The Illusion of Skill in *Camel Up*

David Luk

January 30, 2026

TL;DR: Camel Up is statistically a game of immense skill, but it feels like luck because smart play earns small, invisible gains while risky bets occasionally hit massive jackpots. Even playing perfectly, you will still lose 1 out of 4 games to a beginner because one lucky "lottery ticket" moment can erase an entire game of efficient work. It is not a game of chance, but of patience: the math guarantees you win in the long run, even if variance makes you lose tonight.

Abstract

Camel Up is widely perceived as a luck-driven party game because its core loop is dominated by dice rolls, camel stacking cascades, and highly volatile leg-betting payouts. In contrast, our simulations show that probability calculation provides a measurable *marginal advantage* over competent rule-of-thumb play: `GreedyAgent` (full EV enumeration) consistently outperforms a strong "casual friend" baseline (`HeuristicAgent`) in both win rate and average coin outcome.

We evaluate a four-agent ecosystem and scale the environment from $N = 2$ to $N = 6$ players, then introduce bounded rationality models that emulate human constraints (depth-limited search and noisy probability estimation). The key paradox is resolved by a distributional argument: optimal play yields a small, consistent, and psychologically invisible advantage (about +1.97 coins/turn), while short-term outcome variance is enormous (on the order of ±20 coins), causing casual observers to experience "pure luck" even when the long-run edge is real. Results versus purely random play are reported as a sanity check in the Appendix.

1 Introduction

1.1 The user's dilemma

Camel Up is often described as "chaos in a box": players bet on camel positions while dice are drawn in random order, and a unique stacking mechanic causes one roll to carry multiple camels forward. Volatility is amplified by leg betting, where early bets can be extremely profitable (or sharply negative) depending on a few high-leverage transitions.

A common player sentiment is: "I calculate probabilities, yet I still lose to friends who just guess." This work formalizes that experience: if the game is nearly solved in an expected-value sense, why does it feel random to human players?

1.2 Mathematical background

At any decision point, the game state induces a probability distribution over future camel positions and betting outcomes. Our agents compute these probabilities by enumerating all reachable dice-bag sequences for the remainder of the leg, including the random order in which dice are drawn.

Let k be the number of dice remaining in the leg. Each standard colored die has 3 possible faces (moving 1–3). If we model the remainder of the leg as "draw a die uniformly without replacement, then roll it," the outcome space factorizes into a permutation of dice identities and a vector of face results. Thus the number of distinct outcomes is

$$|\Omega_k| = k!\,3^k. \tag{1}$$

For the standard bag ($k = 5$ colored dice), this gives

$$|\Omega_5| = 5!\,3^5 = 120 \times 243 = 29\,160. \tag{2}$$

When including the grey die, we treat it as having 6 faces (three forward 1–3 and three backward 1–3). The full-bag outcome space becomes

$$|\Omega_{\text{full}}| = 6!\,(3^5 \cdot 6) = 720 \times 1458 = 1\,049\,760 \approx 1.05 \times 10^6. \tag{3}$$

Given an action a in state s (e.g., taking a specific leg bet), we compute its expected value by

$$\text{EV}(a \mid s) = \sum_{\omega \in \Omega(s)} \Pr(\omega \mid s)\, R(s, a, \omega), \tag{4}$$

where $R(s, a, \omega)$ is the coin return if outcome ω occurs. Under the uniform "draw-then-roll" model, $\Pr(\omega \mid s) = 1/|\Omega(s)|$.

1.3 Research gap

Most game AI literature emphasizes perfect-information, low-variance domains (e.g., Chess/Go). Here we analyze a high-variance, imperfect-information party game to quantify whether "skill" is substantive, and to explain how high variance can mask a strong strategic signal.

2 Research Questions

- **RQ1 (Practical ceiling).** What is the win-rate and score advantage of EV-optimal play over a strong heuristic ("casual friend") baseline? (`GreedyAgent` vs `HeuristicAgent`)
- **RQ2 (The Environment).** Does increasing the player count from $N = 2$ to $N = 6$ dilute skill with "noise"?
- **RQ3 (The Human Limit).** Do cognitive limits (e.g., planning depth = 2) or precision errors (sloppy math) destroy the skill edge?
- **RQ4 (The Paradox).** If skill is dominant, why does the game feel random to human players?

3 Methodology

3.1 Simulation engine

We use a Python/PyPy Monte Carlo simulation framework validated by 200+ unit tests, with deterministic seeding, seat alternation to remove first-player bias, and optional multiprocessing for batch runs. The simulator models full game dynamics, including dice-bag sampling, stacking interactions, and betting payouts.

3.2 Probability calculator and EV-optimal policy

The EV engine enumerates reachable outcomes for a decision step and selects the action maximizing expected coin gain.

- **Fast mode:** 29 160 outcomes (excluding the grey die).
- **Full mode:** $\sim$ 1 000 000 outcomes (including the grey die).

EV optimality vs. win probability. Our default objective is risk-neutral: for an action a in state s we maximize

$$\mathrm{EV}(a \mid s) = \mathbb{E}[\Delta C \mid s, a], \tag{5}$$

where ΔC is the immediate (or rollout-estimated) coin change. Importantly, maximizing expected coins is not always equivalent to maximizing the probability of winning,

$$\Pr\big(C_{\text{self}} > \max_{i \neq \text{self}} C_i\big), \tag{6}$$

especially in late-game "winner-takes-all" situations where only first place matters and catch-up requires high variance.

A more general formulation replaces coin return with a utility U:

$$\mathrm{EU}(a \mid s) = \sum_{\omega \in \Omega(s)} \Pr(\omega \mid s)\, U\big(C(s, a, \omega), s'\big), \tag{7}$$

allowing risk-sensitive or win-probability objectives (e.g., during the final leg).

Calibration is verified empirically (reported $R^2 \approx 1.0$), indicating the evaluator's predicted values match realized returns under controlled roll sampling.

3.3 Agent archetypes

We evaluate four canonical agents:

- `RandomAgent`: baseline (no skill).
- `HeuristicAgent`: "casual friend" (leader-following and rule-based betting).
- `GreedyAgent`: "computer" (EV-optimal with full-depth evaluation).
- `Bounded/NoisyAgent`: "human simulator" (depth-limited and error-prone estimates).

4 Results I: Marginal Advantage over Heuristics

4.1 How much does math add over a "casual friend"?

Beating a purely random player is a useful sanity check but a weak bar for "skill." The practically relevant comparison is EV-optimal play versus a competent heuristic player.

In head-to-head play, `GreedyAgent` wins 86.3 % of decisive games against `HeuristicAgent` (2-player, fast mode; 95% CI [0.841, 0.884]) with an average score advantage of roughly +11.5 coins per game. This gap quantifies the *value of information*: how much full probability enumeration adds beyond a rule-of-thumb strategy.

4.2 The "shark tank" effect: N-player scaling

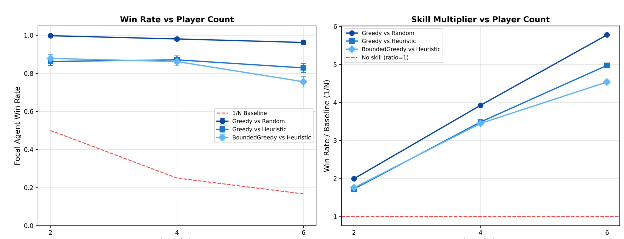

Figure 1: N-player scaling: skilled agent advantage from $N = 2$ to $N = 6$.

Contrary to the intuition "more players means more luck," the skilled agent remains strongly advantaged. In $N = 6$ games, the skilled agent wins $\approx 5.8\times$ more often than the baseline rate, suggesting that chaos provides liquidity: suboptimal actions by opponents increase exploitable mispricings.

5 Results II: The Human Reality (Why People Lose)

5.1 Hypothesis 1: "I can't calculate deep enough"

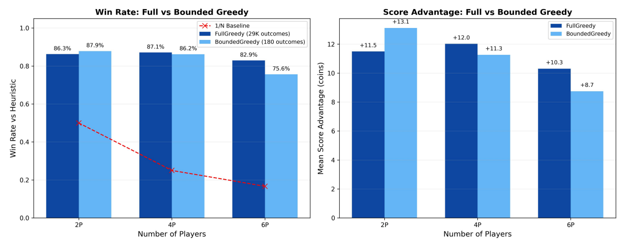

Figure 2: Bounded rationality: depth-limited planning retains most of the edge.

A depth-limited agent (Depth= 2) still achieves a 75.6 % win rate in $N = 6$ games (95% CI [0.729, 0.784]). Thus, deep lookahead is not required to capture most of the strategic value; two steps already recover > 90 % of the advantage relative to full-depth EV play.

5.2 Hypothesis 2: "I make math errors"

We model imprecise probability computation as Gaussian noise injected into EV estimates. At 30 % noise, the estimated win rate is 0.737 (95% CI [0.709, 0.766]) versus the 0% anchor 0.756 (95% CI [0.729, 0.784]); the 1.9 pp difference is within simulation uncertainty and the confidence intervals overlap, so we do not treat this as a statistically significant decrease.

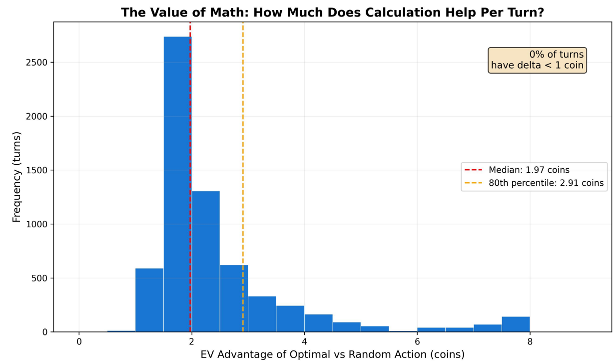

Figure 3: Value of math: optimality manifests as a small, steady per-turn advantage.

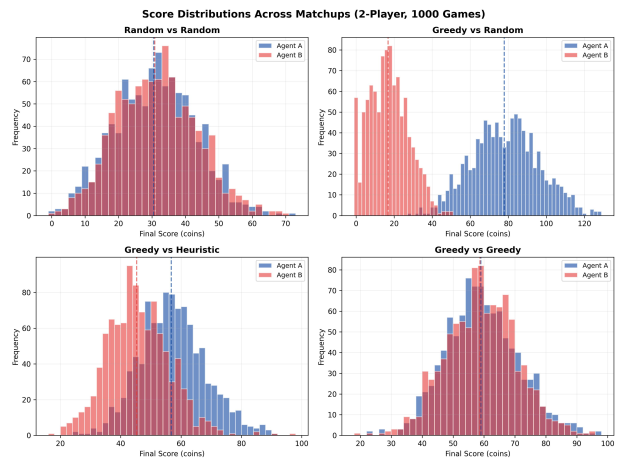

Figure 4: Score distributions: heavy overlap between skilled and casual outcomes.

6 Discussion: The Illusion of Skill

This section answers **RQ4**: the game is strategically dominated, yet psychologically experienced as random.

6.1 The grind (an invisible signal)

Skill in *Camel Up* is not primarily about rare jackpots; it is a persistent $\approx +1.97$ coin advantage per turn. This incremental gain is cognitively "boring" and difficult for humans to attribute to correct decisions.

6.2 The variance (a visible noise source)

Despite a positive mean, the outcome standard deviation is large (on the order of ± 20 coins), so distribution overlap dominates short-run perception. A practical consequence is the "1-in-4" rule: even as a "pro," you should expect to lose roughly 25 % of games to a casual opponent due to variance alone.

Humans overweight salient negative outcomes (the embarrassing loss) relative to routine wins (the expected three), producing an illusion that the system is luck-driven.

6.3 Why humans give up: diminishing perceived returns

Because casual heuristics are already highly effective (e.g., 98 % vs `RandomAgent`), the marginal mental effort required to be truly optimal can feel disproportionate to the perceived reward, even when the long-run EV advantage is real.

7 Limitations

- **No table talk:** real games include negotiation, bluffing, and social manipulation that can change incentives and actions.
- **Static opponents:** our agents do not adapt; humans may learn to target a greedy player.
- **Objective mismatch:** we assume maximizing expected coins; some players optimize for chaos/fun,

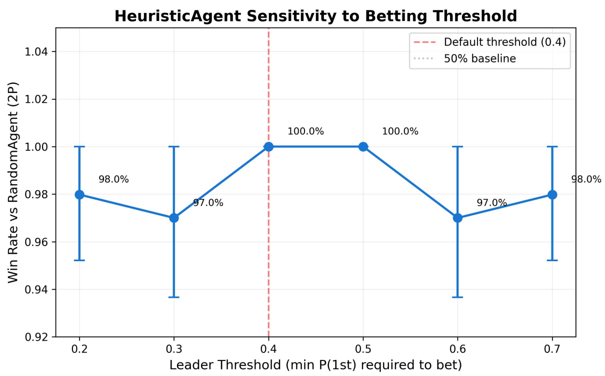

Figure 5: Parameter sensitivity: heuristics already achieve much of the baseline performance.

altering trajectories in ways EV-centric models do not capture.

- **EV vs.** Pr(Win)**:** our strongest agent is coin-EV optimal, not necessarily win-probability optimal. In late-game states where a player is far behind, a lower-EV, higher-variance "lottery ticket" action can increase Pr(Win) even if it decreases expected coins.

8 Conclusion

Camel Up is best characterized as a high-variance efficiency engine: it is strategically exploitable and nearly solved under EV-optimal play, yet experienced as random because variance dominates human salience.

In plain terms: you do not lose because the game is "pure luck." You lose because the skill edge is a slow grind that is frequently overshadowed by lucky windfalls in the short term. The math works, but it requires patience that human psychology often lacks.

A Appendix: Sanity-check baselines

For completeness, we also report results against a purely random policy. These comparisons serve primarily as implementation/validation sanity checks (the agent must exploit obviously mispriced actions).

In 2-player, fast mode, `GreedyAgent` achieves a 99.8 % win rate against `RandomAgent` (95% CI [0.995, 1.000]), confirming that the EV model and action selection are internally consistent in a low-skill environment.