FROM FINITE PLANS TO LONG-RUN OPTIMALITY AGAINST HEDGE

Anonymous authors
Paper under double-blind review

ABSTRACT

We study optimal play against full-history Hedge in infinitely repeated games with arbitrary finite payoff matrices and learning rate $t^{-\alpha}$, where $0 < \alpha < 1$. We construct a computable strategy that maximizes the lower limit of expected average payoff, using one strategy for all evaluation horizons while the learner keeps its actual history. The construction has three steps: finite planning with a payoff guarantee at every prefix, certified extension from short searches to long references, and sparse replacement of references during the continuing game. The finite planner returns global lower and upper bounds and has a complete count-state fallback. A uniform variation bound supports verification of repeated references with logarithmic dependence on the repetition count in a scalar-evaluation model. The execution rule controls the effect of earlier history and weights finite planning errors by the distribution of the latest reference. Experiments check the finite certificates against independent solvers, measure when compression saves verification work, and show how history-induced payoff differences decrease over time.

1 INTRODUCTION

An optimizer facing a known learning rule controls both its current reward and the learner's future responses. Each action changes the experience on which subsequent decisions are based, creating a strategic control problem even when the stage game is completely known. Research on strategizing against no-regret learners shows that anticipating learning dynamics can improve an optimizer's payoff, and that the attainable improvement depends on the learner's algorithm (Deng et al., 2019; Brown et al., 2023; Assos et al., 2024).

We consider a learner that applies its current learning rate to its entire cumulative payoff vector. With rate $t^{-\alpha}$, a fixed historical score discrepancy has diminishing influence, whereas the cumulative effect of new actions can continue to shape responses. An optimal infinite strategy must coordinate these effects in one continuing interaction. Optimizing the payoff at a selected terminal horizon does not control intermediate deficits, and evaluating a new plan from the original prior does not erase the history created by actual play.

Our finite interface is a *prefix-safe plan*: its expected cumulative payoff is at least $\ell T - C$ at every prefix of a prescribed horizon. The budget C allows strategic investment before its returns are realized. These finite problems characterize the original long-run value, but their direct count-state representation becomes large as the horizon grows. We address this computational difficulty by distinguishing the horizon that is *searched* from the horizon for which a policy is *verified*.

Certified Continuation Planning (CCP) searches a shorter prefix of length h, extends a short policy into a finitely represented reference of length $H \geq h$, and verifies its complete prefix reward curve. The shorter problem supplies a global upper bound for the longer one because both use the same deficit budget. The extended reference supplies a feasible lower bound. Their difference bounds the improvement achievable over the plan by any behavioral strategy, including those outside the current candidate library. If the interval remains wide, CCP increases h; reaching $h = H$ supplies a complete finite fallback. A repetition-independent variation bound for the actual softmax rewards supports compressed long-reference verification.

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.

To execute these plans, we introduce Sparse-Refresh Prefix Planning (SRPP). At geometrically spaced calendar times, it refreshes a reference with a decreasing probability and otherwise continues its calendar-indexed suffix. The refresh law makes a reference likely to be recent enough to cover the current round but old enough for the effect of the inherited history to have decayed. The learner is never reset. This also identifies the correct error accounting: finite planning errors are weighted by the distribution of the latest refresh rather than summed over all earlier calls.

Contributions. We establish three connected results. First, CCP returns an executable finite plan and a global ε-certificate, with finite completion for arbitrary finite computable inputs and no periodic-coverage assumption. Second, exact repeated-word dynamics yield simultaneous all-prefix verification with a cost logarithmic in the repetition count in a stated scalar-evaluation model. Third, the integrated CCP–SRPP algorithm attains the original optimal lower limit of expected average payoff when certified errors vanish, and has a quantified residual guarantee when they do not.

The complete-count fallback can be expensive, so we measure when a short search or compressed verification reduces the required work. Our experiments verify all 28 finite targets, compare direct and compressed evaluation on fixed repeated references, and measure the payoff difference caused by earlier histories. The results connect each part of the construction to a directly observed quantity.

2 Problem Formulation and Related Work

2.1 Repeated-Game Model

Game and information. A strategist and a learner repeatedly play a finite game with action sets $[m]$ and $[n]$, where $[k] = \{1, \dots, k\}$. We write Δ_k for the set of probability distributions on $[k]$. The payoff matrices $A, B \in \mathbb{R}^{m\times n}$ give rewards A_{ab} to the strategist and B_{ab} to the learner when they choose (a, b). We scale the strategist's payoffs to $A \in [0, 1]^{m\times n}$ by a positive affine transformation, which preserves optimal strategies; the scale of B is unchanged. At each round, the players choose their actions simultaneously and observe both actions afterward. The strategist knows A, B, and the learner's update rule and parameters.

Hedge update. The learner uses Hedge (Freund & Schapire, 1997) with a prior $q \in \Delta_n$ satisfying $q_b > 0$ for every b. At round $t \geq 1$, it draws $b_t \sim p_t$, where

$$p_t(b) = \frac{q_b \exp\big(\eta_t \sum_{s<t} B_{a_s b}\big)}{\sum_{d=1}^{n} q_d \exp\big(\eta_t \sum_{s<t} B_{a_s d}\big)}, \qquad \eta_t = t^{-\alpha}, \quad 0 < \alpha < 1. \tag{1}$$

After round t, it observes the full payoff vector $(B_{a_t b})_{b\in[n]}$ and adds it to its cumulative payoffs. The current rate η_t multiplies the entire payoff history, so every past reward has the same weight at the current round. We refer to this update as full-history Hedge.

Strategist's choices. The public history before round t is $\mathcal{H}_{t-1} = ((a_s, b_s))_{s<t}$, with $\mathcal{H}_0 = \varnothing$. A behavioral strategy $\pi = (\pi_t)_{t\geq 1}$ assigns an action distribution $\pi_t(\cdot \mid \mathcal{H}_{t-1}) \in \Delta_m$ to each history. Conditional on that history, the players draw their current actions independently: $a_t \sim \pi_t(\cdot \mid \mathcal{H}_{t-1})$ and $b_t \sim p_t(\cdot \mid \mathcal{H}_{t-1})$. The strategist can compute p_t from the past but chooses a_t before observing b_t.

2.2 Long-Run Objective

Let Π be the set of all history-dependent randomized behavioral strategies. For $\pi \in \Pi$, define

$$F_T(\pi) = \frac{1}{T}\mathbb{E}_\pi\left[\sum_{t=1}^{T} A_{a_t b_t}\right], \qquad J(\pi) = \liminf_{T\to\infty} F_T(\pi), \qquad V = \sup_{\pi\in\Pi} J(\pi). \tag{2}$$

The expectation includes both players' random choices and is taken before the lower limit. We seek one strategy $\pi^\star$ with $J(\pi^\star) = V$, to be used at every evaluation horizon. For the computational results, the entries of A, B, and q, and the exponent α, are computable: each is given by a procedure that returns rational approximations with any requested error bound. Computation occurs between rounds and does not advance the game clock.

Each action affects both the current reward and the learner's later choices. A plan that maximizes payoff at one terminal horizon can have poor intermediate averages, whereas J requires high expected averages at every sufficiently large horizon. A plan introduced later also faces the history created by earlier play. The method must therefore control intermediate payoff losses and transfer finite-plan guarantees to a learner that keeps its history.

2.3 Related Work

Strategizing against learning rules. Deng et al. (2019) connect no-regret learning to Stackelberg payoffs and formulate asymptotic non-adaptive play against mean-based learners as an optimal control problem. Brown et al. (2023) show that learners with no-regret guarantees can differ in the payoffs available to a strategic opponent, while Arunachaleswaran et al. (2024) study the trade-offs through asymptotic menus. Planning against fictitious play has also been studied through finite-state models and explicit long-run strategies (Dong & Mu, 2022a;b; Vundurthy et al., 2023). These works motivate studying the specified learning rule, beyond its regret bound alone.

Optimization against Hedge and multiplicative weights. For zero-sum games, Assos et al. (2024) solve the continuous-time control problem against replicator dynamics and give performance bounds for discrete-time multiplicative weights. Guo & Mu (2025) derive globally optimal finite-horizon responses to Hedge in 2×2 zero-sum games using a state graph and Bellman recursion. Assos et al. (2025) prove finite-horizon approximation hardness in a model whose learning rate stays fixed during play and whose action sets can grow with the horizon. Our setting fixes the finite game and uses the changing rate in (1).

Learning the opponent's model. Zhang et al. (2025) study how to reach Stackelberg outcomes when the learner's payoffs are unknown. Handina et al. (2026) study how approximate simulators and payoff advice help compute Stackelberg strategies. Here the learner's model is known, and the target is the optimal value over all behavioral strategies in (2).

Recovery in general environments. Ryabko & Hutter (2008) study optimal average-reward policies in environments where earlier actions affect later outcomes. Their value-stability condition requires a uniform recovery cost that is sublinear in the previous history length. Our construction uses the decay of a fixed score difference under full-history Hedge and schedules reference changes to control its effect on the lower-limit objective.

Focus of this work. We construct one computable infinite strategy for general payoff matrices and arbitrary finite action sets. The strategy achieves V under the specified full-history Hedge update while preserving the learner's actual history. Finite count-state planning provides the required payoff bounds; the main step is to carry these bounds into one continuing interaction at every sufficiently large horizon.

3 Proposed Method

We construct a strategy achieving V in three steps. First, we define finite problems that require a payoff guarantee at every prefix, meaning every initial segment of a plan. Second, we solve these problems with a short search and verification over the full target horizon. Third, we combine the finite plans into one infinite strategy while controlling the effect of the learner's earlier history.

3.1 Step 1: Define a Finite Planning Problem

We first consider the following problem: for a horizon H and budget C, find a randomized plan with the largest rate ℓ such that its expected cumulative payoff is at least $\ell T - C$ for every $T \le H$. The budget allows early payoff losses while the plan influences the learner's later responses, and the prefix constraints make the guarantee useful throughout the plan.

Reference rewards. A reference sequence is a finite action sequence evaluated from round one, using the original prior q and zero initial counts. Let $N_a(t-1) = \sum_{s<t} \mathbf{1}\{a_s = a\}$ and $N(t-1) =$

$(N_a(t-1))_{a\in[m]}$. Hedge's cumulative payoff vector is $B^\top N(t-1)$, so the counts determine its current response. Writing $\mathrm{softmax}(z)_b = e^{z_b}/\sum_{d=1}^n e^{z_d}$ for softmax, the expected stage reward and prefix reward of a reference $w \in [m]^H$ are

$$r_t(c,a) = A_{a,\cdot}\,\mathrm{softmax}\big(\log q + t^{-\alpha}B^\top c\big)\,, \qquad S_T(w) = \sum_{t=1}^{T} r_t(N_w(t-1), w[t])\,, \tag{3}$$

where $N_w(t-1)$ counts the first $t-1$ actions of w and $T \in [H]$. A distribution μ over reference sequences specifies a randomized plan. Distributions over reference sequences capture all expected prefix rewards achievable by behavioral strategies, because each conditional stage reward depends only on the strategist's action prefix (Section A.1).

The finite objective. For an integer $C \geq 0$, define

$$\Lambda_H(C) = \max_{\mu\in\Delta([m]^H),\, \ell\in[0,1]} \ell \qquad \text{s.t.} \quad \mathbb{E}_\mu S_T \geq \ell T - C, \quad T \in [H]. \tag{4}$$

Here $\Delta([m]^H)$ is the set of distributions on length-H action sequences. A feasible solution guarantees expected average payoff at least $\ell - C/T$ at every covered horizon. Increasing H adds constraints, while increasing C allows larger temporary losses. In particular, $\Lambda_H(C) \leq \Lambda_h(C)$ for $h \leq H$ when the budget is unchanged.

An exact finite formulation. The problem is a linear program (LP) on count states $\mathcal{C}_s = \{c \in \mathbb{N}_0^m : \mathbf{1}^\top c = s\}$. Its variable $y_t(c,a)$ is the probability of reaching state $c \in \mathcal{C}_{t-1}$ and taking action a. Probability flows from c to $c + e_a$, where e_a is the ath coordinate vector, and the prefix constraints use rewards $r_t(c,a)$. Conversely, any feasible flow gives a reference policy by setting $\pi_t^{\mathrm{ref}}(a \mid c) = y_t(c,a)/\sum_d y_t(c,d)$ at positive-mass states. Section A.2.1, (19), gives the complete LP and proves that its value is $\Lambda_H(C)$. This gives an exact finite problem; we next address the cost of solving it for large H.

3.2 Step 2: Certify a Long Plan from a Short Search

The count graph has $\binom{H+m-1}{m}$ nonterminal states, so solving the full LP can be costly. We instead solve a shorter problem with horizon $h \leq H$ and extend its policy to length H. We verify the extended plan and compare it with a global upper bound on $\Lambda_H(C)$, since extension need not preserve the short solution's payoff rate. We call this procedure *Certified Continuation Planning* (CCP); the target (H, C) stays fixed while h grows.

Bound the target value from above. The short LP gives a distribution μ_h over length-h reference sequences and verified bounds $L_h \leq \Lambda_h(C) \leq U_h$, with $U_h - L_h \leq \varepsilon/8$. This distribution gives L_h; dual weights on the prefix constraints give a weighted-reward problem on the full short count graph. A backward Bellman recursion with upper reward bounds gives U_h (Section A.2). The ordering from Step 1 then gives $\Lambda_H(C) \leq U_h$.

Construct and verify a long reference. We extend μ_h by repeating short action sequences and also include repeated sequences that start at round one. The library contains all primitive words of length at most three, where a primitive word is an action sequence that is not a repetition of a shorter one. An extension uses the distribution of counts after a sequence drawn from μ_h and keeps the original round indices. A small LP selects a mixture of candidates using a finite set of prefix checkpoints. After rounding its sampling probabilities to dyadic rationals, whose denominators are powers of two, we verify all prefixes of the distribution that will actually be sampled. For long periodic parts, Theorem 3 lets us bound rewards over groups of repetitions; direct finite evaluation is the fallback when these bounds cannot be resolved. The verified rate L_H satisfies

$$\mathbb{E}_\mu S_T \geq L_H T - C \quad (T \in [H]), \qquad L_H \leq \Lambda_H(C) \leq \Lambda_h(C) \leq U_h. \tag{5}$$

Thus $U_h - L_H$ bounds the plan's loss relative to all strategies for the target finite problem, including those outside the library.

Increase the search horizon when needed. CCP retains the best verified reference and lower bound L, together with the smallest upper bound U. It returns (μ, L, U) when $U - L \le \varepsilon$; otherwise, it doubles h, capped at H. At $h = H$, the complete LP closes the gap directly, guaranteeing finite completion regardless of the quality of the proposed continuations. Section A.2 gives the initialization, verification details, and termination proof. The result is a finite plan with a verified error bound; executing such plans in the original infinite game is the remaining step.

3.3 Step 3: Use Finite Plans in One Infinite Game

A certified finite plan still differs from the original task in two ways: its guarantee ends at H, and it was evaluated from zero counts while the actual learner keeps its earlier history. We address the first issue by increasing the horizons and budgets, and the second by choosing when to replace a plan and which reference action to play.

Choose increasingly accurate plans. Proposition 1 shows that, for every $v < V$, some finite budget C_v gives $\Lambda_H(C) \ge v$ for all H once $C \ge C_v$. For each index $j \ge 1$, round $t_j = 2^j$ is an opportunity to replace the current reference; we call a replacement a *refresh*. A refresh at t_j uses

$$H_j = 2^{j^2}, \qquad C_j = j, \qquad \varepsilon_j = \min\left\{\frac{1}{2}, \varepsilon_\infty + \frac{1}{j+1}\right\}, \tag{6}$$

where rational $\varepsilon_\infty \in [0, 1)$ is the target long-run error. The growing budget and CCP certificate give $L_j \ge v - \varepsilon_j$ for every sufficiently large j, without knowing v or C_v.

Keep the original round indices. When a refresh occurs at t_j, compute $(\mu_j, L_j, U_j) = \mathrm{CCP}(H_j, C_j, \varepsilon_j)$ and draw $W \sim \mu_j$ independently of the refresh decisions and the learner. Generate the virtual actions through index t_j internally and play $W[t_j]$. At each later round t, play $W[t]$ until the next refresh or until $t > H_j$. The learner continues from its actual counts $N(t-1)$. The two action sequences now agree at the same round indices, so their count difference stays fixed and its contribution to Hedge's exponent decreases with $t^{-\alpha}$.

Let $s = t_j - 1$ and $\kappa = \max_a(\max_b B_{ab} - \min_b B_{ab})$. Conditional on a refresh history in which reference j remains active through $T \le H_j$, its distribution remains μ_j, and

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} A_{a_t b_t} \,\middle|\, \text{refresh history}\right] \ge L_j - \frac{C_j}{T} - \frac{s}{T} - \frac{2\kappa s}{1-\alpha} T^{-\alpha}. \tag{7}$$

The loss terms are the reference budget, the rounds before the refresh, and the effect of their counts on later responses (Section A.4). A reference therefore needs enough time for the earlier history's effect to become small, while its horizon must still cover the current round.

Refresh slowly enough for history errors to shrink. At opportunity j, refresh independently with dyadic probability ϱ_j, where $\varrho_1 = 1$ and $\varrho_j = \Theta(1/(j\sqrt{\log j}))$ for large j; otherwise, keep the current reference. The exact probabilities are given in (48). Together with the growing H_j, this rule makes the reference both old enough for its history error to be small and long enough to cover the current round, with probability tending to one. Before the first reference or after its horizon ends, play a fixed action until the next refresh. Each planner call is deterministic given the model and index; refresh decisions, reference sampling, and learner sampling use independent random streams. This rule, called *Sparse-Refresh Prefix Planning* (SRPP), samples its decisions with finitely many random bits and generates virtual prefixes between game rounds. Theorem 4 proves $J(\pi_{\varepsilon_\infty}) \ge V - \varepsilon_\infty$, and setting $\varepsilon_\infty = 0$ gives $J(\pi_0) = V$. Section A.4 gives the exact refresh law and proves the transfer guarantee.

4 Theoretical Guarantees

We establish long-run optimality for one computable strategy while the learner keeps its history. The finite problems provide a value characterization and global error certificates; the main step is to turn these separate plans into one strategy that works at every sufficiently large horizon. We use the model and computational assumptions in Section 2; proofs are in Appendix A.

4.1 Finite Problems Characterize the Long-Run Value

We characterize the original value V through the finite values $\Lambda_H(C)$.

Proposition 1 (Finite value characterization)**.** *The original value satisfies*

$$V = \sup_{C\in\mathbb{N}_0} \inf_{H\geq 1} \Lambda_H(C). \tag{8}$$

For every $v < V$, there is a finite integer C_v such that

$$\Lambda_H(C) \geq v, \qquad H \geq 1, \quad C \geq C_v. \tag{9}$$

Why the budget is fixed before the horizon grows. The same C must cover early payoff losses for every horizon. Reversing the order in (8) gives $\inf_H \sup_C \Lambda_H(C) = 1$ for every game, since choosing $C \geq H$ makes rate one feasible. Fixing C first prevents the budget from paying for the entire interaction. For that fixed budget, compactness extracts consistent prefix distributions with rate $\inf_H \Lambda_H(C)$. Conversely, any strategy with long-run payoff above v has finitely many early deficits, which a finite C_v covers. Section A.1 proves both directions. This characterization supports growing budgets without knowing V or C_v; the later construction supplies an executable infinite strategy.

4.2 Finite Plans Have Global Error Certificates

Step 2 must solve each finite problem with an error bound that covers the full strategy space.

Theorem 2 (Certified finite planning)**.** *For every finite $H \geq 1$, $C \in \mathbb{N}_0$, and rational $0 < \varepsilon < 1$, CCP with the complete finite fallback finishes and returns a reference distribution μ and rational $L, U \in [0, 1]$ satisfying*

$$\mathbb{E}_\mu S_T \geq LT - C \quad (1 \leq T \leq H), \qquad L \leq \Lambda_H(C) \leq U, \qquad U - L \leq \varepsilon. \tag{10}$$

The reference can be sampled with finitely many random bits, and its guarantee includes all numerical and probability-rounding errors. Every intermediate verified interval is valid for the same target (H, C).

What a short search can certify. A candidate library may omit good strategies, so optimizing within it only supplies a lower bound. The full short count graph supplies an upper bound on every length-H strategy through $\Lambda_H(C) \leq \Lambda_h(C)$ at the same budget. Thus a small gap provides a global certificate for the continuation, including when the candidate library is incomplete. At $h = H$, the complete LP closes the gap (Section A.2); an interrupted call retains its verified interval. The stopping depth is determined by the certificate, with no assumption that a short search always suffices.

Verifying long repeated sequences. For a repeated word $v = (v_0, \ldots, v_{P-1})$, let $f_r(t)$ be the expected reward at position r, extended to real calendar times by the exact formula in Section A.3. The total variation $\mathrm{TV}_I(f_r)$ measures its total change on I, counting increases and decreases positively.

Theorem 3 (Compressed verification)**.** *For every position r and positive-time interval I,*

$$\mathrm{TV}_I(f_r) \leq 2 \sum_{b<d} |A_{v_r,b} - A_{v_r,d}| \leq n(n-1). \tag{11}$$

For exact rational model data and rational tolerance $\eta \in (0, 1)$, a path formed from repeated-word blocks of lengths P_ℓ and repetition counts R_ℓ has verified bounds $\underline{S}_T \leq S_T \leq \overline{S}_T$ with $\overline{S}_T - \underline{S}_T \leq \eta T$ at all prefixes, using

$$O\left(n^2 \sum_\ell P_\ell \left[1 + \frac{n(n-1)}{\eta} \right] \log(1 + R_\ell) \right) \tag{12}$$

scalar function evaluations under the model in Section A.3. For any fixed finite mixture, the resulting lower reward bounds reduce all-prefix verification to finitely many endpoint checks on intervals where these bounds are periodic.

Why repeated actions admit compressed verification. Repeated actions still produce changing rewards: cumulative scores grow and the learning rate depends on the absolute round index. At each word position, the logits contain both $t^{1-\alpha}$ and $t^{-\alpha}$ terms, and the reward need not be monotone. The useful structure is that each pairwise logit difference has at most one turning point. Its logistic transform has bounded total variation, yielding (11) independently of the repetition count and initial score size. This bounds the number of intervals that need further refinement. After merging their change times, the reward enclosures are periodic on each resulting interval, so the lower prefix margin is affine across repetitions with the same remainder; its first and last values suffice. These endpoint checks use the constructed enclosures, while the true rewards continue to change. The bound counts scalar function evaluations; precision, prefix components, and endpoint checks have separate costs (Section A.3). General computable inputs retain direct finite verification.

4.3 The Actual Infinite Strategy Is Optimal

Step 3 transfers the finite guarantees for references evaluated from the original state to the actual game.

Theorem 4 (Long-run optimality)**.** *For every rational $\varepsilon_\infty \in [0, 1)$, the CCP–SRPP construction in Section 3.3, with schedules* (6) *and* (48)*, defines a computable randomized strategy satisfying*

$$J(\pi_{\varepsilon_\infty}) \geq V - \varepsilon_\infty. \tag{13}$$

In particular, $J(\pi_0) = V$. The learner retains its actual history, and the action at each round uses finite computation and finitely many random bits.

What the optimality statement adds. With $\varepsilon_\infty = 0$, the strategy is fixed before the evaluation horizon is chosen and attains the supremum over all behavioral strategies, including those that observe the learner's realized actions. It uses only the known model, the calendar index, and its own random bits, without requiring the value V or access to an optimal comparison strategy. The guarantee concerns the lower limit of expected averages in (2).

Why history decay alone is insufficient. The shared-suffix error in (7) vanishes for each fixed history length s, but later refreshes introduce longer histories. The term $sT^{-\alpha}$ can therefore remain large under repeated switching. In the two-action example of Section B.4, identical subsequent actions still leave an average-payoff gap of at least $1/8$ for $2s \leq T \leq ((s-1)/\log 3)^{1/\alpha}$. Under this shared suffix, recovery can take a time that grows faster than the initial history length. The refresh rule must make the latest reference both recent enough to cover T and old enough for its history error to shrink.

Why the construction works. Fix $v < V$. Proposition 1 and Theorem 2 give $L_j \geq v - \varepsilon_j$ for all sufficiently large j. For $2^k \leq T < 2^{k+1}$, let I_k be the latest refresh index through opportunity k. For any fixed $0 < \beta < \alpha$, Section A.4 proves

$$\Pr\Big(\Big\lceil\sqrt{k+1}\Big\rceil \leq I_k \leq \lfloor \beta k \rfloor\Big) \longrightarrow 1. \tag{14}$$

On this event, the lower bound gives $H_{I_k} \geq 2^{k+1} > T$, so the reference still covers the current round. The upper bound gives $t_{I_k} T^{-\alpha} \leq 2^{-(\alpha-\beta)k}$, so the history term in (7) tends to zero; its budget and initial-round terms also vanish. Conditional on $I_k = j$, independence of reference sampling and refresh decisions preserves the reference distribution μ_j. Averaging the transfer bound gives $F_T \geq v - \varepsilon_\infty - o(1)$ at all sufficiently large horizons. Taking the lower limit and then $v \uparrow V$ proves the theorem; the definition of V gives the reverse inequality when $\varepsilon_\infty = 0$.

How finite planning errors enter. For deterministic solvers with verified gaps $e_j = U_j - L_j$, let $r_k = \lceil\sqrt{k+1}\rceil$ and $u_k = \lfloor \beta k \rfloor$. Proposition 15 shows that finite-planning errors enter through $\Gamma_k = \sum_{j=r_k}^{u_k} \Pr(I_k = j) e_j$. Because the weights sum to at most one, $e_j \to 0$ suffices without a summability requirement. Earlier references affect the inherited history, already covered by the transfer bound; their planning gaps are not added again. All planning uses finite computation between rounds under the game-clock convention in Section 2.2.

Why short planning alone can fail. Consider

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \qquad q = (1/2, 1/2). \tag{15}$$

Action 2 gives zero immediate reward but trains the learner to choose action 1, which benefits the strategist when it plays action 1. For this game, $V = 1$, while exact model predictive control that maximizes cumulative reward over a window $h(t) = o(t^\alpha)$ achieves only $1/2$ (Proposition 16). A vanishing fraction of training actions can still drive the learner's desired-action probability to one, but a short window misses the later returns from that training. This example shows why CCP verifies a long continuation before accepting a short search.

5 Experiments

We test finite-plan accuracy, verification cost, and the effect of earlier history. The finite suite contains a training game, a coordination game, and six fixed-seed random games with 3×3 or 3×4 payoff matrices. The main settings use $\alpha = 1/2$, uniform priors, $H = 32$, $C \in \{1, 4\}$, and $\varepsilon = 0.01$; additional training horizons and sensitivity checks give 28 targets in total. We compute conditional expected rewards directly and use one CPU thread, with 120 seconds per planner call and 30 seconds per reference verification. Appendix B gives all inputs, checks, and results.

5.1 Finite Planning and Certificate Accuracy

CCP certifies all 28 targets within the requested tolerance, with a largest returned gap of 0.00216. Its exported distributions and global upper bounds also pass an independent interval-arithmetic replay. Table 1 compares CCP with the complete count-flow LP: four targets stop before $h = H$, while the other 24 use the complete fallback. For the training game with $H = 128$ and $C = 4$, CCP obtains $\Phi_{H,C} = 0.7463$, compared with 0.6526 for MPC-16 and 0.5313 for the best constant and greedy policies. Figure 1(a) shows the early payoff cost and later gain of training.

Table 1: Finite certificates at tolerance 0.01. Closed counts verified gaps meeting the tolerance; $h < H$ counts early stops. Times are medians over all targets in each row. All full-LP runs also close their gaps.

Cases	N	Closed	$h < H$	Max. $U - L$	CCP (s)	Full (s)
Training	8	8	0	$< 10^{-10}$	0.62	0.43
Coordination	2	2	2	$< 10^{-10}$	0.02	0.09
Random 3×3	6	6	0	$< 10^{-10}$	4.68	4.48
Random 3×4	6	6	2	0.0022	5.84	6.37
Sensitivity	6	6	0	$< 10^{-10}$	2.47	2.26

5.2 Cost of Verifying Long References

We compare direct and compressed interval evaluation on the same two fixed reference templates in five games. Compressed verification completes all 50 references through $H = 2^{64}$ and keeps the normalized reward-envelope width below 0.01. Direct evaluation completes 20 of 30 cases through 2^{18} under the 30-second limit; unfinished cases remain in the record. All completed paired cases pass a direct interval check at every prefix. At $H = 2^{14}$, compression speeds up verification by a median paired factor of $24.6\times$ across the ten references. Figure 1(b) shows the scalar evaluation counts for the fixed training reference; Appendix B reports every game and template. Compression has overhead on short references, while its cost grows slowly with the number of repetitions. These horizons describe represented references; the experiment measures verification work.

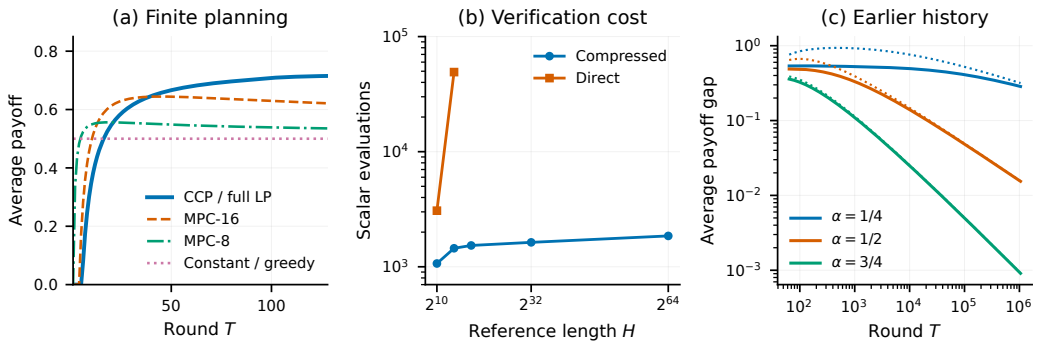


Figure 1: Three direct checks. (a) Training game, $H = 128$, $C = 4$: CCP and the full LP have the same reward curve; constant and greedy play also coincide. (b) A fixed training reference: 16 training actions followed by action 1. Direct points show completed runs; the 2^{18} direct run reaches its time limit. (c) Shared-suffix transfer with $s = 32$: solid curves are observed gaps, and dotted curves are the tanh-sum bounds.

5.3 Effect of Earlier History

We next use the shared-suffix game in Section B.4, with different initial histories of length $s \in \{8, 32\}$ followed by identical calendar-indexed actions. Figure 1(c) compares the resulting average-payoff gap with the state-dependent bound in (47). At $T = 2^{20}$ and $s = 32$, the gaps are 0.2851, 0.0154, and 0.00091 for $\alpha = 1/4$, $1/2$, and $3/4$, respectively. The six cases satisfy all three tested transfer bounds at every checked prefix. This directly shows why a newly introduced reference needs time for the effect of earlier history to shrink.

6 Discussion and Conclusion

We use finite prefix guarantees to connect a computable planning problem to optimal long-run play against a learner that keeps its history. CCP supplies global finite certificates, compressed evaluation reduces the work of verifying long repeated references, and sparse refresh transfers these guarantees into one continuing interaction. The resulting strategy attains the original lower limit of expected average payoff.

The experiments check finite certificates, repeated-reference verification, and shared-suffix transfer directly. All finite targets close their requested gaps, four permit early stopping, and long repeated references can be verified with far fewer scalar evaluations than direct computation. The history experiment also shows that a fixed earlier discrepancy can remain important for many rounds when the learning rate decreases slowly.

Limitations. The guarantees apply to known payoff matrices and the specified full-history Hedge update; extensions to unknown games or other learning rules remain outside their scope. The long-run theorem gives asymptotic optimality, while slow recovery from earlier play can delay good performance at practical horizons. Our experiments test finite certificates, verification cost, and history transfer; they do not establish how quickly the complete sparse-refresh strategy approaches the optimal long-run value.

AI USE STATEMENT

Generative AI assisted with experimental code generation and manuscript polishing. The authors manually reviewed the entire manuscript and take full responsibility for its content.

ETHICS STATEMENT

The reported experiments use synthetic objectives and involve no personal data or human participants.

REPRODUCIBILITY STATEMENT

We provide the code to reproduce all experiments reported in this paper.

REFERENCES

Eshwar Ram Arunachaleswaran, Natalie Collina, and Jon Schneider. Pareto-optimal algorithms for learning in games. In *Proceedings of the 25th ACM Conference on Economics and Computation*, pp. 490–510, 2024.

Angelos Assos, Yuval Dagan, and Constantinos Daskalakis. Maximizing utility in multi-agent environments by anticipating the behavior of other learners. *Advances in Neural Information Processing Systems*, 37:38769–38798, 2024.

Angelos Assos, Yuval Dagan, and Nived Rajaraman. Computational intractability of strategizing against online learners. *arXiv preprint arXiv:2503.04202*, 2025.

William Brown, Jon Schneider, and Kiran Vodrahalli. Is learning in games good for the learners? *Advances in Neural Information Processing Systems*, 36:54228–54249, 2023.

Yuan Deng, Jon Schneider, and Balasubramanian Sivan. Strategizing against no-regret learners. *Advances in neural information processing systems*, 32, 2019.

Hongcheng Dong and Yifen Mu. The optimal strategy against fictitious play in infinitely repeated games. In *2022 41st Chinese Control Conference (CCC)*, pp. 6852–6857. IEEE, 2022a.

Hongcheng Dong and Yifen Mu. The optimal strategy against the opponent adopting fictitious play algorithm in infinitely repeated $2\times$ 2 games. *Available at SSRN 4201849*, 2022b.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Xinxiang Guo and Yifen Mu. Globally optimal strategy against the hedge algorithm in repeated games. *Journal of Global Optimization*, 93(3):861–895, 2025.

Tinashe Handina, Tongxin Li, Kishan Panaganti, Eric Mazumdar, and Adam Wierman. Leveraging machine-learned advice in strategic interactions with no-regret learners. *arXiv preprint arXiv:2606.10261*, 2026.

Daniil Ryabko and Marcus Hutter. On the possibility of learning in reactive environments with arbitrary dependence. *Theoretical Computer Science*, 405(3):274–284, 2008.

Bhaskar Vundurthy, Aris Kanellopoulos, Vijay Gupta, and Kyriakos G Vamvoudakis. Intelligent players in a fictitious play framework. *IEEE Transactions on Automatic Control*, 69(1):479–486, 2023.

Yizhou Zhang, Yian Ma, and Eric Mazumdar. Learning to steer learners in games. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25, 2025.

A PROOFS OF THE THEORETICAL RESULTS

We prove the finite value characterization, certified finite planning, compressed verification, and long-run optimality in the order used in Section 4. The final subsection proves the short-window MPC claim. All prefix rewards below are sums of conditional expected rewards under the model in Section 2.

A.1 PROOF OF PROPOSITION 1: FINITE VALUE CHARACTERIZATION

A.1.1 PATH DISTRIBUTIONS PRESERVE EVERY EXPECTED PREFIX

Proposition 5 (Path-mixture equivalence)**.** *For every behavioral strategy π and finite H, there exists a distribution μ on $[m]^H$ such that*

$$\mathbb{E}_\pi \sum_{t=1}^{T} A_{a_t b_t} = \mathbb{E}_{w\sim\mu} S_T(w), \qquad 1 \le T \le H. \tag{16}$$

Conversely, sampling a word independently of the learner and executing it realizes the right-hand side at all these prefixes.

Proof. Let $\mathcal{F}_{t-1}$ contain the actual public history and the strategist's private state before round t. Conditional independence of simultaneous action draws gives

$$\mathbb{E}[A_{a_t b_t} \mid \mathcal{F}_{t-1}, a_t] = r_t(N(t-1), a_t).$$

Take μ to be the marginal law of $(a_1, \dots, a_H)$ under π. For each t, the displayed right-hand side depends only on the strategist's first t actions. Marginalizing this finite-word function and summing over $t \le T$ proves (16). In particular, we do not assume that learner samples remain independent after conditioning on a complete future strategist word. For the converse, first draw $w \sim \mu$ using randomness independent of the learner. Given w, the learner's conditional response at t is exactly the distribution in (3); iterated expectation proves the claim. □

The finite-dimensional distributions of the strategist's word induced by an infinite behavioral strategy are consistent. Thus the same comparison can be made on the compact product space $[m]^{\mathbb{N}}$. The effective algorithm, however, only samples finite references or generates their finite prefixes lazily.

A.1.2 MONOTONICITY

For any finite-reference law define its feasible prefix rate

$$\Phi_{H,C}(\mu) = \min\left\{1, \min_{1\le T\le H} \frac{\mathbb{E}_\mu S_T + C}{T}\right\}. \tag{17}$$

The expectation is taken before minimizing over T. In general, $\Phi_{H,C}(\mu)$ is not the mean of the individual sampled paths' feasible rates.

Lemma 6 (Monotonicity)**.** *$\Lambda_H(C)$ is nonincreasing in H and nondecreasing in C. For $h \le H$, a reliable $U_h \ge \Lambda_h(C)$ is an upper bound for $\Lambda_H(C)$ only with the same budget C. If $C \ge H$, then $\Lambda_H(C) = 1$.*

Proof. Truncating a feasible length-H law retains each of its first h constraints, giving $\Lambda_H(C) \le \Lambda_h(C)$. Increasing C relaxes every constraint. If $C \ge H$, nonnegative rewards make rate one feasible since $T - C \le 0$ for all $T \le H$. □

A.1.3 VALUE CHARACTERIZATION

Proof. Fix C and set $\ell_C = \inf_H \Lambda_H(C)$. The finite linear program (4) has a nonempty compact feasible set and attains its optimum. Extend an optimal length-H word law arbitrarily to infinite words. Choose successively convergent subsequences on one-letter marginals, two-letter marginals, and so on. A diagonal subsequence converges on each fixed finite prefix, and its limiting marginals are consistent. Their conditional probabilities define an infinite strategist word law, with arbitrary actions

after zero-probability prefixes. For each fixed T, sufficiently long members of this subsequence satisfy

$$\mathbb{E}S_T \geq \Lambda_H(C)T - C \geq \ell_C T - C.$$

Because S_T is a function of finitely many letters, the inequality passes to the limit. The limiting strategy has $J \geq \ell_C$, proving $\sup_C \ell_C \leq V$.

For the converse, choose $v < V$ and a strategy π with $J(\pi) > v$. Some finite T_0 satisfies $\mathbb{E}_\pi S_T \geq vT$ for every $T \geq T_0$. Define

$$C_v = \left\lceil \max\left\{0, \max_{1 \leq T < T_0} (vT - \mathbb{E}_\pi S_T)\right\}\right\rceil, \tag{18}$$

where an empty inner maximum is omitted. Then $\mathbb{E}_\pi S_T \geq vT - C_v$ for all T. Proposition 5 and truncation show $\Lambda_H(C_v) \geq v$ for every finite H. Monotonicity proves (9), and letting v increase to V proves (8). The algorithm neither computes C_v nor accesses this comparison strategy. □

A.2 Proof of Theorem 2: Certified Finite Planning

We first derive an exact finite formulation and a global upper bound, then control reward evaluation and probability rounding. These ingredients certify each continuation and ensure that the complete fallback terminates.

A.2.1 Complete count-flow formulation

For a search horizon h, let

$$\mathcal{C}_{t-1} = \{c \in \mathbb{N}_0^m : \mathbf{1}^\top c = t - 1\}.$$

The occupancy variable $y_t(c, a)$ is the probability of state c and action a in the *reference* process. The exact short problem is

$$\begin{aligned}
\max \quad & \ell && && (19)\\
\text{subject to} \quad & y_t(c,a) \geq 0, \quad \sum_a y_1(0,a) = 1, \quad 0 \leq \ell \leq 1, && \\
& \sum_a y_{t+1}(c,a) = \sum_{a: c_a > 0} y_t(c - e_a, a), && c \in \mathcal{C}_t,\ 1 \leq t < h,\\
& R_0 = 0, \quad R_t - R_{t-1} = \sum_{c \in \mathcal{C}_{t-1}, a} y_t(c,a) r_t(c,a), && 1 \leq t \leq h,\\
& R_t \geq \ell t - C, && 1 \leq t \leq h.
\end{aligned}$$

Every word law induces such a flow. Conversely, normalize the outgoing flow at positive-mass states:

$$\pi_t^{\mathrm{ref}}(a \mid c) = \frac{y_t(c,a)}{\sum_{a'} y_t(c,a')}. \tag{20}$$

Assign arbitrary actions at zero-mass states. Induction over the acyclic state graph recovers the prescribed flow, hence all R_t. Thus (19) has value $\Lambda_h(C)$ without restricting the original behavioral comparison class.

The nonterminal state count is

$$S(h) = \sum_{t=1}^{h} \binom{t+m-2}{m-1} = \binom{h+m-1}{m}. \tag{21}$$

There are $mS(h)$ state–action variables, in addition to R_t and ℓ. For fixed m, the graph size is polynomial in h.

A.2.2 A dual upper bound that is independent of the candidate library

Set $v_0(w) = 1$ and $v_s(w) = (S_s(w) + C)/s$ for $1 \leq s \leq h$. Finite minimax gives

$$\Lambda_h(C) = \min_{\lambda \in \Delta(\{0,\dots,h\})} \max_{w \in [m]^h} \sum_{s=0}^{h} \lambda_s v_s(w). \tag{22}$$

Indeed, these are the dual multipliers of the common-rate constraints, and their sum is one by balancing the rate variable. The coordinate v_0 enforces the upper cap of one; the lower cap is redundant since all evaluations are nonnegative. For any feasible λ, define

$$b_\lambda = \lambda_0 + C \sum_{s=1}^{h} \frac{\lambda_s}{s}, \qquad \omega_t = \sum_{s=t}^{h} \frac{\lambda_s}{s}. \tag{23}$$

Then the inner problem is $b_\lambda + \max_w \sum_{t=1}^{h} \omega_t r_t(w)$. It is solved on *all* count states by

$$D_{h+1}(c) = 0, \qquad D_t(c) = \max_a \{\omega_t r_t(c,a) + D_{t+1}(c+e_a)\}. \tag{24}$$

Replacing rewards by upper enclosures and rounding additions and nonnegative products upward yields a global upper bound $b_\lambda + D_1(0)$. Taking its minimum with one is valid. An arbitrary dyadic dual vector is admissible: optimality of the proposed dual is not required for upper-bound validity. In contrast, the dual value of a master restricted to continuation candidates is not an upper bound on (4).

A.2.3 Computable reward enclosures

For fixed finite (t, c, a), the weights can be evaluated as

$$q_b \exp\big(t^{-\alpha}(B^\top c)_b\big)\,, \tag{25}$$

so rational priors do not require evaluating exact logarithms. Powers, exponentials, and division by their positive sum are computable. Consequently, successive refinement returns rational $\underline{r}_t(c,a), \overline{r}_t(c,a)$ with

$$0 \le \underline{r}_t(c,a) \le r_t(c,a) \le \overline{r}_t(c,a) \le 1, \qquad \overline{r}_t(c,a) - \underline{r}_t(c,a) \le \nu$$

for any prescribed rational $\nu > 0$.

For example, a nonnegative rational x admits the finite Taylor enclosure

$$\sum_{k=0}^{M} \frac{x^k}{k!} \le e^x \le \sum_{k=0}^{M} \frac{x^k}{k!} + \frac{x^{M+1}}{(M+1)!} \frac{1}{1 - x/(M+2)}, \quad M + 2 > x. \tag{26}$$

The geometric remainder bounds all subsequent term ratios. Reciprocal bounds handle negative arguments, and range reduction can improve practical evaluation. If $\alpha = p/q$ is rational, bisection using integer-power comparisons encloses $t^{-\alpha}$. For a computable real $\alpha \in (0,1)$, computable logarithms and exponentials provide the same finite refinement property. Positive priors and denominators permit certified division after finite refinement.

A.2.4 Perturbation of the finite rate

Lemma 7 (Uniform reward perturbation)**.** *If two finite reward tables differ by at most ν entrywise, their prefix values at the same (h, C) differ by at most ν. Every fixed reference's normalized expected prefix rewards also differ by at most ν, simultaneously at all prefixes.*

Proof. The transition law of the count process depends on actions but not on the reward table. For any fixed word, the difference of its cumulative rewards through T is at most νT. This survives averaging over words, and hence changes each expression in (17) by at most ν. Taking a minimum and then a supremum preserves the bound. Equivalently, the same argument uses that each time layer of a feasible occupancy flow has total mass one. □

Solve (19) once with the lower reward table and once with the upper table using rational arithmetic, obtaining optimal values ℓ^- and U^+. The lower optimizer is a reference policy with true prefix rate at least ℓ^-, and

$$\ell^- \le \Lambda_h(C) \le U^+, \qquad 0 \le U^+ - \ell^- \le \nu. \tag{27}$$

A finite rational LP can be solved by exact algorithms; finite basis enumeration is a complete fallback if necessary.

A.2.5 DYADIC POLICIES AND THE LAW ACTUALLY EXECUTED

Lemma 8 (Rounding finite distributions). *For $p \in \Delta_s$, choose a power of two D and round the first $s-1$ probabilities down to multiples of $1/D$, assigning the remaining mass to the last coordinate. The resulting $\widehat{p}$ satisfies*

$$\mathrm{TV}(p, \widehat{p}) \le \frac{s-1}{D}. \tag{28}$$

If every conditional action law of a length-h count policy changes by at most θ/h in total variation, the induced word laws differ by at most θ.

Proof. Less than $(s-1)/D$ mass is removed from the first coordinates and added to the last one. The total variation distance equals that transferred mass. For conditional policies, couple the next actions maximally whenever the prefixes still coincide. They then have identical count states; the probability of the first discrepancy at any given step is at most θ/h. A union bound over the h steps gives the path-law bound. □

Convert (20) to dyadic conditionals with a common denominator $D \ge mh/\theta$. For the resulting law $\widehat{\mu}$, every S_T/T is a function in $[0,1]$, and thus

$$\mathbb{E}_{\widehat{\mu}} S_T \ge \mathbb{E}_{\mu} S_T - \theta T, \qquad 1 \le T \le h. \tag{29}$$

Return $L^- = \max\{0, \ell^- - \theta\}$. The resulting certified gap is at most $\nu + \theta$. All reachable states of the rounded policy are evaluated under the original model, including states that had zero probability before rounding.

Proposition 9 (A complete short solver). *For computable model data, finite h, C, and rational $\zeta > 0$, a finite deterministic computation returns a dyadically sampleable reference policy and rational L_h, U_h satisfying*

$$\mathbb{E} S_T \ge L_h T - C \ (T \le h), \qquad L_h \le \Lambda_h(C) \le U_h, \qquad U_h - L_h \le \zeta.$$

Proof. Use $\nu = \theta = \zeta/4$ in (27)–(29). There are finitely many coefficients and states, and every enclosure and rational solve terminates. The returned interval has width at most $\zeta/2$, leaving positive slack in the requested tolerance. Each conditional action is sampled with exactly $\log_2 D$ fair bits. The entire word requires finitely many such draws, or can be generated lazily. □

A.2.6 REPRESENTING RANDOMIZED PREFIXES AND REPEATED TAILS

Let a dyadic short count policy have horizon h. Exact forward probability propagation gives its endpoint law

$$\rho_h(c) = \Pr(N_W(h) = c), \qquad \mathbf{1}^\top c = h,$$

and reliable intervals for its expected cumulative rewards through h. There are at most $\binom{h+m-1}{m-1}$ endpoint states. Appending the same word v repeatedly, truncated to length $H-h$, gives for every $T > h$

$$\mathbb{E} S_T = \mathbb{E} S_h + \sum_{c \in \mathrm{supp}\, \rho_h} \rho_h(c)\, S_{h+1:T}(c, v). \tag{30}$$

The right-hand side needs no enumeration of all m^h prefix words. More generally, a finite tail program may depend on c, provided this dependence is fixed before actual play.

The initial family contains all primitive words with lengths $1, 2, 3$ (delete a word that is itself a repetition of a shorter word), used directly or after the short policy. Words start at their true reference calendar positions, so cyclic rotations need not be identified. Previously generated candidates may be retained. Additional finitely represented tails can be proposed, but none supplies an upper bound on the unsearched policy space.

A.2.7 THE CANDIDATE-MIXTURE OPTIMIZATION

For candidates $\pi_1, \dots, \pi_K$ and a finite proposal checkpoint set $\mathcal{G}$, solve the small LP

$$\max_{w \in \Delta_K,\, 0 \le z \le 1} z \quad \text{subject to} \quad \sum_{i=1}^{K} w_i \frac{\widehat{S}_T(\pi_i) + C}{T} \ge z, \quad T \in \mathcal{G}. \tag{31}$$

The proposal values may be high-precision approximations or conservative lower values. This LP only selects a candidate mixture; its value and dual are not global certificates. Each coefficient $(\widehat{S}_T + C)/T$ may exceed one. The coefficients are kept unchanged when mixing; the common variable $z \le 1$ imposes the intended cap.

Round the actual mixture weights to a common dyadic denominator and recompute the required reward enclosures for its nonzero support. Section A.3 provides a finite set $\mathcal{T}_*$ of endpoints sufficient to verify all prefixes. Return the rational lower rate

$$L_H = \max\left\{0, \min\left\{1, \min_{T \in \mathcal{T}_*} \frac{\sum_i \widehat{w}_i \underline{S}_T(\pi_i) + C}{T}\right\}\right\}. \tag{32}$$

If direct verification is used, take $\mathcal{T}_* = [H]$. The returned law is the rounded law $\widehat{w}$, not the original floating-point vector. Consequently all rounding error is already reflected in L_H; there is no unreported post-certification change to the strategy.

Proposition 10 (Continuation certificate)**.** *If $U_h \ge \Lambda_h(C)$ with $h \le H$, and a reference law μ has verified $\mathbb{E}_\mu S_T \ge L_H T - C$ for all $T \le H$, then*

$$L_H \le \Phi_{H,C}(\mu) \le \Lambda_H(C) \le U_h, \qquad \Lambda_H(C) - \Phi_{H,C}(\mu) \le U_h - L_H. \tag{33}$$

Proof. The lower inequality is the definition of a feasible rate. The middle inequality follows because the reference is one admissible word distribution. The upper inequality is Lemma 6. Subtract the lower bound from the upper bound. No optimality of the continuation, mixture, or proposal checkpoints is needed. □

A.2.8 FINITE TERMINATION

If $C \ge H$, any constant reference certifies $L = U = 1$. Otherwise, initialize $h_0 = \min\{H, \max\{8, C+1\}\}$, a constant reference, and the interval $[0, 1]$. At each depth h, call Proposition 9 with tolerance $\varepsilon/8$, keep the smallest verified upper bound, and retain the largest verified lower bound together with its reference. For $h < H$, form and verify the candidate mixture above; return when the retained interval has width at most ε, and otherwise replace h by $\min\{2h, H\}$. At $h = H$, use the full short policy as a length-H reference and return its certificate, or a tighter retained interval.

Proof. Each short solve terminates by Proposition 9. At $h < H$, the candidate family is finite, every prefix program is finite, and all-prefix verification terminates by Section A.3 or by the finite direct fallback. Proposition 10 gives a valid finite interval at each iteration. Taking the largest previously verified lower rate, while retaining the associated policy, and the smallest previously verified upper rate preserves validity. All such bounds refer to the same (H, C); CCP never combines bounds from different deficit budgets.

There are at most $1 + \lceil \log_2(H/h_0) \rceil$ short-solve depths, where h_0 is the initial depth. If no early interval closes, the process reaches $h = H$. The returned short policy then covers the entire target problem, with $U_h - L_h \le \varepsilon/8$. The incumbent interval is no wider, so the terminal branch also satisfies the requested gap. Each proposed mixture and each conditional policy has exactly normalized dyadic probabilities and is executable with finitely many random bits. This proves the theorem.

For the anytime statement, any earlier valid lower policy and upper certificate remain valid if a deterministic work budget interrupts refinement. One can always return the trivial constant law and interval $[0, 1]$ if no nontrivial short solve has completed. The returned gap records the accuracy reached before interruption. □

A.3 Proof of Theorem 3: Compressed Verification

We derive the repeated-word reward formula, bound its variation, and use the bound to construct all-prefix enclosures. All time coordinates are the original calendar indices; extending the formula to real $t > 0$ provides variation bounds between integer evaluation points.

A.3.1 Counts within a repeated word

Consider a block beginning at round τ with count c, where $\mathbf{1}^\top c = \tau - 1$. Let $v = (v_0, \dots, v_{P-1})$, let k be its action-count vector, let h_r count the first r letters, and set $u = k/P$. At position r in repetition j,

$$t = \tau + jP + r, \qquad N_w(t-1) = c + jk + h_r = ut + c + h_r - u(\tau + r). \tag{34}$$

Substituting into the actual Hedge law gives the exact logits

$$\begin{aligned} z_b(t) &= \log q_b + \beta_b t^{1-\alpha} + \gamma_{r,b} t^{-\alpha}, \\ \beta &= B^\top u, \qquad \gamma_r = B^\top [c + h_r - u(\tau + r)]. \end{aligned} \tag{35}$$

A final incomplete word is treated as a finite suffix. If a candidate has multiple blocks, update c exactly at each block boundary. For random prefixes, apply this calculation separately to every endpoint-count component in (30).

A.3.2 Proof of the uniform variation bound

Fix r and abbreviate $a_b = A_{v_r,b}$ and $d_{bc}(t) = z_b(t) - z_c(t)$. Differentiating (35) gives

$$d'_{bc}(t) = t^{-\alpha-1} \left[(1-\alpha)(\beta_b - \beta_c)t - \alpha(\gamma_{r,b} - \gamma_{r,c}) \right]. \tag{36}$$

The expression in brackets is affine in t, so the score difference has at most two monotone pieces on a positive interval, including all zero-coefficient and constant cases. Writing $p = \mathrm{softmax}(z)$, direct differentiation and symmetrization yield

$$f'_r(t) = \sum_{b<c} p_b(t) p_c(t) (a_b - a_c) d'_{bc}(t). \tag{37}$$

For $\sigma(x) = (1 + e^{-x})^{-1}$, the two corresponding denominator terms imply

$$p_b p_c \le \frac{e^{z_b + z_c}}{(e^{z_b} + e^{z_c})^2} = \sigma'(d_{bc}). \tag{38}$$

Therefore a computable local variation majorant is

$$\mathcal{V}_r(I) = \sum_{b<c} |a_b - a_c| \int_I \sigma'(d_{bc}(t)) |d'_{bc}(t)| \, dt, \qquad \mathrm{TV}_I(f_r) \le \mathcal{V}_r(I). \tag{39}$$

Each integral is the variation of $\sigma \circ d_{bc}$. It is at most two because $\sigma \in [0,1]$ and there are at most two monotone pieces. Thus

$$\mathcal{V}_r(I) \le 2 \sum_{b<c} |a_b - a_c| \le n(n-1),$$

proving (11). The bound also holds on unbounded positive intervals by monotone limits.

For rational data, the only possible interior turning point is

$$t^*_{bc} = \frac{\alpha(\gamma_{r,b} - \gamma_{r,c})}{(1-\alpha)(\beta_b - \beta_c)} \tag{40}$$

when the denominator is nonzero. Its location is tested exactly with rational arithmetic. The integral in (39) is then a sum of absolute changes of $\sigma(d_{bc})$ at interval endpoints and an included turning point. The prior contributes a constant to d_{bc} and does not alter its derivative.

A.3.3 A TERMINATING INTERVAL-ATOM EVALUATOR

For a fixed residue r, bisect the integer repetition set $\{0, \dots, R-1\}$. A node $[l, u]$ corresponds to $I = [\tau + lP + r, \tau + uP + r]$. Compute an enclosure of $\mathcal{V}_r(I)$ of width at most $\eta/16$. If its upper endpoint is at most $\eta/4$, evaluate f_r at the left endpoint with interval width at most $\eta/4$ and enlarge that interval in both directions by the variation upper bound. Intersect with $[0, 1]$ and assign it to every repetition in the node. The assigned interval contains every represented reward and has width at most $3\eta/4$. If the variation test fails, split the integer node into its two halves. A singleton is evaluated directly to width at most η.

Each failed nonsingleton test has $\mathcal{V}_r(I) > 3\eta/16$. At a fixed depth the time intervals have disjoint interiors, and their variation majorants sum to at most the whole block's majorant. Hence at most $16\mathcal{V}_r/(3\eta)$ nodes at that depth require splitting. The tree depth is at most $\lceil \log_2 R \rceil$, so the visited-node count is

$$O\left(\left[1 + \frac{\mathcal{V}_r}{\eta}\right] \log(1 + R)\right). \tag{41}$$

Each node evaluates $O(n^2)$ scalar expressions. Summing over positions and blocks proves (12). The slack in the acceptance threshold avoids an equality decision at the threshold. Reliable endpoint quantization must be outward; it is charged to the stated interval width.

Store each accepted atom as integer repetition endpoints and rational reward endpoints. Multiplying a reward endpoint by the number of complete or partial repetitions gives a cumulative enclosure using integer/rational arithmetic. Because each round has enclosure width at most η,

$$\underline{S}_T \le S_T \le \overline{S}_T, \qquad \overline{S}_T - \underline{S}_T \le \eta T, \qquad 1 \le T \le H. \tag{42}$$

The same statement holds under exact convex combinations of paths or endpoint-count components. Prefix rewards can be prepared directly to the same per-round tolerance.

For general computable-real parameters, determining a zero denominator in (40) need not be decidable. A conservative interval test may be used where it succeeds. If it cannot resolve a local bound, finite subdivision and ultimately direct single-round evaluation still give (42) for finite H. The scalar bound (12) is not asserted for that fallback, and numerical bit costs are separate even in the rational case.

A.3.4 ALL-PREFIX VERIFICATION FROM FINITELY MANY ENDPOINTS

Lemma 11 (Endpoint reduction for periodic reward enclosures)**.** *Suppose a fixed finite candidate mixture has rational lower reward enclosures that are periodic between a finite set of integer change times. Let P_* be a common period, and merge all change times into E intervals. For each such interval and each residue modulo P_* that appears, checking the first and last prefix in that residue suffices to verify*

$$\underline{S}_T \ge LT - C$$

at every prefix. There are at most $2P_*E$ checks, plus any explicitly separated initialization prefixes.

Proof. Fix an interval with an unchanged reward vector for each absolute-time residue, and write $T = P_*j + r$. For a fixed r, the cumulative lower reward equals $a_r j + b_r$ because each complete period contributes the same reward and all preceding intervals only change the intercept. The fixed mixture preserves this affine form. Then

$$\underline{S}_{P_*j+r} - L(P_*j + r) + C$$

is affine in j, so its minimum over the finite integer range is attained at an endpoint. □

Finite prefixes contribute their explicit round boundaries to the merged set. For an atom describing one residue, include the first represented round and the first subsequent round at which that residue's enclosure changes. This handles arbitrary tail offsets; blocks need not start at the same calendar residue. If candidate periods differ, use their least common multiple, not an assumed common starting point. The cost of a large least common multiple is real and must be recorded. For words of lengths at most three, $P_* \le 6$.

A.3.5 Proof of Theorem 3

Proof. Equations (36)–(39) prove the uniform variation statement. The integer bisection construction yields (41), and summing its scalar evaluation cost over positions and blocks gives (12). All accepted reward intervals have per-round width at most η, yielding (42) simultaneously for every prefix. Exact probability averaging preserves these inequalities. Lemma 11 establishes the claimed finite verification set for a fixed mixture. □

Evaluation model and additional costs. The bound in (12) counts scalar powers and exponentials at the precision needed for each enclosure. For a randomized prefix, the tail cost is summed over its endpoint-count support in (30), whose size is at most $\binom{h+m-1}{m-1}$. Probability propagation, candidate-mixture optimization, atom merging, and the endpoint checks of Lemma 11 are additional operations. The endpoint count depends on the common period and the number of merged intervals, including all candidates with positive mixture weight. Calendar indices require $O(\log H)$ bits, while reward refinement and exact probability arithmetic may require further precision. Thus the scalar evaluation bound describes the repeated-block verification step; a bit-operation bound must also account for those precision and representation costs.

A.4 Proof of Theorem 4: Long-Run Optimality

We first bound the effect of different initial histories, then show that the sparse refresh rule selects a covered reference whose history error is small with probability tending to one. The final step transfers finite certificates to the actual infinite strategy.

A.4.1 Stability under inherited histories

Let $\kappa = \max_a \operatorname{osc}_b B_{ab}$. Subtracting a constant from one row of B shifts every cumulative logit by a common scalar and hence leaves Hedge unchanged. In stability proofs only, subtract each row minimum so that $0 \le B_{ab} \le \kappa$. This does not change the learning-rate schedule.

Lemma 12 (Normalized reward stability)**.** *For $a \in [0,1]^n$ and $f(z) = a^\top \operatorname{softmax}(z)$,*

$$|f(z) - f(\widetilde{z})| \le \|z - \widetilde{z}\|_\infty. \tag{43}$$

Proof. With $p = \operatorname{softmax}(z)$, $\partial f/\partial z_b = p_b(a_b - f)$. Since both a_b and f lie in $[0,1]$, $\|\nabla f\|_1 \le 1$. Integrate the derivative along the line segment and use ℓ_1–ℓ_∞ duality. □

Lemma 13 (Shared calendar suffix)**.** *Two action paths that may differ through round s but agree at each calendar index $s+1, \dots, T$ satisfy*

$$\frac{|S_T - \widetilde{S}_T|}{T} \le \frac{s}{T} + \frac{2\kappa s}{1-\alpha} T^{-\alpha}, \qquad T > s. \tag{44}$$

The bound is uniform in both initial prefixes and the shared suffix.

Proof. The count difference after s is a constant D with $\|D\|_1 \le 2s$. At round $t > s$ its logit effect has infinity norm at most $2\kappa s t^{-\alpha}$ after the row translation. The actions coincide, so Lemma 12 bounds the reward difference by the same quantity. The first s rewards differ by at most s. Finally,

$$\sum_{t=s+1}^{T} t^{-\alpha} \le \sum_{t=1}^{T} t^{-\alpha} \le T^{1-\alpha}/(1-\alpha).$$

Divide by T to conclude. □

If the reference law obeys $\mathbb{E}S_T^{\mathrm{ref}} \ge L_j T - j$ for $T \le H_j$, and the actual path follows its calendar suffix after s, then

$$F_T^{\mathrm{actual}} \ge L_j - \frac{j+s}{T} - \frac{2\kappa s}{1-\alpha} T^{-\alpha} \tag{45}$$

up to the next switch and while $T \le H_j$. The two paths may be probabilistically dependent; the comparison bound is pointwise for their conditional-reward sums. The reference must still retain its prescribed law under whatever event is conditioned on.

A.4.2 STATE-DEPENDENT HISTORY BOUND USED IN THE EXPERIMENTS

For $z, d \in \mathbb{R}^n$, softmax satisfies

$$\mathrm{TV}(\mathrm{softmax}(z+d), \mathrm{softmax}(z)) \le \tanh(\mathrm{osc}(d)/4) \le \mathrm{osc}(d)/4. \tag{46}$$

To prove this, put $M = e^{\mathrm{osc}(d)}$. For any event with probability x under $\mathrm{softmax}(z)$, its probability under $\mathrm{softmax}(z+d)$ is at most $Mx/(1-x+Mx)$. Maximizing the increase over $x \in [0,1]$ gives $(\sqrt{M}-1)/(\sqrt{M}+1)$; taking the supremum over events proves the total variation bound, and $\tanh u \le u$ gives the second inequality.

For two paths that agree after round s, define $D = N(s) - N^{\mathrm{ref}}(s)$ and $\Delta_B = \mathrm{osc}_b(B^\top D)_b$. Their logit difference at each later round is $t^{-\alpha} B^\top D$. The common action has a reward vector in $[0,1]^n$, so its expected reward difference is at most the total variation distance between the learner distributions. Summing (46) and using $\sum_{t=s+1}^{T} t^{-\alpha} \le T^{1-\alpha}/(1-\alpha)$ therefore gives

$$\frac{|S_T - S_T^{\mathrm{ref}}|}{T} \le \frac{|S_s - S_s^{\mathrm{ref}}|}{T} + \frac{1}{T}\sum_{t=s+1}^{T} \tanh\left(\frac{\Delta_B}{4t^\alpha}\right) \le \frac{|S_s - S_s^{\mathrm{ref}}|}{T} + \frac{\Delta_B}{4(1-\alpha)} T^{-\alpha}. \tag{47}$$

If $\Delta_B = 0$, the two logits differ by a common scalar, so all subsequent learner distributions and rewards coincide. This also supplies the equal-relative-score control in the history experiment.

A.4.3 AN INTEGER-DEFINED SPARSE REFRESH LAW

Use the deterministic schedules

$$t_j = 2^j, \quad H_j = 2^{j^2}, \quad C_j = j, \quad d_j = \left\lceil \sqrt{\lceil \log_2(j+1) \rceil} \right\rceil, \quad b_j = \lceil \log_2(j d_j) \rceil, \quad \varrho_j = 2^{-b_j}. \tag{48}$$

All ceilings are computable by comparisons of integers and their squares or powers of two. At $j = 1$, $b_1 = 0$, so the first opportunity refreshes with probability one. For every j,

$$\frac{1}{2jd_j} < \varrho_j \le \frac{1}{jd_j}. \tag{49}$$

Read b_j fair bits and refresh exactly when all are zero. Keep the refresh stream independent of the reference-sampling streams and of the learner. The finite solver has deterministic inputs and output for every j; any heuristic seed and any work limit are fixed from the model and index before play. The program may defer the solve until a refresh actually occurs.

A reference that has expired is not used outside its certified index range. The algorithm plays a fixed action until the next refresh.

A.4.4 A COVERED REFERENCE WITH A SMALL HISTORY ERROR

Fix a constant $0 < \beta < \alpha$ for analysis only. For $2^k \le T < 2^{k+1}$, let I_k denote the latest successful refresh index through k, and define

$$r_k = \lceil \sqrt{k+1} \rceil, \qquad u_k = \lfloor \beta k \rfloor, \qquad G_k = \{ r_k \le I_k \le u_k \}. \tag{50}$$

For sufficiently large k, $2 \le r_k \le u_k$. On G_k, the latest reference has horizon

$$H_{I_k} = 2^{I_k^2} \ge 2^{k+1} > T,$$

and its switch occurred after at most $s = 2^{I_k} - 1 \le 2^{u_k}$ prior actual actions. Thus it is both unexpired and sufficiently old for history recovery.

Lemma 14 (Probability of a recovered, unexpired reference)**.** *The good event satisfies*

$$\Pr(G_k^c) \le \mathcal{B}_k := \prod_{j=r_k}^{u_k} (1-\varrho_j) + \sum_{j=u_k+1}^{k} \varrho_j \longrightarrow 0. \tag{51}$$

More explicitly, with $c_0 = \sqrt{\log 2}/8$*,*

$$\begin{aligned} \mathcal{B}_k \le\ & \exp\Big[-2c_0\big(\sqrt{\log(u_k+1)} - \sqrt{\log r_k}\big)\Big] \\ & + 2\sqrt{\log 2}\big(\sqrt{\log k} - \sqrt{\log u_k}\big). \end{aligned} \tag{52}$$

Proof. Independence gives the exact expression

$$\Pr(G_k) = \left[1 - \prod_{j=r_k}^{u_k} (1 - \varrho_j)\right] \prod_{j=u_k+1}^{k} (1 - \varrho_j).$$

A union bound proves (51). For $j \geq 2$, elementary ceiling inequalities give

$$\sqrt{\log_2 j} \leq d_j \leq 4\sqrt{\log_2 j}, \qquad \frac{c_0}{j\sqrt{\log j}} \leq \varrho_j \leq \frac{\sqrt{\log 2}}{j\sqrt{\log j}}.$$

The function $x \mapsto 1/(x\sqrt{\log x})$ is decreasing for $x > 1$. Use $1 - p \leq e^{-p}$ for the product and compare both sums with integrals to obtain (52). Since $\log r_k \sim (\log k)/2$ and $\log u_k \sim \log k$, the exponent's magnitude diverges, while the second difference of square roots tends to zero. □

A.4.5 Residual errors of the actual reference distributions

Let the finite computation at index j return a dyadic law μ_j and a valid interval

$$\mathbb{E}_{\mu_j} S_T \geq L_j T - j \quad (T \leq H_j), \qquad L_j \leq \Lambda_{H_j}(j) \leq U_j, \qquad e_j = U_j - L_j \in [0, 1]. \tag{53}$$

The law in this inequality is the one actually sampled, after all probability rounding. Define

$$a_{k,j} = \varrho_j \prod_{i=j+1}^{k} (1 - \varrho_i), \qquad \Gamma_k = \sum_{j=r_k}^{u_k} a_{k,j} e_j, \tag{54}$$

and

$$\mathcal{E}_k = k2^{-k} + 2^{u_k - k} + \frac{2\kappa}{1-\alpha} 2^{u_k - \alpha k}. \tag{55}$$

The first term accounts for the finite deficit budget and the other two for the inherited prefix. Because $\beta < \alpha < 1$, $\mathcal{E}_k \to 0$.

Proposition 15 (Weighted transfer of finite computational errors)**.** *For the deterministic finite subroutine with certificates in* (53)*, every $v < V$, all sufficiently large k, and every $2^k \leq T < 2^{k+1}$,*

$$F_T \geq v - \mathcal{E}_k - \mathcal{B}_k - \Gamma_k. \tag{56}$$

Proof. Proposition 1 supplies a finite C_v such that $\Lambda_H(j) \geq v$ for all H once $j \geq C_v$. For sufficiently large k, $r_k \geq C_v$. Then (53) gives, for each $j \in [r_k, u_k]$,

$$L_j = U_j - e_j \geq \Lambda_{H_j}(j) - e_j \geq v - e_j.$$

Condition on the entire refresh stream. If $I_k = j$ lies in this range, the reference has not expired, and the actual actions coincide with its calendar suffix after $s = 2^j - 1$. Because reference-sampling streams are independent of the entire refresh stream, W_j still has law μ_j under this conditioning. The actual prefix may depend on earlier references, but Lemma 13 is uniform in that prefix. Apply (45), $j \leq k$, $T \geq 2^k$, and $s \leq 2^{u_k}$ to obtain the conditional bound

$$F_T(\text{given the refresh stream}) \geq v - e_j - \mathcal{E}_k.$$

The probability of $I_k = j$ is exactly $a_{k,j}$. On the bad event use only nonnegative rewards. Thus

$$F_T \geq \Pr(G_k)(v - \mathcal{E}_k) - \Gamma_k \geq v - \mathcal{E}_k - \mathcal{B}_k - \Gamma_k,$$

where the last step uses $v - \mathcal{E}_k \leq 1$. The argument holds at every calendar index in the stated interval, not only at refresh opportunities. □

A.4.6 Long-run guarantee and finite computation

Proof. For the exact integrated algorithm, Theorem 2 gives (53) with

$$e_j \leq \varepsilon_j = \min\{1/2, \varepsilon_\infty + 1/(j+1)\}.$$

Since $\sum_{j=r_k}^{u_k} a_{k,j} = \Pr(G_k) \leq 1$,

$$\Gamma_k \leq \varepsilon_\infty + \frac{1}{r_k + 1}.$$

Take the lower limit in (56), use $\mathcal{E}_k, \mathcal{B}_k \to 0$, and then let v increase to V. This proves $J(\pi_{\varepsilon_\infty}) \geq V - \varepsilon_\infty$. When $\varepsilon_\infty = 0$, the reverse inequality $J(\pi_0) \leq V$ follows from admissibility, proving equality. The more general residual statement is Proposition 15.

For effectiveness, every calendar integer and dyadic refresh probability in (48) is computed by a finite integer calculation. Each required CCP call terminates by Theorem 2, with a finite reference program and dyadic sampling laws. Only finitely many refresh opportunities precede any fixed finite round. At a refresh, simulate the new reference's unexecuted virtual prefix to the accessed calendar index, or evaluate that index directly when its representation permits it. This uses finite work and finitely many random bits, although no practical uniform latency bound is asserted. The actual count vector is never replaced by that virtual count. The construction uses neither V, C_v, an unknown optimal strategy, nor an equality test on computable-real model coefficients. □

A.5 Proof of the Short-Window MPC Claim

The training example in Section 4.3 separates long-run optimality from optimization over a short terminal window.

Proposition 16 (A sub-t^α terminal window can miss investment)**.** *For*

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \qquad q = (1/2, 1/2), \tag{57}$$

the long-run optimum is $V = 1$. An exact terminal-reward MPC policy with prediction length $h(t) = o(t^\alpha)$ has long-run payoff $1/2$.

Proof. Only action 2 increases the learner's relative score in favor of its action 1, and action 2 gives the strategist zero reward. Let $f_t = (t+1)^\alpha \log(t+1)$, put $\theta_t = \min\{1, f_t - f_{t-1}\}$, and $K_t = \lfloor \sum_{s=1}^{t} \theta_s \rfloor$. Since f_t is increasing and its increments tend to zero, $K_t = t^\alpha \log t + O(1)$. Also $K_t - K_{t-1} \in \{0, 1\}$ and $K_t \leq t$. Play action 2 exactly when K_t increases, and action 1 otherwise. The training fraction tends to zero, while the round-t relative logit K_{t-1}/t^α tends to infinity. Learner action 1 then has probability tending to one, so the average strategist reward tends to one. The upper bound $V \leq 1$ is immediate.

Now fix an MPC decision time t and any candidate plan with r training actions in its next $h(t)$ rounds. Compare it with the plan that replaces every one of those training actions by action 1. The existing training count is nonnegative, so each replacement gives at least $1/2$ immediate expected reward. The logistic derivative is at most $1/4$, and all future calendar times are at least t. All training-induced improvements during the window are therefore at most $rh(t)/(4t^\alpha)$ in total. The original candidate's advantage over never training is at most

$$r\left(\frac{h(t)}{4t^\alpha} - \frac{1}{2}\right).$$

For sufficiently large t, this is negative whenever $r > 0$. Every optimal terminal MPC plan then contains no training, and the executed policy eventually stops training. Its finite remaining cumulative score divided by t^α tends to zero, so learner action 1 has probability tending to $1/2$. The average reward tends to $1/2$. □

B EXPERIMENTAL DETAILS

The released artifact contains the fixed inputs, executable solvers, independent checks, exported certificates, and measurement records for the three experiments in Section 5. All reported runs use the current finite planner and verification routines.

B.1 GAMES AND FIXED PARAMETER GRIDS

The finite suite contains the training game (57), the coordination game $A = B = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 3/4 \end{smallmatrix}\right)$, and six random games. For $n \in \{3, 4\}$ and $s \in \{0, 1, 2\}$, generate $3 \times n$ matrices with entries $A_{ab} = X_{ab}/100$ and $B_{ab} = Y_{ab}/100$, where X is uniform on $\{0, \dots, 100\}$ and Y is uniform on $\{-50, \dots, 50\}$. We use NumPy PCG64 with seed $20260919 + 1000n + s$, draw all entries of X in row order before Y, and store the matrices as exact rational strings. The primary prior is uniform. The verification-cost suite uses the first random instance of each dimension and adds the cyclic game

$$A = \begin{pmatrix} 0 & 0 & 1/4 & 1 \\ 1/4 & 0 & 0 & 1 \\ 0 & 1/4 & 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 2/3 & -1/3 & -1/3 & 0 \\ -1/3 & 2/3 & -1/3 & 0 \\ -1/3 & -1/3 & 2/3 & 0 \end{pmatrix}, \qquad q_b = 1/4. \tag{58}$$

The primary finite grid has $H = 32$, $C \in \{1, 4\}$, $\alpha = 1/2$, and $\varepsilon = 0.01$ in all eight games. The training game also uses $H \in \{64, 128\}$ and $C \in \{1, 4, 16\}$. On the training game and the first 3×3 random game, we add $\alpha \in \{1/4, 3/4\}$ at $H = 32, C = 4$ with uniform priors, and $q_b = b/\sum_{d=1}^{n} d$ at $\alpha = 1/2$. This gives 28 targets, each evaluated with CCP, the complete count-flow LP, and four strategy baselines.

B.2 FINITE IMPLEMENTATION AND INDEPENDENT VERIFICATION

The floating-point LP proposes a count policy and a vector of prefix dual weights. We convert both to exactly normalized dyadic distributions with denominator 2^{40}. The proposed policy is propagated with exact rational probability masses, including newly reachable states created by rounding. Reward coefficients use 50-decimal-digit interval arithmetic; conversion to binary64 and subsequent nonnegative products and sums round outward. A separate Bellman recursion on the full short count graph verifies the upper bound for the same budget C. The final lower bound uses all prefixes of the actual rounded reference.

CCP starts at $h = \min\{H, \max\{8, C + 1\}\}$ and doubles h up to H. At each depth below H, it considers every primitive word of length at most three, both directly and after the short policy, and selects a mixture using all target prefixes as proposal checkpoints. The endpoint-count distribution propagates the random short prefix into each attached tail. For these small finite targets, direct interval evaluation verifies the selected mixture; the separate cost experiment tests the compressed evaluator. At $h = H$, the complete count policy supplies the fallback. Numerical proposal or verification failures retain the last valid interval and a failure status. Every formal target closes before the 120-second cap.

An independent implementation compares the word-enumeration LP with a separately formulated occupancy LP on the eight games at $H \in \{4, 6\}$ and $C \in \{0, 1\}$. All 32 comparisons agree to within 1.05×10^{-14}; this is a floating-point consistency check. A second independent check replays all 56 formal solver records using a separate 80-digit interval reward formula, exact rational probability propagation, and an exact rational Bellman evaluation of the exported dual witness. It checks the saved cumulative enclosures, final rates, and reference probability normalization without calling the planner's verification routines. All 56 records pass. The separately formulated occupancy LP is also run on all 28 formal targets; its numerical values agree with the complete planner intervals to the specified 10^{-8} diagnostic tolerance.

The constant baseline selects the best constant action by $\Phi_{H,C}$, and greedy play maximizes the current expected reward. MPC-8 and MPC-16 exhaustively solve the next $\min\{K, H - t + 1\}$ rounds on the count graph using the actual counts and absolute calendar indices, then execute the first action. Ties use the smallest action index; MPC rewards are evaluated in binary64. All baseline curves use the same model, target horizon, and deficit budget.

Table 2: All finite-prefix scores $\Phi_{H,C}$, rounded to four decimals. IDs 00–01 are training, 02–03 coordination, 04–09 the three 3×3 random games, and 10–15 the three 3×4 random games; each pair uses $C = 1, 4$. IDs 16–21 extend the training horizon, and 22–27 are sensitivity cases. The last case in each sensitivity group uses the nonuniform prior.

ID	H	C	α	CCP	Constant	Greedy	MPC-8	MPC-16
00	32	1	$1/2$	0.6308	0.5312	0.5312	0.5847	0.2500
01	32	4	$1/2$	0.7775	0.6250	0.6250	0.6784	0.7651
02	32	1	$1/2$	0.9137	0.9137	0.9137	0.9137	0.9137
03	32	4	$1/2$	1.0000	1.0000	1.0000	1.0000	1.0000
04	32	1	$1/2$	0.7754	0.7754	0.7754	0.7754	0.7754
05	32	4	$1/2$	0.8691	0.8691	0.8691	0.8691	0.8691
06	32	1	$1/2$	0.5797	0.5797	0.5797	0.5797	0.5797
07	32	4	$1/2$	0.6734	0.6734	0.6734	0.6734	0.6734
08	32	1	$1/2$	0.6241	0.4907	0.5968	0.6225	0.6041
09	32	4	$1/2$	0.7182	0.5844	0.6905	0.7162	0.7182
10	32	1	$1/2$	0.5588	0.5521	0.5566	0.5586	0.5588
11	32	4	$1/2$	0.6525	0.6459	0.6504	0.6523	0.6525
12	32	1	$1/2$	0.7690	0.7690	0.7690	0.7690	0.7690
13	32	4	$1/2$	0.8669	0.8669	0.8669	0.8669	0.8669
14	32	1	$1/2$	0.6412	0.6356	0.5996	0.5996	0.6356
15	32	4	$1/2$	0.7949	0.7947	0.6934	0.6934	0.7949
16	64	1	$1/2$	0.6274	0.5156	0.5156	0.5605	0.2500
17	64	4	$1/2$	0.7550	0.5625	0.5625	0.6074	0.7033
18	64	16	$1/2$	0.9656	0.7500	0.7500	0.7949	0.8908
19	128	1	$1/2$	0.6267	0.5078	0.5078	0.5432	0.2500
20	128	4	$1/2$	0.7463	0.5312	0.5312	0.5666	0.6526
21	128	16	$1/2$	0.8964	0.6250	0.6250	0.6603	0.7463
22	32	4	$1/4$	0.8982	0.6250	0.6250	0.8151	0.8936
23	32	4	$3/4$	0.6745	0.6250	0.6250	0.6490	0.6631
24	32	4	$1/2$	0.6826	0.4583	0.4583	0.5998	0.6667
25	32	4	$1/4$	0.8609	0.8609	0.8609	0.8609	0.8609
26	32	4	$3/4$	0.8702	0.8702	0.8702	0.8702	0.8702
27	32	4	$1/2$	0.8242	0.8239	0.8241	0.8242	0.8242

B.3 Repeated-reference verification

Each of the five games uses two fixed templates: 16 rounds of action 2 followed by repetitions of action 1, and 16 rounds of action 1 followed by repetitions of the word $(1, \dots, m)$. Both evaluators use $\alpha = 1/2$ and required normalized envelope width $\eta = 0.01$. Direct and compressed methods are paired at $H \in \{2^{10}, 2^{14}, 2^{18}\}$; compression additionally uses $H \in \{2^{32}, 2^{64}\}$. The compressed method implements Section A.3, including exact rational turning-point tests, interval variation bounds, integer bisection, and the common-period endpoint checks. Long horizon indices remain exact integers throughout. Direct verification sums interval rewards one round at a time. For every completed direct run, its exact rational prefix enclosures are checked against the compressed bounds at every prefix. Independent binary64 trajectory calculations additionally check every manageable compressed prefix through 2^{18}. A separate 80-digit interval implementation replays all 52 saved compressed certificates, including timing repetitions, and checks all 13,791 atoms from explicit counts, their variation bounds, coverage, and endpoint sets. Every certificate passes this replay, including those at $H = 2^{64}$.

Table 3 reports completion counts over the fixed grid. Evaluation counts include exponential and real-power calls; reused expressions are counted once. Times include reference evaluation and, for compression, construction of the all-prefix endpoints and width check; interpreter startup, output writing, and independent cross-checks are excluded. Each timed verification starts in a fresh process and has a 30-second limit. Incomplete direct runs retain their partial sums and a valid bound for the remaining rewards. Short references can cost more under compression: for the cyclic game with the second template at $H = 2^{10}$, compression takes 0.56 seconds and direct evaluation 0.33 seconds; at $H = 2^{14}$, the times are 0.97 and 5.44 seconds.

B.4 Shared-suffix history experiment

The history experiment uses the two-action game

$$B = \begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix}, \qquad A = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \qquad q = (1/2, 1/2). \tag{59}$$

Table 3: Verification over ten game–template pairs at each horizon. Counts are completed/attempted. Calls and times are medians over completed runs, with their denominators shown explicitly; all time-limited runs remain in the raw results. Direct evaluation is attempted only through 2^{18}.

Horizon	Comp.	Direct	Comp. calls	Direct calls	Comp. (s)	Direct (s)
2^{10}	10/10	10/10	2,051	4,096	0.16	0.30
2^{14}	10/10	10/10	2,249	65,536	0.17	4.88
2^{18}	10/10	0/10	2,387	–	0.18	–
2^{32}	10/10	–	2,809	–	0.20	–
2^{64}	10/10	–	3,618	–	0.25	–

Table 4: Every game–template pair: compressed / direct verification time in seconds. Template 1 has an action-2 prefix and action-1 tail; template 2 has an action-1 prefix and a cyclic tail. > 30 denotes a retained time-limited run; – denotes a method not attempted at that horizon. Exact evaluation counts and all intervals are in the raw records.

Game / template	2^{10}	2^{14}	2^{18}	2^{32}	2^{64}
Training / 1	0.06 / 0.13	0.08 / 2.05	0.09 / > 30	0.10 / –	0.11 / –
Training / 2	0.02 / 0.13	0.02 / 2.24	0.02 / > 30	0.04 / –	0.06 / –
Coordination / 1	0.02 / 0.15	0.02 / 2.43	0.03 / > 30	0.03 / –	0.05 / –
Coordination / 2	0.04 / 0.16	0.06 / 2.69	0.06 / > 30	0.08 / –	0.12 / –
Cyclic / 1	0.22 / 0.33	0.24 / 5.59	0.26 / > 30	0.27 / –	0.33 / –
Cyclic / 2	0.56 / 0.33	0.97 / 5.44	1.04 / > 30	1.19 / –	1.27 / –
Random 3×3 / 1	0.09 / 0.29	0.10 / 4.75	0.11 / > 30	0.13 / –	0.17 / –
Random 3×3 / 2	0.23 / 0.31	0.25 / 5.00	0.27 / > 30	0.32 / –	0.44 / –
Random 3×4 / 1	0.25 / 0.38	0.27 / 6.36	0.28 / > 30	0.31 / –	0.37 / –
Random 3×4 / 2	0.74 / 0.40	1.20 / 6.46	1.35 / > 30	1.47 / –	1.70 / –

We take $s \in \{8, 32\}$, $\alpha \in \{1/4, 1/2, 3/4\}$, and $T \le 2^{20}$. The actual prefix always chooses action 1, while the reference prefix alternates actions 2 and 1. After round s, both paths alternate actions 2 and 1 at the same absolute calendar indices. The reference relative score is then zero or -1, while the actual relative score is s or $s - 1$. Thus their learner distributions differ even though every subsequent action agrees.

This example also gives a direct lower bound on the time needed for the history effect to fade. For $t \le ((s-1)/\log 3)^{1/\alpha}$, the reference probability of learner action 2 is at least $1/2$, and the actual probability is at most $1/4$. The reference also earns at least as much during the first s rounds, so the cumulative gap is at least $(T - s)/4$ over this range. Consequently,

$$\frac{S_T^{\mathrm{ref}} - S_T}{T} \ge \frac{1}{8}, \qquad 2s \le T \le ((s-1)/\log 3)^{1/\alpha}. \tag{60}$$

This lower bound describes the specified shared suffix and the initial histories above.

We compute conditional expected reward curves in binary64 and check the generic bound (44), the tanh-sum bound, and the relaxed state bound (47) at every integer prefix after s. Section A.4.2 proves the two state-dependent bounds. All 6,291,336 checked prefixes satisfy these inequalities. The lower-bound interval in (60) is also checked wherever nonempty. A separate control uses distinct count vectors with zero relative-score difference and confirms identical subsequent rewards to numerical precision. These are deterministic mechanism measurements, so they require no Monte Carlo confidence intervals.

B.5 Computation and reproducibility

All experiments run locally on an Apple M5 CPU, with one numerical-library thread. The recorded environment uses Python 3.9, NumPy 2.0.2, SciPy 1.13.1, mpmath 1.4.1, and Matplotlib 3.9.4; complete versions and hardware details are stored with the results. We clear planner-specific count caches between calls and use fresh verifier processes. Three timing repetitions with planner caches cleared are recorded for finite targets 01, 05, and 20. The first training template at $H = 2^{14}$ is also repeated three times under both verification methods, each in a fresh process. Table 6 reports the observed timing variation. The fixed matrices, manifests, raw records, timing repetitions, source hashes, certificate checker, and plotting scripts are included in the artifact. The raw tables retain every attempted case and distinguish completed, time-limited, and failed verification outcomes.

Table 5: History-induced average-payoff gaps and upper bounds at $T = 2^{20}$. The state-dependent bounds capture the observed differences much more closely than the general bound.

s	α	Observed gap	Tanh bound	State bound	General bound
8	$1/4$	0.082365	0.083024	0.083337	0.666674
8	$1/2$	0.003893	0.003896	0.003909	0.031258
8	$3/4$	0.000233	0.000233	0.000246	0.001961
32	$1/4$	0.285090	0.317125	0.333349	2.666697
32	$1/2$	0.015422	0.015506	0.015639	0.125031
32	$3/4$	0.000909	0.000913	0.000988	0.007843

Table 6: Three repeated timings for each fixed target. Finite timings clear planner-specific caches within the same process; reference-verification timings use fresh processes. Imports and output writing are outside the measured routines.

Target	Method	Min. (s)	Median (s)	Max. (s)
Finite 01	FULL	0.080	0.086	0.086
Finite 01	CCP	0.121	0.136	0.138
Finite 05	FULL	4.264	4.814	4.962
Finite 05	CCP	4.549	5.315	5.402
Finite 20	FULL	3.738	4.240	4.275
Finite 20	CCP	4.488	5.009	5.076
Training reference	Compressed	0.081	0.106	0.108
Training reference	Direct	2.054	2.800	2.902

The implementation checks cover word/count LP agreement, dyadic normalization, all-count upper bounds, exported-policy replay, nonzero histories, absolute calendar indices, turning points, periodic endpoint reduction, and exact long-horizon indexing. Corrupted probability masses, reference states, lower curves, and upper certificates are rejected by dedicated checks. The finite and compression certificates use interval arithmetic; the history curves and strategy-baseline scores are numerical measurements. The experiments assess finite certificates, verification work, and shared-suffix transfer under the stated conditions; the infinite optimality result follows from Theorem 4.