A GENERALIZED THEORY OF CARD COUNTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing card counting systems have largely been developed through heuristic design, without a unified mathematical definition of what constitutes an optimal count. This paper introduces a framework for defining and deriving optimal card-counting systems by formulating card counting as a long-term growth-rate maximization problem. The central idea is to view a count as a low-dimensional representation of the high-dimensional information contained in the exposed-card composition: the representation should preserve the information most relevant to future returns while remaining sufficiently compact for practical use. Based on this formulation, an alternating optimization procedure jointly adapts the count representation and the playing policy, combining a growth-optimal count update with reinforcement learning for count-dependent strategy deviations. The framework establishes connections between card counting, Kelly criterion, information-theoretic value of side information, and representation learning, and can be generalized beyond classical blackjack to variants with different rules and payoff structures. Experiments improve both long-term growth and fixed-spread return, increasing growth by up to 30.92% over Hi-Lo with the Illustrious 18 under 3:2 blackjack and by 76.98% under 6:5; the framework also demonstrates its value on new variants, achieving positive growth and spread in the evaluated Spanish 21 game where baseline methods loses.

1 INTRODUCTION

The mathematical analysis of blackjack predates card counting itself. In the 1950s, Baldwin, Cantey, Maisel, and McDermott, later known as the "Four Horsemen" of blackjack, used early computational methods to determine how a player should act from a given hand against a given dealer upcard Baldwin et al. (1956). Their work laid the foundation for what is now known as *basic strategy*: the composition-averaged policy that selects the action with the highest expected value under a fixed ruleset. In this sense, basic strategy forms the foundation of blackjack decision-making. It answers the question of how to play a hand before any additional information from previously exposed cards is incorporated.

The modern study of card counting begins with Edward O. Thorp, whose book *Beat the Dealer* Thorp (1966) demonstrated that blackjack could be beaten by exploiting the finite composition of the shoe. Thorp's central insight was simple but profound: the cards that have already been played are not merely past events; they provide information about the cards that remain. To convert this idea into a practical counting system, Thorp studied the effect of removing a single card of each rank from the shoe and simulated play using basic strategy. Each removal was classified according to whether it benefited the player, harmed the player, or had little effect. This grouping led to the familiar coarse weights: cards whose removal tends to improve the player's position were assigned +1, approximately neutral cards were assigned 0, and cards that tend to favor the house were assigned -1. By accumulating these weights as cards were exposed, Thorp transformed the discard tray into a compact statistic that could guide both betting and playing decisions.

This statistic is known as the *running count*. In the familiar Hi-Lo system, cards 2 through 6 are assigned +1, cards 7 through 9 are assigned 0, and tens and aces are assigned -1. Summing these values over the exposed cards provides a compact summary of the discard tray. Because the same running count carries different significance early and late in the shoe, counting systems typically convert it into a *true count* by dividing by the estimated number of decks remaining. The true count

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

is then used to determine wager sizes and to identify situations in which the player should depart from basic strategy.

Such departures are known as *deviations*, or *index plays*. A deviation replaces the basic-strategy action for a particular player hand and dealer upcard when the count indicates that changes in the remaining shoe composition make another action more profitable. Don Schlesinger's Illustrious 18 is perhaps the best-known collection of such deviations for traditional blackjack Schlesinger (2017); the canonical index set is listed in Appendix K.2.

Later work refined both count design and evaluation. Wong's balanced Wong Halves count uses half-point tags for finer rank discrimination Wong (1994). Griffin introduced *betting correlation* (BC), which measures alignment with rank-removal effects on expectancy, and *playing efficiency* (PE), which measures support for deviations relative to full-composition decisions Griffin (1999). Vancura and Fuchs's unbalanced Knock-Out (KO) count eliminates explicit true-count conversion Vancura & Fuchs (1998). These developments expose tradeoffs among simplicity, betting and playing accuracy, and robustness. Appendix K summarizes some of the classical tag systems.

Spanish 21 has also motivated game-specific counting methods. Katarina Walker's *The Pro's Guide to Spanish 21 and Australian Pontoon* develops a Hi-Lo-based system together with Spanish 21-specific betting guidance and playing indices Walker (2007). Appendix K.3 gives the corresponding count initialization and compact deviation baseline.

However, these systems are not derived from a unified criterion of count optimality; they are designed through manual weight adjustments and evaluated using auxiliary metrics such as BC and PE. The absence of such a criterion also limits their systematic adaptation to new variants whose rules differ from classical blackjack, where changes in deck composition, available actions, or payoffs can cause classical counting systems to perform poorly (Appendix A). We instead define and jointly optimize the count weights and deviations for the ruleset, treating the count as a low-dimensional statistic that preserves the composition information most relevant to wagering and play.

Contributions. This work makes three contributions: (i) theoretically, it connects card counting to long-term growth optimization and derives the deck-removal conservation law $\mathbf{1}^\top w = 0$; (ii) algorithmically, it introduces alternating optimization of the count and playing policy together with a tractable representation of split hands; and (iii) it provides a C++20 simulation, training, and evaluation framework with complete experiment logs.

2 Problem Formulation

2.1 Ruleset, State, and Basic Strategy

Fix a blackjack ruleset R. The ruleset determines the initial deck composition, dealer drawing rules, blackjack payout, and the set of legal player actions.

Without composition information, a blackjack decision state is represented by $s_0 = (h, u)$, where h denotes the player's current hand information and u denotes the dealer upcard. Let $\mathcal{A}_R(s_0)$ denote the set of legal actions in state s_0. For each legal action $a \in \mathcal{A}_R(s_0)$, let $Q^0_R(s_0, a)$ denote the expected return from taking a and continuing optimally without observing the shoe composition. This expectation averages over compositions conditioned only on the visible state s_0.

Let Π_0 denote the class of no-information policies mapping each visible state to a legal action:

$$\pi : s_0 \mapsto a, \qquad a \in \mathcal{A}_R(s_0).$$

The *basic strategy* is therefore any policy satisfying

$$\pi^B_R(s_0) \in \arg\max_{a \in \mathcal{A}_R(s_0)} Q^0_R(s_0, a).$$

To incorporate composition information, define the full state $s = (\mathbf{c}, h, u)$, where $\mathbf{c} = (c_2, c_3, \ldots, c_A)$ records the discarded cards. For a fixed initial shoe $\mathbf{D} = (D_2, D_3, \ldots, D_A)$, this is equivalent to knowing the remaining composition. Let Π_F denote the class of policies that condition on s. Let $Q^*_R(s, a)$ denote the expected return from taking a in the full state s and continuing with

an optimal policy in Π_F. The corresponding full-information optimal policy is any policy satisfying

$$\pi_R^*(s) \in \arg\max_{a \in \mathcal{A}_R(s)} Q_R^*(s,a).$$

2.2 Card Counting as Linear Information Compression

Although the full vector $\mathbf{c}$ contains detailed information about the exposed cards, maintaining and using the complete vector during live play is impractical. Card counting replaces this high-dimensional information with a low-dimensional statistic.

In complete generality, a count could be any statistic $S(\mathbf{c})$ of the exposed-card information. This paper restricts attention to one-dimensional linear counts because they can be updated efficiently as cards are exposed and remain practical for live use. A weight vector $w \in \mathbb{R}^p$ assigns a tag to each tracked rank, yielding the running count $r_w(\mathbf{c}) = w^\top \mathbf{c}$. Nonlinear or multidimensional counts may preserve more information but are less practical to track during live play.

The same running count has a larger composition effect when fewer cards remain, so it must be normalized by shoe depth. If one deck contains β cards, then the estimated decks remaining are $d_{\mathrm{rem}}(\mathbf{c}) = \frac{|\mathbf{D}|-|\mathbf{c}|}{\beta}$. Define the normalized discard vector $\mathbf{d} = \beta \frac{\mathbf{c}}{|\mathbf{D}|-|\mathbf{c}|}$. The conventional true count can then be written as

$$Y_w(\mathbf{c}) = \frac{w^\top \mathbf{c}}{d_{\mathrm{rem}}(\mathbf{c})} = \beta \frac{w^\top \mathbf{c}}{|\mathbf{D}| - |\mathbf{c}|} = w^\top \mathbf{d}.$$

Because $\mathbf{d}$ incorporates shoe-depth normalization, Y_w serves as the generalized true count.

For notational convenience, the generalized true count is written in augmented form. Specifically, the normalized discard vector is augmented by a constant coordinate and the corresponding coefficient vector includes the bias:

$$\widetilde{\mathbf{d}} = \begin{pmatrix} 1 \\ \mathbf{d} \end{pmatrix}, \qquad \widetilde{w} = \begin{pmatrix} b \\ w \end{pmatrix}, \qquad \widetilde{w}^\top \widetilde{\mathbf{d}} = b + w^\top \mathbf{d}.$$

To avoid additional notation, the tildes are generally suppressed, so an expression of the form $w^\top \mathbf{d}$ includes the bias term through this augmented coordinate. The bias is written separately only when it must be constrained explicitly, as in Section 2.7.

2.3 Count-Dependent Policies and Deviations

Given a count Y_w, a count-dependent playing policy takes the form

$$\pi_{R,w} : (h, u, Y_w) \mapsto a, \qquad a \in \mathcal{A}_R(h,u).$$

Let $\Pi_{R,w}$ denote the class of policies whose composition dependence is restricted to the information contained in Y_w. The policy therefore has access to more information than basic strategy.

A *deviation* occurs when the composition-dependent best action differs from basic strategy: $\pi_R^*(s) \neq \pi_R^B(s_0)$ under full composition, or $\pi_{R,w}(s_w) \neq \pi_R^B(s_0)$ under a linear count.

2.4 Optimal Count as Growth-Relevant Side Information

A count is useful because it compresses the discard tray into side information available before both the wager and the playing decisions are selected. Since the policy observes Y_w, its induced behavior depends on the count weights; we therefore write $\pi(w) \in \Pi_{R,w}$. Let $X^{\pi(w)}$ be the net payoff of the next resolved round, and let Y_w be the scalar side information obtained from the random discard vector. For fixed w and decision rule π, define the optimal logarithmic growth rate

$$G(w, \pi(w)) = \sup_f \mathbb{E}\left[\log\left(1 + f(Y_w) X^{\pi(w)}\right)\right],$$

where the supremum is over admissible betting functions for which wealth remains positive, i.e., $1 + f(Y_w)X^{\pi(w)} > 0$ almost surely. This condition is fully enforced despite the multiple wagers created by splits and doubles; see Appendix I.

The complete problem is coupled because w determines both the side information available to the policy and the argument of the betting function, while π determines the payoff distribution. The joint objective is

$$\max_{w} \max_{\pi \in \Pi_{R,w}} G(w, \pi(w)) = \max_{w} \max_{\pi \in \Pi_{R,w}} \sup_{f} \mathbb{E}\left[\log\left(1 + f(Y_w) X^{\pi(w)}\right)\right].$$

Without count information, the corresponding optimal growth rate is

$$G^0(\pi) = \sup_{f} \mathbb{E}\left[\log\left(1 + f X^{\pi}\right)\right].$$

For a fixed policy π independent of w, the count is side information about its future payoff. The Cover-Thomas value-of-information theorem (Cover & Thomas, 2006, Chapter 16) gives

$$G(w, \pi) - G^0(\pi) \leq I(X^{\pi}; Y_w).$$

The even-money Bernoulli model is a simplified blackjack-like game with binary win-or-loss outcomes and no multi-valued payoffs. In this setting the bound is exact: mutual information characterizes the payoff-relevant information preserved by the count, while logarithmic growth provides the operational objective. Appendix C gives the full portfolio argument and definitions, the equality proof, and the extension to multi-valued blackjack payoffs.

From this point on, we restrict the betting function to the direct linear rule

$$f(Y_w) = Y_w = w^{\top} \mathbf{d}.$$

2.5 Expected-Return Approximation for Policy Optimization

Although the playing policy can be optimized directly for logarithmic growth (Appendix E), blackjack wagers are typically small relative to bankroll. The low-exposure distribution measured in Appendix I therefore supports a first-order approximation linking logarithmic growth to conventional expected-return optimization. When $|fX| \ll 1$, the logarithmic objective admits the expansion

$$\begin{aligned}\mathbb{E}\left[\log(1 + fX)\right] &= f\,\mathbb{E}[X] - \frac{f^2}{2}\mathbb{E}[X^2] + O\left(f^3 \mathbb{E}[|X|^3]\right) \\ &= f\,\mathbb{E}[X] + O\left(f^2 \mathbb{E}[X^2]\right).\end{aligned}$$

For a fixed positive wager fraction f, it follows that

$$\arg\max_{\pi} \mathbb{E}\left[\log(1 + fX^{\pi})\right] \approx \arg\max_{\pi} \mathbb{E}[X^{\pi}].$$

Thus, in the small-wager regime, expected-return maximization is the first-order approximation to logarithmic-growth maximization. This provides the theoretical justification for using conventional Q-learning (Sutton & Barto, 2018, Chapter 6) to optimize the playing policy while retaining logarithmic growth as the objective for optimizing the count weights.

2.6 Linear Growth Optimal Count

We refer to the direct first-order linear approximation of the growth-optimal betting function as the *Linear Growth Optimal Count* (L-GOC). Recall that the optimization is based on the direct betting rule $f(Y_w) = Y_w = w^{\top}\mathbf{d}$. For a fixed playing policy π, define

$$J_{\pi}(w) = \mathbb{E}\left[\log\left(1 + X^{\pi} w^{\top} \mathbf{d}\right)\right].$$

Its exact gradient is

$$\nabla_w J_{\pi}(w) = \mathbb{E}\left[\frac{X^{\pi}\mathbf{d}}{1 + X^{\pi} w^{\top}\mathbf{d}}\right].$$

When $|X^{\pi} w^{\top}\mathbf{d}| \ll 1$, the expansion $\frac{1}{1+u} \approx 1 - u$ gives the following local approximation; see Appendix I.

$$\begin{aligned}\nabla_w J_{\pi}(w) &\approx \mathbb{E}\left[\left(1 - X^{\pi} w^{\top}\mathbf{d}\right) X^{\pi}\mathbf{d}\right] \\ &= \mathbb{E}[X^{\pi}\mathbf{d}] - \mathbb{E}\left[(X^{\pi})^2 \mathbf{d}\mathbf{d}^{\top}\right] w = \mathbf{0}.\end{aligned}$$

Assuming invertibility, the unconstrained first-order solution is

$$w_\pi^* \approx \left(\mathbb{E}\left[(X^\pi)^2 \mathbf{d}\mathbf{d}^\top\right]\right)^{-1} \mathbb{E}[X^\pi \mathbf{d}].$$

Thus, the learned count balances expected return against payoff risk: $\mathbb{E}[X^\pi \mathbf{d}]$ favors discard directions associated with positive returns, whereas $\mathbb{E}[(X^\pi)^2 \mathbf{d}\mathbf{d}^\top]$ penalizes large payoffs. This matters in blackjack because doubles and splits increase payoff magnitude and risk, while surrender limits losses; the count therefore optimizes long-term growth rather than expected return alone.

2.7 Balanced Count Constraint

The L-GOC approximation does not by itself impose structural constraints on the resulting count. To preserve the interpretation of the count throughout the alternating optimization, the rank weights are constrained to satisfy $\mathbf{1}^\top w = 0$. This balance condition follows from a symmetry of the blackjack shoe: removing a complete deck changes the number of cards remaining but does not change their relative composition, and therefore should not change the information represented by the count. The corresponding conservation law and its relation to true-count normalization are derived in Appendix F.1. Let $f(\mathbf{d}) = b + w^\top \mathbf{d}$, where the bias b remains free and is re-estimated together with the rank weights at every iteration. The algorithm therefore maximizes the constrained quadratic L-GOC approximation obtained by applying $\log(1+z) = z - \frac{1}{2}z^2 + O(z^3)$ to the exposure $z = X^\pi(b + w^\top \mathbf{d})$:

$$\max_{b,w:\,\mathbf{1}^\top w = 0} \mathbb{E}\left[X^\pi(b + w^\top \mathbf{d}) - \frac{1}{2}(X^\pi)^2 (b + w^\top \mathbf{d})^2\right]$$

at every count update. The constrained solution and a comparison with alternative constraint variants are given in Appendix F.3.

2.8 Alternating Optimization of Policy and Count

The joint optimization of w and π is coupled. The count weights w determine both the side information available to the policy and the wager fraction, while the policy π determines the payoff distribution X^π. Optimizing either object therefore changes the objective faced by the other, making simultaneous optimization difficult. Writing the growth objective as $G(b, w, \pi)$, the exact alternating updates and their approximations are

$$\pi_k \in \arg\max_{\pi \in \Pi^R_{w_{k-1}}} G(b_{k-1}, w_{k-1}, \pi) \approx \arg\max_{\pi \in \Pi^R_{w_{k-1}}} \mathbb{E}[X^\pi],$$

$$(b_k, w_k) \in \arg\max_{b,w} G(b, w, \pi_k) \approx \arg\max_{b,w:\,\mathbf{1}^\top w = 0} \mathbb{E}\left[X^{\pi_k}(b + w^\top \mathbf{d}) - \frac{1}{2}(X^{\pi_k})^2 (b + w^\top \mathbf{d})^2\right].$$

Thus, each iteration first updates the playing policy using the previous count and then updates the L-GOC weights for the resulting payoff distribution. Section 3 describes the resulting implementation.

2.9 Representation Learning and State Abstraction

More generally, the proposed formulation can be viewed as a representation-learning problem for reinforcement learning under a dimensionality constraint. Let $s \in \mathcal{S}$ denote the full environment state and $z = \phi_\theta(s) \in \mathbb{R}^k$, with $k \ll \dim(\mathcal{S})$, a compressed representation available to the agent. Related work on state abstraction seeks mappings that aggregate states while preserving properties required for near-optimal control Li et al. (2006); Abel et al. (2016), while continuous representation-learning approaches seek low-dimensional embeddings that preserve particular properties of the underlying MDP, such as reward and transition dynamics Gelada et al. (2019) or bisimulation similarity Zhang et al. (2021). In contrast, the present formulation evaluates the compressed representation directly through the downstream objective,

$$(\theta^*, \pi^*) \in \arg\max_{\theta,\pi} J(\pi \circ \phi_\theta), \qquad \dim(\phi_\theta(s)) \ll \dim(s),$$

where J is the downstream control objective. The policy selected for a fixed representation depends on its parameters θ, while the representation selected for a fixed policy depends on which information that policy uses to achieve high return. Writing these conditional optimizers as $\pi^*(\theta)$ and

$\theta^*(\pi)$, with a choice of optimizer when multiple solutions exist, gives the alternating updates

$$\pi_{t+1} = \pi^*(\theta_t) \in \arg\max_{\pi} J(\pi \circ \phi_{\theta_t}),$$
$$\theta_{t+1} = \theta^*(\pi_{t+1}) \in \arg\max_{\theta} J(\pi_{t+1} \circ \phi_{\theta}).$$

The first step holds the representation fixed and learns how to act on it; the second holds the policy's mapping from representations to actions fixed and changes the representation supplied to that mapping. The updated representation then defines the next policy-learning problem. In the blackjack setting considered here, $\phi_{\mathbf{w}}(\mathbf{d}) = \mathbf{w}^\top \mathbf{d}$ restricts the high-dimensional shoe composition to a one-dimensional linear statistic, so count weights and count-dependent play are optimized jointly according to the final growth-rate objective.

3 Algorithm

3.1 Alternating Optimization

We implement the alternating updates from Section 2.8, initialized at $(b_0, W_0) = (0, 0)$:

$$(b_0, W_0) \longrightarrow P_1 \longrightarrow (b_1, W_1) \longrightarrow P_2 \longrightarrow (b_2, W_2) \longrightarrow \cdots .$$

Given W_{k-1}, the algorithm trains the count-dependent policy P_k. Holding P_k fixed, it then simulates resolved rounds and estimates the next bias and balanced count (b_k, W_k). Algorithm 1 summarizes the procedure.

3.2 Policy Update for a Fixed Count

Given W_{k-1}, the algorithm trains a Q-table using simulated hands whose states include the player's hand, dealer upcard, legal actions, and a true-count value computed using W_{k-1} and discretized to an integer by rounding to the nearest integer. Exploration is controlled by either an epsilon-greedy rule or Boltzmann sampling, and the learning rate follows the configured decay schedule (Sutton & Barto, 2018, Chapter 2). Training stops after a fixed number of rounds or when sampled Q-values satisfy the selected convergence criterion. The learned Q-table is then converted into the greedy deterministic policy P_k.

The split-hand update introduces an intermediate zombie state representing one child hand; after a forced hit, its continuation is sampled and the update target doubled, retaining a single-hand Q-table. Appendix B gives the complete construction and proof. Previous blackjack Q-learning approaches either disable resplitting de Granville (2005) or encode the complete set of active split hands, causing the state space to grow rapidly as additional splits are permitted Buramdoyal & Gebbie (2023). The learned actions are compared with published basic-strategy tables in Appendix D.2, including a separate analysis of pair states targeted at the zombie split-hand update.

3.3 Count-Weight Update Step

With P_k fixed, each flat-bet simulated round produces the normalized composition vector $\mathbf{d}_t$ and total payoff X_t. Define the augmented feature and parameter vectors

$$\mathbf{z}_t = \begin{bmatrix} 1 \\ \mathbf{d}_t \end{bmatrix}, \qquad \theta_k = \begin{bmatrix} b_k \\ w_k \end{bmatrix}.$$

The implementation does not store the N individual observations. Instead, it maintains the sufficient statistics

$$S_A \leftarrow S_A + X_t^2 \mathbf{z}_t \mathbf{z}_t^\top,$$
$$S_q \leftarrow S_q + X_t \mathbf{z}_t.$$

After N rounds, the moments used by the constrained solve are $\widehat{A}_{k,N} = \frac{S_A}{N}$, $\widehat{q}_{k,N} = \frac{S_q}{N}$. Thus only one $(p+1) \times (p+1)$ matrix, one $(p+1)$-vector, and a counter are retained: the memory cost is $O(p^2)$, or $O(1)$ with respect to N, rather than $O(Np)$. Statistics produced by separate workers are combined by addition. The resulting moments are inserted into the constrained S0 solution derived in Appendix F.3.

Finally, the ten-valued ranks are averaged and all tags are rescaled so their common tag is -1, preserving balance and matching Hi-Lo units for easy comparison with classical counts.

Algorithm 1 Alternating Optimization for L-GOC-S0

Require: Blackjack ruleset R, number of iterations K, and simulation budgets
Ensure: Final greedy policy P_K, bias b_K, and balanced count W_K
1: Initialize the null system $(b_0, W_0) = (0, 0)$
2: **for** $k = 1, 2, \ldots, K$ **do**
3: Train a Q-table using true-count states induced by W_{k-1}
4: Convert the Q-table into the greedy policy P_k
5: Simulate flat-bet rounds under P_k
6: Stream the sufficient statistics S_A, S_q
7: Form $\widehat{A}_{k,N}$ and $\widehat{q}_{k,N}$
8: Apply the constrained L-GOC-S0 solution to obtain (b_k, W_k)
9: Average the ten-valued-rank tags, then rescale every tag so their shared value is -1
10: **end for**
11: **return** P_K, b_K, and W_K

4 Results

4.1 Evaluation Metrics

Each learned policy-count pair is evaluated by logarithmic growth and fixed-spread profit.

(i) Growth rate. For wager fraction f_t and per-unit payoff x_t, the average logarithmic growth is

$$g_T = \frac{1}{T}\sum_{t=1}^{T}\log(1 + f_t x_t) = \frac{1}{T}\log\frac{B_T}{B_0}.$$

Writing $G = \exp(g_T)$, the tables report the scaled growth $10^4(G - 1)$.

(ii) $1{:}m$ spread. Under a fixed $1{:}m$ spread, the wager and average profit per hand are

$$a_t = \begin{cases} 1, & b + w^\top \mathbf{d}_t \le 0, \\ m, & b + w^\top \mathbf{d}_t > 0, \end{cases} \qquad \widehat{P}_{1:m} = \frac{1}{T}\sum_{t=1}^{T} a_t x_t.$$

The score $b + w^\top \mathbf{d}_t$ predicts the player advantage. When its power-series expansion is valid, logarithmic growth depends on all payoff moments $\mathbb{E}[X^k]$, whereas fixed-spread profit depends only on the first moment. These metrics are standardized evaluation axes, not complete live betting policies.

4.2 Blackjack

We test one-, two-, and six-deck shoes under the American ruleset in Appendix A.1.1, with 3:2 and 6:5 blackjack payouts. L-GOC-S0 is compared with Hi-Lo plus the Illustrious 18 (Appendix K.2), since Hi-Lo is the most widely used practical count and the Illustrious 18 provide a well-established compact deviation set; comparable public deviation sets for other counts are less standardized. The reinforcement-learning hyperparameters and their selection rationale are reported in Appendix G.1. Complete constraint ablations appear in Appendix F.2.

Each count fit uses 5×10^9 retained samples, recording one every five hands; policies train in 10^8-hand blocks until the mean absolute Q-table change is at most 0.005; and each $1{:}10$ spread evaluation uses 5×10^9 hands. Tables report pointwise 95% intervals for Monte Carlo evaluation error only, excluding uncertainty from Q-learning, count fitting, and other sources of training randomness. Appendix G separately evaluates this broader stochastic variability across independent end-to-end L-GOC-S0 runs. Logs and budgets accompany the code described in Section 5.

For growth evaluation, L-GOC uses its untuned wager $f = b + w^\top \mathbf{d}$. Hi-Lo uses $f^{\mathrm{eval}}_{\mathrm{HL}}(\mathrm{TC}) = b + \lambda^*(0.005\,\mathrm{TC})$, where b is the baseline edge (Appendix D). To give Hi-Lo its best linear wager option, λ^* is selected from eleven values spanning $1/\mathbb{E}[X^2] \pm 0.25$ in steps of 0.05, motivated by Appendix C.5; each candidate uses 1,000 experiments of 10^6 hands.

Table 1: Hi-Lo with the Illustrious 18 versus L-GOC-S0 under (a) 3:2 and (b) 6:5 payouts. Entries are estimates ± pointwise 95% confidence bounds; positive improvements favor L-GOC-S0.

	(a) Standard 3:2 payout			**(b)** 6:5 payout		
Decks	I18	L-GOC-S0	Improvement (%)	I18	L-GOC-S0	Improvement (%)
Growth rate, $10^4(G-1)$						
1	1.571030 ±0.010567	2.056782 ±0.012192	30.92 ±1.17	0.543180 ±0.006300	0.961310 ±0.008390	76.98 ±2.57
2	0.686013 ±0.007041	0.833096 ±0.007778	21.44 ±1.69	0.175720 ±0.003850	0.288340 ±0.004730	64.09 ±4.49
6	0.150840 ±0.003335	0.180274 ±0.003454	19.51 ±3.50	0.018340 ±0.001190	0.030290 ±0.001390	65.15 ±13.13
1:10 *spread edge*						
1	0.091575 ±0.000191	0.111640 ±0.000254	21.91 ±0.38	0.024671 ±0.000138	0.045496 ±0.000171	84.41 ±1.24
2	0.052270 ±0.000204	0.060050 ±0.000208	14.88 ±0.60	0.000144 ±0.000123	0.007970 ±0.000134	5434.58* ±4728.37
6	0.013964 ±0.000195	0.017779 ±0.000168	27.32 ±2.15	-0.016799 ±0.000086	-0.014908 ±0.000081	11.26 ±0.66

*The two-deck 6:5 relative spread improvement is numerically unstable because the Hi-Lo baseline is close to zero; the absolute difference is more informative.

L-GOC-S0 improves both metrics for every deck count and payout, with strictly positive improvement intervals. Gains are larger under 6:5, which changes the payoff importance of aces and weakens the match of fixed Hi-Lo tags and deviations, whereas L-GOC adapts both to the altered payoff distribution. The American rules provide a strong established benchmark through the Illustrious 18.

4.3 SPANISH 21

Spanish 21 is a key test of the theory because it changes both the deck composition and the game rules, precisely where transferred counting systems may become mismatched. We evaluate the six-deck game under the ruleset specified in Appendix A.4.1. Table 2 compares L-GOC-S0 with conventional Hi-Lo and Walker's Spanish-specific system (Appendix K.3); full one- and six-deck results appear in Appendix Table 8. The evaluated variant omits the full bonus schedule, but this does not change the Walker action mapping implemented here; Appendix A.4.1 explains the resulting common payoff shift. We also include *full-deviation* baselines, each obtained by training a Q-table conditioned on its fixed count until convergence using the hyperparameters and criterion in Appendix G.1. Wager calibration requires a substantially wider search range here: Walker's selected multipliers are typically near 2, reaching 2.10 for six decks, indicating greater expected-return sensitivity per true-count unit.

Table 2: Six-deck Spanish 21 results. Each entry reports the mean ± its pointwise 95% confidence-interval half-width. Bold values are the best point estimates.

Method	Policy/count	Growth rate, $10^4(G-1)$	1:10 spread edge
Hi-Lo	Full deviations	-0.067130 ± 0.003650	-0.040305 ± 0.000299
Walker	Published deviations	0.043150 ± 0.001840	-0.001804 ± 0.000065
Walker	Full deviations	0.047500 ± 0.001860	0.000638 ± 0.000065
L-GOC-S0	Final iteration	$\mathbf{0.193150 \pm 0.003630}$	$\mathbf{0.017608 \pm 0.000167}$

For six decks, L-GOC-S0 changes the negative growth and spread of the evaluated Hi-Lo baseline into positive values. Relative to Walker's published deviations, it improves excess Kelly growth by 347.6% and turns negative spread edge positive. Relative to Walker with full deviations, it improves excess Kelly growth by 306.6% and spread by 2659.9%. Hi-Lo's selected multiplier remains at a questionable lower search boundary (Appendix Table 8).

Spanish 21 exposes the limitation of transferring hand-designed systems across rulesets. Removing the numeral tens breaks standard Hi-Lo's balance; its evaluated six-deck implementation remains unprofitable even with full deviations, although its wager calibration is unresolved. Walker mitigates this mismatch through Spanish-specific initialization and indices but retains heuristic Hi-Lo tags rather than optimizing the count. L-GOC-S0 instead learns rules-specific balanced tags and deviations from the payoff distribution.

4.4 Ablation Study

The ablation study separates two sources of performance: the learned count representation and its structural constraints. To isolate count quality from policy richness, we compare L-GOC-S0 with the full-deviation baselines defined above. To assess the constraints, we compare unconstrained (U), sum-zero (S0), and fixed-bias sum-zero (S0-B1) variants; full results appear in Appendix F.2. L-GOC-S0 outperforms full-deviation Hi-Lo for every blackjack deck count and metric: in six-deck 3:2, growth and spread edge improve by 14.94% and 25.06%, while under 6:5 the gains reach 45.10% for six-deck growth and 106.95% for two-deck spread. It also outperforms Walker with full deviations in both the six-deck Spanish 21 game and the single-deck setting included for mathematical completeness; for six decks, excess Kelly growth improves by 306.6%, and spread rises from 0.000638 to 0.017608 (Table 8). Across scenarios, S0 provides the strongest overall performance: competing variants occasionally lead by small margins, whereas S0 can yield substantially larger gains, motivating its use throughout.

4.5 Count Quantization

Real-valued weights may be impractical to track, so we measure the cost of jointly quantizing the final W_3 weights to multiples of q while preserving the zero-sum constraint, holding the P_2 policy fixed and performing no wager recalibration. Complete results for all deck counts and payouts appear in Appendix J.

Rounding to $q = 0.01$ is essentially lossless. At $q = 0.5$, degradation begins but remains modest at most 2.60% in spread edge and 7.26% in growth across the tested settings suggesting a useful tradeoff between tracking complexity and performance. Integer weights ($q = 1$) are substantially coarser, with losses reaching 53.36% in spread edge and 63.77% in growth.

4.6 Effect of Deck Count on Alternating Optimization

Most count-fitting samples lie in low-count regimes, whereas deviations primarily change play in rarer high-count regimes. With fewer decks, composition fluctuates more strongly, coupling these regimes and causing the count re-estimated after the first policy update to move farther. Appendix H confirms that this change is larger with fewer decks for both payouts; subsequent updates contract in every setting.

5 Conclusion

This paper introduced a rules-dependent framework for jointly deriving card-counting systems and playing deviations. L-GOC improves growth by up to 30.92% over Hi-Lo with the Illustrious 18 under 3:2 blackjack and by 76.98% under 6:5; in Spanish 21, it achieves positive growth and spread where the evaluated Hi-Lo baseline loses. L-GOC is intended as an additional option, particularly for professional counters willing to trade the simplicity of Hi-Lo for greater precision and more extensive deviations.

However, the main contribution is theoretical: count optimality is defined as preserving the composition information most relevant to long-term growth, together with a procedure for jointly optimizing the count representation and policy. This paper studies a one-dimensional linear representation, $w^\top \mathbf{d}$, producing a practical scalar count, but the definition permits nonlinear and multidimensional representations. The framework therefore supports systematic count design for new variants and unfavorable rules, such as 6:5, where transferred classical systems may be poorly matched or ineffective. To the best of the author's knowledge, no previous framework has jointly derived a count representation and count-dependent policy directly from a ruleset under a long-term growth objective.

Future directions include side bets, whose decisions and payoffs can be added to the policy and return so the main game, side bet, and shared count are optimized jointly. Richer representations could include higher-dimensional linear counts or nonlinear embeddings learned jointly with deep Q-networks Mnih et al. (2015). Comparing them with scalar linear counts would quantify the useful composition information lost through one-dimensional compression.

AI USE STATEMENT

In this work, generative AI tools were used to aid and polish writing, assist with the retrieval and discovery of related work, and assist in writing code for the simulations. All AI-assisted prose was reviewed and revised by the author, retrieved sources were checked against their original publications, and AI-assisted simulation code was reviewed and tested for correctness. Generative AI tools were not used for research ideation, drafting complete sections, generating synthetic datasets, or generating mathematical proofs. The author takes responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies wagering decisions and may therefore have implications for gambling behavior and financial risk. The methods and results are presented for mathematical and scientific analysis and should not be interpreted as a promise of profit or as encouragement to gamble. Even a strategy with positive expected value remains subject to substantial variance, estimation error, implementation error, table limits, and the possibility of severe financial loss. Anyone who chooses to gamble should comply with applicable laws and venue rules, wager only amounts they can afford to lose, and seek support if gambling becomes harmful. The experiments in this paper are simulation based and involve no human participants, personal data, or real-money wagering.

REPRODUCIBILITY STATEMENT

The source code used to implement the blackjack simulator, train count-dependent policies, perform the alternating count optimization, and evaluate the resulting strategies is available in the accompanying repository:

`Anonymous code repository`

The repository contains the C++20 simulation engine, Q-learning and count-optimization applications, comparison and measurement tools, checkpoint formats, build and test instructions, and the complete experiment logs. These logs support the results in Section 4 and the validation, ablation, reproducibility, coupling, exposure, clipping, and quantization studies in Appendices D, F, G, H, I, and J. These materials provide the implementation reference for reproducing the blackjack experiments and for extending the framework to other game variants.

REFERENCES

David Abel, D. Ellis Hershkowitz, and Michael L. Littman. Near optimal behavior via approximate state abstraction. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2915–2923. PMLR, 2016.

Roger R. Baldwin, Wilbert E. Cantey, Herbert Maisel, and James P. McDermott. The optimum strategy in blackjack. *Journal of the American Statistical Association*, 51(275):429–439, 1956.

Avish Buramdoyal and Tim Gebbie. Variations on the reinforcement learning performance of Blackjack. *arXiv preprint arXiv:2308.07329*, 2023.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2 edition, 2006.

C. D. de Granville. Applying reinforcement learning to Blackjack using Q-Learning. Master's thesis, University of Oklahoma, 2005.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G. Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2170–2179. PMLR, 2019.

Peter A. Griffin. *The Theory of Blackjack: The Compleat Card Counter's Guide to the Casino Game of 21*. Huntington Press, 6 edition, 1999.

Lihong Li, Thomas J. Walsh, and Michael L. Littman. Towards a unified theory of state abstraction for MDPs. In *Proceedings of the Ninth International Symposium on Artificial Intelligence and Mathematics*, 2006.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

Don Schlesinger. *Blackjack Attack: Playing the Pros' Way*. RGE Publishing, 4 edition, 2017.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018.

Edward O. Thorp. *Beat the Dealer: A Winning Strategy for the Game of Twenty-One*. Random House, revised edition, 1966.

Olaf Vancura and Ken Fuchs. *Knock-Out Blackjack: The Easiest Card-Counting System Ever Devised*. Huntington Press, 1998.

Katarina Walker. *The Pro's Guide to Spanish 21 and Australian Pontoon*. Lulu.com, 2007. ISBN 9781435710658.

Christopher J. C. H. Watkins and Peter Dayan. Q-Learning. *Machine Learning*, 8(3–4):279–292, 1992. doi: 10.1007/BF00992698.

Wizard of Odds. Blackjack house edge calculator. `https://wizardofodds.com/games/blackjack/calculator/`, 2026a. Accessed August 29, 2026.

Wizard of Odds. Casino games. `https://wizardofodds.com/games/`, 2026b. Accessed September 4, 2026.

Wizard of Odds. Blackjack hand calculator. `https://wizardofodds.com/games/blackjack/hand-calculator/`, 2026c. Accessed September 19, 2026.

Wizard of Odds. Blackjack basic strategy calculator. `https://wizardofodds.com/games/blackjack/strategy/calculator/`, 2026d. Accessed September 19, 2026.

Stanford Wong. *Professional Blackjack*. Pi Yee Press, revised edition, 1994. ISBN 0935926216.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021.

A BLACKJACK RULES AND VARIANTS

This appendix specifies the game features that are relevant to the optimization problem studied in this paper. There is no single universal casino blackjack ruleset: deck count, dealer procedure, legal player actions, and settlement rules vary by table. Accordingly, the ruleset R used throughout the paper should be understood as a complete specification of the game being simulated, rather than as the game name alone.

A.1 REFERENCE BLACKJACK

CARDS, OBJECTIVE, AND SETTLEMENT

Reference blackjack uses one or more standard 52-card decks. Cards 2 through 10 have their face values, jacks, queens, and kings count as ten, and an ace counts as either one or eleven. The player attempts to defeat the dealer without exceeding twenty one. A player bust loses immediately; otherwise the hand wins by exceeding the dealer's non-busted total or when the dealer busts. Ordinary wins pay $1:1$, while a two-card blackjack commonly pays $3:2$, although some tables use $6:5$ or another posted payout Wizard of Odds (2026b).

The player normally receives two initial cards. In a hole-card game, the dealer receives one upcard and one face-down card. Depending on the rules, the dealer may check the hole card for blackjack before player actions when showing an ace or a ten-valued card. In a no-hole-card game, the dealer's second card is delayed until the players have completed their hands. This distinction matters whenever a player doubles or splits against a possible dealer blackjack, because the rules determine whether the additional wagers are also lost or whether only the original wager is at risk.

PLAYER AND DEALER ACTIONS

The standard player actions are hit, stand, double down, split, and, where offered, surrender. A conventional double adds up to one additional original wager and commits the hand to exactly one additional card. Splitting creates two separately played hands and normally requires a second wager. Surrender ends the hand in exchange for losing one half of the original wager. Exact restrictions on doubling, resplitting, split aces, and surrender are table-specific.

After all player hands are complete, the dealer follows a fixed drawing rule regardless of the player's situation. The dealer normally hits totals of sixteen or less and stands on hard seventeen or more. A ruleset must additionally specify whether the dealer hits or stands on soft seventeen.

RULES THAT AFFECT THE LEARNED COUNT

The following rule dimensions are especially important for this work:

- **Deck composition and deck count.** These determine the initial rank frequencies and how strongly a particular removal changes the remaining shoe.
- **Dealer soft-seventeen and hole-card rules.** These change dealer outcome probabilities and the risk attached to doubles and splits.
- **Blackjack payout.** Changing $3:2$ to $6:5$, for example, directly changes the value of aces and ten-valued cards.
- **Doubling and splitting permissions.** These determine when favorable composition information can be converted into additional wager exposure.
- **Surrender.** Surrender adds a payoff of -0.5 units and changes the optimal action near otherwise strongly negative states.
- **Dealer blackjack check (ENHC).** Under European no-hole-card rules, the dealer does not check for blackjack before the player acts. Splitting or doubling against an ace or ten can therefore expose additional wagers to a later dealer blackjack, requiring more conservative play.

Thus even within ordinary blackjack, the card tags and deviation thresholds are functions of the full ruleset.

A.1.1 Experimental American Ruleset

For the blackjack experiments, the reference game is instantiated with the following American ruleset: the dealer receives a hole card and peeks for blackjack, hits soft seventeen, and pays player blackjack at 3:2; surrender is unavailable, doubling after a split is allowed, up to three resplits are permitted (SAS $= 3$, four hands maximum), penetration is 75%, aces cannot be resplit, and each split ace receives one card.

A.2 Double Down Madness

Double Down Madness changes both the initial state and the doubling mechanism. Each player begins with one face-up card, while the dealer receives an upcard and a hole card. The dealer checks for blackjack, and if the dealer has blackjack, the player's wager is lost. Unlike conventional blackjack, the player cannot match the dealer's blackjack because the player begins with only one card. From the one-card starting hand, the player may stand, hit, or double. After each additional card, the player again receives the option to stand, hit, or double; doubling does not automatically terminate the hand. A double may be made for less or may match the amount already in action, so repeated doubles can create substantially larger exposure than a conventional one-time blackjack double Wizard of Odds (2026b).

An initial player ace is treated specially: the player receives only one additional card, whether that card is requested by hitting or doubling. A two-card blackjack produced after a double receives the posted blackjack payout. The game also uses a Push 22 rule: if the dealer busts with exactly twenty two, unresolved main wagers push, while player blackjacks are paid before the dealer completes the hand and are exempt from that rule Wizard of Odds (2026b). The optional Push 22 side wager is not part of the main-game model considered here.

These changes are directly relevant to optimal counting. The state space contains one-card player hands that do not exist in standard blackjack, and the action set permits repeated decisions about increasing exposure. Card composition therefore affects not only the probability of winning but also the number and timing of profitable doubling opportunities. In addition, cards that alter the probability of a dealer total of twenty two affect settlement differently from cards that merely alter the dealer's general bust probability. Consequently, the payoff is naturally multi-valued and path-dependent, and a count fitted for conventional blackjack need not preserve the most valuable information.

Repeated doubling also creates an important limitation of the growth approximation. In the theoretical action model, a player may double as many as nine times, and sequences containing five doubles can occur in practice. Such a sequence can place a very large fraction of the player's bankroll, potentially the entire bankroll, at risk. It may therefore violate the admissibility condition

$$1 + X^{\pi} w^{\top} \mathbf{d} > 0 \quad \text{almost surely},$$

used in 2.4, and it also calls into question the small-exposure assumption

$$\left| X^{\pi} w^{\top} \mathbf{d} \right| \ll 1$$

used in the approximation of Section 2.6. Although such extreme doubling sequences are rare, they are worth noting because their large payoff magnitude can materially affect both bankroll risk and the accuracy of the first-order approximation.

A.3 Free Bet Blackjack

Free Bet Blackjack begins with the same two-card player and dealer hands as standard blackjack, but qualifying splits and doubles are funded by a free-bet marker rather than by an additional player wager. In the published rules used here, pairs may be split for free except for ten-valued pairs, with resplitting permitted subject to table limits. A two-card hard total of 9, 10, or 11 may be doubled for free, including after a split when the qualifying total is present Wizard of Odds (2026b).

If a free double or free split wins, it is paid as though the additional wager had been supplied by the player; if it loses, the player did not risk the free amount. This asymmetric benefit is offset by the Push 22 rule: when the dealer busts with exactly twenty two, main wagers still in action push rather

than win. Player blackjacks are exempt from Push 22. The cited implementation also has the dealer hit soft seventeen Wizard of Odds (2026b). Other installations may vary, so those table-specific parameters must remain explicit in R.

In practical terms, the free-wager mechanism makes many qualifying decisions nearly automatic. Because the additional split or double wager carries no direct capital risk, taking the free action is often an obvious choice, substantially reducing the strategic complexity of the game relative to conventional blackjack.

For this paper, the essential difference is that wager exposure is no longer determined solely by the player's capital. The value of reaching a hard 9, 10, or 11, or a splittable non-ten pair, includes the option value of a free wager. At the same time, the dealer-22 exception changes the value of cards associated with particular dealer draw sequences. A useful count must therefore learn both effects: the composition dependence of qualifying free actions and the composition dependence of Push 22.

A.4 Spanish 21

Spanish 21 changes the shoe itself. It is played with 48-card Spanish decks formed by removing the four ten-spot cards from each ordinary deck; jacks, queens, and kings remain and still count as ten. The proportion of ten-valued cards is therefore reduced from $16/52 = 4/13$ in a standard deck to $12/48 = 1/4$ in a Spanish deck Wizard of Odds (2026b). This is not a minor rules adjustment: it changes the baseline rank distribution before any cards are observed.

The missing tens are offset by player-favorable action and settlement rules. Published Spanish 21 rules commonly provide a $3:2$ player blackjack, priority for a player's twenty one over a dealer twenty one, doubling on a wider range of hands, doubling after splitting, and a double-down rescue option that allows the player to withdraw the added wager and forfeit the original wager after a non-busted double. Certain multi-card twenty ones and combinations such as 6-7-8 or 7-7-7 receive bonus payouts; those bonuses may be unavailable after doubling Wizard of Odds (2026b). Exact bonus schedules, redoubling rules, split restrictions, and dealer soft-seventeen rules vary by installation and must be included in the simulated ruleset.

Spanish 21 differs from standard blackjack in every component relevant to the present framework. The rank alphabet is different because the ten rank is absent. The action space is larger because doubling and rescue may remain available in states where standard blackjack offers only hit or stand. The payoff function depends on the number and identities of cards in the final hand, not only its total. In particular, removals of sixes, sevens, and eights can affect bonus-hand probabilities in ways that have no analogue in ordinary blackjack. These features make Spanish 21 a strong example of why card tags should be derived from the rules rather than transferred from a classical system. Appendix K.3 summarizes Walker's rules-specific practical count for comparison.

A.4.1 Experimental Spanish 21 Ruleset

To limit rule complexity, the simulator omits the multi-card and composition-based bonuses integral to casino Spanish 21. It retains the other principal player-favorable rules: doubling on any number of cards, one redouble (up to four times the original wager), double-down rescue, player twenty one beating dealer twenty one, and player blackjack beating dealer blackjack.

This omission does not alter the published Walker policy implemented in the experiments. Walker's basic-strategy and deviation rules condition on the player total or pair, dealer upcard, and true count, but do not condition on the number of cards in the hand or on eligibility for a bonus payout. Removing the bonuses therefore leaves the Walker action mapping unchanged rather than invalidating the baseline. It does increase the house edge by removing player-favorable payoffs, but that reduced payoff schedule is applied consistently to Walker, Hi-Lo, and L-GOC, so every compared strategy is evaluated under the same harder game.

B ZOMBIE REPRESENTATION OF SPLIT HANDS

B.1 WHY SPLITS CREATE A STATE-SPACE PROBLEM

A split produces two active hands rather than one. A literal representation would therefore have to store the complete joint state of both descendants. For example, after splitting $8, 8$, a later state might contain the two hands $(8, 10, 8)$ and $(8, 7)$, together with the dealer upcard, shoe information, and the legal actions for each hand. Resplitting creates three or four simultaneous hands and increases the number of possible joint configurations still further. Most such states are visited rarely, so a multi-hand Q-table becomes large, sparse, and slow to converge.

Previous blackjack Q-learning approaches address this complexity by either disabling resplitting de Granville (2005) or encoding the complete set of active split hands, which causes the state space to grow rapidly as more hands are permitted Buramdoyal & Gebbie (2023). These limitations motivate the compact representation developed here.

B.2 THE ZOMBIE STATE

To retain a single-hand Q-table, the algorithm replaces the two new one-card hands by one representative *zombie state*. If

$$s = \mathrm{hand}(c, c)$$

is split, then z_c represents one child containing the single card c. A zombie hand is not yet a strategic blackjack hand: its only legal action is a forced hit, which deals the mandatory second card and converts it into an ordinary hand state. The value of the original split is then estimated by doubling the value of this representative continuation.

The notation z_c suppresses information that remains part of the state, including the dealer upcard, count or shoe information, wager type, and split restrictions. These variables must be retained whenever they change the continuation value.

B.3 BELLMAN UPDATE AND DOUBLING THEOREM

For an ordinary transition (s, a, r, s'), the Q-learning update is

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right].$$

The reward term must remain part of the general Bellman equation. Under the reward convention used here, however, splitting does not settle either hand and therefore produces no immediate reward: $r_{\mathrm{split}} = 0$. The rewards arrive only when the resulting hands are resolved.

Theorem 1 (Zombie-hand representation)**.** *Let $G^{(1)}$ and $G^{(2)}$ be the returns of the two hands created by splitting $s = \mathrm{hand}(c, c)$. Assume that their rewards are additive, that the two child hands are exchangeable in expectation, and that $Q^\pi(z_c, \mathrm{hit})$ is the continuation value of one child after the mandatory post-split deal. Then*

$$Q^\pi(s, \mathrm{split}) = 2\gamma Q^\pi(z_c, \mathrm{hit}).$$

For episodic blackjack, where $\gamma = 1$, this reduces to

$$Q^\pi(s, \mathrm{split}) = 2Q^\pi(z_c, \mathrm{hit}).$$

Proof. Additivity gives $G_{\mathrm{split}} = G^{(1)} + G^{(2)}$. Since both children begin with the same card and are represented by the same zombie continuation,

$$\mathbb{E}_\pi[G^{(1)} \mid s, \mathrm{split}] = \mathbb{E}_\pi[G^{(2)} \mid s, \mathrm{split}] = \gamma Q^\pi(z_c, \mathrm{hit}).$$

Therefore, by linearity of expectation,

$$\begin{aligned} Q^\pi(s, \mathrm{split}) &= \mathbb{E}_\pi \left[G^{(1)} + G^{(2)} \mid s, \mathrm{split} \right] \\ &= 2\gamma Q^\pi(z_c, \mathrm{hit}). \end{aligned}$$

Independence of the two hands is not required. □

The corresponding split update, with the reward term shown explicitly, is

$$Q(s, \text{split}) \leftarrow Q(s, \text{split}) + \alpha \left[r_{\text{split}} + 2\gamma Q(z_c, \text{hit}) - Q(s, \text{split}) \right].$$

Since $r_{\text{split}} = 0$, the update used by the algorithm is

$$Q(s, \text{split}) \leftarrow (1 - \alpha) Q(s, \text{split}) + 2\alpha\gamma Q(z_c, \text{hit}).$$

Thus one sampled zombie continuation, doubled in the target, is an unbiased estimate of the expected total split return under the theorem's assumptions.

Caveat for count-conditioned states. The two child hands are not exactly exchangeable when the state includes an evolving count or shoe composition. The cards exposed while playing the first child update the information state before the second child is played, so the second child's decisions condition on more exposed-card information. Its continuation value should therefore be evaluated from the updated state rather than from the same zombie state used for the first child. The additional information has nonnegative decision value because the policy can ignore it, although the physical depletion of the shoe can move the realized conditional payoff in either direction. Consequently, the second child can have a higher optimized expectancy from its richer information state, and replacing the two distinct continuation values by twice one representative value is an approximation rather than an exact identity in count-conditioned Q-learning.

Appendix D.2 provides the complementary empirical test by comparing the learned greedy actions with published basic strategy and reporting pair-state errors separately.

C Information-Theoretic Interpretation of Card Counting

Section 2.6 derived the linear betting-fraction solution used by the algorithm. This appendix places that result in the broader information-theoretic framework of Cover and Thomas (Cover & Thomas, 2006, Chapter 16). It then examines the exact even-money Bernoulli case, its equality with mutual information, its relation to ordinary least squares, and the risk-aware small-wager rule for general payoffs.

C.1 Portfolio Optimization with Side Information

Let $\mathbf{X} = (X_1, \ldots, X_m)^T$ be a random vector of nonnegative price relatives with distribution F_X, where X_j is the one-period multiplicative return of asset j. A portfolio $\mathbf{b} = (b_1, \ldots, b_m)^T$ belongs to the simplex

$$\mathcal{B}_m = \left\{ \mathbf{b} \in \mathbb{R}^m : b_j \geq 0,\ \sum_{j=1}^{m} b_j = 1 \right\}.$$

For $\mathbf{b} \in \mathcal{B}_m$, the one-period wealth multiplier and expected logarithmic growth rate are

$$\mathbf{b}^T\mathbf{X} = \sum_{j=1}^{m} b_j X_j, \qquad W(\mathbf{b}, F_X) = \mathbb{E}_{F_X}\left[\log(\mathbf{b}^T\mathbf{X})\right].$$

For a return sequence $\mathbf{X}_{1:n} = (\mathbf{X}_1, \ldots, \mathbf{X}_n)$ and initial wealth S_0, wealth under the fixed portfolio $\mathbf{b}$ is

$$S_n(\mathbf{X}_{1:n}, \mathbf{b}) = S_0 \prod_{i=1}^{n} \mathbf{b}^T\mathbf{X}_i.$$

Now suppose that side information Y_i is observed before the portfolio for period i is selected, allowing the investor to choose $\mathbf{b}(Y_i) \in \mathcal{B}_m$. Wealth then becomes

$$S_n(\mathbf{X}_{1:n}, \mathbf{b}(Y_{1:n})) = S_0 \prod_{i=1}^{n} \mathbf{b}(Y_i)^T\mathbf{X}_i.$$

Without side information, the optimal growth rate is

$$W^*(\mathbf{X}) = \max_{\mathbf{b}\in\mathcal{B}_m} \mathbb{E}\left[\log\left(\mathbf{b}^T\mathbf{X}\right)\right].$$

With side information Y, the optimal portfolio may depend on Y, giving

$$W^*(\mathbf{X} \mid Y) = \max_{\mathbf{b}(\cdot):\, \mathbf{b}(Y)\in\mathcal{B}_m} \mathbb{E}\left[\log\left(\mathbf{b}(Y)^T\mathbf{X}\right)\right].$$

The increase in achievable growth rate is therefore

$$\Delta W = W^*(\mathbf{X} \mid Y) - W^*(\mathbf{X}).$$

The following classical theorem appears in Cover and Thomas (Cover & Thomas, 2006, Chapter 16).

Theorem 2 (Value of Side Information)**.** *For the portfolio model with side information,*

$$\Delta W \leq I(\mathbf{X}; Y).$$

In the fair-odds horse-race model, this bound is attained.

C.2 Optimal Even-Money Betting

Assume that the player may take either side of an even-money wager. Let the outcome be

$$X = \begin{cases} +1, & \text{with probability } p, \\ -1, & \text{with probability } 1-p. \end{cases}$$

For $f \in (-1, 1)$, the logarithmic growth rate obtained by betting the signed fraction f of the bankroll is

$$\begin{aligned} W(f) &= \mathbb{E}\left[\log(1 + fX)\right] \\ &= p\log(1 + f) + (1 - p)\log(1 - f). \end{aligned}$$

Differentiating and setting the derivative equal to zero gives

$$W'(f) = \frac{p}{1 + f} - \frac{1 - p}{1 - f} = 0.$$

Therefore,

$$f^* = 2p - 1.$$

Since

$$\mathbb{E}[X] = (+1)p + (-1)(1 - p) = 2p - 1,$$

the optimal Kelly fraction is

$$f^* = \mathbb{E}[X].$$

For $0 < p < 1$, the second derivative is strictly negative on $(-1, 1)$, so this stationary point is the unique maximizer. The limiting cases $p \in \{0, 1\}$ place the optimum at the boundary. If negative wagers are prohibited, the constrained solution is $f^* = \max\{0, 2p - 1\}$.

C.3 Equality Between Growth-Rate Gain and Mutual Information

Under the signed-betting assumption of the previous subsection, the optimal fraction for the even-money Bernoulli game is

$$f^* = 2p - 1.$$

Therefore,

$$1 + f^* = 2p, \qquad 1 - f^* = 2(1 - p).$$

Substituting these identities into the optimal logarithmic growth rate gives

$$\begin{aligned} W^*(X) &= p\log(2p) + (1 - p)\log\big(2(1 - p)\big) \\ &= \log 2 + p\log p + (1 - p)\log(1 - p). \end{aligned}$$

Define the binary entropy function, using natural logarithms, by

$$h_2(q) = -q\log q - (1 - q)\log(1 - q).$$

Since $H(X) = h_2(p)$, it follows that

$$W^*(X) = \log 2 - H(X).$$

Now suppose that side information Y is observed before the bet is placed, and define

$$p(y) = \mathbb{P}(X = +1 \mid Y = y).$$

Conditioned on $Y = y$, the game is again an even-money Bernoulli game. Applying the result above to the conditional distribution gives

$$f^*(y) = 2p(y) - 1 = \mathbb{E}[X \mid Y = y]$$

and

$$W^*(X \mid Y = y) = \log 2 - h_2\big(p(y)\big).$$

Averaging over Y,

$$\begin{aligned} W^*(X \mid Y) &= \mathbb{E}_Y\left[W^*(X \mid Y = y)\right] \\ &= \log 2 - \mathbb{E}_Y\left[h_2\big(p(Y)\big)\right] \\ &= \log 2 - H(X \mid Y). \end{aligned}$$

Consequently, the gain in optimal logarithmic growth is

$$\begin{aligned} \Delta W &= W^*(X \mid Y) - W^*(X) \\ &= H(X) - H(X \mid Y) \\ &= I(X; Y). \end{aligned}$$

Thus,

$$\boxed{\Delta W = I(X; Y).}$$

For the even-money Bernoulli game, the Cover-Thomas upper bound is therefore attained with equality: mutual information is exactly the additional optimal logarithmic growth made possible by observing Y.

C.4 Even-Money Bernoulli Limit and Ordinary Least Squares

Consider the even-money Bernoulli game $X \in \{-1, +1\}$ and the linear betting fraction

$$f_w(\mathbf{d}) = w^\top \mathbf{d}.$$

The exact logarithmic-growth objective is

$$J(w) = \mathbb{E}\left[\log\left(1 + Xw^\top \mathbf{d}\right)\right],$$

with first-order condition

$$\mathbb{E}\left[\frac{X\mathbf{d}}{1 + Xw^\top \mathbf{d}}\right] = \mathbf{0}.$$

This condition is exact. To connect it with classical regression, suppose that the wager is small, $|f_w(\mathbf{d})| \ll 1$. Since $X^2 = 1$, the approximation

$$\frac{1}{1 + Xw^\top \mathbf{d}} \approx 1 - Xw^\top \mathbf{d}$$

gives

$$\begin{aligned}\mathbf{0} &\approx \mathbb{E}\left[\left(1 - Xw^\top \mathbf{d}\right) X\mathbf{d}\right] \\ &= \mathbb{E}[X\mathbf{d}] - \mathbb{E}[\mathbf{d}\mathbf{d}^\top]w.\end{aligned}$$

Therefore,

$$w^* \approx \left(\mathbb{E}[\mathbf{d}\mathbf{d}^\top]\right)^{-1} \mathbb{E}[X\mathbf{d}].$$

To show the equivalence explicitly, consider the population least-squares problem

$$\min_w \mathbb{E}\left[\left(X - w^\top \mathbf{d}\right)^2\right].$$

Expanding the objective gives

$$\begin{aligned}\mathbb{E}\left[\left(X - w^\top \mathbf{d}\right)^2\right] = \mathbb{E}[X^2] - 2w^\top \mathbb{E}[X\mathbf{d}] \\ + w^\top \mathbb{E}[\mathbf{d}\mathbf{d}^\top]w.\end{aligned}$$

Its first-order condition is

$$-2\mathbb{E}[X\mathbf{d}] + 2\mathbb{E}[\mathbf{d}\mathbf{d}^\top]w = \mathbf{0},$$

which is the same moment equation obtained from the small-wager approximation of logarithmic growth. Therefore,

$$w_{\mathrm{OLS}} = \left(\mathbb{E}[\mathbf{d}\mathbf{d}^\top]\right)^{-1} \mathbb{E}[X\mathbf{d}] \approx w^*.$$

For simulated observations collected in a design matrix Z and outcome vector $\mathbf{x}$, the corresponding empirical problem and, when $\mathsf{Z}^\top\mathsf{Z}$ is nonsingular, its solution are

$$\begin{aligned}\widehat{w}_{\mathrm{OLS}} &= \arg\min_w \|\mathbf{x} - \mathsf{Z}w\|_2^2 \\ &= \left(\mathsf{Z}^\top\mathsf{Z}\right)^{-1} \mathsf{Z}^\top\mathbf{x}.\end{aligned}$$

Thus, in the even-money Bernoulli case, the exact logarithmic-growth solution approaches classical OLS in the small-wager limit.

C.5 General Payoffs and Risk-Aware Wagering

For a general payoff distribution and constant wager fraction f, let

$$g(f) = \mathbb{E}\left[\log(1 + fX)\right],$$

The exact first-order condition is

$$g'(f) = \mathbb{E}\left[\frac{X}{1 + fX}\right] = 0.$$

When $|fX| \ll 1$, the approximation

$$\frac{1}{1 + fX} \approx 1 - fX$$

gives

$$g'(f) \approx \mathbb{E}[X] - f\mathbb{E}[X^2].$$

Hence, provided $\mathbb{E}[X^2] > 0$ and the resulting fraction remains admissible, the local growth-optimal fraction is approximately

$$f^* \approx \frac{\mathbb{E}[X]}{\mathbb{E}[X^2]}.$$

The numerator rewards positive expectancy, whereas the second moment in the denominator penalizes outcomes with greater magnitude and dispersion. Logarithmic-growth optimization can therefore be viewed as risk-aware expectancy maximization: it favors profitable opportunities while reducing exposure when payoffs are more variable.

C.6 Symmetric Payoffs: Equality of the Approximate and Exact Solutions

Consider the symmetric payoff distribution

$$X = \begin{cases} +a, & \text{with probability } p, \\ -a, & \text{with probability } 1-p, \end{cases} \qquad a > 0.$$

For a constant wager fraction f, the expected logarithmic growth is

$$g(f) = p\log(1+af) + (1-p)\log(1-af), \qquad |af| < 1.$$

Differentiating gives

$$g'(f) = \frac{pa}{1+af} - \frac{(1-p)a}{1-af}.$$

The first-order condition $g'(f) = 0$ is equivalent to

$$p(1-af) = (1-p)(1+af),$$

and therefore, for $0 < p < 1$, the exact growth-optimal wager fraction is

$$f^* = \frac{2p-1}{a}.$$

For the same distribution,

$$\frac{\mathbb{E}[X]}{\mathbb{E}[X^2]} = \frac{a(2p-1)}{a^2} = \frac{2p-1}{a},$$

so the general small-wager approximation coincides with the exact logarithmic-growth solution.

D SIMULATOR VALIDATION

D.1 HOUSE-EDGE VALIDATION

To assess the reliability of the simulator used in this study, we compared the empirical house edge it produces with theoretical benchmarks obtained from the Wizard of Odds Blackjack House Edge Calculator Wizard of Odds (2026a). For each rule configuration, we evaluated both the continuous-shuffling-machine (CSM) model and the traditional cut-card shoe model, using the matching theoretical values reported by the calculator.

The validation grid crosses deck count $\{1, 2, 6\}$, dealer soft-seventeen behavior $\{\mathrm{S17}, \mathrm{H17}\}$, double after split $\{\mathrm{no}, \mathrm{yes}\}$, player doubling totals $\{\mathrm{all}, \text{9-11}, \text{9-10}\}$, split-after-split limit $\mathrm{SAS} \in \{2, 3, 4\}$, resplitting aces $\{\mathrm{no}, \mathrm{yes}\}$, drawing beyond one card to split aces $\{\mathrm{no}, \mathrm{yes}\}$, dealer peek $\{\mathrm{false}, \mathrm{true}\}$, late surrender $\{\mathrm{no}, \mathrm{yes}\}$, and blackjack payout $\{3{:}2, 6{:}5\}$. The Cartesian product contains 3456 possible rule configurations. A random subset of 1510 configurations was simulated, with each configuration producing both a CSM estimate and a cut-card estimate, for a total of 3020 validation experiments.

For every sampled rule configuration and each shoe model (CSM or cut card), we simulated a total of $N = 10^9$ independent hands. During each run, we tracked the cumulative net result from the player's perspective, denoted by M_{total}, assuming a fixed base wager of 1 per hand. The empirical player edge was computed as

$$\widehat{E}_{\text{player}} = \frac{M_{\mathrm{total}}}{N},$$

and the corresponding house edge estimate was defined as

$$\widehat{H} = -\widehat{E}_{\text{player}}.$$

For each rule set and shoe model, the theoretical benchmark value H_{theory} was extracted from the Wizard of Odds calculator Wizard of Odds (2026a), matching all options listed above. The error in percentage points was computed as

$$\varepsilon = 100 \left(\widehat{H} - H_{\text{theory}} \right).$$

Across most configurations, the simulator aligned closely with the theoretical values. The remaining systematic discrepancy occurs when dealer peek is disabled and surrender is enabled; these configurations show an offset of approximately $+0.15$ percentage points, with the simulator predicting a more favorable outcome for the player. We contacted the Wizard of Odds maintainers for clarification but received no response. The mismatch may arise because the calculator describes its option as "late surrender," a designation that is ambiguous in a no-peek game because the dealer's blackjack status is not determined before the player acts. Consequently, no-peek configurations with surrender are displayed for completeness but excluded from the accuracy calculations.

Validation employed a two-sided 99% normal-approximation confidence interval. The standard deviation of the return from one blackjack round was approximated as $\sigma = 1.2$ betting units, based on published blackjack variance estimates Griffin (1999); Schlesinger (2017). For $N = 10^9$ rounds, the estimated standard error was

$$\mathrm{SE}(\hat{\mu}) = \frac{\sigma}{\sqrt{N}} = \frac{1.2}{\sqrt{10^9}} \approx 3.7947 \times 10^{-5}.$$

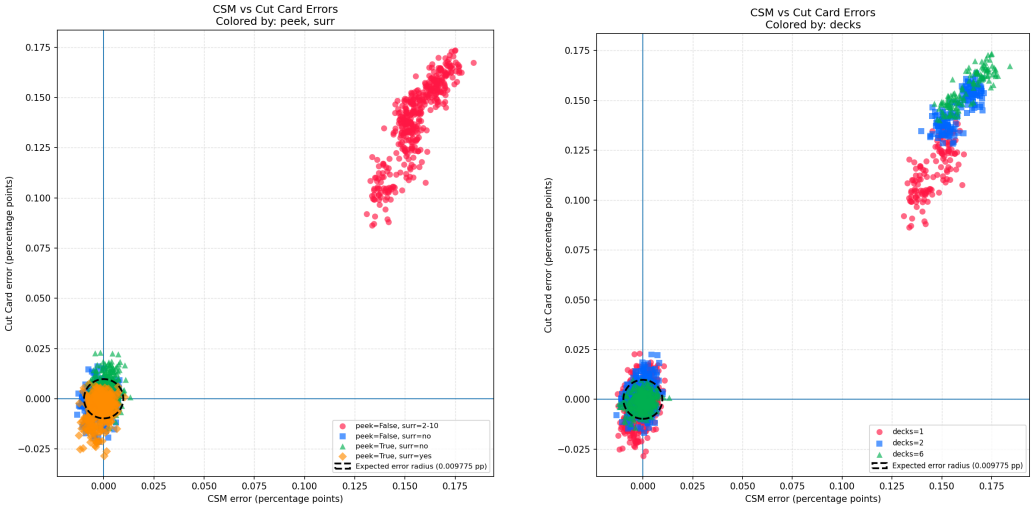
Using the 99% standard-normal critical value $z_{0.995} = 2.576$, the acceptance threshold was

$$\delta = 100 \times 2.576 \times \frac{1.2}{\sqrt{10^9}} \approx 0.0097752327$$

percentage points. Consequently, a regression result was accepted when

$$|\varepsilon| < 0.0097752327\%.$$

This criterion was applied separately to the cut-card and CSM estimates. Before excluding the unmatched no-peek-with-surrender configurations, 1005 of 3020 estimates exceed the bound (33.28%). The exceedances comprise $422/1510$ CSM estimates (27.95%) and $583/1510$ cut-card



(a) Peek and surrender.

(b) Number of decks.

Figure 1: CSM house-edge error versus cut-card house-edge error for the blackjack validation grid. The dashed circle marks the expected error radius of 0.0097752327 percentage points.

estimates (38.61%), accounting for 41.99% and 58.01% of all exceedances, respectively. By deck count, the rates are $401/1032$ for one deck (38.86%), $355/1028$ for two decks (34.53%), and $249/960$ for six decks (25.94%).

Approximately one quarter of the experiments use the unmatched no-peek-with-surrender combination discussed above. After removing this group, approximately 88.9% of the directly comparable estimates lie within the stated bound. For the CSM experiments alone, the exclusion reduces the exceedance rate from 27.95% to approximately 3.9%. The six-deck stratum illustrates the residual error particularly clearly: removing its 240 unmatched cases leaves only 9 exceedances among 720 estimates, an exceedance rate of 1.25%. The larger remaining discrepancies are concentrated in the cut-card and lower-deck settings, where the benchmark depends more strongly on the assumed shoe penetration. The use of a common $\sigma = 1.2$ for every ruleset may also make the nominal interval too narrow for some configurations. Despite these limitations, the results show strong overall agreement between the simulator and the theoretical benchmarks.

Figure 1 compares the paired CSM and cut-card errors under two colorings. Panel (a) isolates the unmatched no-peek-with-surrender cluster. Panel (b) shows that the residual dispersion is greatest for smaller shoes, consistent with sensitivity to the penetration convention.

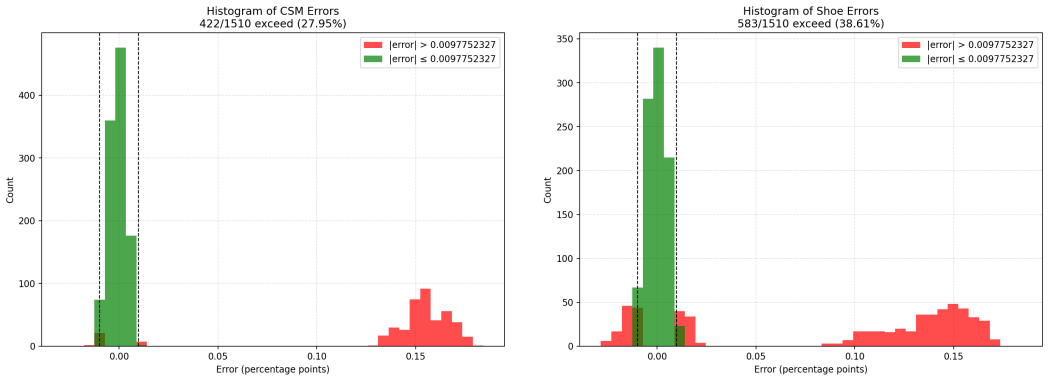
Figure 2 separates the marginal error distributions by shoe model. The higher cut-card exceedance rate and its wider central distribution are consistent with greater sensitivity to the penetration convention.

D.2 Q-Learning Policy Accuracy

The house-edge comparison above validates the game simulator, but it does not test the Q-learning update or the zombie representation used to learn split decisions. We therefore perform a separate policy-level validation against the basic-strategy tables published by the Wizard of Odds. For each evaluated ruleset, the greedy action of the trained no-count Q-table is compared state by state with the corresponding published basic-strategy action.

Let $\mathcal{S}_R$ be the set of decision states for ruleset R that can be matched unambiguously to an entry in the reference table, and let $a_Q(s)$ and $a_{\mathrm{ref}}(s)$ denote the learned and reference actions. The overall error set and accuracy are

$$\mathcal{E}_R = \{s \in \mathcal{S}_R : a_Q(s) \neq a_{\mathrm{ref}}(s)\}, \qquad \mathrm{Acc}_R = 100\left(1 - \frac{|\mathcal{E}_R|}{|\mathcal{S}_R|}\right).$$



(a) CSM errors.

(b) Cut-card shoe errors.

Figure 2: House-edge error distributions by shoe model. Green bars fall within ± 0.0097752327 percentage points, while red bars exceed this bound.

Table 3: Q-learning policy errors relative to the original reference table, grouped by deck count. Pair-state errors are shown separately as the targeted test of decisions whose split values use the zombie representation.

Decks	Shoe errors	Shoe pair errors	CSM errors	CSM pair errors
1	63/2,880 (2.188%)	26/800 (3.250%)	60/2,880 (2.083%)	25/800 (3.125%)
2	43/2,880 (1.493%)	12/800 (1.500%)	48/2,880 (1.667%)	15/800 (1.875%)
6	29/2,880 (1.007%)	3/800 (0.375%)	29/2,880 (1.007%)	5/800 (0.625%)
All	135/8,640 (1.563%)	41/2,400 (1.708%)	137/8,640 (1.586%)	45/2,400 (1.875%)

In addition to reporting the full mismatch list and overall accuracy for each ruleset, we evaluate pair states separately. With $\mathcal{S}_R^{\text{pair}} \subseteq \mathcal{S}_R$, define

$$\mathcal{E}_R^{\text{pair}} = \{s \in \mathcal{S}_R^{\text{pair}} : a_Q(s) \neq a_{\text{ref}}(s)\}, \qquad \text{Acc}_R^{\text{pair}} = 100\left(1 - \frac{|\mathcal{E}_R^{\text{pair}}|}{|\mathcal{S}_R^{\text{pair}}|}\right).$$

The comparison uses the 24 matching rule configurations for which surrender is enabled in both the cut-card shoe and CSM tests. Following the rule dimensions available in the Wizard of Odds basic-strategy calculator Wizard of Odds (2026d), these configurations form the Cartesian product

$$\{1, 2, 6\} \text{ decks} \times \{\text{S17}, \text{H17}\} \times \{\text{DAS}, \text{no DAS}\} \times \{\text{peek}, \text{no peek}\},$$

with surrender enabled throughout. Thus, each shoe model contains 24 configurations in total, with eight at each deck count. Each configuration contributes 360 evaluated decisions, including 100 pair decisions. Tables 3 and 4 report mismatches relative to the original reference table; percentages are mismatch rates rather than accuracy rates.

Across all configurations, the mismatch rate is 1.563% for the cut-card shoe and 1.586% for the CSM, corresponding to agreement rates of 98.437% and 98.414%. For pair states, the mismatch rates are 1.708% and 1.875%, corresponding to agreement rates of 98.292% and 98.125%. Errors decrease substantially with deck count, while the rule-subset breakdown shows broadly similar performance across H17/S17, DAS/no-DAS, and peek/no-peek conditions. The high pair-state agreement provides direct empirical support for the zombie split-hand update.

The reported mismatch rates also include errors in the reference data. In particular, for the pair $9, 9$ against a dealer ace, the original reference table used in this comparison prescribes splitting, whereas the Wizard of Odds hand calculator prescribes standing for the corresponding configuration Wizard of Odds (2026c). This inconsistency shows that the reference table is not a perfect ground truth; consequently, at least some reported mismatches may reflect reference errors rather than Q-learning errors.

Table 4: Q-learning policy errors relative to the original reference table, grouped by rule subset. Each row pools all deck counts and the configurations satisfying the indicated rule.

Rule subset	Shoe errors	Shoe pair errors	CSM errors	CSM pair errors
H17	72/4,320 (1.667%)	23/1,200 (1.917%)	73/4,320 (1.690%)	24/1,200 (2.000%)
S17	63/4,320 (1.458%)	18/1,200 (1.500%)	64/4,320 (1.481%)	21/1,200 (1.750%)
No DAS	68/4,320 (1.574%)	23/1,200 (1.917%)	72/4,320 (1.667%)	26/1,200 (2.167%)
DAS	67/4,320 (1.551%)	18/1,200 (1.500%)	65/4,320 (1.505%)	19/1,200 (1.583%)
No peek	64/4,320 (1.481%)	18/1,200 (1.500%)	64/4,320 (1.481%)	19/1,200 (1.583%)
Peek	71/4,320 (1.644%)	23/1,200 (1.917%)	73/4,320 (1.690%)	26/1,200 (2.167%)

E LOGARITHMIC-GROWTH Q-LEARNING

The main algorithm optimizes the playing policy using expected net return. Section 2.5 shows that this is the first-order approximation to logarithmic-growth optimization when the wager fraction is small. This appendix describes how the playing policy could instead be optimized directly with respect to logarithmic bankroll growth.

E.1 LOGARITHMIC WEALTH AS AN EPISODIC REWARD

Let W_t denote bankroll immediately before a blackjack round, let f denote the fraction of bankroll wagered, and let X denote the terminal net return measured in units of the wager. After the round,

$$W_{t+1} = W_t(1 + fX).$$

The logarithmic change in wealth is therefore

$$\Delta \log W_t = \log W_{t+1} - \log W_t = \log(1 + fX).$$

Consequently, maximizing the expected logarithmic growth rate is equivalent, on a per-round basis, to maximizing

$$\mathbb{E}\left[\log(1 + fX)\right].$$

This suggests defining the terminal reinforcement-learning reward as

$$R_{\log} = \log(1 + fX).$$

The reward is well defined whenever

$$1 + fX > 0.$$

Accordingly, the wagering rule must remain within the admissible region that prevents bankruptcy for all outcomes assigned positive probability. Appendix I describes a practical implementation based on bankroll-feasible actions and wager clipping.

E.2 Q-LEARNING UPDATE

Consider a finite blackjack episode consisting of states and actions

$$s_0, a_0, s_1, a_1, \ldots, s_T, a_T,$$

followed by terminal payoff X. Intermediate actions do not directly change bankroll; rather, their financial consequence is realized through the terminal payoff of the hand. The reward sequence can therefore be defined as

$$r_t = 0, \qquad t < T,$$

and

$$r_T = \log(1 + fX).$$

Standard Q-learning uses the update

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right],$$

where α is the learning rate and γ is the discount factor.

Because blackjack is a finite episodic problem and the objective does not discount rewards occurring later within the same hand, one may take

$$\gamma = 1.$$

For a nonterminal transition, the update becomes

$$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha \max_{a'} Q(s_{t+1}, a').$$

At the terminal transition,

$$Q(s_T, a_T) \leftarrow (1 - \alpha)Q(s_T, a_T) + \alpha \log(1 + fX).$$

Under the usual conditions for Q-learning convergence, the resulting action-value function estimates the optimal expected logarithmic reward,

$$Q^*_{\log}(s,a) = \sup_{\pi} \mathbb{E}\left[\log(1 + fX^{\pi}) \mid s, a\right],$$

with the expectation taken over subsequent game transitions and outcomes.

The corresponding growth-optimal policy is

$$\pi^*_{\log}(s) \in \arg\max_{a \in \mathcal{A}_R(s)} Q^*_{\log}(s,a).$$

E.3 Count-Dependent Wagering

Within the linear counting framework of Section 2.4, the wager fraction is

$$f = Y_w(\mathbf{d}) = w^{\top}\mathbf{d}.$$

The terminal reward can therefore be written directly as

$$R_{\log} = \log\left(1 + X^{\pi} w^{\top}\mathbf{d}\right).$$

For a fixed count vector w, Q-learning can thus optimize the playing policy directly according to the same logarithmic-growth objective used in the count optimization:

$$\pi^*_w \in \arg\max_{\pi \in \Pi_{R,w}} \mathbb{E}\left[\log\left(1 + X^{\pi} w^{\top}\mathbf{d}\right)\right].$$

In principle, this produces an exact alignment between the two conditional steps of the alternating optimization:

$$w_k \in \arg\max_{w} \mathbb{E}\left[\log\left(1 + X^{\pi_{k-1}} w^{\top}\mathbf{d}\right)\right],$$

$$\pi_k \in \arg\max_{\pi \in \Pi^R_{w_k}} \mathbb{E}\left[\log\left(1 + X^{\pi} w_k^{\top}\mathbf{d}\right)\right].$$

However, changing w changes both the information supplied to the policy and the transformation from terminal payoff to reinforcement-learning reward. Direct logarithmic-growth Q-learning therefore couples the policy-learning problem more tightly to the count optimization.

Split-hand caveat. The zombie-state update used for expected-return Q-learning cannot be transferred directly to logarithmic rewards. Let x_1 and x_2 be the terminal net returns of the two child hands created by a split, measured in units of the initial wager. Treating the children as separate zombie continuations would add their logarithmic rewards,

$$\log(1 + fx_1) + \log(1 + fx_2),$$

whereas the actual round-level change in log wealth is

$$\log(1 + fx_1 + fx_2).$$

These quantities are not equal: the sum of the two logarithms corresponds to the logarithm of $(1 + fx_1)(1 + fx_2)$, which introduces an artificial cross term $f^2 x_1 x_2$. Consequently, neither adding nor doubling a single child's logarithmic value gives the correct split reward. Although this discrepancy is second order in the wager fraction f and may therefore be small in the low-wager regime, it remains a deviation from full correctness and is important to state explicitly.

A possible, though less elegant, implementation is to buffer the original split state and action until every descendant hand has terminated. Once the combined split return $x_1 + x_2$ is known, the algorithm can update the split entry in the Q-table using the single round-level target $\log(1 + fx_1 + fx_2)$. This special delayed update preserves the correct growth objective but removes the simple local zombie-state update available under additive expected-return rewards.

F BALANCED AND CONSTRAINED L-GOC

F.1 DECK-REMOVAL SYMMETRY AND THE BALANCE CONDITION

This subsection gives two versions of the balancing principle. The first symmetry argument applies to an unnormalized running count, and the second applies after true-count normalization. In both cases, invariance in a direction that should not contain rank information forces the card weights to sum to zero. The balanced constrained solution is derived in Subsection F.3.

The relevant symmetry is the idea that adding the same amount to every rank should not create directional information about the composition of the shoe. In the unnormalized model, this is represented by the all-ranks direction $\mathbf{1}$, which changes the overall scale of the discard vector without favoring one rank over another. In the true-count model, the transformation also includes a depth correction: after the uniform shift, the vector is rescaled so that the effect of shoe depth is accounted for through the true-count normalization.

Throughout this subsection, let $\mathbf{c} \in \mathbb{R}^s$ denote the discarded-card vector over s ranks, let $\mathbf{1} \in \mathbb{R}^s$ be the all-ones vector, and write the signed algebraic sum explicitly as

$$\mathrm{sum}(v) := \mathbf{1}^T v = \sum_{i=1}^{s} v_i.$$

Thus $\mathrm{sum}(\mathbf{1}) = s$ and $\mathrm{sum}(w) = w^T\mathbf{1} = \sum_{i=1}^{s} w_i$. For example, if all physical ranks are tracked separately, then classical blackjack has $s = 13$, while Spanish 21 has $s = 12$ because the ten rank is removed.

Theorem 3 (Balancing without true-count conversion)**.** *Consider the local unnormalized linear response model*

$$L_0(\mathbf{c}) = b + w^T\mathbf{c}.$$

Let

$$\phi_\alpha(\mathbf{c}) = \mathbf{c} + \alpha\mathbf{1}$$

be the transformation that shifts the discarded-card vector equally in every rank. If the model is invariant under this transformation,

$$L_0(\phi_\alpha(\mathbf{c})) = L_0(\mathbf{c}),$$

then differentiating with respect to α at $\alpha = 0$ gives the infinitesimal symmetry condition. The transformation is an infinitesimal symmetry of L_0 at $\alpha = 0$ if and only if

$$\mathrm{sum}(w) = w^T\mathbf{1} = 0.$$

Equivalently, the components of the running-count weight vector sum to zero.

Proof. The assumed invariance is

$$L_0(\phi_\alpha(\mathbf{c})) = L_0(\mathbf{c}).$$

Differentiating with respect to α gives

$$0 = \frac{d}{d\alpha} L_0(\phi_\alpha(\mathbf{c})) = \nabla_{\mathbf{c}} L_0(\phi_\alpha(\mathbf{c}))^T \frac{d}{d\alpha}\phi_\alpha(\mathbf{c}).$$

Evaluating at $\alpha = 0$ gives the infinitesimal symmetry condition

$$\nabla_{\mathbf{c}} L_0(\mathbf{c})^T v(\mathbf{c}) = 0.$$

The infinitesimal generator of the transformation is

$$v(\mathbf{c}) = \left.\frac{d}{d\alpha}\phi_\alpha(\mathbf{c})\right|_{\alpha=0} = \mathbf{1}.$$

Since

$$\nabla_{\mathbf{c}} L_0(\mathbf{c}) = w,$$

we obtain

$$0 = w^T\mathbf{1} = \mathrm{sum}(w).$$

Thus infinitesimal invariance implies the count is balanced:

$$\boxed{\sum_{i=1}^{s} w_i = 0}.$$

Conversely, if $\mathrm{sum}(w) = 0$, then $\nabla_{\mathbf{c}} L_0(\mathbf{c})^T v(\mathbf{c}) = 0$, so the transformation is an infinitesimal symmetry. □

Theorem 4 (Balancing with true-count conversion)**.** *Let* $\mathbf{D}$ *denote the initial shoe-composition vector, so* $\mathrm{sum}(\mathbf{D})$ *is the initial number of cards. Consider the local true-count response model*

$$L(\mathbf{c}) = b + \frac{w^T \mathbf{c}}{\mathrm{sum}(\mathbf{D}) - \mathrm{sum}(\mathbf{c})}.$$

Define the one-parameter transformation

$$\begin{aligned}\phi_\alpha(\mathbf{c}) &= (\mathbf{c} + \alpha \mathbf{1}) \left(1 - \alpha \frac{\mathrm{sum}(\mathbf{1})}{\mathrm{sum}(\mathbf{D})}\right) \\ &= (\mathbf{c} + \alpha \mathbf{1}) \left(1 - \alpha \frac{s}{\mathrm{sum}(\mathbf{D})}\right).\end{aligned}$$

If the model is invariant under this transformation,

$$L(\phi_\alpha(\mathbf{c})) = L(\mathbf{c}),$$

then differentiating with respect to α *at* $\alpha = 0$ *gives the infinitesimal symmetry condition. For states with* $\mathrm{sum}(\mathbf{D}) - \mathrm{sum}(\mathbf{c}) > 0$*, this transformation is an infinitesimal symmetry of* L *at* $\alpha = 0$ *if and only if*

$$\mathrm{sum}(w) = w^T \mathbf{1} = 0.$$

Equivalently, the true-count weight vector must also be balanced.

Proof. The assumed invariance is

$$L(\phi_\alpha(\mathbf{c})) = L(\mathbf{c}).$$

Differentiating with respect to α gives

$$0 = \frac{d}{d\alpha} L(\phi_\alpha(\mathbf{c})) = \nabla_{\mathbf{c}} L(\phi_\alpha(\mathbf{c}))^T \frac{d}{d\alpha} \phi_\alpha(\mathbf{c}).$$

Evaluating at $\alpha = 0$ gives

$$\nabla_{\mathbf{c}} L(\mathbf{c})^T v(\mathbf{c}) = 0,$$

where

$$v(\mathbf{c}) = \left. \frac{d}{d\alpha} \phi_\alpha(\mathbf{c}) \right|_{\alpha=0}.$$

Expanding the transformation gives

$$\begin{aligned}\phi_\alpha(\mathbf{c}) &= (\mathbf{c} + \alpha \mathbf{1}) \left(1 - \alpha \frac{s}{\mathrm{sum}(\mathbf{D})}\right) \\ &= \mathbf{c} + \alpha \mathbf{1} - \alpha \frac{s}{\mathrm{sum}(\mathbf{D})} \mathbf{c} - \alpha^2 \frac{s}{\mathrm{sum}(\mathbf{D})} \mathbf{1}.\end{aligned}$$

Therefore,

$$v(\mathbf{c}) = \mathbf{1} - \frac{s}{\mathrm{sum}(\mathbf{D})} \mathbf{c}.$$

Let

$$A = \mathrm{sum}(\mathbf{D}) - \mathrm{sum}(\mathbf{c}).$$

Then

$$L(\mathbf{c}) = b + \frac{w^T \mathbf{c}}{A}.$$

Since $\nabla_{\mathbf{c}} A = -\mathbf{1}$, differentiating gives

$$\nabla_{\mathbf{c}} L(\mathbf{c}) = \frac{w}{A} + \frac{w^T \mathbf{c}}{A^2}\mathbf{1}.$$

Substituting the gradient and generator into the infinitesimal symmetry condition yields

$$\left(\frac{w}{A} + \frac{w^T \mathbf{c}}{A^2}\mathbf{1}\right)^T \left(\mathbf{1} - \frac{s}{\mathrm{sum}(\mathbf{D})}\mathbf{c}\right) = 0.$$

Expanding the left-hand side gives

$$\frac{1}{A} w^T \left(\mathbf{1} - \frac{s}{\mathrm{sum}(\mathbf{D})}\mathbf{c}\right) + \frac{w^T \mathbf{c}}{A^2}\mathbf{1}^T \left(\mathbf{1} - \frac{s}{\mathrm{sum}(\mathbf{D})}\mathbf{c}\right).$$

The first inner product is

$$w^T \left(\mathbf{1} - \frac{s}{\mathrm{sum}(\mathbf{D})}\mathbf{c}\right) = \mathrm{sum}(w) - \frac{s}{\mathrm{sum}(\mathbf{D})} w^T \mathbf{c}.$$

The second inner product is

$$\begin{aligned}
\mathbf{1}^T \left(\mathbf{1} - \frac{s}{\mathrm{sum}(\mathbf{D})}\mathbf{c}\right) &= s - \frac{s}{\mathrm{sum}(\mathbf{D})} \mathrm{sum}(\mathbf{c}) \\
&= \frac{s}{\mathrm{sum}(\mathbf{D})} \left(\mathrm{sum}(\mathbf{D}) - \mathrm{sum}(\mathbf{c})\right) \\
&= \frac{s}{\mathrm{sum}(\mathbf{D})} A.
\end{aligned}$$

Hence

$$\begin{aligned}
\nabla_{\mathbf{c}} L(\mathbf{c})^T v(\mathbf{c}) &= \frac{1}{A}\left(\mathrm{sum}(w) - \frac{s}{\mathrm{sum}(\mathbf{D})} w^T \mathbf{c}\right) + \frac{s\, w^T \mathbf{c}}{\mathrm{sum}(\mathbf{D}) A} \\
&= \frac{\mathrm{sum}(w)}{A} - \frac{s\, w^T \mathbf{c}}{\mathrm{sum}(\mathbf{D}) A} + \frac{s\, w^T \mathbf{c}}{\mathrm{sum}(\mathbf{D}) A} \\
&= \frac{\mathrm{sum}(w)}{A}.
\end{aligned}$$

The symmetry condition therefore implies

$$0 = \frac{\mathrm{sum}(w)}{\mathrm{sum}(\mathbf{D}) - \mathrm{sum}(\mathbf{c})},$$

so

$$\mathrm{sum}(w) = 0.$$

Conversely, if $\mathrm{sum}(w) = 0$, then $\nabla_{\mathbf{c}} L(\mathbf{c})^T v(\mathbf{c}) = 0$, so the transformation is an infinitesimal symmetry. Equivalently,

$$\boxed{\sum_{i=1}^{s} w_i = 0}.$$

□

The normalization in Theorem 4 uses total remaining cards rather than decks remaining. This is without loss of generality: if decks remaining are defined as $(\mathrm{sum}(\mathbf{D}) - \mathrm{sum}(\mathbf{c}))/\beta$ for a fixed conversion factor β, then dividing by decks remaining only multiplies the numerator by β, and that constant can be absorbed into the weight vector by replacing w with $\tilde{w} = \beta w$.

Both theorems rely on the approximation that the all-ranks direction contains no strategically relevant rank information, and this approximation can fail in an extremely shallow shoe. When very few cards remain, the discard composition may reveal the exact or nearly exact remaining cards, so even a nominally uniform change can alter the optimal decision. Theorem 3 has an additional limitation: without normalization, a uniform change also alters the magnitude of the response simply by changing shoe depth. This scale effect is small relative to composition effects in a heavy shoe but can become substantial near the end of the shoe. Theorem 4 corrects this second issue through normalization by $\mathrm{sum}(\mathbf{D}) - \mathrm{sum}(\mathbf{c})$, but it does not eliminate the first, decision-theoretic limitation caused by near-complete information in a very shallow shoe.

Table 5: L-GOC formulations used to evaluate the effect of the balance and fixed-bias constraints.

Variant	Balance constraint	Bias	Description
L-GOC-U	None	Free	Unconstrained
L-GOC-S0	$\mathbf{1}^\top w = 0$	Free	Balanced
L-GOC-S0-B1	$\mathbf{1}^\top w = 0$	$b = b_1$	Balanced, fixed bias

Table 6: Ablation of the L-GOC variants. Each numerical entry reports the estimate $\pm$ its pointwise 95% confidence bound. Improvements are measured relative to Hi-Lo with the Illustrious 18 (I18) and Hi-Lo with full deviations (FD). Bold values indicate the largest point estimate for each deck and column.

		Growth rate			Spread edge		
Decks	Variant	Value	Δ_{I18} (%)	Δ_{FD} (%)	Value	Δ_{I18} (%)	Δ_{FD} (%)
1	L-GOC-U	2.007214 $\pm$0.011957	27.76 $\pm$1.15	10.59 $\pm$0.94	0.111163 $\pm$0.000221	21.39 $\pm$0.35	9.09 $\pm$0.31
1	L-GOC-S0	2.056782 $\pm$0.012192	30.92 $\pm$1.17	13.32 $\pm$0.96	**0.111640** $\pm$0.000254	**21.91** $\pm$0.38	**9.56** $\pm$0.33
1	L-GOC-S0-B1	**2.061267** $\pm$0.012179	**31.20** $\pm$1.18	**13.57** $\pm$0.96	0.110757 $\pm$0.000246	20.95 $\pm$0.37	8.70 $\pm$0.32
2	L-GOC-U	0.832690 $\pm$0.007588	21.38 $\pm$1.67	12.85 $\pm$1.52	**0.060172** $\pm$0.000203	**15.12** $\pm$0.60	**6.69** $\pm$0.53
2	L-GOC-S0	**0.833096** $\pm$0.007778	**21.44** $\pm$1.69	**12.91** $\pm$1.54	0.060050 $\pm$0.000208	14.88 $\pm$0.60	6.48 $\pm$0.54
2	L-GOC-S0-B1	0.832416 $\pm$0.008215	21.34 $\pm$1.73	12.82 $\pm$1.58	0.059787 $\pm$0.000201	14.38 $\pm$0.59	6.01 $\pm$0.53
6	L-GOC-U	0.178588 $\pm$0.003540	18.40 $\pm$3.52	13.87 $\pm$3.54	0.017895 $\pm$0.000168	28.15 $\pm$2.15	25.87 $\pm$2.09
6	L-GOC-S0	0.180274 $\pm$0.003454	19.51 $\pm$3.50	14.94 $\pm$3.53	0.017779 $\pm$0.000168	27.32 $\pm$2.15	25.06 $\pm$2.09
6	L-GOC-S0-B1	**0.181492** $\pm$0.003580	**20.32** $\pm$3.57	**15.72** $\pm$3.59	**0.017919** $\pm$0.000165	**28.32** $\pm$2.15	**26.04** $\pm$2.08
Mean improvement: L-GOC-U		-	22.51 $\pm$1.35	12.44 $\pm$1.32	-	**21.55** $\pm$0.75	**13.88** $\pm$0.72
Mean improvement: L-GOC-S0		-	23.96 $\pm$1.35	13.72 $\pm$1.32	-	21.37 $\pm$0.75	13.70 $\pm$0.73
Mean improvement: L-GOC-S0-B1		-	**24.29** $\pm$1.38	**14.03** $\pm$1.35	-	21.22 $\pm$0.75	13.58 $\pm$0.72

F.2 Constrained L-GOC Formulations

L-GOC can be implemented with different levels of constraint. Three formulations are considered.

The first, denoted *L-GOC-U*, is fully unconstrained and allows both the bias and rank weights to adapt to the payoff distribution.

The second, denoted *L-GOC-S0*, imposes the balance condition

$$\mathbf{1}^\top w = 0,$$

while allowing the bias to be re-estimated.

The third, denoted *L-GOC-S0-B1*, additionally fixes the bias to the value b_1 obtained from the initial optimization:

$$\mathbf{1}^\top w = 0, \qquad b = b_1.$$

These formulations are summarized in Table 5.

L-GOC-U provides the direct unconstrained approximation derived in Section 2.6. L-GOC-S0 isolates the effect of enforcing the deck-removal symmetry, while L-GOC-S0-B1 additionally tests the effect of holding the initial intercept fixed. The final alternating algorithm uses L-GOC-S0 because it is more stable across the broader rule scenarios considered in this study; L-GOC-S0-B1 is retained here only as an ablation variant.

The empirical effect of these three choices is reported in Table 6.

For a fixed reference counting system, a *full-deviation* baseline is obtained by training a count-conditioned Q-table until convergence, using the same policy-training hyperparameters and convergence criterion as L-GOC (Appendix G.1). Thus, unlike a compact index set such as the Illustrious 18, the learned policy may deviate from basic strategy in any represented decision state.

Table 6 compares the L-GOC variants against both the Illustrious 18 and full-deviation baselines. Each variant and deck count is evaluated using both growth rate and spread edge.

Table 7: Ablation of the L-GOC variants under a 6:5 blackjack payout. Each metric is reported with its pointwise 95% confidence bound, and improvements are relative percentages. Bold values indicate the largest selected point estimate for each deck and metric.

		Growth rate			Spread edge		
Decks	Variant	Value	Δ_{I18} (%)	Δ_{FD} (%)	Value	Δ_{I18} (%)	Δ_{FD} (%)
1	L-GOC-U	0.867730 ±0.008070	59.75 ±2.37	14.89 ±1.58	0.043000 ±0.000167	74.29 ±1.19	24.62 ±0.71
1	L-GOC-S0	**0.961310** ±0.008390	**76.98** ±2.57	**27.28** ±1.70	**0.045496** ±0.000171	**84.41** ±1.24	**31.85** ±0.74
1	L-GOC-S0-B1	0.924950 ±0.008370	70.28 ±2.51	22.46 ±1.66	0.044128 ±0.000158	78.87 ±1.19	27.89 ±0.71
2	L-GOC-U	0.280780 ±0.004300	59.79 ±4.27	27.98 ±2.90	0.007561 ±0.000136	5150.88 ±4486.12	96.35 ±7.33
2	L-GOC-S0	**0.288340** ±0.004730	**64.09** ±4.49	**31.43** ±3.08	**0.007970** ±0.000134	**5434.58** ±4728.37	**106.95** ±7.61
2	L-GOC-S0-B1	0.281450 ±0.004040	60.17 ±4.20	28.29 ±2.82	0.007843 ±0.000127	5346.87 ±4653.37	103.67 ±7.44
6	L-GOC-U	0.028700 ±0.001850	56.50 ±14.31	37.49 ±11.16	**-0.014888** ±0.000083	**11.37** ±0.67	**8.10** ±0.71
6	L-GOC-S0	**0.030290** ±0.001390	**65.15** ±13.13	**45.10** ±9.78	-0.014908 ±0.000081	11.26 ±0.66	7.98 ±0.70
6	L-GOC-S0-B1	0.029220 ±0.001360	59.30 ±12.72	39.96 ±9.50	-0.014973 ±0.000080	10.87 ±0.66	7.58 ±0.70
Mean improvement: L-GOC-U		-	58.68 ±5.04	26.79 ±3.88	-	1745.52 ±1495.37	43.02 ±2.47
Mean improvement: L-GOC-S0		-	**68.74** ±4.70	**34.60** ±3.46	-	**1843.42** ±1576.12	**48.93** ±2.56
Mean improvement: L-GOC-S0-B1		-	63.25 ±4.54	30.24 ±3.35	-	1812.20 ±1551.12	46.38 ±2.50

Table 8: Results for one- and six-deck Spanish 21. Growth is reported as $10^4(G-1)$, and each raw metric is given as the mean ± its 95% confidence-interval half-width. For each baseline row, the delta columns report the relative improvement achieved by L-GOC-S0. Bold values indicate the largest point estimate for each deck and metric.

Decks	Method	Growth rate, $10^4(G-1)$	1:10 spread edge	$\Delta(G-1)$ (%)	Δ spread (%)
1	Walker + published deviations	0.761220 ±0.008210	0.043019 ±0.000117	209.6	181.5
1	Walker + full deviations	1.915050 ±0.011720	0.079523 ±0.000125	23.1	52.3
1	Hi-Lo + full deviations	1.929130 ±0.012500	0.106535 ±0.000256	22.2	13.7
1	L-GOC-S0 W_4	**2.356550** ±0.012850	**0.121087** ±0.000281	—	—
6	Walker + published deviations	0.043150 ±0.001840	-0.001804 ±0.000065	347.6	Negative → positive
6	Walker + full deviations	0.047500 ±0.001860	0.000638 ±0.000065	306.6	2659.9
6	Hi-Lo + full deviations	-0.067130 ±0.003650	-0.040305 ±0.000299	Loss → growth	Negative → positive
6	L-GOC-S0 W_4	**0.193150** ±0.003630	**0.017608** ±0.000167	—	—

No single constraint variant dominates every deck count and metric. L-GOC-S0-B1 produces the largest mean growth-rate improvements in this specific 3:2 ablation, while L-GOC-U has the largest mean spread improvements. L-GOC-S0 leads the single-deck spread result and the two-deck growth-rate result. Because L-GOC-S0 is more stable across the broader rule scenarios, it is used in the body of the paper and in the final algorithm.

Table 7 provides the corresponding ablation for blackjack paying 6:5. The implementation names L-GOC-U, L-GOC-S0, and L-GOC-S0-B1 correspond to L-GOC-U, L-GOC-S0, and L-GOC-S0-B1, respectively. Each family is represented by its available iteration with the largest estimated Kelly growth.

L-GOC-S0 has the largest mean Kelly improvement, $68.74\% \pm 4.70\%$ relative to I18 and $34.60\% \pm 3.46\%$ relative to full deviations. It also has the largest mean relative spread improvements, although the I18 mean is dominated by the two-deck result. L-GOC-S0 leads Kelly growth for every deck count and spread edge for one and two decks, while L-GOC-U has the largest six-deck spread edge. Because the two-deck I18 spread is only 0.000144, the corresponding relative spread improvements and their uncertainty are numerically very large and should be interpreted together with the raw spread values.

Table 8 reports the current comparisons, including the single-deck setting for mathematical completeness. Percentage improvements use $100(m_{\text{L-GOC}} - m_{\text{baseline}})/m_{\text{baseline}}$ for positive baseline metrics, with $m = G - 1$ or spread edge; sign changes are reported explicitly instead. These deltas are point estimates without propagated confidence bounds. L-GOC-S0 W_4 has the largest growth and spread estimates in both deck settings.

F.3 Derivation of the Balanced S0 Solution

This subsection derives the closed-form approximation used by L-GOC-S0. For a fixed playing policy π, the bias and rank weights are optimized jointly through

$$J_\pi(b, w) = \mathbb{E}\left[\log\left(1 + X^\pi\left(b + w^\top \mathbf{d}\right)\right)\right],$$

subject to the balance constraint

$$\mathbf{1}^\top w = 0.$$

Differentiating with respect to w gives

$$\nabla_w J_\pi(b, w) = \mathbb{E}\left[\frac{X^\pi \mathbf{d}}{1 + X^\pi(b + w^\top \mathbf{d})}\right].$$

When

$$\left|X^\pi(b + w^\top \mathbf{d})\right| \ll 1,$$

the reciprocal can be approximated by

$$\frac{1}{1 + X^\pi(b + w^\top \mathbf{d})} \approx 1 - X^\pi(b + w^\top \mathbf{d}).$$

Hence,

$$\nabla_w J_\pi(b, w) \approx \mathbb{E}\left[X^\pi \mathbf{d} - (X^\pi)^2 \mathbf{d}\left(b + w^\top \mathbf{d}\right)\right].$$

Expanding the expectation yields

$$\begin{aligned}\nabla_w J_\pi(w; b) \approx\ & \mathbb{E}[X^\pi \mathbf{d}] - b\,\mathbb{E}[(X^\pi)^2 \mathbf{d}] \\ & - \mathbb{E}\left[(X^\pi)^2 \mathbf{d}\mathbf{d}^\top\right] w.\end{aligned}$$

Define

$$A_\pi = \mathbb{E}\left[(X^\pi)^2 \mathbf{d}\mathbf{d}^\top\right]$$

and

$$q_\pi(b) = \mathbb{E}\left[X^\pi \mathbf{d} - b(X^\pi)^2 \mathbf{d}\right].$$

For a fixed value of b, the approximate first-order condition with respect to w is therefore

$$q_\pi(b) - A_\pi w = \mathbf{0}.$$

To impose the balance condition, introduce the Lagrangian

$$\mathcal{L}(b, w, \lambda) = J_\pi(b, w) - \lambda\, \mathbf{1}^\top w.$$

Under the same approximation, stationarity with respect to w gives

$$q_\pi(b) - A_\pi w - \lambda \mathbf{1} = \mathbf{0}.$$

Therefore,

$$A_\pi w = q_\pi(b) - \lambda \mathbf{1},$$

and, assuming A_π is nonsingular,

$$w = A_\pi^{-1}\left(q_\pi(b) - \lambda \mathbf{1}\right).$$

The balance constraint requires

$$\mathbf{1}^\top w = 0.$$

Substituting the expression for w gives

$$\mathbf{1}^\top A_\pi^{-1}\left(q_\pi(b) - \lambda \mathbf{1}\right) = 0,$$

and hence

$$\lambda = \frac{\mathbf{1}^\top A_\pi^{-1} q_\pi(b)}{\mathbf{1}^\top A_\pi^{-1} \mathbf{1}}.$$

Substituting this value back into the expression for w gives

$$w_\pi^* \approx A_\pi^{-1}\left[q_\pi(b) - \frac{\mathbf{1}^\top A_\pi^{-1} q_\pi(b)}{\mathbf{1}^\top A_\pi^{-1} \mathbf{1}} \mathbf{1}\right].$$

For any candidate bias b, substituting the expectations explicitly gives

$$w_\pi^* \approx \left(\mathbb{E}\left[(X^\pi)^2\mathbf{d}\mathbf{d}^\top\right]\right)^{-1} \times \left[\mathbb{E}\left[X^\pi\mathbf{d} - b(X^\pi)^2\mathbf{d}\right] - \lambda\mathbf{1}\right],$$

where

$$\lambda = \frac{\mathbf{1}^\top\left(\mathbb{E}\left[(X^\pi)^2\mathbf{d}\mathbf{d}^\top\right]\right)^{-1}\mathbb{E}\left[X^\pi\mathbf{d} - b(X^\pi)^2\mathbf{d}\right]}{\mathbf{1}^\top\left(\mathbb{E}\left[(X^\pi)^2\mathbf{d}\mathbf{d}^\top\right]\right)^{-1}\mathbf{1}}.$$

The free bias is determined by the additional stationarity condition

$$\frac{\partial J_\pi}{\partial b} = \mathbb{E}\left[\frac{X^\pi}{1 + X^\pi(b + w^\top\mathbf{d})}\right] \approx \mathbb{E}[X^\pi] - \mathbb{E}\left[(X^\pi)^2(b + w^\top\mathbf{d})\right] = 0.$$

To express the joint solution compactly, define

$$\mathbf{z} = \begin{bmatrix}1\\ \mathbf{d}\end{bmatrix}, \qquad \theta = \begin{bmatrix}b\\ w\end{bmatrix}, \qquad c = \begin{bmatrix}0\\ \mathbf{1}_p\end{bmatrix}.$$

Here $\mathbf{1}_p \in \mathbb{R}^p$ is the all-ones vector, so the lower block has the same dimension as w. Then the balance constraint is $c^\top\theta = 0$. Let

$$\overline{A}_\pi = \mathbb{E}\left[(X^\pi)^2\mathbf{z}\mathbf{z}^\top\right], \qquad \overline{q}_\pi = \mathbb{E}[X^\pi\mathbf{z}].$$

Under the first-order approximation, the constrained stationarity equations are

$$\overline{A}_\pi\theta = \overline{q}_\pi - \lambda c, \qquad c^\top\theta = 0.$$

Assuming $\overline{A}_\pi$ is nonsingular, the balanced S0 solution is

$$\theta_\pi^* \approx \overline{A}_\pi^{-1}\left[\overline{q}_\pi - \frac{c^\top\overline{A}_\pi^{-1}\overline{q}_\pi}{c^\top\overline{A}_\pi^{-1}c}c\right].$$

Thus each alternating count update re-estimates both b and w, with the single structural restriction $\mathbf{1}^\top w = 0$.

Table 9: Default reinforcement-learning hyperparameters used throughout the experiments.

Hyperparameter	Value
Algorithm	Tabular Q-learning
Exploration	Epsilon-greedy
Initial epsilon	1.0
Minimum epsilon	0.1
Epsilon decay	0.99999 per state visit
Initial learning rate α	0.01
Minimum learning rate	0.0001
Learning-rate decay parameter N	100
Discount factor γ	1.0
Count-state resolution	1
Count-state range	$[-5, 5]$
Stopping mode	Q-table difference
Convergence check interval	100,000,000 rounds
Convergence threshold	0.005
Main-suite threads	5
Reproducibility-suite threads	10
Alternating iterations	3

G Reproducibility of L-GOC-S0

G.1 Hyperparameter Selection

Unless stated otherwise, every policy phase in the alternating-optimization experiments and every full-deviation baseline uses the tabular Q-learning configuration in Table 9. The values were selected through preliminary empirical trials to balance exploration, convergence stability, state coverage, and computational cost. Tabular Q-learning was used because the represented blackjack state space is finite and discrete. Epsilon-greedy exploration begins at 1 to cover the action space broadly and retains a minimum of 0.1 to prevent rarely visited states from becoming permanently deterministic. The undiscounted choice $\gamma = 1$ matches the finite episodic objective, in which rewards are evaluated at the end of a round.

The integer count resolution matches the policy representation used throughout the experiments. Restricting count states to $[-5, 5]$ limits sparsity in the Q-table while retaining the count range most frequently encountered in training. The convergence interval and threshold were selected as a practical compromise: checking every 10^8 rounds makes the diagnostic inexpensive relative to training, while the 0.005 mean absolute Q-table-change threshold was found to provide stable learned policies. Thread counts affect simulation throughput rather than the learning objective.

The learning rate decays independently for each state. If n denotes its state-visit counter, the implemented schedule is

$$\alpha_n = \max\left(\frac{100}{n + 100}, 0.0001\right),$$

with the counter initialized so that the first applied learning rate is 0.01. Three alternating iterations were selected empirically because preliminary trials showed that the learned weights and policies stabilized by $k = 3$ and did not improve materially afterward. The Spanish 21 experiments use $k = 4$ as an additional convergence check.

Recall that tabular Q-learning converges to the optimal action-value function Q^* under standard assumptions, including sufficient exploration and state–action learning rates satisfying the Robbins–Monro conditions

$$\sum_{t=1}^{\infty} \alpha_t(s, a) = \infty, \qquad \sum_{t=1}^{\infty} \alpha_t^2(s, a) < \infty,$$

for every state–action pair (s, a) (Watkins & Dayan, 1992). The unfloored term $100/(n + 100)$ has the required harmonic decay and therefore follows this classical convergence prescription. The implemented lower bound of 0.0001 is a finite-training safeguard; if retained indefinitely, it prevents the second condition from holding asymptotically. Thus, the Robbins–Monro result motivates

the decay schedule but does not constitute a literal infinite-horizon convergence guarantee for the clipped implementation.

G.2 Experimental Design

We evaluated whether independent executions of alternating optimization converge to equivalent final count systems and playing policies. The experiment covered all six combinations of blackjack payout in $\{3{:}2, 6{:}5\}$ and deck count in $\{1, 2, 6\}$. Each configuration was trained independently five times, giving 30 completed runs. No random seed was fixed, so each repetition was a fresh stochastic execution rather than a bit-for-bit replay of the same random sequence.

All runs used American H17 blackjack with 75% penetration, dealer peek, no surrender, double after split, at most four hands after splitting, doubling on any initial two-card total, no resplitting aces, and no hitting split aces. Each run performed three alternating iterations,

$$P_0 \to W_1 \to P_1 \to W_2 \to P_2 \to W_3.$$

Each W phase retained five billion regression observations. Because one observation was retained every five rounds, a W phase simulated 25 billion played rounds. Each P phase used Q-learning in batches of 100 million rounds and continued until the average absolute change in the Q-table between consecutive samples was at most 0.005.

G.3 Data and Comparison Objects

Each run produced artifacts for all three count-fitting and policy-learning phases. The reproducibility analysis uses the final outputs: `W3.json`, containing the final normalized 13-element count-weight vector, bias, and factor; and `P2_agent.json`, containing the final Q-table and the number of completed training rounds. The 13 count weights correspond, in order, to

$$2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A.$$

The Q-table maps each observable state—defined by the count bucket, hand type, player total, and dealer upcard—to Q-values for the actions available in that state.

The runs also included five-billion-round performance evaluations, 500-million-round EV/count evaluations, and Kelly measurements comprising 1,000 repetitions of one million rounds. Those measurements are not used in the weight- and policy-reproducibility metrics reported here.

G.4 Weight Reproducibility Metrics

For each configuration, five final W_3 vectors yield $\binom{5}{2} = 10$ distinct run pairs. All vectors use the same normalization and can therefore be compared directly. For a pair of final weight vectors w_i and w_j, cosine similarity is

$$C_{ij} = \frac{w_i^\top w_j}{\|w_i\|_2 \|w_j\|_2}.$$

It measures agreement in the relative weight pattern across ranks, with one denoting identical direction. We report the least similar pair,

$$C_{\text{worst}} = \min_{i<j} C_{ij},$$

rather than the average, providing a conservative measure of directional consistency.

For the same pair, the per-rank weight RMSE is

$$\text{RMSE}_{ij}^{W} = \sqrt{\frac{1}{13}\sum_{k=1}^{13}(w_{ik} - w_{jk})^2}.$$

We report the largest distance among the ten pairs,

$$\text{RMSE}_{\text{worst}}^{W} = \max_{i<j} \text{RMSE}_{ij}^{W}\ .$$

The worst cosine similarity and worst RMSE need not be attained by the same pair: cosine similarity measures direction, whereas RMSE measures absolute numerical distance.

Table 10: Reproducibility across five independent L-GOC-S0 runs per configuration. Weight and Q-table columns report the least favorable result among the ten run pairs; policy columns compare states present in all five final Q-tables.

Payout	Decks	Worst W_3 cosine	Worst W_3 weight RMSE	Worst P_2 Q RMSE	Unanimous $5/5$	At least $4/5$
3:2	1	0.999918	0.011637	0.017109	97.12%	98.71%
3:2	2	0.999976	0.008502	0.015907	95.71%	97.67%
3:2	6	0.999861	0.017051	0.016648	95.27%	97.66%
6:5	1	0.999949	0.008832	0.018682	96.99%	98.84%
6:5	2	0.999950	0.008896	0.015801	95.44%	97.63%
6:5	6	0.999808	0.019010	0.016617	94.93%	97.44%

G.5 Q-Table and Policy Reproducibility Metrics

For each pair of final P_2 Q-tables, numerical comparison is restricted to state–action entries appearing in both tables. If E_{ij} denotes this common set, then

$$\mathrm{RMSE}_{ij}^{Q} = \sqrt{\frac{1}{|E_{ij}|}\sum_{e\in E_{ij}}\big(Q_i(e)-Q_j(e)\big)^2}, \qquad \mathrm{RMSE}_{\mathrm{worst}}^{Q} = \max_{i<j}\mathrm{RMSE}_{ij}^{Q}\,.$$

This is the largest numerical Q-table difference across the ten pairs. Q-table RMSE does not by itself determine policy agreement: small value differences may leave the maximizing action unchanged, or may reverse two nearly tied actions. We therefore also compare greedy actions directly.

For every state present in all five final Q-tables, define the greedy action in run r as

$$a_r(s) = \arg\max_a Q_r(s,a).$$

Unanimous agreement is the percentage of these states satisfying

$$a_1(s) = a_2(s) = a_3(s) = a_4(s) = a_5(s).$$

We also report the percentage for which at least four of the five runs choose the same greedy action. This second measure includes unanimous states while allowing one stochastic repetition to differ. No exact maximum-Q ties occurred in the final P_2 tables, so every evaluated greedy action was unambiguous.

G.6 Results

The worst cosine similarity exceeds 0.9998 in every configuration, showing that even the least-similar repetitions learn practically identical weight-vector directions. The largest weight RMSE is 0.019010; relative to normalized weights with characteristic magnitude near one, this is a root-mean-square per-rank difference below 2%.

The final Q-tables also show small numerical differences, with worst-case RMSE at most 0.018682. More importantly, these differences seldom change the induced policies. All five repetitions select the same greedy action in 94.93–97.12% of evaluated states, and at least four agree in 97.44–98.84% of states.

The remaining policy disagreements may be concentrated in infrequently visited states or in states whose two leading actions have nearly equal Q-values, where a small numerical perturbation can change the maximizer. This interpretation was not tested directly; verifying it would require stratifying disagreements by state visitation frequency and by the gap between the two largest Q-values. Overall, the independently trained repetitions produce effectively equivalent count systems and playing policies.

Table 11: Euclidean distances between successive L-GOC-S0 weight vectors across deck counts and payouts.

Payout	Decks	$\|W_2 - W_1\|_2$	$\|W_3 - W_2\|_2$	Distance ratio	Contraction
3:2	1	0.251149	0.035201	0.1402	85.98%
3:2	2	0.179664	0.024602	0.1369	86.31%
3:2	6	0.092005	0.029933	0.3253	67.47%
6:5	1	0.232573	0.073457	0.3158	68.42%
6:5	2	0.172507	0.024863	0.1441	85.59%
6:5	6	0.151028	0.028146	0.1864	81.36%

H Deck Count and Alternating-Optimization Coupling

Table 11 measures changes between successive L-GOC-S0 weight vectors. Distances use the 13 unrounded rank weights and exclude the bias and the null initialization W_0. The distance ratio is $\|W_3 - W_2\|_2 / \|W_2 - W_1\|_2$, and contraction is one minus this ratio.

For both payouts, $\|W_2 - W_1\|_2$ decreases as the deck count increases, consistent with stronger coupling in shallower shoes. The second update is smaller in every setting, with contractions between 67.47% and 86.31%. These distances are computed from single fitted vectors; independent replicate fits or covariance estimates were not retained, so confidence intervals are unavailable.

Table 12: Diagnostic effect of clipping the maximum initial wager fraction in one-deck 3:2 blackjack.

Metric	$c = 0.10$	$c = 0.25$	$c = 0.50$	$c = 0.75$	$c = 1.00$ reference
Growth Rate	0.999893844052	1.000106020180	1.000172802874	1.000178108521	1.000178304910
Δ growth vs. $c = 1$, 10^{-6}	-284.461	-72.285	-5.502	-0.196	0

I Admissibility, Exposure, and Wager Clipping

Let $f = b + w^\top \mathbf{d}$ denote the nominal initial wager fraction. The logarithmic-growth objective requires $1 + X^\pi f > 0$ almost surely, whereas the regression approximation additionally assumes that

$$|X^\pi f| \ll 1$$

over the region carrying most of the probability mass. These statements have different status. The first defines the exact admissible domain, whereas the second is a local approximation whose accuracy depends on realized wager exposure. The experiments reported elsewhere in the paper use $\eta = 0$ and $c = 1$, with no positive safety margin or diagnostic clipping. This appendix examines stricter bankroll feasibility and clipping only to test whether enforcing lower exposure would improve realized logarithmic growth.

I.1 Bankroll-Feasible Splits and Doubles

Even when the initial wager is a moderate fraction of bankroll, subsequent actions can increase the total amount at risk. Under a four-hand split limit, splitting to four hands and doubling every hand can produce a total commitment equal to eight times the initial wager. Thus an unrestricted action policy can, in theory, select a legal game action that violates the positivity condition.

Strict positivity could be enforced by making the action set bankroll dependent. Let B be bankroll before the round, let C_t be the amount already committed at decision time t, and let $L(a)$ be the largest additional loss made possible by action a. For a fixed safety margin $\eta > 0$, replace the ruleset action set by

$$\mathcal{A}_{R,B}(s_t, C_t) = \{a \in \mathcal{A}_R(s_t) : C_t + L(a) < (1 - \eta)B\}.$$

A split or double is then unavailable whenever funding it could reduce terminal wealth below ηB. For $\eta > 0$, the worst possible loss remains strictly smaller than bankroll, guaranteeing positive wealth for every terminal outcome. The reported experiments instead use $\eta = 0$; the positive-margin formulation is presented here as a possible strict safeguard rather than as part of the evaluated method.

A separate diagnostic clips the initial wager fraction. For a selected cap c, define

$$f_c = \min\{\max\{f, 0\}, c\}.$$

Only the clipping study below varies c; all other reported experiments use the $c = 1$ reference. The purpose is to test whether limiting the high-exposure tail, where the local approximation is least accurate, improves the exact logarithmic-growth objective.

I.2 Effect of Clipping on Growth

The clipping diagnostic uses one-deck 3:2 blackjack under the reference American rules, the most composition-volatile blackjack configuration studied here. The learned count and policy are held fixed while only the cap is varied. Table 12 reports Kelly growth at the nominal multiplier 1. The final row gives the change from the unclipped $c = 1$ reference in units of 10^{-6}.

No tested cap improves growth over the $c = 1$ reference. Clipping at $c = 0.75$ has a negligible measured cost of 0.196×10^{-6}, while $c = 0.50$ reduces growth by 5.502×10^{-6}; more aggressive caps increasingly discard profitable high-count opportunities. Thus there is no empirical benefit from enforcing the small-exposure regime through clipping or from adding a separate cap-optimization stage to the wagering procedure. Together with the exposure distribution below, this strengthens the empirical justification for the local approximation and for retaining the untuned L-GOC wager, although it does not establish $|X^\pi f| \ll 1$ uniformly.

Table 13: Exposure distribution for the $c = 1.00, \eta = 0$ reference experiment over 10^9 rounds of one-deck 3:2 blackjack.

Exposure bin	Gross wager / bankroll	Absolute return $\lvert fX \rvert$
Exactly 0%	65.2354995%	68.1304328%
$>$ 0-1%	9.0061840%	8.0726441%
1-2%	8.8142%	8.0280%
2-5%	11.7230%	10.8220%
5-10%	4.3922%	4.1271%
10-20%	0.7252%	0.7439%
20-40%	0.093280%	0.069075%
40-60%	0.009302%	0.006111%
60-80%	0.0009909%	0.0006776%
80-90%	0.0000868%	0.0000651%
90-100%	0.0000250%	0.0000191%
$\geq$ 100%	0%	0%

Table 14: Distribution of total committed wager multiples conditional on placing a wager in the $c = 1.00, \eta = 0$ reference experiment.

Wager multiple	Conditional probability
1×	86.1323297%
2×	12.6799216%
3×	0.7586725%
4×	0.3591529%
5×	0.0625955%
6×	0.0069959%
7×	0.0003121%
8×	0.0000198%
Total	100%

I.3 Empirical Exposure Distribution

Table 13 reports two round-level quantities from the $c = 1, \eta = 0$ reference experiment over 10^9 rounds. Gross exposure is the total wager committed during the round, including splits and doubles, divided by bankroll. The second quantity is the absolute realized bankroll return $|fX|$. The zero-return bin includes both zero-wager rounds and wagered rounds with zero net return. Gross exposure is below 5% in 94.7789% of rounds, while $|fX| < 5\%$ in 95.0531%; neither quantity reaches 100%.

These results do not establish $|X^\pi f| \ll 1$ uniformly or almost surely: rare observations extend beyond the range in which a first-order expansion is accurate. They do show that the approximation operates in a low-exposure regime for the overwhelming majority of rounds. Moreover, because clipping this tail does not improve measured growth, these departures do not appear to create a practically important failure of the approximate wager in the tested setting.

I.4 Total Wager Multiples

Of the 10^9 simulated rounds, $652{,}354{,}995$ have no initial wager. Table 14 removes these rounds and reports the total committed wager, relative to the initial wager, among the remaining $347{,}645{,}005$ wagered rounds. The conditional frequencies are renormalized to 100%. A one-unit commitment means that no additional wager is placed, while larger multiples arise from splits and doubles.

Overall, large exposures are rare, and removing them through tighter caps does not improve measured growth. These diagnostics support the local growth approximation and the decision not to introduce additional wager tuning, while a positive safety margin remains available when strict pathwise admissibility is required.

I.5 Variance-Induced Reversal of Expected-Value Decisions

The logarithmic objective can prefer a lower-expectancy action when the higher-expectancy action also introduces sufficiently greater variance. To second order in the wager fraction f,

$$\mathbb{E}[\log(1 + fX)] \approx f\mathbb{E}[X] - \frac{f^2}{2}\mathbb{E}[X^2].$$

Consider the classical choice between doubling and hitting. Let X_d and X_h denote the respective round payoffs, and suppose doubling has the greater expected payoff,

$$\mathbb{E}[X_d] - \mathbb{E}[X_h] = \mu > 0.$$

In the simplified even-money case, hitting produces payoffs of magnitude one and doubling produces payoffs of magnitude two, so

$$\mathbb{E}[X_d^2] - \mathbb{E}[X_h^2] = 4 - 1 = 3.$$

This calculation ignores pushes, for which the payoff is zero. Push probability reduces $\mathbb{E}[X^2]$ for both actions, so the actual second-moment difference need not equal three. More generally, writing

$$\Delta m_2 = \mathbb{E}[X_d^2] - \mathbb{E}[X_h^2],$$

the reversal boundary below becomes $f > 2\mu/\Delta m_2$ whenever $\Delta m_2 > 0$; the value $2\mu/3$ is the no-push special case. The second-order logarithmic-growth difference between doubling and hitting is therefore

$$\begin{aligned}\Delta g(f) &\approx \left(f\mathbb{E}[X_d] - \frac{f^2}{2}\mathbb{E}[X_d^2]\right) - \left(f\mathbb{E}[X_h] - \frac{f^2}{2}\mathbb{E}[X_h^2]\right) \\ &= f\mu - \frac{3}{2}f^2.\end{aligned}$$

Although doubling has the higher expectancy, the approximation favors hitting whenever $\Delta g(f) < 0$. For $f > 0$, this occurs when

$$f\mu - \frac{3}{2}f^2 < 0 \qquad \Longleftrightarrow \qquad f > \frac{2}{3}\mu.$$

Thus $f = 2\mu/3$ is the approximate decision boundary: below it, the expectancy advantage dominates, while above it, the additional second moment of doubling dominates the logarithmic objective. In particular, when μ is close to zero, even a small wager fraction can reverse the expected-value preference. This boundary is specific to the second-order approximation and the assumed second-moment difference of three; exact logarithmic decisions depend on the complete payoff distributions and must also satisfy $1 + fX > 0$.

As a marginal example, consider soft 13 (ace–two) against a dealer five, with

$$\mathbb{E}[X_h] = 0.137618, \qquad \mathbb{E}[X_d] = 0.141030.$$

The doubling advantage is

$$\mu = 0.141030 - 0.137618 = 0.003412,$$

or 0.3412 percentage points per unit wager. Under the simplified second-moment difference $\Delta m_2 = 3$, the boundary is

$$f_{\text{crit}} = \frac{2}{3}(0.003412) = 0.0022747,$$

approximately 0.2275% of bankroll. Above this fraction, the second-order logarithmic approximation prefers hitting despite the slightly higher expected return from doubling. When pushes are included, the corresponding threshold is instead $2\mu/\Delta m_2$, using the actual payoff distributions of the two actions.

For practical context, under a conventional classical Hi-Lo betting ramp, reaching a wager near 0.2275% of bankroll generally requires a true count of approximately $+1$ to $+1.5$, although the exact mapping depends on the chosen bankroll and ramp. At such a favorable true count, the composition-dependent expected-return advantage of doubling over hitting is larger than the off-the-top difference used above. Consequently, treating $\mu = 0.003412$ as fixed understates the advantage of doubling at the point where this wager fraction is reached: the relevant μ, and hence the boundary $2\mu/\Delta m_2$, are higher. The composition adjustment therefore shifts the practical reversal point further upward and strengthens the preference for doubling.

J COUNT QUANTIZATION EFFECT

This appendix reports the full quantization experiments summarized in Subsection 4.5. Every case uses the final W_3 count with the P_2 policy. For $q > 0$, the quantized vector is selected jointly from the grid $(q\mathbb{Z})^p$ to minimize its mean squared error relative to the exact learned vector while retaining the balance constraint:

$$\widehat{w}^{(q)} \in \arg\min_{v \in (q\mathbb{Z})^p} \frac{1}{p}\|v - w\|_2^2 \qquad \text{subject to} \qquad \mathbf{1}^\top v = 0.$$

Thus the quantized tags continue to sum to zero. This joint constrained quantization is not generally equivalent to rounding each tag independently to its nearest multiple of q, because independent rounding need not preserve balance. The case $q = 0$ denotes the exact learned count. The 1:10 spread and Kelly growth are then evaluated without retraining the policy. Kelly growth is measured at the original nominal count scale, using multiplier 1 for every q, rather than selecting a new empirical optimum after quantization.

Within each performance row, Δ is the signed percentage change from the corresponding exact result. Spread changes are normalized by the absolute exact edge so that the sign remains intuitive when the edge itself is negative. Kelly changes are computed from excess growth, $G - 1$. Negative values indicate degradation and positive values indicate an improvement attributable to simulation variation or a favorable rounding perturbation.

Table 15: Quantization results for one-, two-, and six-deck 3:2 blackjack using $W_3 + P_2$.

Row	Exact ($q = 0$)	$q = 0.01$	$q = 0.05$	$q = 0.1$	$q = 0.5$	$q = 1$
One deck						
2	0.652784	0.650000	0.650000	0.700000	0.500000	1.000000
3	0.805437	0.810000	0.800000	0.800000	1.000000	1.000000
4	1.128447	1.130000	1.150000	1.100000	1.000000	1.000000
5	1.393210	1.390000	1.400000	1.400000	1.500000	1.000000
6	0.899254	0.900000	0.900000	0.900000	1.000000	1.000000
7	0.522141	0.520000	0.500000	0.500000	0.500000	0.000000
8	0.034590	0.030000	0.050000	0.000000	0.000000	0.000000
9	-0.303412	-0.300000	-0.300000	-0.300000	-0.500000	0.000000
10	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
J	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
Q	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
K	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
A	-1.132452	-1.130000	-1.150000	-1.100000	-1.000000	-1.000000
1:10 spread edge	0.111618903	0.111629793	0.111440558	0.111769981	0.111720336	0.107295224
Δ spread (%)	-	+0.0098	-0.1598	+0.1354	+0.0909	-3.8736
Kelly growth at 1	1.000205115	1.000205112	1.000204826	1.000204846	1.000203267	1.000185016
Δ Kelly (%)	-	-0.0016	-0.1411	-0.1309	-0.9009	-9.7988
Two decks						
2	0.714347	0.710000	0.700000	0.700000	0.500000	1.000000
3	0.865129	0.870000	0.850000	0.900000	1.000000	1.000000
4	1.191015	1.190000	1.200000	1.200000	1.000000	1.000000
5	1.444079	1.440000	1.450000	1.400000	1.500000	1.000000
6	0.851819	0.850000	0.850000	0.900000	1.000000	1.000000
7	0.463069	0.460000	0.450000	0.500000	0.500000	0.000000
8	-0.063583	-0.060000	-0.050000	-0.100000	0.000000	0.000000
9	-0.362565	-0.360000	-0.350000	-0.400000	-0.500000	0.000000
10	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
J	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
Q	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
K	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
A	-1.103311	-1.100000	-1.100000	-1.100000	-1.000000	-1.000000
1:10 spread edge	0.059976455	0.059989240	0.059989685	0.059965740	0.059890585	0.056382030
Δ spread (%)	-	+0.0213	+0.0221	-0.0179	-0.1432	-5.9931
Kelly growth at 1	1.000082170	1.000082208	1.000082273	1.000082091	1.000081670	1.000074316
Δ Kelly (%)	-	+0.0460	+0.1256	-0.0955	-0.6084	-9.5588
Six decks						
2	0.733943	0.740000	0.750000	0.700000	0.500000	1.000000
3	0.888184	0.890000	0.900000	0.900000	1.000000	1.000000
4	1.201259	1.200000	1.200000	1.200000	1.000000	1.000000
5	1.440238	1.440000	1.450000	1.400000	1.500000	1.000000
6	0.833696	0.830000	0.850000	0.800000	1.000000	1.000000
7	0.457456	0.460000	0.450000	0.500000	0.500000	0.000000
8	-0.108880	-0.110000	-0.100000	-0.100000	0.000000	0.000000
9	-0.419707	-0.420000	-0.450000	-0.400000	-0.500000	0.000000
10	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
J	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
Q	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
K	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
A	-1.026189	-1.030000	-1.050000	-1.000000	-1.000000	-1.000000
1:10 spread edge	0.017899316	0.017896996	0.017857754	0.017923167	0.017693883	0.016209713
Δ spread (%)	-	-0.0130	-0.2322	+0.1333	-1.1477	-9.4395
Kelly growth at 1	1.000017812	1.000017803	1.000017769	1.000017842	1.000017562	1.000015901
Δ Kelly (%)	-	-0.0510	-0.2387	+0.1710	-1.4019	-10.7299

Table 16: Quantization results for one-, two-, and six-deck 6:5 blackjack using $W_3 + P_2$.

Row	Exact ($q = 0$)	$q = 0.01$	$q = 0.05$	$q = 0.1$	$q = 0.5$	$q = 1$
One deck						
2	0.588643	0.590000	0.600000	0.600000	0.500000	1.000000
3	0.748539	0.750000	0.750000	0.800000	0.500000	1.000000
4	1.106962	1.110000	1.100000	1.100000	1.000000	1.000000
5	1.396535	1.390000	1.400000	1.400000	1.500000	1.000000
6	0.808978	0.810000	0.800000	0.800000	1.000000	1.000000
7	0.519414	0.520000	0.500000	0.500000	0.500000	0.000000
8	-0.078154	-0.080000	-0.100000	-0.100000	0.000000	0.000000
9	-0.421747	-0.420000	-0.400000	-0.400000	-0.500000	0.000000
10	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
J	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
Q	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
K	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
A	-0.669171	-0.670000	-0.650000	-0.700000	-0.500000	-1.000000
1:10 spread edge	0.045594062	0.045586303	0.045648330	0.045346787	0.045405463	0.035980989
Δ spread (%)	-	-0.0170	+0.1190	-0.5423	-0.4136	-21.0840
Kelly growth at 1	1.000094536	1.000094536	1.000094180	1.000094010	1.000094100	1.000074309
Δ Kelly (%)	-	-0.0004	-0.3769	-0.5569	-0.4620	-21.3961
Two decks						
2	0.651243	0.650000	0.650000	0.600000	0.500000	1.000000
3	0.812307	0.810000	0.800000	0.800000	1.000000	1.000000
4	1.187699	1.190000	1.200000	1.200000	1.000000	1.000000
5	1.465822	1.470000	1.450000	1.500000	1.500000	1.000000
6	0.771266	0.770000	0.800000	0.800000	0.500000	1.000000
7	0.387239	0.390000	0.400000	0.400000	0.500000	0.000000
8	-0.179816	-0.180000	-0.200000	-0.200000	0.000000	0.000000
9	-0.510148	-0.510000	-0.500000	-0.500000	-0.500000	0.000000
10	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
J	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
Q	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
K	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
A	-0.585612	-0.590000	-0.600000	-0.600000	-0.500000	-1.000000
1:10 spread edge	0.008006176	0.008001563	0.007958047	0.007973949	0.007798227	0.003734424
Δ spread (%)	-	-0.0576	-0.6011	-0.4025	-2.5974	-53.3557
Kelly growth at 1	1.000028055	1.000028056	1.000028033	1.000028085	1.000027677	1.000022086
Δ Kelly (%)	-	+0.0015	-0.0779	+0.1042	-1.3474	-21.2782
Six decks						
2	0.676476	0.680000	0.700000	0.700000	0.500000	1.000000
3	0.840922	0.840000	0.850000	0.800000	1.000000	1.000000
4	1.207531	1.210000	1.200000	1.200000	1.000000	1.000000
5	1.471409	1.470000	1.500000	1.500000	1.500000	2.000000
6	0.766254	0.760000	0.750000	0.800000	1.000000	1.000000
7	0.371073	0.370000	0.350000	0.400000	0.500000	0.000000
8	-0.241132	-0.240000	-0.250000	-0.300000	-0.500000	0.000000
9	-0.563378	-0.560000	-0.550000	-0.600000	-0.500000	-1.000000
10	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
J	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
Q	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
K	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
A	-0.529155	-0.530000	-0.550000	-0.500000	-0.500000	-1.000000
1:10 spread edge	-0.014885757	-0.014884831	-0.014908778	-0.014928456	-0.015142146	-0.016612599
Δ spread (%)	-	+0.0062	-0.1547	-0.2868	-1.7224	-11.6006
Kelly growth at 1	1.000002824	1.000002823	1.000002804	1.000002797	1.000002619	1.000001023
Δ Kelly (%)	-	-0.0431	-0.7148	-0.9533	-7.2635	-63.7702

Table 17: Card tags, balance per standard deck, and normalization for the seven reference counting systems. Halves denotes Wong Halves, and KO denotes Knock-Out.

Card rank	Hi-Lo	Halves	Hi-Opt I	Hi-Opt II	Omega II	Zen	KO
2	$+1$	$+0.5$	0	$+1$	$+1$	$+1$	$+1$
3	$+1$	$+1$	$+1$	$+1$	$+1$	$+1$	$+1$
4	$+1$	$+1$	$+1$	$+2$	$+2$	$+2$	$+1$
5	$+1$	$+1.5$	$+1$	$+2$	$+2$	$+2$	$+1$
6	$+1$	$+1$	$+1$	$+1$	$+2$	$+2$	$+1$
7	0	$+0.5$	0	$+1$	$+1$	$+1$	$+1$
8	0	0	0	0	0	0	0
9	0	-0.5	0	0	-1	0	0
10, J, Q, K	-1	-1	-1	-2	-2	-2	-1
A	-1	-1	0	0	0	-1	-1
Balance per deck	0	0	0	0	0	0	$+4$
Normalization	True count	True count	True count	True count	True count	True count	Running count

K Baseline Counting Systems and Deviations

This appendix summarizes the practical counting systems and compact deviation sets used as historical or experimental reference points in the paper. The descriptions are intended to specify the baseline statistics and decision rules, not to claim that one fixed index table is optimal under every blackjack ruleset.

K.1 Reference Counting Systems

All seven systems assign a fixed tag to each exposed card and update a scalar running count. Hi-Lo, Wong Halves, Hi-Opt I, Hi-Opt II, Omega II, and Zen are balanced: the tags sum to zero over a complete standard deck, and the evaluations below use true-count normalization. Knock-Out (KO) is unbalanced and uses the running count; its conventional initial and key counts depend on the shoe size and rules Wong (1994); Vancura & Fuchs (1998). Table 17 specifies the tags, balance per deck, and normalization used in these comparisons.

Hi-Lo is a balanced level-one system that groups low cards 2-6, neutral cards 7-9, and high cards 10-A. Wong Halves uses half-point increments to distinguish ranks more finely while remaining balanced Wong (1994). Knock-Out assigns $+1$ to sevens as well as 2-6; the resulting nonzero full-deck sum is what permits running-count use without explicit true-count conversion Vancura & Fuchs (1998).

K.2 Illustrious 18 Deviations

The Illustrious 18 is a compact collection of high-value Hi-Lo index plays introduced by Schlesinger Schlesinger (2017). Table 18 gives the commonly quoted canonical indices. At or above an index, the listed deviation replaces the ordinary basic-strategy action. Exact thresholds can change with deck count, dealer soft-seventeen behavior, surrender availability, and the convention used to round the true count; the table should therefore be read as a reference specification rather than a universal rules-independent optimum.

These deviations illustrate how a single count can support both betting and play. They are deliberately compact: a full-deviation policy contains many more count-dependent action boundaries, while the Illustrious 18 retain only a small set with high practical value.

K.3 Walker's Spanish 21 Baseline

Walker adapts Hi-Lo to the 48-card Spanish deck, in which the numeral ten is absent but jacks, queens, and kings remain ten-valued Walker (2007). The tags are shown in Table 19. Removing the four ten spots makes the resulting system unbalanced relative to a standard Hi-Lo starting point, so an N-deck shoe begins at

$$RC_0 = -4N.$$

Table 18: Canonical Hi-Lo Illustrious 18 index plays. A dealer ten denotes any ten-valued upcard.

No.	Player situation	Index	Action at or above the index
1	Insurance against dealer ace	$+3$	Take insurance
2	Hard 16 vs. 10	0	Stand
3	Hard 15 vs. 10	$+4$	Stand
4	Pair 10,10 vs. 5	$+5$	Split
5	Pair 10,10 vs. 6	$+4$	Split
6	Hard 10 vs. 10	$+4$	Double
7	Hard 12 vs. 3	$+2$	Stand
8	Hard 12 vs. 2	$+3$	Stand
9	Hard 11 vs. A	$+1$	Double
10	Hard 9 vs. 2	$+1$	Double
11	Hard 10 vs. A	$+4$	Double
12	Hard 9 vs. 7	$+3$	Double
13	Hard 16 vs. 9	$+5$	Stand
14	Hard 13 vs. 2	-1	Stand
15	Hard 12 vs. 4	0	Stand
16	Hard 12 vs. 5	-2	Stand
17	Hard 12 vs. 6	-1	Stand
18	Hard 13 vs. 3	-2	Stand

Table 19: Walker's Hi-Lo tags for Spanish 21. The numeral ten is absent from the Spanish deck.

Rank	2	3	4	5	6	7	8	9	J/Q/K	A
Tag	$+1$	$+1$	$+1$	$+1$	$+1$	0	0	0	-1	-1

Thus a six-deck shoe begins at $RC_0 = -24$, and an eight-deck shoe begins at $RC_0 = -32$. The true count is

$$TC = \frac{RC}{\text{decks remaining}},$$

so a fresh shoe corresponds to $TC = -4$, rather than zero. Walker uses this scalar true count both for betting and for count-dependent playing deviations.

Table 20 records the Walker deviations used in our experiments, which were provided as true-count thresholds Walker (2007).

K.4 Cross-System Comparison

Tables 21 and 22 compare seven reference counts under basic strategy and full count-dependent deviations in one-, two-, and six-deck blackjack. They report the selected search parameter f^*, the best Kelly growth factor observed in the search, and the 1:10 fixed-spread edge. All intervals are 95% confidence intervals. Growth factors above one indicate positive logarithmic growth; they are not themselves logarithmic growth rates. Superscripts L and U identify lower and upper search boundaries, respectively; ‡ indicates that the boundary value was confirmed by a complementary or wider sweep, whereas † indicates that it remains questionable. Boundary selections should not be interpreted as unrestricted optima. These results describe the evaluated implementations and calibration range, rather than the best achievable performance of each counting system.

Table 20: Walker Spanish 21 true-count deviations used as a practical baseline.

Player hand	Dealer	Index	Action at or above the index
Hard 16	X	-2	Surrender
Hard 16	X	$+3$	Stand*
Hard 17	A	-6	Surrender
Hard 17	A	-3	Stand*
Doubled 17	A	$+2$	Forfeit the double
A,6	4	-5	Double; otherwise hit
A,7	4	-5	Double
8,8	9	-11	Split; otherwise hit
Hard 9	6	-3	Double; otherwise hit
Hard 11	9	-5	Double; otherwise hit

Table 21: Reference-count performance in blackjack paying 3:2. Kelly growth is the best observed growth factor within the search range; superscripts mark boundary selections and their confirmation status. All intervals are 95% CIs.

Decks	Count	Policy	f^*	Kelly growth factor	Spread 1:10
1	Hi-Lo	Basic	0.75	$1.000108903 \pm 0.000000913$	0.074534 ± 0.000187
		Full deviations	0.80	$1.000183111 \pm 0.000001118$	0.101911 ± 0.000197
	Halves	Basic	0.70	$1.000113704 \pm 0.000000952$	0.078371 ± 0.000196
		Full deviations	0.80	$1.000202567 \pm 0.000001177$	0.109565 ± 0.000205
	Hi-Opt I	Basic	0.70	$1.000082662 \pm 0.000000810$	0.067136 ± 0.000185
		Full deviations	0.90	$1.000159269 \pm 0.000001159$	0.097908 ± 0.000195
	Hi-Opt II	Basic	0.425	$1.000088751 \pm 0.000000902$	0.071784 ± 0.000193
		Full deviations	$0.50^{L\dagger}$	$1.000170503 \pm 0.000001201$	0.102252 ± 0.000203
	Omega II	Basic	0.375	$1.000089822 \pm 0.000000811$	0.072471 ± 0.000194
		Full deviations	0.450	$1.000173181 \pm 0.000001106$	0.103336 ± 0.000204
	Zen	Basic	0.425	$1.000104433 \pm 0.000000971$	0.076067 ± 0.000194
		Full deviations	0.475	$1.000188298 \pm 0.000001187$	0.105525 ± 0.000205
	KO	Basic	$1.00^{U\ddagger}$	$1.000109881 \pm 0.000000931$	0.075331 ± 0.000230
		Full deviations	1.15	$1.000189743 \pm 0.000001168$	0.108278 ± 0.000238
2	Hi-Lo	Basic	0.70	$1.000049028 \pm 0.000000581$	0.042250 ± 0.000185
		Full deviations	0.85	$1.000074822 \pm 0.000000770$	0.055993 ± 0.000192
	Halves	Basic	0.70	$1.000051751 \pm 0.000000612$	0.043657 ± 0.000186
		Full deviations	0.85	$1.000082487 \pm 0.000000811$	0.059175 ± 0.000192
	Hi-Opt I	Basic	0.70	$1.000037303 \pm 0.000000501$	0.037321 ± 0.000182
		Full deviations	0.95	$1.000065790 \pm 0.000000730$	0.053333 ± 0.000188
	Hi-Opt II	Basic	0.425	$1.000039789 \pm 0.000000578$	0.038733 ± 0.000179
		Full deviations	$0.50^{L\dagger}$	$1.000069170 \pm 0.000000778$	0.055156 ± 0.000202
	Omega II	Basic	0.400	$1.000040486 \pm 0.000000578$	0.039432 ± 0.000181
		Full deviations	0.475	$1.000069062 \pm 0.000000692$	0.054149 ± 0.000187
	Zen	Basic	0.400	$1.000046845 \pm 0.000000575$	0.041790 ± 0.000181
		Full deviations	0.475	$1.000075696 \pm 0.000000702$	0.055927 ± 0.000188
	KO	Basic	0.35	$1.000039030 \pm 0.000000531$	0.022252 ± 0.000257
		Full deviations	0.45	$1.000061684 \pm 0.000000719$	0.036967 ± 0.000260
6	Hi-Lo	Basic	0.75	$1.000011438 \pm 0.000000284$	0.011688 ± 0.000153
		Full deviations	0.85	$1.000015784 \pm 0.000000356$	0.015909 ± 0.000156
	Halves	Basic	0.65	$1.000012325 \pm 0.000000278$	0.012536 ± 0.000156
		Full deviations	0.75	$1.000017636 \pm 0.000000321$	0.017374 ± 0.000159
	Hi-Opt I	Basic	0.75	$1.000008583 \pm 0.000000246$	0.009019 ± 0.000147
		Full deviations	0.85	$1.000013270 \pm 0.000000307$	0.013830 ± 0.000149
	Hi-Opt II	Basic	0.450	$1.000009270 \pm 0.000000268$	0.009726 ± 0.000142
		Full deviations	$0.50^{U\ddagger}$	$1.000014518 \pm 0.000000308$	0.014677 ± 0.000145
	Omega II	Basic	0.425	$1.000009584 \pm 0.000000291$	0.010208 ± 0.000146
		Full deviations	$0.50^{U\ddagger}$	$1.000014318 \pm 0.000000340$	0.014751 ± 0.000149
	Zen	Basic	0.425	$1.000011014 \pm 0.000000293$	0.011339 ± 0.000147
		Full deviations	$0.50^{U\ddagger}$	$1.000015583 \pm 0.000000337$	0.015541 ± 0.000149
	KO	Basic	$0.25^{L\dagger}$	$0.999893187 \pm 0.000001015$	-0.031260 ± 0.000281
		Full deviations	$0.25^{L\dagger}$	$0.999898799 \pm 0.000001035$	-0.027452 ± 0.000281

Table 22: Reference-count performance in blackjack paying 6:5, using the same conventions as Table 21.

Decks	Count	Policy	f^*	Kelly growth factor	Spread 1:10
1	Hi-Lo	Basic	$0.50^{L\dagger}$	$1.000022987 \pm 0.000000445$	0.010055 ± 0.000141
		Full deviations	0.75	$1.000074209 \pm 0.000000724$	0.035562 ± 0.000151
	Halves	Basic	$0.50^{L\dagger}$	$1.000025909 \pm 0.000000479$	0.012158 ± 0.000144
		Full deviations	0.80	$1.000088638 \pm 0.000000830$	0.041465 ± 0.000156
	Hi-Opt I	Basic	$0.50^{L\dagger}$	$1.000017304 \pm 0.000000380$	0.007618 ± 0.000135
		Full deviations	0.90	$1.000069803 \pm 0.000000776$	0.036263 ± 0.000145
	Hi-Opt II	Basic	0.300	$1.000019337 \pm 0.000000405$	0.008871 ± 0.000132
		Full deviations	$0.50^{L\dagger}$	$1.000073068 \pm 0.000000968$	0.039079 ± 0.000173
	Omega II	Basic	0.275	$1.000020119 \pm 0.000000383$	0.009841 ± 0.000135
		Full deviations	0.450	$1.000077053 \pm 0.000000745$	0.038303 ± 0.000146
	Zen	Basic	0.300	$1.000023165 \pm 0.000000459$	0.010678 ± 0.000136
		Full deviations	0.475	$1.000081587 \pm 0.000000790$	0.037867 ± 0.000147
	KO	Basic	1.00	$1.000024134 \pm 0.000000427$	0.011208 ± 0.000143
		Full deviations	1.60	$1.000083172 \pm 0.000000805$	0.041185 ± 0.000153
2	Hi-Lo	Basic	0.55	$1.000006968 \pm 0.000000257$	-0.008106 ± 0.000115
		Full deviations	0.85	$1.000022237 \pm 0.000000480$	0.003825 ± 0.000123
	Halves	Basic	$0.50^{L\dagger}$	$1.000007998 \pm 0.000000265$	-0.007037 ± 0.000119
		Full deviations	0.85	$1.000026544 \pm 0.000000506$	0.006567 ± 0.000127
	Hi-Opt I	Basic	0.55	$1.000005141 \pm 0.000000213$	-0.009421 ± 0.000108
		Full deviations	0.95	$1.000021594 \pm 0.000000425$	0.003931 ± 0.000115
	Hi-Opt II	Basic	0.300	$1.000005986 \pm 0.000000214$	-0.008663 ± 0.000103
		Full deviations	0.475	$1.000021302 \pm 0.000000349$	0.003618 ± 0.000109
	Omega II	Basic	0.275	$1.000006315 \pm 0.000000218$	-0.008098 ± 0.000107
		Full deviations	0.475	$1.000021965 \pm 0.000000393$	0.004271 ± 0.000114
	Zen	Basic	0.300	$1.000007248 \pm 0.000000243$	-0.007710 ± 0.000108
		Full deviations	0.475	$1.000023224 \pm 0.000000398$	0.004185 ± 0.000114
	KO	Basic	$0.25^{L\dagger}$	$1.000001112 \pm 0.000000244$	-0.029310 ± 0.000198
		Full deviations	0.30	$1.000013260 \pm 0.000000322$	-0.015012 ± 0.000204
6	Hi-Lo	Basic	$0.50^{L\dagger}$	$1.000000610 \pm 0.000000080$	-0.018783 ± 0.000073
		Full deviations	0.75	$1.000002192 \pm 0.000000136$	-0.015961 ± 0.000077
	Halves	Basic	$0.50^{L\dagger}$	$1.000000729 \pm 0.000000090$	-0.018559 ± 0.000077
		Full deviations	0.75	$1.000002719 \pm 0.000000146$	-0.015299 ± 0.000080
	Hi-Opt I	Basic	$0.50^{L\dagger}$	$1.000000409 \pm 0.000000061$	-0.018967 ± 0.000065
		Full deviations	0.75	$1.000001958 \pm 0.000000099$	-0.015989 ± 0.000069
	Hi-Opt II	Basic	0.275	$1.000000491 \pm 0.000000062$	-0.018705 ± 0.000061
		Full deviations	$0.50^{U\ddagger}$	$1.000001909 \pm 0.000000112$	-0.015940 ± 0.000063
	Omega II	Basic	0.275	$1.000000548 \pm 0.000000073$	-0.018598 ± 0.000065
		Full deviations	0.425	$1.000001898 \pm 0.000000113$	-0.015911 ± 0.000068
	Zen	Basic	0.275	$1.000000611 \pm 0.000000074$	-0.018520 ± 0.000065
		Full deviations	0.425	$1.000002007 \pm 0.000000110$	-0.015889 ± 0.000067
	KO	Basic	$0.25^{L\dagger}$	$0.999827418 \pm 0.000000836$	-0.121967 ± 0.000261
		Full deviations	$0.25^{L\dagger}$	$0.999833057 \pm 0.000000876$	-0.118137 ± 0.000262