcomes for varying discard thresholds.

Threshold $x \leq t$	mean score	standard deviation
0	3.08	8.88
1	4.37	8.52
2	7.02	7.68

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2026 doi:10.20944/preprints202601.0112.v1

3	11.12	7.66
4	15.51	7.75

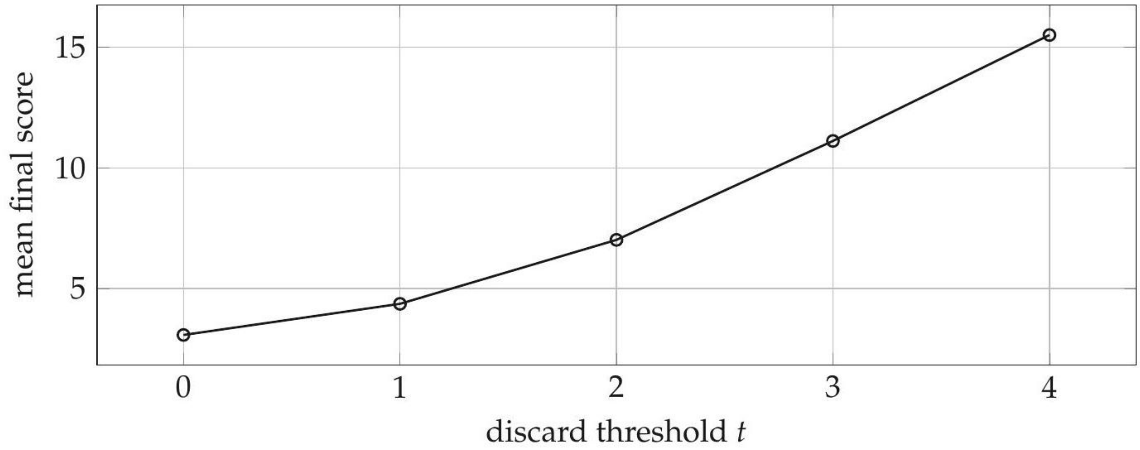

Figure 5. Mean score as a function of the discard threshold.

16. Discussion

The analytical results explain the monotonic behavior observed in simulation. The discard-threshold parameter directly controls whether a player replaces an expected-value draw with a known quantity. Since the deck-belief mean remains positive for most of the round, aggressive discard selection for small values leads to sustained score reduction.

Column completion introduces a linear dependence on the repeated value. This creates a strong preference ordering among otherwise similar replacement opportunities. Simulation outcomes match these structural predictions without requiring tuning beyond the threshold parameter.

17. Limitations

The belief model assumes uniform uncertainty over unseen cards and does not track opponent-specific information. Multiplayer interactions affect round termination timing, which is approximated here by a single additional turn. These simplifications isolate local decision quality rather than full equilibrium behavior.

18. Concluding remarks

This study identifies turn-level rules in Skyjo that admit exact mathematical comparison. The combination of closed-form expected-score expressions and Monte Carlo evidence explains why simple threshold-based play performs well. The framework generalizes to other replacement games with partial information and pattern-based removal.

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 random
from statistics import mean
# ------------------------------
# Game Configuration
```

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 January 2026 doi:10.20944/preprints202601.0112.v1


```
# ------------------------------
ROWS = 3
COLUMNS = 4
TABLEAU_SIZE = ROWS * COLUMNS
INITIAL_REVEALS = 2
MAX_TURNS = 500
# ------------------------------
# Deck Construction (Skyjo)
# ------------------------------
def create_draw_pile():
    """
    Creates a Skyjo draw pile with standard card distribution.
    """
    pile = []
    pile += [-2] * 5
    pile += [-1] * 10
    pile += [0] * 15
    for value in range(1, 13):
        pile += [value] * 10
    return pile
# -----------------------------
# Tableau Utilities
# ------------------------------
def column_indices(column):
    return [column + COLUMNS * row for row in range(ROWS)]

def remove_completed_columns(tableau, revealed):
    """
    Removes a column if all cards are revealed and identical.
    """
    for column in range(COLUMNS):
        indices = column_indices(column)
        values = [tableau[i] for i in indices]
        if any(v is None for v in values):
            continue
        if all(revealed[i] for i in indices) and len(set(values)) == 1:
            for i in indices:
                tableau[i] = None
                revealed[i] = True
def tableau_score(tableau):
    return sum(card for card in tableau if card is not None)
# ------------------------------
```

 doi:10.20944/preprints202601.0112.v1


```
# Dealing
# ------------------------------
def deal_tableau(draw_pile):
    """
    Deals a Skyjo tableau and reveals two random cards.
    """
    tableau = [draw_pile.pop() for _ in range(TABLEAU_SIZE)]
    revealed = [False] * TABLEAU_SIZE
    for index in random.sample(range(TABLEAU_SIZE), INITIAL_REVEALS):
        revealed[index] = True
    remove_completed_columns(tableau, revealed)
    return tableau, revealed
# ------------------------------
# Draw Logic
# ------------------------------
def expected_value_of_draw_pile(draw_pile):
    return mean(draw_pile)

def draw_card(draw_pile, discard_pile):
    """
    Draws a card, reshuffling discard pile if needed.
    """
    if draw_pile:
        return draw_pile.pop()
    top_discard = discard_pile.pop()
    draw_pile.extend(discard_pile)
    random.shuffle(draw_pile)
    discard_pile[:] = [top_discard]
    return draw_pile.pop()
# -----------------------------
# Player Turn
# ----------------------------
def play_turn(draw_pile, discard_pile, tableau, revealed, threshold):
    """
    Executes a single Skyjo turn using a threshold-based policy.
    """
    unseen_mean = expected_value_of_draw_pile(draw_pile)
    top_discard = discard_pile[-1]
    if top_discard <= threshold:
        drawn_card = discard_pile.pop()
    else:
        drawn_card = draw_card(draw_pile, discard_pile)
```

```
    revealed_indices = [
        i for i in range(TABLEAU_SIZE)
        if tableau[i] is not None and revealed[i]
    ]
    hidden_indices = [
        i for i in range(TABLEAU_SIZE)
        if tableau[i] is not None and not revealed[i]
    ]
    replacement_index = select_replacement_index(
        drawn_card,
        unseen_mean,
        tableau,
        revealed_indices,
        hidden_indices
    )

    if replacement_index is None:
        discard_pile.append(drawn_card)
        return
    discard_pile.append(tableau[replacement_index])
    tableau[replacement_index] = drawn_card
    revealed[replacement_index] = True
    remove_completed_columns(tableau, revealed)
def select_replacement_index(card, unseen_mean, tableau, revealed, hidden):
    """
    Decides which card to replace, if any.
    """
    if revealed:
        worst_revealed = max(revealed, key=lambda i: tableau[i])
        if tableau[worst_revealed] > card:
            return worst_revealed
    if hidden and card < unseen_mean:
        return random.choice(hidden)
    return None
# -----------------------------
# Game Loop
# -----------------------------
def tableau_complete(revealed, tableau):
    return all(revealed[i] or tableau[i] is None for i in range(TABLEAU_SIZE))
def play_game(threshold):
    """
    Simulates a two-player Skyjo game.
```

 doi:10.20944/preprints202601.0112.v1


```
    """
    draw_pile = create_draw_pile()
    random.shuffle(draw_pile)
    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, threshold)
        if tableau_complete(revealed_a, tableau_a):
            end_player = 0
            break
    else:
        play_turn(draw_pile, discard_pile, tableau_b, revealed_b, 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, threshold)
elif end_player == 1:
    play_turn(draw_pile, discard_pile, tableau_a, revealed_a, threshold)
return tableau_score(tableau_a), tableau_score(tableau_b)
```

Appendix B. Technical Proofs

All results are derived under the deck-belief assumption and concern local, turn-level expected-score comparisons.

Appendix B.1. Belief Consistency and Linearity

Lemma 2 (Linearity of Expected Score). *Expected tableau score is additive across positions.*

Proof. Each tableau position contributes additively to the final score. Linearity of expectation implies that expected total score equals the sum of expected contributions of individual positions, independently of correlations.

This justifies all subsequent marginal comparisons.

Appendix B.2. Hidden Card Expectation

Lemma 3 (Expected Value of Hidden Cards). *Under the deck-belief assumption, the expected value of a hidden card equals* $\mu(\mathcal{U})$.

Proof. Hidden cards are drawn uniformly from the multiset $\mathcal{U}$. Therefore, the expected value equals the arithmetic mean of $\mathcal{U}$ by definition.

 doi:10.20944/preprints202601.0112.v1

Appendix B.3. Replacement of Hidden Cards

Proposition 6 (Hidden Replacement Criterion). *Replacing a hidden card with a known card of value y reduces expected score if and only if $y < \mu(\mathcal{U})$.*

Proof. The hidden card contributes expected value $\mu(\mathcal{U})$. Replacing it with value y changes expected score by $y - \mu(\mathcal{U})$. The change is negative if and only if $y < \mu(\mathcal{U})$. □

Appendix B.4. Discard vs. Unknown Draw

Proposition 7 (Discard Dominance). *If the discard pile shows a card $x < \mu(\mathcal{U})$ and at least one hidden card remains, then taking the discard card and replacing a hidden card yields lower expected score than drawing from the draw pile and discarding.*

Proof. Replacing a hidden card with x changes expected score by $x - \mu(\mathcal{U}) < 0$. Drawing from the draw pile and discarding produces zero expected score change. Hence discard replacement strictly dominates. □

Appendix B.5. Replacement of Revealed Cards

Proposition 8 (Optimal Revealed Replacement). *Given a drawn card of value y, replacing the revealed card with maximal value exceeding y minimizes the immediate resulting score.*

Proof. Replacing revealed card g_i changes score by $y - g_i$. Among admissible replacements ($g_i > y$), this is minimized by choosing the largest g_i. □

Appendix B.6. Column Completion Incentives

Proposition 9 (Expected Column Removal Gain). *Suppose a column contains two revealed cards of value v and one hidden card. Completing the column yields expected score change*

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

Proof. Before completion, the expected column contribution equals $2 * v + \mu(\mathcal{U})$. After completion, it equals 0. The difference gives the stated expression. □

Corollary 2 (Value Ordering). *For $v_1 < v_2$, we have $\Delta(v_1) > \Delta(v_2)$.*

Proof. $\Delta(v)$ is strictly decreasing in v. □

Appendix B.7. Dominance of Conservative Discarding

Proposition 10 (Dominance of Replacement). *Within the class of belief-consistent strategies, discarding a card $y < \mu(\mathcal{U})$ without replacement is weakly dominated.*

Proof. Discarding produces zero expected score change. Replacing a hidden card produces expected change $y - \mu(\mathcal{U}) < 0$. Thus replacement weakly dominates discarding. □

Appendix B.8. Threshold Monotonicity

Proposition 11 (Threshold Monotonicity). *Expected final score is weakly increasing in the discard threshold t.*

 doi:10.20944/preprints202601.0112.v1

Proof. A larger threshold weakly reduces the set of discard cards taken. This weakly increases reliance on unknown draws, which have expected value $\mu(\mathcal{U})$. Thus expected score weakly increases.

Appendix B.9. Simulation Correctness

Proposition 12 (Implementation Fidelity). *The simulation code in the Appendix correctly implements the formal model defined in Section 4.*

Proof. Each component of the state, action space, transition kernel, and payoff function has a direct implementation: deck draws are uniform, replacements deterministic, and column removal rule exact. Termination conditions coincide.

Appendix B.10. Scope of Validity

All proofs rely solely on:

- the deck-belief assumption,
- linearity of expectation,
- local one-step comparisons.

No claim is made regarding global equilibrium or multiplayer optimality.

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.