EXACT AVERAGING AND DISCOUNTING FOR MONTE CARLO COUNTERFACTUAL REGRET MINIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Counterfactual regret minimization (CFR) is the standard method for solving two-player zero-sum games with imperfect information, and its error bound concerns an average strategy that weights iteration t at an information set by the probability that the player's own actions lead there. Tabular CFR accumulates that average exactly, whereas outcome-sampling Monte Carlo CFR updates only the information sets on one sampled playthrough per player and iteration, while the average keeps changing at the information sets it skips. The rules in use account for them differently, and which average they return depends on where they are applied. On the player's own playthrough the opponent does not explore, so an information set can stay unvisited while its weight keeps growing. We show that the missed weight is already recorded in the player's tables, as the growth of the exact sum at the predecessor's action leading to the information set. Reading it at each visit, on the way down, and once at the end returns the exact reach-weighted average of the strategies the sampled run produced, for uniform, linear and polynomial weights alike. A prefix-product table replays skipped discount factors at one subtraction. The coefficient distortion of a skipped positive-regret factor is bounded when discounted CFR's exponent exceeds one, and unbounded below that and for the strategy sum. On nine games from 12 to 524,288 information sets, our implementation of OpenSpiel's rule returns strategies 1.7 to 9 times as exploitable as the exact average after 10^7 iterations, a gap that grows with training, while the opponent's-playthrough rule stays within five per cent and the exact rule costs 0.3–4.5% more time. Both are cheap enough to be defaults for outcome-sampling implementations.

1 INTRODUCTION

An opponent who also learns from the game cannot be modelled as a fixed objective to optimize. The strongest poker agents are built on equilibrium computation, not value estimation (Brown & Sandholm, 2018; 2019b). The model is the two-player zero-sum extensive-form game with imperfect information (Section 3): a player cannot tell apart the histories inside one of its information sets and must play them all with one strategy, and an approximate Nash equilibrium bounds what any opponent can gain against it. Computing one is well posed; the obstacle is scale.

Counterfactual regret minimization (CFR) made such computations routine (Zinkevich et al., 2007): it runs regret matching (Hart & Mas-Colell, 2000) at every information set, and its guarantee concerns the average strategy, which weights iteration t at information set I by the reach probability $\pi_i^{\sigma^t}(I)$. Outcome-sampling Monte Carlo CFR replaces each player's full traversal with a sampled playthrough (Lanctot et al., 2009), and discounted variants multiply the tables by a factor of the global iteration (Brown & Sandholm, 2019a); a visit must then account for the iterations in between.

Two rules are in use for that accounting. The optimistic rule of the original algorithm adds the missed iterations at the current weight; the stochastic rule adds the current iteration's weight divided by the probability of having sampled the visit (Lanctot et al., 2009; Lanctot, 2013). Implementations disagree about which to use and where to apply them: OpenSpiel accumulates on the update player's own information sets (Lanctot et al., 2019), the original algorithm on the opponent's (Lanctot, 2013, Alg. 3), and ports of discounted CFR apply the current factor once per visit. Unbiased regret estimates do not make the stored average exact; we measure the accumulation error of the rules in use on matched trajectories.

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.

The average is exactly recoverable from the tables a solver already stores. The weight an information set accrues between two visits equals the growth of the exact sum at its predecessor, for the action that leads to it (Lemma 1), so reading that growth at each visit and once per information set in a final pass reproduces the exact average for uniform, linear and polynomial weights alike, at one stored number per information set (Section 4.2 and Theorem 1). Discount factors compose, so the product of skipped factors is the exponential of a difference of shared log-prefix sums (Section 5), and Proposition 5 separates the exponents at which a skipped positive-regret factor distorts its coefficient by at most a constant from those at which no bound exists.

On nine games from 12 to $524{,}288$ information sets, with every rule on one shared regret trajectory per seed, OpenSpiel's rule returns strategies 1.7–9 times as exploitable as the exact average after 10^7 iterations (Table 1 and Figure 3). OpenSpiel's own solver reproduces the effect on Leduc at 10^6 iterations (1.41; Kuhn 1.32, interval covering 1) (Table 11). The gap grows with training. On the own playthrough an information set's visit probability need not track its weight and can be zero while the weight is positive, so regret matching leaves 6.1% of the exact weight unvisited on Goofspiel(4) (Proposition 1); on the opponent's playthrough the visit probability is proportional to the weight, and the same rule stays within five per cent (Proposition 2). The exact rule matches brute force to 10^{-9} at 0.3–4.5% overhead (Table 20), and on linear trajectories the single-current-factor discount semantics is 1.5–6.5 times as exploitable as exact replay (Table 18).

- **What each averaging rule estimates.** On the own playthrough the stochastic rule's increment estimates the exact average up to a support constant that moves with the opponent's play; on the opponent's playthrough the constant is fixed by the game tree (Propositions 1 and 2).
- **An exact average and exact discounting from stored tables.** Reading the predecessor's growth at each visit, and settling the visited information sets once at read-out, with the uniform fallback for never-visited ones (Proposition 6), reproduces the exact average for uniform and weighted averages alike, with no sampling correction (Lemma 1 and Theorem 1); Section 5 does the same for discount factors.
- **A controlled measurement of the rules in use.** One shared regret trajectory per seed isolates the averaging, and the exact rule agrees with brute force to 10^{-9} at 0.3–4.5% overhead, so every difference between rules is due to the averaging estimator and its placement.

2 Related work

The averaging rules as they stand. Lanctot et al. (2009) introduced MCCFR and proved that sampled counterfactual regrets are unbiased. The average strategy is treated in the thesis (Lanctot, 2013, §4.4), which names the optimistic and stochastically-weighted rules and places both on the opponent's information sets "to enforce the unbiasedness of the update". It calls the optimistic rule a heuristic and proposes lazy-weighted averaging as a correct alternative, which it then finds slower on Bluff and Goofspiel. Lazy-weighted averaging is not exact either (Appendix A and Section 4.4). OpenSpiel (Lanctot et al., 2019) applies the stochastic rule on the update player's own information sets, a placement that biases the average; its Python solver's documentation still says that "the other players update average strategies".

Weighted and discounted variants. CFR+ (Tammelin, 2014; Bowling et al., 2015), linear and discounted CFR (Brown & Sandholm, 2019a), predictive variants (Farina et al., 2021; Xu et al., 2024), and hyperparameter schedules (Zhang et al., 2026) change the weights of the average and the factors applied to the regrets as functions of the global iteration. They are analysed in full-tree CFR, where every table sees every factor. Brown & Sandholm (2019a) port discounting to sampling by applying it in bulk at period boundaries; Section 5 treats the per-visit port. Burch et al. (2019) repair the regret-to-equilibrium argument for alternating updates, and our exact average is the profile their analysis covers. Pruning (Brown & Sandholm, 2017) and Lazy-CFR (Zhou et al., 2020) skip information sets deliberately and patch their averages on return; the recurrence applies to their catch-up updates too.

Other estimators and representations. Pure CFR (Gibson, 2014) counts visits of sampled pure strategies. Variance reduction for the regrets (Gibson et al., 2012; Schmid et al., 2019; Davis et al.,

2020), alternative samplers (Johanson et al., 2012b), and the regression-fitted averages of Deep CFR and its successors (Brown et al., 2019; Steinberger et al., 2020; Li et al., 2020; McAleer et al., 2023) change the estimates or the representation, not which iterations a sampled visit stands for. Stochastic regret minimization in the sequence form (Farina et al., 2020) averages sequence-form strategies, whose weights are the reach probabilities themselves; the recurrence maintains exactly that average from a behavioral sampler. The recurrence is sequence-form flow conservation summed over iterations. Its novelty is the lazy maintenance along sampled visits, with one watermark per information set. The average-strategy bookkeeping of the Alberta theses (Gibson, 2014; Burch, 2018; Burch et al., 2012) counts sampled pure realizations or propagates reach along visits; none reads the predecessor's stored sum. Table 2 lists each rule's event, how the missed iterations enter, and whether it stays exact when the local policy changed between two of its events. CFR-BR and purification (Johanson et al., 2012a; Ganzfried et al., 2012) show that a strategy other than the exact average can be less exploitable, so fidelity to the target and quality of the strategy are separate questions; we measure both: policy distance to the exact average and exploitability.

Deployed solvers. Averaged sampled CFR computes the blueprints of Libratus (Brown & Sandholm, 2018) and Pluribus (Brown & Sandholm, 2019b); DeepStack (Moravčík et al., 2017) and online search (Lisý et al., 2015; Schmid et al., 2023) re-solve with full-width variants. OpenSpiel's solver is the reference implementation many papers report against.

3 Background

Games. A two-player zero-sum extensive-form game has a finite tree of histories h with terminal utilities $u_1(z) = -u_2(z)$. A non-terminal history belongs to chance or to a player $i \in \{1, 2\}$, who observes only its information set $I \in \mathcal{I}_i$, with legal actions $\mathcal{A}(I)$. A behavioral strategy σ_i assigns a distribution $\sigma_i(I, \cdot)$ over $\mathcal{A}(I)$ to every $I \in \mathcal{I}_i$. The probability of reaching h factors as $\pi^\sigma(h) = \pi_c(h)\,\pi_1^\sigma(h)\,\pi_2^\sigma(h)$ into chance and the two players' own contributions. Under perfect recall every $h \in I$ passes through the same sequence of player i's information sets and actions, so $\pi_i^\sigma(h)$ is the same for all $h \in I$, written $\pi_i^\sigma(I)$. Every $I \in \mathcal{I}_i$ other than a first decision point then has a unique *predecessor* $(I^\uparrow, a^\uparrow)$, the last information set of player i and the action taken there before I, and

$$\pi_i^\sigma(I) = \pi_i^\sigma(I^\uparrow)\,\sigma_i(I^\uparrow, a^\uparrow), \qquad \pi_i^\sigma(I) = 1 \text{ at first decision points.} \tag{1}$$

Exploitability of a profile σ is $\frac{1}{2}\sum_i \max_{\sigma_i'} u_i(\sigma_i', \sigma_{-i})$, the average gain of a best response, and is zero exactly at Nash equilibria.

CFR and its average. CFR (Zinkevich et al., 2007) runs regret matching at every I: after iteration t it adds the counterfactual regret of each action to a cumulative regret table $R(I, a)$, and $\sigma^{t+1}(I, \cdot)$ is the positive part of $R(I, \cdot)$, normalized (uniform if no entry is positive). The theorem is about the profile

$$\bar\sigma_i^T(I, a) = \frac{S^T(I, a)}{\sum_b S^T(I, b)}, \qquad S^T(I, a) = \sum_{t=1}^{T} \pi_i^{\sigma^t}(I)\,\sigma_i^t(I, a), \tag{2}$$

which we call the *exact average*: if both players' average regrets are at most δ, $\bar\sigma^T$ is a 2δ-equilibrium. Only player i's own strategy enters the weights of $\bar\sigma_i$, and a set's contribution changes whenever the player changes an earlier action even if its own strategy is unchanged. Tabular CFR accumulates S exactly.

Outcome-sampling MCCFR as implemented in OpenSpiel. Outcome sampling (Lanctot et al., 2009) samples a single playthrough to a terminal history and updates only the information sets on it. Algorithm 2 (Appendix A) shows OpenSpiel's solver (Lanctot et al., 2019), which we reproduce; its C++ and Python versions differ only in the constant that initializes the tables, 10^{-6} against 0, and we use 0. An iteration is one playthrough per *update player* i: i samples from the exploring policy $\varepsilon\,\mathrm{Unif} + (1-\varepsilon)\sigma_i^t$ with $\varepsilon = 0.6$, the opponent from its current strategy, and chance from nature; regrets are updated only at i's information sets, with the standard importance weights, so $\sigma_i(I, \cdot)$ changes only at own visits to I; throughout, σ_i^t is the strategy player i carries into its own playthrough of iteration t. The average strategy is also accumulated at i's own information sets on i's

Algorithm 1 Recurrence rule. VISIT is called from Algorithm 2 where the update player's strategy $\sigma(I, \cdot)$ is computed (line 4), before the recursive call and the regret update; the averaging line 11 is dropped. $\hat{S}$ is the stored table, P_I and c_I are one stored number per information set (only c_I at first decision points, only P_I elsewhere), initialized to 0.

1: **procedure** VISIT(I, t) ▷ at player i's information set I on i's playthrough, before the regret update
2: **if** I is a first decision point **then** $g \leftarrow t - c_I$; $c_I \leftarrow t$
3: **else** $g \leftarrow \hat{S}(I^\uparrow, a^\uparrow) - P_I$; $P_I \leftarrow \hat{S}(I^\uparrow, a^\uparrow)$ ▷ growth of the predecessor's sum since the last visit
4: $\hat{S}(I, b)$ += $g\,\sigma(I, b)$ for all $b \in \mathcal{A}(I)$
5: **end procedure**
6: **procedure** FINISH(T) ▷ once, when the average is read out
7: **for** $I \in \mathcal{I}_i$ in order of increasing own depth **do** VISIT(I, T)
8: **end procedure**

own playthrough (line 11), with the increment

$$\text{stochastic rule:} \qquad S(I, a) \mathrel{+}= \frac{\pi_i^{\sigma^t}(I)\,\sigma_i^t(I, a)}{q^t(h)}, \tag{3}$$

where $h \in I$ is the visited history and $q^t(h)$ is the probability that the playthrough reaches h. The original algorithm (Lanctot, 2013, Alg. 3) instead uses the *optimistic rule*

$$\text{optimistic rule:} \qquad S(I, a) \mathrel{+}= (t - c_I)\,\pi_i^{\sigma^t}(I)\,\sigma_i^t(I, a), \quad c_I \leftarrow t, \tag{4}$$

where c_I is the iteration of the previous visit, and it applies the rule at the *opponent's* information sets of the playthrough: player i's average is accumulated when $-i$ is the update player. Either rule can be placed on either playthrough, giving four combinations: STOCH/OWN (OpenSpiel's default), STOCH/OPP, OPT/OWN, and OPT/OPP. The placement decides on whose playthrough player i's average is accumulated, never whose strategy is averaged. The stochastic rules are read out as stored, with no closing update, as in OpenSpiel; the others close once: the optimistic rules add $(T - c_I)$ times the current weight and strategy at every information set, the lazy-weighted rule pushes its pending weights down, and the recurrence rule runs FINISH (Appendix A). Each closing update is applied to a copy of the tables at every checkpoint.

4 WHAT EACH RULE AVERAGES

Fix player i and $I \in \mathcal{I}_i$. Write $w_t(I) = \pi_i^{\sigma^t}(I)$ for the exact weight of iteration t, and $S^T(I, a) = \sum_{t \le T} w_t(I)\sigma_i^t(I, a)$ for the exact sum (2). Two facts drive everything below. The strategy $\sigma_i^t(I, \cdot)$ changes only when I is visited on player i's own playthrough, the only place $R(I, \cdot)$ is updated, so it is constant between two such visits $c < v$; the weight $w_t(I)$ does change between visits, through the ancestors of I. The first fact makes the missed iterations recoverable, and Section 4.1 derives the exact rule from it; the second is what the rules in use get wrong.

4.1 THE EXACT AVERAGE IS A RECURRENCE ON THE PLAYER'S OWN TREE

Lemma 1 (Weight between visits)**.** *Let $I \in \mathcal{I}_i$ have predecessor $(I^\uparrow, a^\uparrow)$. For any $c < T$,*

$$\sum_{t=c+1}^{T} w_t(I) \;=\; S^T(I^\uparrow, a^\uparrow) - S^c(I^\uparrow, a^\uparrow).$$

If I is a first decision point of player i, then $w_t(I) = 1$ and the left side equals $T - c$.

The weight a set accrues while unvisited is thus already recorded in its predecessor's sum; with the constancy of $\sigma_i(I, \cdot)$ between visits, this gives an exact update that needs no sampling probability.

Theorem 1 (Exactness)**.** *Under perfect recall and alternating updates, after FINISH(T) the stored table of Algorithm 1 satisfies $\hat{S}(I, a) = S^T(I, a)$ for every $I \in \mathcal{I}_i$ and $a \in \mathcal{A}(I)$, in exact arithmetic. Here S^T is the exact average of the strategies the sampled run actually produced, not of a tabular run.*

The proof is an induction on own depth (Appendix A); the one nontrivial step is that a visit reads its predecessor's table while that table is current.

Two orderings matter. VISIT runs on the way down, so the predecessor's table is current before I reads it, and before the regret update at I, so the strategy it reads is the one in force on $(c, v]$. The stochastic rule sits on the way back up only because its increment does not depend on order. OpenSpiel already computes $\sigma(I, \cdot)$ before the recursive call, which is where the call goes.

The closing pass does not need the game tree: FINISH over the sets visited at least once returns the exact average there, and one never visited keeps the uniform policy such solvers already return (Proposition 6 and Appendix A).

4.2 WEIGHTED AVERAGES

Linear CFR and DCFR return weighted averages, $S_f^T(I, a) = \sum_{t \le T} f(t)\, w_t(I)\, \sigma_i^t(I, a)$ with $f(t) = t$ or $f(t) = t^\gamma$. Lemma 1 holds verbatim with $f(t)w_t(I)$ in place of $w_t(I)$; at a first decision point the gap contributes $F(T) - F(c)$, with F the prefix sum of f, closed-form for polynomials. Algorithm 1 with $\hat{S}_f$ and $F(t) - F(c_I)$ in place of $\hat{S}$ and $t - c_I$ returns the exact weighted average, and Theorem 1 carries over. No factor is applied to the table; the weights enter through the increments, and F is a table for non-integer γ. Discounting the regret table does not disturb the constancy the theorem rests on: regret matching returns the same strategy on a table whose positive entries are scaled by $d^+ > 0$ and non-positive entries by $d^- \ge 0$ (Lemma 2), so Algorithm 1 returns the exact weighted average of a discounted run.

4.3 THE STOCHASTIC RULE: WHERE IT ACCUMULATES DECIDES WHAT IT ESTIMATES

The stochastic rule (3) is an importance-sampling estimator whose target depends on the playthrough it runs on.

Proposition 1 (Own playthrough)**.** *On player i's own playthrough, $q^t(h) = \pi_c(h)\, \pi_{-i}^{\sigma^t}(h)\, \pi_i^{\sigma'^t}(h)$, and the increment of* STOCH/OWN *at I has expectation*

$$\mathbb{E}\big[\Delta S(I, a) \mid \mathcal{F}_t\big] = n_t^{\text{own}}(I)\; w_t(I)\, \sigma_i^t(I, a), \qquad n_t^{\text{own}}(I) = \big|\{h \in I : \pi_c(h)\, \pi_{-i}^{\sigma^t}(h) > 0\}\big|.$$

(The exploring factor $\pi_i^{\sigma'^t}(h) \ge (\varepsilon/|\mathcal{A}|)^{d_i(h)}$ is always positive.) Here $\mathbb{E}[\,\cdot\,]$ is over the playthrough conditional on $\mathcal{F}_t$, the run before it, so $n_t^{\text{own}}(I)$, $w_t(I)$ and $\sigma_i^t(I, a)$ are $\mathcal{F}_t$-measurable; summing, $\mathbb{E}[\hat{S}^T(I, a)] = \mathbb{E}\big[\sum_{t \le T} n_t^{\text{own}}(I)\, w_t(I)\, \sigma_i^t(I, a)\big]$, which is $n(I)$ times $\mathbb{E}[S^T(I, a)]$ if $n_t^{\text{own}}(I) \equiv n(I)$ and in general not: the stored sum tracks a reweighted sum whose weights move with the opponent's support. The realized increment is $\sigma_i^t(I, a)\, (\pi_i^{\sigma^t}(h)/\pi_i^{\sigma'^t}(h))/(\pi_c(h)\pi_{-i}^{\sigma^t}(h))$, unbounded as $\pi_{-i}^{\sigma^t}(h) \to 0$. The visit probability is $\sum_{h \in I} q^t(h) \le \sum_{h \in I} \pi_c(h)\pi_{-i}^{\sigma^t}(h)$, a bound that does not involve $w_t(I)$; it is not proportional to $w_t(I)$ and is 0 whenever $\pi_c(h)\pi_{-i}^{\sigma^t}(h) = 0$ for every $h \in I$.

Proposition 2 (Opponent's playthrough)**.** *On the opponent's playthrough, $q^t(h) = \pi_c(h)\, \pi_i^{\sigma^t}(h)\, \pi_{-i}^{\sigma'^t}(h)$, and, conditional on the run up to that playthrough, the increment of* STOCH/OPP *at I has expectation $n(I)\, w_t(I)\, \sigma_i^t(I, a)$ with $n(I) = |\{h \in I : \pi_c(h) > 0\}|$, a constant of the game tree. Write $\tilde{S}^T(I, a)$ for the exact sum of the strategies player i carries into the opponent's playthroughs of iterations $1, \dots, T$ ($\tilde{S}^T = S^T$ for the player updated second; for the player updated first it is S^T shifted by one iteration, Appendix A). Then $\hat{S}^T(I, a)/n(I) - \tilde{S}^T(I, a)$ is a martingale in T: the stored sum tracks $n(I)$ times the exact sum of the run that was actually produced, and normalization removes $n(I)$ from the target, though not from the sampling error (Proposition 3). Given a visit, the increment is at most $\sigma_i^t(I, a)\, (|\mathcal{A}|/\varepsilon)^{d_{-i}(h)}/\pi_c(h)$, where $d_{-i}(h)$ counts the opponent's decisions before h and $|\mathcal{A}|$ is the largest action count. The visit probability is at least $(\varepsilon/|\mathcal{A}|)^{D_{-i}}\, w_t(I) \sum_{h \in I} \pi_c(h)$, bounded below by a multiple of $w_t(I)$.*

Proposition 3 (Concentration of the opponent-playthrough rule)**.** *Let $B(I) = \max_{h \in I:\, \pi_c(h) > 0} (|\mathcal{A}|/\varepsilon)^{d_{-i}(h)}/\pi_c(h)$, $M_T(I) = \sum_{t \le T} w_t(I)$, and $\lambda = B(I)\sqrt{2T \log(2|\mathcal{A}(I)|/\delta)}$. With probability at least $1 - \delta$, for every $a \in \mathcal{A}(I)$,*

$$\left|\frac{\hat{S}^T(I, a)}{n(I)} - \tilde{S}^T(I, a)\right| \le \lambda, \qquad \big\|\hat{\bar{\sigma}}_i^T(I, \cdot) - \tilde{\bar{\sigma}}_i^T(I, \cdot)\big\|_1 \le \frac{2|\mathcal{A}(I)|\, \lambda}{M_T(I)},$$

where $\hat{\bar{\sigma}}_i^T$ is the normalized stored sum (uniform if it is zero). For the player updated first, $\tilde{\bar{\sigma}}_i^T$ and $\bar{\sigma}_i^T$ differ in two boundary terms out of T.

An information set carrying a constant share of the weight, $M_T(I) \geq \mu T$, therefore has normalized error $O\big(B(I)\sqrt{\log(1/\delta)}/(\mu\sqrt{T})\big)$, the $T^{-1/2}$ rate that Figure 2 shows; $B(I)$ grows with the inverse chance probability and the opponent's exploration depth, but does not order the rules: the chance sweep of Table 15 raises $\max_I B(I)$ within a game (Leduc, $7.5 \times 10^4 \to 8.6 \times 10^{14}$) without the gap between the sampled and exact averages following it.

Both propositions are direct computations (Appendix A). Regret matching assigns probability exactly zero to actions with non-positive cumulative regret whenever some action's regret is positive (and is uniform otherwise, Lemma 2), so whole information sets of player i are unreachable for the opponent for stretches of iterations, while $w_t(I)$ keeps growing and $n_t^{\text{own}}(I)$ changes whenever the opponent's support does. STOCH/OWN therefore weights iterations by how many of I's histories have positive probability under the opponent's current strategy, which player i's own play does not control; on the opponent's playthrough player i's reach cancels and the visit rate tracks the weight (Figure 1).

4.4 THE OPTIMISTIC RULE

The optimistic rule (4) assumes the weight was constant since the last visit. Its error at a visit in iteration v after a previous visit at c is $(v-c)\, w_v(I)\, \sigma_i^v(I,a) - \sum_{t=c+1}^{v} w_t(I)\, \sigma_i^t(I,a)$, which on the own playthrough, where $\sigma_i(I,\cdot)$ is constant on $(c,v]$, equals

$$\sigma_i^v(I,a) \sum_{t=c+1}^{v} \big(w_v(I) - w_t(I)\big), \tag{5}$$

growing with the gap and the ancestors' drift during it; on the opponent's playthrough own visits in between can also change $\sigma_i(I,\cdot)$, adding a policy-drift term. Own-playthrough gaps are long when the opponent's reach of I is small (Proposition 1), independent of $w_t(I)$; on the opponent's playthrough the visit probability is at least a multiple of $w_t(I)$ (Proposition 2), so a set carrying a constant share of the weight is visited at a rate that shrinks with chance rarity and exploration depth. The lazy-weighted averaging of Lanctot (2013, Alg. 5) pushes down the reach accrued while the action was not sampled; its sum at a first decision point counts only the iterations that visited it, so it is exact there only if every first decision point is visited every iteration, which fails once one is reached with probability below one (Appendix A).

5 DISCOUNT FACTORS BETWEEN VISITS

Discounted CFR variants multiply their tables by a factor at every iteration. DCFR with exponents (α,β,γ) (Brown & Sandholm, 2019a) multiplies positive cumulative regrets by $d_t^+ = t^\alpha/(t^\alpha+1)$, negative ones by $d_t^- = t^\beta/(t^\beta+1)$, and the strategy sum by $(t/(t+1))^\gamma$, recommended $(1.5, 0, 2)$; linear CFR is $(1,1,1)$ and hyperparameter schedules (Zhang et al., 2026) move the exponents with t. Under sampling, an information set visited at iterations $v_1 < \cdots < v_m$ misses the factors in between. Ports to MCCFR apply the current iteration's factor once per visit (*single-current-factor*, SCF); *replay* applies every skipped factor, as full-tree CFR would have.

Discount factors compose, so replay costs one subtraction. Fix one table coordinate with increments r_{v_j} at its visits and a global factor sequence $d_t \in (0,1]$. After the m-th visit,

$$R_m^{\text{SCF}} = \sum_{j \leq m} r_{v_j} \prod_{\ell=j+1}^{m} d_{v_\ell}, \qquad R_m^{\text{replay}} = \sum_{j \leq m} r_{v_j} \prod_{t=v_j+1}^{v_m} d_t. \tag{6}$$

Proposition 4 (Application event)**.** *The two accumulators in* (6) *agree for every increment sequence if and only if every skipped factor equals one, $\prod_{t=v_j+1}^{v_{j+1}-1} d_t = 1$ for all $j < m$. For zero-initialized accumulators, $\prod_{\ell>j} d_{v_\ell} \geq \prod_{t=v_j+1}^{v_m} d_t$ coefficientwise, so SCF assigns to every past increment a coefficient at least as large as replay does, by a factor that grows with the visit gaps.*

Replay does not require revisiting the skipped iterations. With the prefix sums $L(t) = \sum_{s \le t} \log d_s$ computed once for the run and shared by every information set, the product of the factors skipped between visits c and v is $\exp(L(v) - L(c))$: one subtraction and one exponential, using the last-visit iteration c_I that the optimistic rule stores anyway. For the DCFR strategy sum the same holds: under (6)'s convention that the factor of iteration t is applied before its increment is inserted, the factor is $d_t = ((t-1)/t)^\gamma$ (DCFR's $(t/(t+1))^\gamma$ re-indexed by one), and there the replay semantics is exactly the weighted exact average of Section 4.2: $\prod_{t=v+1}^{T}((t-1)/t)^\gamma = (v/T)^\gamma$, so replay weights iteration v by v^γ up to a common factor.

Which factors are bounded when skipped. Write $C_j^{\mathrm{SCF}} = \prod_{\ell=j+1}^{m} d_{v_\ell}$ and $C_j^{\mathrm{rep}} = \prod_{t=v_j+1}^{v_m} d_t$ for the coefficients of r_{v_j} in (6), so that $C_j^{\mathrm{SCF}} = \rho_j \, C_j^{\mathrm{rep}}$ with $\rho_j = 1/\prod_{t \in V_j} d_t \ge 1$ and V_j the iterations skipped between v_j and v_m.

Proposition 5 (Coefficient distortion of skipped factors)**.** *Let $\Pi(V) = \prod_{t \in V} d_t$ for a finite set V of iterations, and ζ the Riemann zeta function. For the positive-regret factor $d_t = t^\alpha/(t^\alpha + 1)$: if $\alpha > 1$ then $\Pi(V) \ge e^{-\zeta(\alpha)}$ for every V, hence $1 \le \rho_j \le e^{\zeta(\alpha)}$ for every visit pattern; if $0 < \alpha \le 1$ then $\inf_V \Pi(V) = 0$, and for $\alpha = 1$ and $V = \{1, \dots, T\}$, $\Pi(V) = 1/(T+1)$. For the strategy-sum factor $d_t = ((t-1)/t)^\gamma$ with any $\gamma > 0$, $\Pi(\{c+1, \dots, v\}) = (c/v)^\gamma \to 0$ as the gap grows. Consequently, for one coordinate fed the same increments under both semantics,*

$$R_m^{\mathrm{SCF}} - R_m^{\mathrm{replay}} \;=\; \sum_{j \le m} r_{v_j} \, C_j^{\mathrm{rep}} \, (\rho_j - 1), \qquad \left|R_m^{\mathrm{SCF}} - R_m^{\mathrm{replay}}\right| \;\le\; \left(e^{\zeta(\alpha)} - 1\right) \sum_{j \le m} |r_{v_j}| \, C_j^{\mathrm{rep}}$$

for the positive-regret factor with $\alpha > 1$, while in the other two cases ρ_j is unbounded in the length of the gap.

The object here is two accumulators fed the same increments; two solver runs diverge after their first disagreement, and regret matching's threshold at zero turns a bounded coefficient difference into an arbitrary ratio of coordinates. It locates where the two semantics can be told apart: at most a constant factor for positive regrets at $\alpha > 1$, an unbounded one for $\alpha \le 1$ and every strategy-sum exponent. Section 6 measures the consequence.

For DCFR's $\alpha = 1.5$, $e^{-\zeta(1.5)} \approx 0.073$: an increment propagated only by positive-regret factors is inflated by SCF by at most $14\times$, while one whose coordinate changes sign is propagated by the other factor sequence and is not covered. For linear CFR and every strategy-sum exponent no such constant exists (Appendix A).

6 Experiments

Setup. We expand each game from OpenSpiel (Lanctot et al., 2019) into an explicit tree and run Algorithm 2 with its defaults. All averaging rules and weights share one regret trajectory and one random stream per run, so every rule sees identical strategies and visits. Games: Kuhn poker (Kuhn, 1950), Leduc poker (Southey et al., 2005), Goofspiel with 4, 5 and 6 cards (descending, imperfect information; Ross, 1971), and Liar's Dice with one die of 4, 6, 7, and 8 sides (Lanctot, 2013) (Table 3). We run 20 seeds per game for 10^7 iterations and read every rule's average at 9 checkpoints, applying catch-up updates to copies of the tables. Exploitability is an exact best response on the expanded tree, pinned against OpenSpiel's own before the runs (Appendix A); the recurrence rule matches brute force on all nine games (Table 10). We report the ratio of a rule's exploitability to the exact average's with the same weights, paired within a run, as a geometric mean over seeds with a bootstrap interval. LWA is Lanctot (2013, Alg. 5) on the opponent's playthrough, its original placement; on OpenSpiel's own Python solver, the returned average is 1.32 and 1.41 against 1.34 and 1.49 for ours on Kuhn and Leduc at 10^6 iterations (Table 11).

The drift grows with training. STOCH/OWN is worse than the exact average at every checkpoint on every game, and the gap widens: 1.04–1.82 times as exploitable at 10^5 iterations with the interval above 1 on every game, 1.66–9.0 times at 10^7 (Tables 1 and 4), worse than the exact average in 177 of 180 runs. It bends away between 10^5 and 10^6 and then falls more slowly (Figure 3). The optimistic rule on the own playthrough is within rounding of the exact average on Kuhn poker, at 1.00, and 1.12–6.0 times as exploitable elsewhere. The gap is not monotone in game size: it is widest on Goofspiel(4), the second smallest tree.

Table 1: Exploitability of each averaging rule divided by that of the exact average after 10^7 iterations; geometric mean over 20 seeds with 95% bootstrap interval (10^5 and 10^6 in Table 4). Rules are named by increment (stochastic (3) or optimistic (4)) and by the playthrough on which the average is accumulated. STOCH/OWN is OpenSpiel's default; OPT/OPP the original MCCFR.

	STOCH/OWN	OPT/OWN	STOCH/OPP	OPT/OPP	LWA
Kuhn poker	1.66 [1.32,2.13]	1.00 [1.00,1.00]	0.99 [0.94,1.04]	1.00 [1.00,1.01]	0.98 [0.91,1.06]
Leduc poker	2.89 [2.48,3.39]	1.25 [1.18,1.33]	0.99 [0.98,1.00]	1.11 [1.09,1.13]	1.13 [1.09,1.17]
Goofspiel(4)	9.01 [5.60,15.09]	2.84 [2.08,3.92]	0.99 [0.94,1.04]	1.62 [1.36,1.93]	1.27 [1.14,1.42]
Liar's Dice(1,4)	4.76 [3.78,5.80]	2.17 [1.70,2.82]	0.99 [0.98,1.00]	1.34 [1.25,1.43]	1.14 [1.03,1.26]
Goofspiel(5)	5.87 [5.07,6.71]	5.99 [5.23,6.86]	1.00 [0.98,1.02]	1.54 [1.41,1.68]	1.43 [1.30,1.58]
Liar's Dice(1,6)	2.51 [2.22,2.83]	1.28 [1.25,1.31]	1.00 [0.99,1.00]	1.10 [1.08,1.11]	0.99 [0.98,1.00]
Goofspiel(6)	2.40 [2.21,2.62]	2.12 [1.96,2.29]	1.00 [1.00,1.01]	1.11 [1.09,1.14]	1.09 [1.05,1.13]
Liar's Dice(1,7)	2.09 [1.96,2.26]	1.19 [1.18,1.21]	1.01 [1.00,1.01]	1.07 [1.06,1.08]	0.99 [0.98,1.00]
Liar's Dice(1,8)	1.90 [1.83,1.98]	1.12 [1.11,1.13]	1.00 [1.00,1.01]	1.06 [1.05,1.06]	0.99 [0.99,1.00]

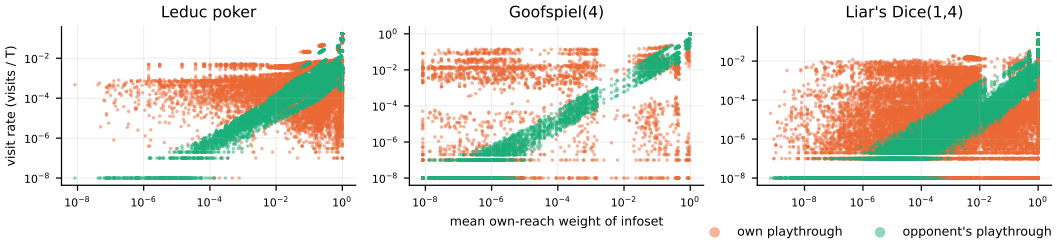

Figure 1: Mechanism, twenty seeds per game after 10^7 iterations. Each point is one information set: horizontal axis its mean exact weight $\frac{1}{T}\sum_t w_t(I)$, vertical axis its visit rate on the own (orange) and opponent (green) playthroughs; unvisited sets drawn at 10^{-8}.

The opponent's playthrough fixes the stochastic rule. Moving OpenSpiel's increment to the opponent's playthrough gives ratios of 0.99–1.05 on every game at 10^5, 10^6 and 10^7 (STOCH/OPP in Tables 1 and 4); the geometric-mean ratio of the estimator of Proposition 2 lies in 0.99–1.01 at 10^7; Proposition 3 bounds this rule's local policy error, not its exploitability, so the residual is consistent with, not explained by, its $T^{-1/2}$ rate. The optimistic rule improves there too but is still up to 1.62 times as exploitable, and Lanctot's lazy-weighted rule up to 1.43; neither error visibly shrinks over the measured range. Tilting every chance node's law away from uniform, its children weighted by the ladder κ^k (Table 12), fails both readings we registered for it: the own-playthrough drift drops below our floor on Kuhn, and the opponent-playthrough rule leaves its registered band on five of the sixteen cells, most visibly Leduc at $\kappa = 0.25$, where twenty seeds per level give 1.32–1.84, every interval excluding one (Table 13); the own-playthrough rule travels by 4.7–333 times as much as the exact rule (Table 14).

Weight and visits decouple on the own playthrough. Figure 1 shows the two facts behind Propositions 1 and 2. On the opponent's playthrough the visit rate of an information set follows its exact weight along a diagonal band (Spearman correlation 0.86–0.97); on the own playthrough it does not (0.24–0.32, a flat cloud), with weight-near-1 sets visited at rates from 10^{-8} to 10^{-1} and 6.1% of the exact weight on Goofspiel(4), 5.9% on Liar's Dice(1,4), carried by sets never visited in 10^7 iterations. Their average is uniform under every rule (Proposition 6), so the error of STOCH/OWN arises at the visited sets, where $n_t^{\text{own}}(I)$ changes from iteration to iteration. Figure 2 shows the consequence: the mass-weighted L_1 distance from STOCH/OWN to the exact average stays near 0.1 from 10^3 to 10^7 iterations, the distance from STOCH/OPP decreases as roughly $T^{-1/2}$, and the optimistic rules and LWA lie between and flatten late.

Opponent exploration removes the gap, at a price. If the mechanism is the opponent's missing exploration, letting the opponent explore with $\varepsilon_{-i} > 0$ should make every rule exact-like, and it does. Sweeping $\varepsilon_{-i} \in \{0.1, 0.25, 0.6\}$, each level paired against the same seed's $\varepsilon_{-i} = 0$ run,

STOCH/OWN falls from 1.65–7.53 to 0.98–1.49 within one level, a drop to 0.14–0.59 of the control by $\varepsilon_{-i} = 0.6$ (Table 5). The direction is unanimous on three of four games at every level (10/10 seeds, sign test $p = 0.002$). The fix is not free: opponent exploration inflates the weights and the exact average itself degrades, from 0.032 to 0.058 on Leduc and 0.0029 to 0.0060 on Liar's Dice(1,4) at $\varepsilon_{-i} = 0.6$, while on Kuhn it is flat. Exploration buys a smaller gap by moving the target both rules are measured against. Regret matching$^+$ (Tammelin, 2014) behaves similarly: own-playthrough rules are within 1.02 of the exact average on three of four games (Goofspiel(4): 1.56), but its exact average is 1.5–3.3 times as exploitable as regret matching's (Table 16). Sweeping the player's own $\varepsilon \in \{0.2, \ldots, 1.0\}$ changes nothing: STOCH/OPP stays within 0.93–1.04, STOCH/OWN ranges from 1.6 to 9.2 (Table 19).

Weighted averages. The top block of Table 18 repeats the comparison for linear and quadratic weights on the regret-matching trajectory: STOCH/OWN is 1.0–3.1 times as exploitable as the exact average with the same weights, STOCH/OPP within 0.98–1.00. The visit-triggered discount of the strategy sum (SCF/OWN) is sometimes below 1, down to 0.32 on Liar's Dice(1,4) with quadratic weights. Lower exploitability here does not mean more accurate accumulation: by Proposition 5 SCF under-discounts old contributions, so its effective weights are closer to uniform than t^2, and on this trajectory the exact uniform average beats the exact linear and quadratic ones by 1.05–8.8× and 1.2–16× (last two columns). Against the uniform exact average, SCF/OWN on Liar's Dice(1,4) is $0.32 \times 16.4 = 5.2$ times as exploitable.

Discounted regrets. With DCFR exponents $(1.5, 0)$, replaying the skipped positive-regret factors or applying the single current factor gives exact averages within 0.98–1.09 of each other; with linear-CFR exponents $(1, 1)$, where no such constant exists, the single-current-factor trajectory is 1.5–6.5 times as exploitable, and replaying costs one subtraction. On the replay trajectories the visit-triggered average is 1.25–3.8 times as exploitable as the exact weighted average and STOCH/OPP within 0.98–1.02; neither discounted trajectory improves on plain regret matching except linear CFR by replay on Liar's Dice(1,6) at 0.76 (Table 17).

The α and β sweeps are in Appendix B.1: the exact average's replay/SCF ratio is 1.12–1.79 at $\alpha = 1$, flat at 0.98–1.12 from $\alpha = 1.25$ to 30, and 0.96–1.18 at $\beta = 30$, where both factors are within 10^{-9} of one, covering 1 on all four games (Tables 6 and 7).

The exact average costs at most 4.5%. Timed in isolation over 10^6 iterations, median of five single-thread runs, the recurrence rule adds 0.3–4.5% over OpenSpiel's rule (Table 20); STOCH/OPP costs the same as STOCH/OWN, so equal iterations is equal wall-clock to within 5%. The closing read is worth paying for: the same rule *without* its end-of-run catch-up is 1.05–3.10 times as exploitable, 5% on Kuhn and 210% on Liar's Dice(1,4) (Table 8).

7 CONCLUSION

A sampled CFR solver settles, at every visit, what an information set is owed for the iterations it missed: its share of the average and the discount factors. Where OpenSpiel applies its rule, the returned strategies are 1.04–1.82 times as exploitable as the exact average at 10^5 iterations and 1.7–9 times at 10^7, a gap that grows with training. The exact amount is the growth of the predecessor's sum, which the solver already stores, and skipped discount factors are settled by a prefix-table lookup, bounded for positive-regret exponents above one and never for linear CFR or the strategy sum (Proposition 5). We recommend both as defaults for implementations that store the expanded tree; for sparse implementations Proposition 6 gives the read-out; a benchmark there is future work.

AI USE STATEMENT

We used generative AI tools to polish and edit the text of this paper and to write and modify parts of the experimental code. We have not used generative AI tools to generate research ideas, and the results and claims of this work are ours. We have reviewed all AI-assisted work, including verifying the code against the reference implementation and against brute-force computation (Table 10 and Appendix A), and we take responsibility for the final content of this work.

REFERENCES

Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold'em poker is solved. *Science*, 347(6218):145–149, 2015.

Noam Brown and Tuomas Sandholm. Reduced space and faster convergence in imperfect-information games via pruning. In *International conference on machine learning*, pp. 596–604. PMLR, 2017.

Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

Noam Brown and Tuomas Sandholm. Solving imperfect-information games via discounted regret minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1829–1836, 2019a.

Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365(6456): 885–890, 2019b.

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *International conference on machine learning*, pp. 793–802. PMLR, 2019.

Neil Burch. *Time and space: Why imperfect information games are hard.* PhD thesis, University of Alberta, 2018.

Neil Burch, Marc Lanctot, Duane Szafron, and Richard Gibson. Efficient monte carlo counterfactual regret minimization in games with many player actions. *Advances in neural information processing systems*, 25, 2012.

Neil Burch, Matej Moravcik, and Martin Schmid. Revisiting cfr+ and alternating updates. *Journal of Artificial Intelligence Research*, 64:429–443, 2019.

Trevor Davis, Martin Schmid, and Michael Bowling. Low-variance and zero-variance baselines for extensive-form games. In *International Conference on Machine Learning*, pp. 2392–2401. PMLR, 2020.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Stochastic regret minimization in extensive-form games. In *International Conference on Machine Learning*, pp. 3018–3028. PMLR, 2020.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Faster game solving via predictive blackwell approachability: Connecting regret matching and mirror descent. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 5363–5371, 2021.

Sam Ganzfried, Tuomas Sandholm, and Kevin Waugh. Strategy purification and thresholding: Effective non-equilibrium approaches for playing large games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 871–878, 2012.

Richard Gibson, Marc Lanctot, Neil Burch, Duane Szafron, and Michael Bowling. Generalized sampling and variance in counterfactual regret minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pp. 1355–1361, 2012.

Richard G Gibson. *Regret minimization in games and the development of champion multiplayer computer poker-playing agents*. PhD thesis, University of Alberta, 2014.

Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

Michael Johanson, Nolan Bard, Neil Burch, and Michael Bowling. Finding optimal abstract strategies in extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pp. 1371–1379, 2012a.

Michael Johanson, Nolan Bard, Marc Lanctot, Richard G Gibson, and Michael Bowling. Efficient nash equilibrium approximation through monte carlo counterfactual regret minimization. In *Aamas*, pp. 837–846, 2012b.

Harold W. Kuhn. A simplified two-person poker. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games, Volume I*, volume 24 of *Annals of Mathematics Studies*, pp. 97–103. Princeton University Press, 1950.

Marc Lanctot. *Monte Carlo sampling and regret minimization for equilibrium computation and decision-making in large extensive form games*. University of Alberta (Canada), 2013.

Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte carlo sampling for regret minimization in extensive games. *Advances in neural information processing systems*, 22, 2009.

Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. Openspiel: A framework for reinforcement learning in games. *arXiv preprint arXiv:1908.09453*, 2019.

Hui Li, Kailiang Hu, Zhibang Ge, Tao Jiang, Yuan Qi, and Le Song. Double neural counterfactual regret minimization. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=lvWE53uu9dH`.

Viliam Lisỳ, Marc Lanctot, and Michael H Bowling. Online monte carlo counterfactual regret minimization for search in imperfect information games. In *AAMAS*, pp. 27–36, 2015.

Stephen Marcus McAleer, Gabriele Farina, Marc Lanctot, and Tuomas Sandholm. Escher: Eschewing importance sampling in games by computing a history value function to estimate regret. In *The Eleventh International Conference on Learning Representations*, 2023.

Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

Sheldon M Ross. Goofspiel—the game of pure strategy. *Journal of Applied Probability*, 8(3): 621–625, 1971.

Martin Schmid, Neil Burch, Marc Lanctot, Matej Moravcik, Rudolf Kadlec, and Michael Bowling. Variance reduction in monte carlo counterfactual regret minimization (vr-mccfr) for extensive form games using baselines. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 2157–2164, 2019.

Martin Schmid, Matej Moravčík, Neil Burch, Rudolf Kadlec, Josh Davidson, Kevin Waugh, Nolan Bard, Finbarr Timbers, Marc Lanctot, G Zacharias Holland, Elnaz Davoodi, Alden Christianson, and Michael Bowling. Student of games: A unified learning algorithm for both perfect and imperfect information games. *Science Advances*, 9(46):eadg3256, 2023.

Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: opponent modelling in poker. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pp. 550–558, 2005.

Eric Steinberger, Adam Lerer, and Noam Brown. Dream: Deep regret minimization with advantage baselines and model-free learning. *arXiv preprint arXiv:2006.10410*, 2020.

Oskari Tammelin. Solving large imperfect information games using cfr+. *arXiv preprint arXiv:1407.5042*, 2014.

Hang Xu, Kai Li, Bingyun Liu, Haobo Fu, Qiang Fu, Junliang Xing, and Jian Cheng. Minimizing weighted counterfactual regret with optimistic online mirror descent. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 5272–5280, 2024.

Naifeng Zhang, Stephen Marcus McAleer, and Tuomas Sandholm. Faster game solving via hyperparameter schedules. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 17319–17326, 2026.

Yichi Zhou, Tongzheng Ren, Jialian Li, Dong Yan, and Jun Zhu. Lazy-cfr: fast and near-optimal regret minimization for extensive games with imperfect information. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rJx4p3NYDB.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. *Advances in neural information processing systems*, 20, 2007.

A ALGORITHM AND PROOFS

Algorithm 2 One playthrough of OpenSpiel's outcome-sampling MCCFR for update player i at iteration t. Reach probabilities: π_i (player i's own actions), π_{-i} (opponent and chance), q (sampling). All start at 1.

1: **function** PLAYTHROUGH($h, i, t, \pi_i, \pi_{-i}, q$)
2: **if** h is terminal **then return** $u_i(h)$
3: **if** h is a chance node **then** sample $a \sim \pi_c(\cdot \mid h)$ and **return** PLAYTHROUGH($ha, i, t, \pi_i, \pi_{-i}\pi_c(a \mid h), q\,\pi_c(a \mid h)$)
4: $I \leftarrow$ information set of h for the acting player p; $\sigma(I,\cdot) \leftarrow$ regret matching on $R(I,\cdot)$
5: **if** $p = i$ **then** VISIT(I, t) ▷ recurrence rule (Algorithm 1); replaces line 11
6: $\sigma'(I,\cdot) \leftarrow \varepsilon/|\mathcal{A}(I)| + (1-\varepsilon)\sigma(I,\cdot)$ if $p = i$, else $\sigma(I,\cdot)$; sample $a \sim \sigma'(I,\cdot)$
7: **if** $p = i$ **then** $v \leftarrow$ PLAYTHROUGH($ha, i, t, \pi_i\sigma(I,a), \pi_{-i}, q\,\sigma'(I,a)$) **else** $v \leftarrow$ PLAYTHROUGH($ha, i, t, \pi_i, \pi_{-i}\sigma(I,a), q\,\sigma'(I,a)$)
8: $\hat{v} \leftarrow \sigma(I,a)\,v/\sigma'(I,a)$ ▷ sampled value of σ at h
9: **if** $p = i$ **then**
10: $R(I,b)$ += $\big(\mathbb{1}[b{=}a]\,v/\sigma'(I,a) - \hat{v}\big)\,\pi_{-i}/q$ for all $b \in \mathcal{A}(I)$
11: $S(I,b)$ += $\pi_i\,\sigma(I,b)\,/\,q$ for all $b \in \mathcal{A}(I)$ ▷ stochastic rule, own playthrough (STOCH/OWN)
12: **end if**
13: **return** $\hat{v}$
14: **end function**

Proof of Lemma 1. By (1), $w_t(I) = w_t(I^\uparrow)\,\sigma_i^t(I^\uparrow, a^\uparrow)$, which is the t-th term of $S(I^\uparrow, a^\uparrow)$. Summing over $t \in (c, T]$ gives the claim; at a first decision point $w_t(I) = 1$. □

Proof of Theorem 1. Say the stored table is *current* at I at iteration t if $\hat{S}(I,\cdot) = S^t(I,\cdot)$. Between visits the stored table is stale while the exact sum grows; the claim is that VISIT(I, v) makes the table current at v, by induction on own depth. First decision points: at a visit in iteration v after a previous visit (or the start) at c, the strategy at I equalled $\sigma_i^v(I,\cdot)$ throughout $(c, v]$ and $w_t(I) = 1$, so $S^v(I,\cdot) - S^c(I,\cdot) = (v-c)\sigma_i^v(I,\cdot)$, which is what VISIT adds to a table that was current at c. Deeper I: on a playthrough that visits I at iteration v, the predecessor $I^\uparrow$ lies on the same path and is processed first, so by induction $\hat{S}(I^\uparrow, a^\uparrow) = S^v(I^\uparrow, a^\uparrow)$ when VISIT(I, v) reads it; P_I was written at the previous visit c of I, on a playthrough that also visited $I^\uparrow$ first, so $P_I = S^c(I^\uparrow, a^\uparrow)$. By Lemma 1, $g = \sum_{t=c+1}^{v} w_t(I)$, and since $\sigma_i(I,\cdot)$ is constant on $(c, v]$, $S^v(I,\cdot) - S^c(I,\cdot) = g\,\sigma_i^v(I,\cdot)$. FINISH applies VISIT($\cdot$, T) in order of increasing own depth, so each predecessor is current at T before its successors read it, and the same argument makes every table current at T. □

Proof of Proposition 1. On player i's playthrough, chance samples from π_c, the opponent from σ_{-i}^t, and player i from the exploring policy $\sigma_i'^t = \varepsilon\,\mathrm{Unif} + (1-\varepsilon)\sigma_i^t$, so $q^t(h) = \pi_c(h)\,\pi_{-i}^{\sigma^t}(h)\,\pi_i^{\sigma'^t}(h)$. The rule adds $w_t(I)\sigma_i^t(I,a)/q^t(h)$ when history $h \in I$ is visited, which happens with probability $q^t(h)$; histories with $q^t(h) = 0$ contribute nothing. Hence

$$\mathbb{E}[\Delta S(I,a) \mid \mathcal{F}_t] = \sum_{h \in I:\, q^t(h) > 0} q^t(h)\,\frac{w_t(I)\,\sigma_i^t(I,a)}{q^t(h)} = n_t^{\mathrm{own}}(I)\,w_t(I)\,\sigma_i^t(I,a),$$

where $q^t(h) > 0$ iff $\pi_c(h)\pi_{-i}^{\sigma^t}(h) > 0$, since $\pi_i^{\sigma'^t}(h) \geq (\varepsilon/|\mathcal{A}|)^{d_i(h)} > 0$. The realized increment is $w_t(I)\sigma_i^t(I,a)/q^t(h)$ with $w_t(I) = \pi_i^{\sigma^t}(h)$, which gives the stated form; it grows without bound as $\pi_{-i}^{\sigma^t}(h) \to 0$ at fixed $w_t(I)$. The visit probability is $\sum_{h\in I} q^t(h) \leq \sum_{h\in I} \pi_c(h)\pi_{-i}^{\sigma^t}(h)$ because $\pi_i^{\sigma'^t}(h) \leq 1$, and it is zero when $\pi_{-i}^{\sigma^t}(h) = 0$ for every $h \in I$. □

Proof of Proposition 2. On the opponent's playthrough the roles swap: $q^t(h) = \pi_c(h)\,\pi_i^{\sigma^t}(h)\,\pi_{-i}^{\sigma'^t}(h)$. The increment is $w_t(I)\sigma_i^t(I,a)/q^t(h) = \sigma_i^t(I,a)/(\pi_c(h)\pi_{-i}^{\sigma'^t}(h))$ because $w_t(I) = \pi_i^{\sigma^t}(h)$ cancels; the bound follows from $\pi_{-i}^{\sigma'^t}(h) \geq (\varepsilon/|\mathcal{A}|)^{d_{-i}(h)}$. For the expectation, $q^t(h) > 0$ iff $\pi_c(h) > 0$ and $w_t(I) > 0$; if $w_t(I) = 0$ both sides of the claimed identity vanish, and otherwise

$$\mathbb{E}[\Delta S(I,a)] = \sum_{h\in I:\,\pi_c(h)>0} q^t(h)\,\frac{w_t(I)\,\sigma_i^t(I,a)}{q^t(h)} = n(I)\,w_t(I)\,\sigma_i^t(I,a).$$

Summing over t and normalizing over a removes the constant $n(I)$. The visit probability is $\sum_h q^t(h) \geq (\varepsilon/|\mathcal{A}|)^{D_{-i}}\,w_t(I)\sum_{h\in I}\pi_c(h)$ with D_{-i} the maximal number of opponent decisions on a path. □

Proof of Proposition 3. Let $\mathcal{F}_t$ be the history of the run before the opponent's playthrough of iteration t (for the player whose target is shifted by one iteration, read $t+1$ throughout). By the proof of Proposition 2, $X_t = \Delta S_t(I,a)/n(I) - w_t(I)\sigma_i^t(I,a)$ satisfies $\mathbb{E}[X_t \mid \mathcal{F}_t] = 0$. The increment is either 0 or $\sigma_i^t(I,a)/(\pi_c(h)\pi_{-i}^{\sigma'^t}(h)) \in [0, B(I)]$, and $w_t\sigma^t \in [0,1]$ with $B(I) \geq 1$ and $n(I) \geq 1$, so $|X_t| \leq B(I)$. Azuma's inequality gives $\mathbb{P}\big(|\sum_{t\leq T} X_t| \geq \lambda\big) \leq 2\exp(-\lambda^2/(2TB(I)^2))$; with $\lambda = B(I)\sqrt{2T\log(2|\mathcal{A}(I)|/\delta)}$ and a union bound over $a \in \mathcal{A}(I)$ the first claim follows. For the second, write $x = \hat{S}^T(I,\cdot)/n(I)$ and $y = S^T(I,\cdot)$, both non-negative with $\|y\|_1 = M_T(I)$. If $x \neq 0$, $\|x/\|x\|_1 - y/\|y\|_1\|_1 \leq \|x-y\|_1/\|y\|_1 + \big|\|y\|_1 - \|x\|_1\big|/\|y\|_1 \leq 2\|x-y\|_1/\|y\|_1$; if $x = 0$ the left side is at most 2 and the right side is at least 2. Finally $\|x-y\|_1 \leq |\mathcal{A}(I)|\max_a |x_a - y_a|$. □

Alternating updates. In iteration t player 1's playthrough comes first and updates R_1, so on player 2's playthrough of the same iteration player 1 already carries σ_1^{t+1}. The opponent-playthrough rules for player 1 therefore average $\sigma_1^{t+1}, \ldots, \sigma_1^{T+1}$ with weights $w_{t+1}(I)$, a version of the exact average shifted by one iteration; for player 2 the iterations align. The shift changes two boundary terms out of T and is invisible in Table 1. The exact average and the recurrence rule use, for both players, the strategy each carries into its own playthrough of iteration t, which is the profile whose regrets are bounded by the analysis of alternating updates (Burch et al., 2019). With weights $f(t)$ the shifted target is $\sum_{t\leq T} f(t)\,w_{t+1}(I)\,\sigma_1^{t+1}(I,\cdot)$, which differs from the exact weighted sum termwise by the factor $f(t)/f(t+1) \in [(t/(t+1))^\gamma, 1]$ for $f(t) = t^\gamma$; the distortion of each coefficient tends to one and is largest on the first iterations.

Lazy-weighted averaging. Lanctot (2013, Alg. 5) maintains a weight $\omega(I,a)$ per state-action pair. On a visit to I with incoming weight ω_{in} (initially 0 at the root) and current reach π, the rule adds $(\omega_{\text{in}} + \pi)\,\sigma(I,\cdot)$ to $S(I,\cdot)$, adds the same quantity to $\omega(I,a)$ for every action a that was not sampled, and passes $\omega_{\text{in}}\sigma(I,a^*) + \omega(I,a^*)$ down the sampled action a^*, resetting $\omega(I,a^*)$ to 0. Unwinding the recursion shows that $S(I,\cdot)$ receives the reach accrued in every iteration in which some ancestor of I, or I itself, was visited, applied to the local policy current at the time it is pushed down; when $\sigma_i(I,\cdot)$ changed between two push-downs, which happens whenever I was visited on the own playthrough in between, the accrued reach multiplies the later policy and the contribution is not the exact one (the difference is $\omega_{\text{in}}\,(\sigma_i^v - \sigma_i^c)(I,\cdot)$ for a push-down at v carrying reach accrued at c).

Iterations in which no information set on the path to I was visited contribute nothing; at a first decision point I of player i, which has no ancestors, the sum after T iterations is $\sum_{t\in\text{visits}(I)} \sigma^t(I,\cdot)$ instead of $\sum_{t\leq T}\sigma^t(I,\cdot)$. Agreement at the first decision points requires that each of them be visited in every iteration, which fails as soon as some first decision point is reached with probability below one on the opponent's playthrough, i.e. when a chance outcome or an opponent action before it leads

to a different first decision point of player i. In Kuhn poker after 300 iterations the two averages already differ by more than 10^{-3} in some entry, and Table 1 shows the effect on exploitability.

The recurrence rule needs none of this: the missing iterations at a first decision point are counted by $t - c_I$, and everywhere else by the predecessor's sum.

Table 2: How the averaging rules maintain the strategy sum. *Event*: when $S(I, \cdot)$ changes. *Missed iterations*: how the iterations between visits enter. *Policy change between visits*: whether the rule is exact when $\sigma_i(I, \cdot)$ changed between two of its events. *State*: per information set.

rule	event	missed iterations	policy change	state
STOCH/OWN (OpenSpiel)	own visit	divided out by $1/q^t(h)$	n/a (own gaps)	sum
STOCH/OPP (Lanctot Alg. 3)	opponent visit	divided out by $1/q^t(h)$	in expectation	sum
AS, Gibson (Gibson, 2014)	opponent visit	divided out by $1/q^t(h)$	in expectation	sum
OPT/OWN	own visit	$(v - c)\, w_v$ at current weight	n/a	sum, c_I
OPT/OPP	opponent visit	$(v - c)\, w_v$ at current weight	not exact	sum, c_I
LWA (Lanctot Alg. 5)	opponent visit	reach carried as $\omega(I, a)$	not exact (Appendix A)	sum, $\omega(I, \cdot)$
EXACT (Algorithm 1)	own visit + closing pass	predecessor's sum growth	exact	sum, watermark

Proof of Proposition 4. Induction on m: $R_m^{\mathrm{SCF}} = d_{v_m} R_{m-1}^{\mathrm{SCF}} + r_{v_m}$ and $R_m^{\mathrm{replay}} = \big(\prod_{t=v_{m-1}+1}^{v_m} d_t\big) R_{m-1}^{\mathrm{replay}} + r_{v_m}$ unwind to (6). Two linear maps agree on every increment vector iff their coefficients agree, i.e. $\prod_{\ell=j+1}^{m} d_{v_\ell} = \prod_{t=v_j+1}^{v_m} d_t$ for all j; dividing the coefficient of r_{v_j} by that of $r_{v_{j+1}}$ reduces this to $d_{v_{j+1}} = \prod_{t=v_j+1}^{v_{j+1}} d_t$, i.e. $\prod_{t=v_j+1}^{v_{j+1}-1} d_t = 1$. Since every $d_t \le 1$, the replay coefficient is the SCF coefficient times a product of skipped factors, which is at most 1, strictly less when some skipped factor is. □

Proof of Proposition 5. $-\log \Pi(V) = \sum_{t \in V} \log(1 + t^{-\alpha}) \le \sum_{t \ge 1} t^{-\alpha} = \zeta(\alpha)$, finite for $\alpha > 1$. For $0 < \alpha \le 1$ use $\log(1 + x) \ge x/2$ on $[0, 1]$: $-\log \Pi(\{1, \dots, T\}) \ge \frac{1}{2} \sum_{t \le T} t^{-\alpha} \to \infty$, so $\inf_V \Pi(V) = 0$; for $\alpha = 1$, $\prod_{t=1}^{T} t/(t + 1) = 1/(T + 1)$. For the strategy sum, $\prod_{t=c+1}^{v} ((t - 1)/t)^\gamma = (c/v)^\gamma$ by telescoping. For the coefficient identity, $C_j^{\mathrm{rep}} = \prod_{t=v_j+1}^{v_m} d_t = \big(\prod_{\ell=j+1}^{m} d_{v_\ell}\big)\big(\prod_{t \in V_j} d_t\big) = C_j^{\mathrm{SCF}}/\rho_j$, and subtracting the two sums in (6) termwise gives the displayed difference; the bound uses $0 \le \rho_j - 1 \le e^{\zeta(\alpha)} - 1$. □

Regret matching under discounting (Lemma 2).

Lemma 2 (Regret matching is invariant under DCFR's factors)**.** *Let R' be obtained from $R \in \mathbb{R}^{\mathcal{A}(I)}$ by multiplying every positive entry by some $d^+ > 0$ and every non-positive entry by some $d^- \ge 0$. Regret matching returns the same strategy on R and on R'.*

Proof. The positive parts of R' are d^+ times those of R, and normalization removes d^+; non-positive entries stay non-positive, so their positive part is zero in both; if no entry of R is positive, both strategies are uniform. □ □

Proposition 6 (Readout touches only visited information sets)**.** *Let $\mathcal{V}_i \subseteq \mathcal{I}_i$ be the information sets visited on player i's own playthrough at least once in T iterations, with regret tables initialized to a constant. (i) For $I \notin \mathcal{V}_i$, $\sigma_i^t(I, \cdot)$ is uniform for every t, and so is $\bar{\sigma}_i^T(I, \cdot)$. (ii) If $I \in \mathcal{V}_i$ is not a first decision point, then $I^\uparrow \in \mathcal{V}_i$. (iii)* FINISH *run over $\mathcal{V}_i$ alone, in order of increasing own depth, returns $S^T(I, \cdot)$ for every $I \in \mathcal{V}_i$; with (i) this is the exact average as a behavioral policy at every information set: at visited sets by finalizing their tables, at never-visited sets by the uniform fallback. The raw sums of Theorem 1 at never-visited sets are not materialized, and the cost is $O(\sum_{I \in \mathcal{V}_i} |\mathcal{A}(I)|)$.*

Proof. (i) $R(I, \cdot)$ is updated only at own visits, so it keeps its constant initialization, regret matching returns the uniform strategy, and every term of $S^T(I, \cdot)$ is a multiple of it. (ii) A playthrough that reaches $h \in I$ passes through $I^{\uparrow}$. (iii) The induction of Theorem 1 reads only predecessors, which by (ii) lie in $\mathcal{V}_i$ and precede I in own depth. □ □

B ADDITIONAL RESULTS

Table 3: Game sizes after expansion into an explicit tree (own decisions: maximal number of a player's decisions on one path), and exploitability of the exact uniform average (geometric mean over 20 seeds). One run of 10^7 iterations takes 60–95 s on the games up to Liar's Dice(1,7) and 160 s on Liar's Dice(1,8), on one core.

	histories	information sets	own decisions	$T = 10^5$	$T = 10^6$	$T = 10^7$
Kuhn poker	58	12	2	4.9×10^{-3}	1.7×10^{-3}	6.7×10^{-4}
Leduc poker	9,457	936	4	4.6×10^{-1}	1.2×10^{-1}	3.2×10^{-2}
Goofspiel(4)	1,077	162	3	1.2×10^{-2}	3.6×10^{-3}	1.0×10^{-3}
Liar's Dice(1,4)	8,181	1,024	5	9.4×10^{-2}	1.5×10^{-2}	2.7×10^{-3}
Goofspiel(5)	26,931	2,124	4	9.0×10^{-2}	2.4×10^{-2}	6.8×10^{-3}
Liar's Dice(1,6)	294,883	24,576	7	3.6×10^{-1}	1.5×10^{-1}	4.9×10^{-2}
Goofspiel(6)	969,523	34,482	5	2.4×10^{-1}	9.4×10^{-2}	2.7×10^{-2}
Liar's Dice(1,7)	1,605,591	114,688	8	4.7×10^{-1}	2.3×10^{-1}	9.2×10^{-2}
Liar's Dice(1,8)	8,388,553	524,288	9	5.7×10^{-1}	2.9×10^{-1}	1.3×10^{-1}

Table 4: As Table 1, after 10^5 and 10^6 iterations.

	STOCH/OWN	OPT/OWN	STOCH/OPP	OPT/OPP	LWA
$T = 100,000$					
Kuhn poker	1.26 [1.07,1.48]	1.00 [1.00,1.00]	1.03 [0.97,1.09]	1.00 [0.99,1.00]	1.02 [0.94,1.11]
Leduc poker	1.11 [1.07,1.16]	1.03 [1.02,1.05]	1.00 [0.99,1.00]	1.02 [1.01,1.03]	1.00 [0.99,1.00]
Goofspiel(4)	1.82 [1.38,2.42]	1.43 [1.12,1.85]	1.02 [0.98,1.07]	1.20 [1.04,1.44]	1.02 [0.93,1.13]
Liar's Dice(1,4)	1.18 [1.11,1.25]	1.09 [1.07,1.12]	1.00 [0.99,1.01]	1.03 [1.01,1.04]	0.98 [0.96,1.00]
Goofspiel(5)	1.55 [1.44,1.67]	1.27 [1.19,1.35]	1.00 [0.99,1.01]	1.03 [0.98,1.07]	1.00 [0.97,1.02]
Liar's Dice(1,6)	1.08 [1.06,1.10]	1.04 [1.03,1.05]	1.01 [1.00,1.01]	1.01 [1.01,1.02]	0.99 [0.98,0.99]
Goofspiel(6)	1.22 [1.17,1.26]	1.12 [1.08,1.17]	1.01 [1.00,1.02]	1.04 [1.02,1.06]	1.01 [0.99,1.03]
Liar's Dice(1,7)	1.06 [1.05,1.08]	1.02 [1.02,1.02]	1.00 [1.00,1.00]	1.01 [1.01,1.01]	0.99 [0.99,1.00]
Liar's Dice(1,8)	1.04 [1.03,1.05]	1.01 [1.01,1.01]	1.00 [1.00,1.01]	1.01 [1.00,1.01]	0.99 [0.99,1.00]
$T = 1,000,000$					
Kuhn poker	1.34 [1.02,1.87]	1.00 [1.00,1.00]	1.03 [0.96,1.11]	1.00 [0.99,1.00]	1.05 [0.90,1.30]
Leduc poker	1.49 [1.34,1.67]	1.09 [1.05,1.13]	1.00 [0.99,1.00]	1.04 [1.03,1.06]	1.03 [1.01,1.04]
Goofspiel(4)	3.61 [2.90,4.54]	2.67 [2.09,3.36]	1.05 [1.00,1.11]	1.39 [1.20,1.63]	1.01 [0.90,1.13]
Liar's Dice(1,4)	1.97 [1.73,2.24]	1.27 [1.19,1.35]	1.00 [0.98,1.01]	1.15 [1.11,1.19]	1.01 [0.97,1.04]
Goofspiel(5)	2.57 [2.31,2.84]	2.42 [2.06,2.85]	1.00 [0.99,1.01]	1.21 [1.13,1.30]	1.13 [1.06,1.21]
Liar's Dice(1,6)	1.37 [1.31,1.44]	1.12 [1.11,1.13]	1.01 [1.00,1.01]	1.05 [1.04,1.06]	0.98 [0.97,0.99]
Goofspiel(6)	1.39 [1.29,1.51]	1.32 [1.25,1.41]	1.00 [0.99,1.00]	1.03 [1.01,1.05]	1.01 [0.99,1.03]
Liar's Dice(1,7)	1.26 [1.23,1.30]	1.06 [1.06,1.07]	1.01 [1.00,1.01]	1.03 [1.03,1.03]	0.98 [0.98,0.99]
Liar's Dice(1,8)	1.23 [1.19,1.26]	1.03 [1.03,1.04]	1.00 [1.00,1.01]	1.02 [1.01,1.02]	0.99 [0.98,1.00]

Agreement with the solver OpenSpiel ships. To separate our implementation of STOCH/OWN from the solver OpenSpiel ships, we pin both exploitability functions against OpenSpiel's Python bindings on a uniform strategy, then run the two solvers side by side (Table 9): their exploitabilities agree to 2.2×10^{-5} on Kuhn and 5.0×10^{-7} on Leduc at the last horizon before they part, $T = 10^3$ and $T = 10^4$ respectively. That horizon is where their random paths separate, not where they disagree about the update rule. Past it the two are no longer on a common random path, so the ratio compares two runs of the same algorithm, and the seed spread at the last horizon ($9.5\times$ on Kuhn, $1.5\times$ on Leduc) is what keeps the agreement before it from being read as convergence to a common point.

Table 5: Ablation: the opponent also explores on the player's own playthrough, at four levels (10 seeds, 10^7 iterations). *own gap* is STOCH/OWN divided by the exact average on the same trajectory; *vs* $\varepsilon_{-i} = 0$ pairs the level with the $\varepsilon_{-i} = 0$ run of the same seed, so it is a drop and not a difference of two marginal means. The last column is the exploitability of the exact average itself: the rule approaches 1 by damaging the algorithm, and that is the price it pays. $\downarrow$ counts the seeds whose paired ratio fell below 1, with the exact two-sided sign test on that count.

game	ε_{-i}	own gap	vs $\varepsilon_{-i}=0$	95% CI	$\downarrow$	EXACT abs.
Kuhn	0.0	1.654 [1.244,2.215]	−−	−−	−−	0.000697
Kuhn	0.1	0.983 [0.886,1.083]	0.594	[0.431,0.814]	9/10, $p = 0.021$	0.000618
Kuhn	0.25	1.010 [0.923,1.132]	0.611	[0.458,0.827]	8/10, $p = 0.109$	0.000606
Kuhn	0.6	0.980 [0.942,1.016]	0.592	[0.446,0.789]	9/10, $p = 0.021$	0.000734
Leduc	0.0	2.522 [2.109,3.161]	−−	−−	−−	0.0322
Leduc	0.1	1.137 [1.088,1.196]	0.451	[0.351,0.555]	10/10, $p = 0.002$	0.0339
Leduc	0.25	1.011 [0.996,1.026]	0.401	[0.323,0.476]	10/10, $p = 0.002$	0.039
Leduc	0.6	1.000 [0.989,1.011]	0.397	[0.315,0.476]	10/10, $p = 0.002$	0.0581
Goofspiel(4)	0.0	7.530 [4.016,16.742]	−−	−−	−−	0.00103
Goofspiel(4)	0.1	1.487 [1.113,1.964]	0.198	[0.079,0.402]	10/10, $p = 0.002$	0.00154
Goofspiel(4)	0.25	1.107 [1.004,1.205]	0.147	[0.067,0.283]	10/10, $p = 0.002$	0.00135
Goofspiel(4)	0.6	1.029 [0.953,1.093]	0.137	[0.060,0.264]	10/10, $p = 0.002$	0.00232
Liar's Dice(1,4)	0.0	5.160 [4.200,6.346]	−−	−−	−−	0.0029
Liar's Dice(1,4)	0.1	1.075 [0.995,1.200]	0.208	[0.166,0.256]	10/10, $p = 0.002$	0.00327
Liar's Dice(1,4)	0.25	1.016 [0.995,1.041]	0.197	[0.159,0.244]	10/10, $p = 0.002$	0.00313
Liar's Dice(1,4)	0.6	1.015 [1.006,1.025]	0.197	[0.160,0.242]	10/10, $p = 0.002$	0.00603

Table 6: Discount sweep: the exact weighted average accumulated on the replay trajectory divided by the same average on the SCF trajectory, paired by seed (10 seeds, $T = 10^7$). α drives the positive-regret factor and β is held at 0; the middle column is $e^{\zeta(\alpha)}$, the coefficient bound of Proposition 5. The measured ratio drops by a factor of 1.1–1.4 in the first step and stays inside 0.98–1.12 from $\alpha = 1.25$ on. Cell format is Table 7's as well.

α	$e^{\zeta(\alpha)}$	Leduc	Goofspiel(4)	Liar's Dice(1,4)	Liar's Dice(1,6)
1.0	∞	1.790 [1.757,1.825]	1.117 [1.102,1.131]	1.572 [1.556,1.586]	1.452 [1.421,1.478]
1.1	3.95×10^4	1.276 [1.256,1.296]	1.036 [1.013,1.060]	1.259 [1.235,1.286]	1.199 [1.174,1.221]
1.25	99	1.100 [1.076,1.125]	1.034 [1.003,1.065]	1.075 [1.052,1.098]	1.100 [1.081,1.117]
1.5	13.6	1.086 [1.054,1.117]	0.985 [0.958,1.010]	1.045 [1.030,1.059]	1.113 [1.091,1.141]
2.0	5.18	1.070 [1.045,1.094]	1.021 [0.980,1.068]	1.044 [1.026,1.062]	1.106 [1.082,1.130]
3.0	3.33	1.098 [1.078,1.119]	0.997 [0.962,1.033]	1.027 [1.012,1.041]	1.098 [1.087,1.110]
4.0	2.95	1.070 [1.051,1.089]	0.989 [0.956,1.028]	1.048 [1.031,1.066]	1.116 [1.094,1.143]
6.0	2.77	1.102 [1.083,1.123]	1.004 [0.972,1.045]	1.028 [1.011,1.043]	1.077 [1.055,1.098]
10.0	2.72	1.085 [1.069,1.101]	1.008 [0.977,1.040]	1.032 [1.017,1.047]	1.081 [1.062,1.103]
30.0	2.72	1.075 [1.052,1.102]	1.035 [1.001,1.084]	1.036 [1.014,1.057]	1.077 [1.052,1.104]

The chance-law tilt. Two readings were registered before the sweep ran: (A) for every game and every tilted level $\kappa < 1$ the own-playthrough rule stays at least 1.3 times as exploitable as the exact average, so that the drift survives a non-uniform chance law; (B) for every game and every level including $\kappa = 1$ the opponent-playthrough rule stays inside $[0.95, 1.05]$. Both fail. (A) fails on Kuhn at $\kappa = 0.5$ (1.09), and (B) on five of the sixteen cells: Kuhn at $\kappa = 0.5$ (0.92) and at $\kappa = 0.1$ (1.11), Leduc at $\kappa = 0.25$ (1.79), and Liar's Dice(1,4) and (1,6) at $\kappa = 0.1$ (1.06 and 1.10). Table 13 re-reads the largest of them at twenty seeds per level and finds a band, not a level, with the levels on either side back inside. The sweep therefore does not license the reading that the mechanism is the playthrough, not the uniform chance law; the ratios are used where they are contrasts between the two placements.

B.1 THE DISCOUNT SCHEDULES ACROSS THE SWEEP

The β sweep of Table 7 moves the exponent that scales negative cumulative regrets while α is held at 30. The two schedules differ at $\beta = 0$: SCF halves a negative regret at every visit, replay multiplies

Table 7: Negative-regret sweep at $\alpha = 30$, where α's factor is within 10^{-9} of one from $t = 2$ on and only the negative side moves across these rows. Cell format as in Table 6: ratio, 95% bootstrap interval, and seeds with ratio ≥ 1. The sweep is not monotone: $\beta = 1$ is below 1 on Leduc and Liar's Dice(1,6) while $\beta = 0.5$ is the highest row on three of the four games.

β	Leduc	Goofspiel(4)	Liar's Dice(1,4)	Liar's Dice(1,6)
0.0	1.075 [1.052,1.102] (10/10)	1.035 [1.001,1.084] (8/10)	1.036 [1.014,1.057] (9/10)	1.077 [1.052,1.104] (10/10)
0.5	2.175 [2.127,2.223] (10/10)	0.958 [0.899,1.021] (4/10)	1.463 [1.406,1.518] (10/10)	1.758 [1.686,1.833] (10/10)
1.0	0.868 [0.819,0.915] (0/10)	1.005 [0.779,1.258] (5/10)	1.060 [0.989,1.125] (8/10)	0.894 [0.821,0.982] (2/10)
2.0	1.072 [0.971,1.189] (7/10)	1.200 [0.951,1.525] (7/10)	1.051 [0.845,1.370] (3/10)	0.950 [0.889,1.007] (2/10)
4.0	0.994 [0.909,1.075] (6/10)	0.982 [0.620,1.602] (3/10)	0.919 [0.810,1.050] (3/10)	0.982 [0.945,1.019] (4/10)
30.0	0.960 [0.875,1.045] (4/10)	1.179 [0.778,1.701] (7/10)	1.023 [0.822,1.259] (4/10)	0.995 [0.928,1.075] (5/10)

Table 8: What the end-of-run read is worth ($T = 10^7$, 10 seeds): the recurrence rule with its closing read dropped, divided by the same rule with it.

run	with the closing read dropped	EXACT absolute
Kuhn	1.046 [1.000,1.144]	0.000697
Leduc	1.249 [1.060,1.645]	0.0322
Goofspiel(4)	1.500 [1.137,2.013]	0.00103
Liar's Dice(1,4)	3.099 [2.219,4.503]	0.0029

it by $2^{-(t-c_I)}$. Its intermediate levels are not monotone: $\beta = 0.5$ gives 1.4–2.2 on three games. The exponent moves the trajectory as well as the two schedules, so the ratio is a property of two runs and not of the semantics alone. The near-tie at $\alpha = \beta = 30$ localizes the same effect. There the discounted run reduces to the undiscounted one to 5.7×10^{-9} at 10^3 iterations, and only there: the residual is amplified by regret matching's threshold at zero into an 86% difference by 3×10^4 iterations. This is why the sweep is read as a distribution over seeds and not run against run.

Reproducibility. The code expands OpenSpiel games into arrays and runs the sampler and all averaging rules in compiled kernels; the games come from OpenSpiel 2.0.1, and a single run of 10^7 iterations takes 1–3 minutes on one core. The recurrence rule matches the brute-force exact average to 10^{-9} on four games, and on all nine games at 10^3, 10^4 and 10^5 iterations (Table 10); the best-response computation matches OpenSpiel's to 10^{-9}; and lazy-weighted averaging differs from the exact average. An anonymous code release accompanies this submission: the sampler and the averaging rules ship as a precompiled extension, together with the game expansion, the run driver, the per-run outputs behind the main experiments and their sweeps, and the seeds and checkpoints each of them used.

Table 9: STOCH/OWN against the outcome-sampling solver OpenSpiel ships, both run to the same horizons and scored by the same exploitability function, pinned against OpenSpiel's own before the runs, so a metric mismatch cannot pass for a solver difference. The ratio is the geometric mean of theirs/ours over 12 seeds per game, printed to six digits because at three an exact match and a gap of 10^{-7} are the same string. The *regime* of each row, and why past the separation horizon a ratio is a distributional comparison, are in the text above.

game	T	theirs	ours	ratio	max rel. dev.	regime
Kuhn	100	2.023187e-01	2.023186e-01	1.000000	1.76e-06	same trajectory
Kuhn	1,000	1.016900e-01	1.016898e-01	1.000002	2.24e-05	same trajectory
Kuhn	10,000	2.561820e-02	2.311113e-02	1.108479	4.05e+00	independent
Kuhn	100,000	6.267807e-03	5.632288e-03	1.112835	2.36e+00	independent
Kuhn	1,000,000	2.835757e-03	1.781558e-03	1.591728	2.71e+00	independent
Leduc	100	2.519699e+00	2.519699e+00	1.000000	8.73e-09	same trajectory
Leduc	1,000	2.316277e+00	2.316277e+00	1.000000	7.42e-09	same trajectory
Leduc	10,000	1.159729e+00	1.159729e+00	1.000000	4.97e-07	same trajectory
Leduc	100,000	5.045276e-01	5.034738e-01	1.002093	1.91e-01	independent
Leduc	1,000,000	1.715608e-01	1.809964e-01	0.947868	2.65e-01	independent

Table 10: The recurrence rule against brute force on all nine games, two seeds each: $\max |S_{\text{rec}} - S_{\text{bf}}|$ divided by the largest entry of the brute-force table, maximized over the two seeds and the three weight families (uniform, linear, quadratic). Brute force replays every iteration on every information set; the recurrence rule needs one term per visit. The deviations grow with T and with the dynamic range of the table, which is floating-point accumulation and not a missing term: the own playthrough never visits 96% of Liar's Dice(1,8)'s information sets at $T = 10^5$ and both rules average the uniform policy there.

game	$T = 1{,}000$	$T = 10{,}000$	$T = 100{,}000$
Kuhn poker	1.5×10^{-15}	7.3×10^{-15}	2.5×10^{-14}
Leduc poker	1.2×10^{-14}	1.6×10^{-13}	1.3×10^{-12}
Goofspiel(4)	7.2×10^{-15}	5.1×10^{-14}	2.7×10^{-13}
Goofspiel(5)	5.5×10^{-15}	1.2×10^{-14}	7.6×10^{-14}
Goofspiel(6)	4.4×10^{-15}	1.4×10^{-14}	6.2×10^{-14}
Liar's Dice(1,4)	1.7×10^{-14}	1.1×10^{-13}	1.3×10^{-12}
Liar's Dice(1,6)	1.4×10^{-14}	1.7×10^{-13}	1.5×10^{-12}
Liar's Dice(1,7)	1.3×10^{-14}	1.8×10^{-13}	1.5×10^{-12}
Liar's Dice(1,8)	1.4×10^{-14}	1.7×10^{-13}	2.0×10^{-12}

Table 11: OpenSpiel's own Python outcome-sampling solver, 10 seeds: the exploitability of the average it returns against that of the recurrence rule run as a side table on the same trajectory, both measured with OpenSpiel's exploitability function, and the mass-weighted L_1 distance between the two returned policies. The ratio is native over exact, so a value above 1 means the solver's own average is the more exploitable one. Kuhn's intervals cover 1 because its absolute numbers are an order of magnitude smaller; Leduc's clears 1 from 10^5 on, and both match the magnitudes Table 4 reports (1.34 and 1.49 at 10^6) without sharing a line of code.

game	T	native/exact	L_1	native	exact
Kuhn poker	10,000	1.042 [0.923,1.162]	0.0086	0.0289	0.0277
Kuhn poker	100,000	1.227 [0.969,1.547]	0.0026	0.0065	0.0053
Kuhn poker	1,000,000	1.318 [0.794,2.160]	0.0012	0.0028	0.0022
Leduc poker	10,000	1.013 [0.988,1.036]	0.0762	1.1709	1.1560
Leduc poker	100,000	1.095 [1.059,1.134]	0.0484	0.5072	0.4634
Leduc poker	1,000,000	1.405 [1.330,1.493]	0.0394	0.1688	0.1202

Table 12: The chance law tilted away from uniform: each chance node's child probabilities become the normalised ladder κ^k, $k = 0, 1, \ldots$ over its children, which leaves the tree, the information sets and both playthroughs untouched and changes only how likely each branch is dealt. Ratios to the exact average at 10^7 iterations, 10 seeds per cell (seeds 0–9; the $\kappa = 1$ rows coincide with the $\varepsilon_{-i} = 0$ rows of Table 5 on the games the two tables share); L_1 is the mass-weighted policy distance to the exact average, mean over seeds 0–2 (the finer Leduc grid of Table 13 uses twenty). OPT/OPP is the rule that moves most with the tilt (Leduc: $1.12 \to 3.32$ at $\kappa = 0.25$); STOCH/OPP stays within 1.8 in every cell where the same stochastic increment on the own playthrough reaches 841. The tilt also moves the exact rule's own exploitability, downward by up to $96\times$ as the deep branches that carry the error lose weight, so these ratios are read against a moving denominator and Table 14 reports the two sides separately.

game	κ	STOCH/OWN	STOCH/OPP	OPT/OPP	L1 own	L1 opp
Kuhn poker	1.0	1.65 [1.24,2.21]	1.00 [0.96,1.05]	1.00 [1.00,1.01]	0.0010	0.0001
Kuhn poker	0.5	1.09 [0.84,1.44]	0.92 [0.84,0.99]	1.00 [0.99,1.01]	0.0008	0.0003
Kuhn poker	0.25	1.43 [1.07,2.06]	0.97 [0.87,1.10]	1.01 [1.00,1.03]	0.0015	0.0002
Kuhn poker	0.1	27.02 [4.88,144.92]	1.11 [1.01,1.24]	1.00 [1.00,1.01]	0.0307	0.0007
Leduc poker	1.0	2.52 [2.11,3.16]	0.99 [0.98,1.00]	1.12 [1.09,1.15]	0.0534	0.0025
Leduc poker	0.5	2.73 [2.25,3.36]	1.04 [1.01,1.07]	1.48 [1.35,1.64]	0.0488	0.0041
Leduc poker	0.25	18.81 [9.50,36.44]	1.79 [1.31,2.49]	3.32 [2.35,4.74]	0.0814	0.0207
Leduc poker	0.1	840.96 [463.21,1354.11]	1.03 [0.79,1.50]	1.59 [1.12,2.80]	0.1559	0.0170
Liar's Dice(1,4)	1.0	5.16 [4.20,6.35]	0.98 [0.97,1.00]	1.28 [1.18,1.41]	0.1177	0.0010
Liar's Dice(1,4)	0.5	16.21 [7.35,33.34]	0.99 [0.97,1.01]	1.09 [1.08,1.10]	0.1607	0.0009
Liar's Dice(1,4)	0.25	36.01 [12.31,85.85]	1.00 [0.98,1.02]	1.00 [0.99,1.00]	0.1778	0.0013
Liar's Dice(1,4)	0.1	24.52 [9.16,60.82]	1.06 [0.87,1.35]	0.99 [0.98,1.00]	0.1052	0.0050
Liar's Dice(1,6)	1.0	2.27 [1.95,2.66]	1.00 [0.99,1.01]	1.09 [1.08,1.11]	0.1474	0.0075
Liar's Dice(1,6)	0.5	27.07 [19.68,36.87]	1.01 [1.00,1.03]	1.03 [0.95,1.11]	0.2203	0.0051
Liar's Dice(1,6)	0.25	50.62 [33.07,75.98]	1.04 [1.00,1.07]	1.04 [1.00,1.11]	0.1401	0.0069
Liar's Dice(1,6)	0.1	15.64 [7.91,33.18]	1.10 [0.93,1.44]	1.02 [1.00,1.04]	0.0870	0.0135

Table 13: The one game with a cell outside noise in Table 12, swept finely: Leduc poker, 20 seeds per level (seeds 0–19; the runs of seeds 0–9 are those behind the $\kappa = 0.25$ row of Table 12, which averages ten seeds there against twenty here, and three for its L_1 columns against twenty here). Ratios to the exact average at 10^7 iterations with 95% bootstrap intervals. Every interval excludes one, and the level effect is a band, not a monotone trend: the ratio is 1.32 at $\kappa = 0.35$, rises to 1.84 at 0.25, and is 1.47 and 1.80 again at 0.20 and 0.15. The own-playthrough rule is 5–546 times as exploitable over the same five levels, so the opponent-playthrough rule's excursion is not the tilt moving every rule.

κ	STOCH/OWN	STOCH/OPP	OPT/OPP	L1 own	L1 opp
0.35	5.03 [4.03,6.37]	1.32 [1.24,1.45]	1.68 [1.52,1.89]	0.0553	0.0090
0.30	8.03 [5.98,11.15]	1.58 [1.39,1.83]	1.93 [1.66,2.30]	0.0678	0.0124
0.25	28.30 [17.11,46.72]	1.84 [1.47,2.32]	3.02 [2.31,3.99]	0.0968	0.0177
0.20	136.96 [89.35,208.17]	1.47 [1.16,1.97]	2.45 [1.96,3.20]	0.1568	0.0237
0.15	546.08 [401.87,734.17]	1.80 [1.36,2.43]	3.73 [2.65,5.33]	0.1673	0.0196

Table 14: The two sides of every ratio in Table 12, as the factor by which each rule's own exploitability changes between $\kappa = 1$ and $\kappa = 0.1$. The opponent-playthrough stochastic rule travels by 1.04–1.11 times whatever factor the exact rule travels by, on all four games; the own-playthrough rule travels by 4.7–333 times as much, and on Kuhn and Leduc it rises while the exact rule falls. The same three rules ran on the same trajectories throughout, so the only difference between a row and the reference is where the average is accumulated.

game	exact	STOCH/OPP	STOCH/OWN	OPT/OPP	STOCH/OPP/exact	STOCH/OWN/exact
Kuhn poker	0.2403	0.2665	3.9251	0.2394	1.11	16.33
Leduc poker	0.0131	0.0137	4.3788	0.0186	1.04	333.50
Liar's Dice(1,4)	0.0415	0.0448	0.1972	0.0320	1.08	4.75
Liar's Dice(1,6)	0.0104	0.0115	0.0719	0.0097	1.10	6.89

Table 15: The constant of Proposition 3 against the exploitability ratio of STOCH/OPP, a descriptive comparison: the proposition bounds the local policy error, which is the L_1 column of Table 12, not the ratio. One row per (game, κ): $\max_I B(I)$ on the tilted chance law with $\varepsilon = 0.6$ and d_{-i} the opponent's decision count above h, and the ratio of the opponent-playthrough rule to the exact average at 10^7 iterations. The constant moves by up to ten orders of magnitude within a game (Leduc, $7.5 \times 10^4 \to 8.6 \times 10^{14}$) and the ratio does not follow: the rank correlation between $\log B(I)$ and $|\text{ratio} - 1|$ is $+0.19$ pooled over the sixteen cells. $B(I)$ is a worst-case constant, already 2.3×10^9 at $\kappa = 1$ on Liar's Dice(1,6) and looser than that on the tilted laws, so it certifies a rule without ordering these cells.

game	κ	$\max_I B(I)$	$\text{median}_I B(I)$	STOCH/OPP	$\lvert\text{ratio} - 1\rvert$
Kuhn poker	1.0	2.0×10^1	2.0×10^1	1.001	0.001
Kuhn poker	0.5	7.0×10^1	3.5×10^1	0.915	0.085
Kuhn poker	0.25	3.5×10^2	8.8×10^1	0.971	0.029
Kuhn poker	0.1	4.1×10^3	4.1×10^2	1.110	0.110
Leduc poker	1.0	7.5×10^4	3.0×10^3	0.986	0.014
Leduc poker	0.5	1.8×10^7	1.0×10^5	1.039	0.039
Leduc poker	0.25	2.5×10^{10}	1.6×10^7	1.793	0.793
Leduc poker	0.1	8.6×10^{14}	1.7×10^{10}	1.030	0.030
Liar's Dice(1,4)	1.0	5.1×10^5	2.8×10^3	0.984	0.016
Liar's Dice(1,4)	0.5	7.1×10^6	2.0×10^4	0.992	0.008
Liar's Dice(1,4)	0.25	2.3×10^8	3.2×10^5	1.000	0.000
Liar's Dice(1,4)	0.1	3.9×10^{10}	2.2×10^7	1.062	0.062
Liar's Dice(1,6)	1.0	2.3×10^9	2.9×10^5	0.999	0.001
Liar's Dice(1,6)	0.5	2.5×10^{11}	1.6×10^7	1.015	0.015
Liar's Dice(1,6)	0.25	1.2×10^{14}	9.3×10^8	1.035	0.035
Liar's Dice(1,6)	0.1	7.9×10^{17}	9.9×10^{11}	1.104	0.104

Table 16: Regret matching$^+$ trajectory (regrets clamped at zero after each update), 10 seeds, $T = 10^7$: ratios to the exact uniform average as in Table 1. The exact average on this trajectory is 2.35 [2.22, 2.47] (Leduc), 1.59 [1.17, 2.07] (Goofspiel(4)), 3.27 [2.86, 3.72] (Liar's Dice(1,4)), and 1.45 [1.40, 1.50] (Liar's Dice(1,6)) times as exploitable as the exact average of the regret-matching trajectory with the same seeds.

	STOCH/OWN	OPT/OWN	STOCH/OPP	OPT/OPP	LWA
Leduc poker	1.00 [0.99,1.01]	1.00 [1.00,1.00]	1.00 [1.00,1.00]	1.01 [1.01,1.01]	1.00 [0.99,1.00]
Goofspiel(4)	1.56 [1.22,2.29]	1.00 [1.00,1.00]	1.00 [1.00,1.00]	1.00 [0.99,1.00]	1.00 [1.00,1.01]
Liar's Dice(1,4)	1.00 [1.00,1.00]	1.00 [1.00,1.00]	1.00 [1.00,1.00]	1.00 [1.00,1.00]	0.98 [0.98,0.98]
Liar's Dice(1,6)	1.02 [1.01,1.02]	1.01 [1.01,1.02]	1.00 [1.00,1.01]	1.01 [1.01,1.01]	0.99 [0.98,0.99]

Table 17: Discounted trajectories against plain regret matching, $T = 10^7$, 20 seeds, paired by seed: exploitability of the exact weighted average on the discounted trajectory divided by exploitability of the exact uniform average on the regret-matching trajectory. Quadratic weights for DCFR $(1.5, 0)$, linear weights for linear CFR $(1, 1)$.

	DCFR, replay	DCFR, SCF	linear CFR, replay	linear CFR, SCF
Leduc poker	1.87 [1.78,1.97]	1.80 [1.70,1.90]	0.99 [0.93,1.06]	2.82 [2.57,3.09]
Goofspiel(4)	1.10 [0.93,1.30]	1.08 [0.90,1.28]	1.16 [0.92,1.50]	1.98 [1.44,2.85]
Liar's Dice(1,4)	2.26 [1.99,2.55]	2.19 [1.94,2.47]	0.96 [0.79,1.18]	6.23 [5.03,7.78]
Goofspiel(5)	1.52 [1.34,1.72]	1.54 [1.37,1.74]	1.17 [1.04,1.31]	3.16 [2.68,3.76]
Liar's Dice(1,6)	1.13 [1.07,1.19]	1.03 [0.98,1.09]	0.76 [0.71,0.80]	1.13 [1.04,1.22]

Table 18: Weighted averages and discounted regrets, $T = 10^7$, 20 seeds. Top: regret-matching trajectory; each rule's exploitability divided by that of the exact average *with the same weights*, $f(t) = t$ or t^2; SCF/OWN is the visit-triggered discount of the strategy sum, $\hat{S} \leftarrow ((t-1)/t)^\gamma \hat{S} +$ stochastic increment; the last two columns divide the exact weighted average's exploitability by the exact uniform average's. Bottom: regrets discounted with DCFR exponents $(1.5, 0)$ or linear-CFR exponents $(1, 1)$, by replay or by the single current factor (SCF); "replay/SCF" divides the exact weighted average's exploitability on the replay trajectory by that on the SCF trajectory, paired by seed; the other columns are ratios to the exact weighted average on the replay trajectory.

	linear, vs exact linear			quadratic, vs exact quadratic			exact weighted / exact uniform	
	STOCH/OWN	SCF/own	STOCH/OPP	STOCH/OWN	SCF/own	STOCH/OPP	linear	quadratic
Kuhn poker	1.49 [[1.17,1.94]]	1.34 [[1.07,1.69]]	0.99 [[0.96,1.02]]	1.41 [[1.13,1.80]]	1.09 [[0.89,1.37]]	0.99 [[0.97,1.01]]	1.79	2.59
Leduc poker	1.79 [[1.63,1.99]]	1.08 [[0.94,1.25]]	1.00 [[1.00,1.01]]	1.41 [[1.29,1.54]]	0.67 [[0.58,0.77]]	1.00 [[1.00,1.01]]	2.69	4.37
Goofspiel(4)	3.07 [[1.82,5.43]]	1.77 [[0.91,3.55]]	1.00 [[0.97,1.03]]	1.86 [[1.23,2.83]]	0.84 [[0.46,1.50]]	0.98 [[0.96,1.00]]	6.28	15.2
Liar's Dice(1,4)	1.10 [[0.89,1.35]]	0.56 [[0.43,0.70]]	1.00 [[0.99,1.00]]	1.04 [[0.89,1.20]]	0.32 [[0.26,0.38]]	1.00 [[1.00,1.00]]	8.78	16.4
Goofspiel(5)	2.51 [[2.13,2.96]]	1.66 [[1.36,1.98]]	1.00 [[0.99,1.01]]	1.87 [[1.60,2.18]]	1.01 [[0.86,1.18]]	1.00 [[0.99,1.00]]	3.69	6.39
Liar's Dice(1,6)	1.67 [[1.52,1.86]]	2.24 [[1.97,2.54]]	1.00 [[1.00,1.00]]	1.41 [[1.29,1.55]]	1.78 [[1.56,2.03]]	1.00 [[0.99,1.00]]	1.12	1.40
Liar's Dice(1,7)	1.61 [[1.50,1.73]]	2.00 [[1.87,2.15]]	1.00 [[1.00,1.01]]	1.43 [[1.35,1.52]]	1.75 [[1.64,1.88]]	1.00 [[1.00,1.01]]	1.05	1.20

	DCFR regrets $(\alpha, \beta) = (1.5, 0)$, quadratic average				linear-CFR regrets $(\alpha, \beta) = (1, 1)$, linear average			
	replay/SCF	STOCH/OWN	SCF/own	STOCH/OPP	replay/SCF	STOCH/OWN	SCF/own	STOCH/OPP
Leduc poker	1.04 [[1.02,1.07]]	1.01 [[1.00,1.02]]	1.28 [[1.25,1.31]]	1.00 [[1.00,1.00]]	0.35 [[0.32,0.38]]	2.34 [[2.08,2.65]]	1.49 [[1.38,1.61]]	0.99 [[0.99,1.00]]
Goofspiel(4)	1.02 [[0.97,1.08]]	3.51 [[2.59,4.94]]	3.47 [[2.67,4.66]]	1.00 [[0.99,1.01]]	0.59 [[0.40,0.87]]	5.32 [[3.62,7.93]]	3.82 [[2.70,5.48]]	1.02 [[0.97,1.06]]
Liar's Dice(1,4)	1.03 [[1.00,1.06]]	1.00 [[1.00,1.00]]	1.25 [[1.23,1.27]]	1.00 [[1.00,1.00]]	0.15 [[0.12,0.20]]	5.24 [[4.12,6.62]]	2.53 [[1.84,3.45]]	0.98 [[0.95,1.00]]
Goofspiel(5)	0.98 [[0.93,1.04]]	1.67 [[1.46,1.99]]	1.85 [[1.62,2.13]]	1.00 [[0.99,1.00]]	0.37 [[0.33,0.42]]	4.88 [[4.27,5.59]]	3.31 [[2.84,3.85]]	0.99 [[0.97,1.01]]
Liar's Dice(1,6)	1.09 [[1.06,1.13]]	1.02 [[1.01,1.02]]	1.32 [[1.30,1.34]]	1.01 [[1.00,1.01]]	0.67 [[0.62,0.73]]	2.03 [[1.86,2.22]]	3.00 [[2.75,3.28]]	1.01 [[1.00,1.01]]

Table 19: Exploration-rate sweep: ratio to the exact average after 10^7 iterations, 10 seeds per cell ($\varepsilon = 0.6$ rows use seeds 0–9 of the main runs).

	ε	STOCH/OWN	OPT/OWN	STOCH/OPP	OPT/OPP	LWA
Leduc poker	0.2	1.59 [1.48,1.71]	1.10 [1.02,1.22]	0.99 [0.98,1.00]	1.03 [1.02,1.05]	1.20 [1.13,1.27]
	0.4	2.13 [1.90,2.40]	1.15 [1.08,1.22]	0.99 [0.98,1.00]	1.10 [1.05,1.14]	1.14 [1.09,1.20]
	0.6	2.52 [2.11,3.16]	1.26 [1.17,1.35]	0.99 [0.98,1.00]	1.12 [1.09,1.15]	1.12 [1.08,1.17]
	0.8	3.05 [2.82,3.31]	1.27 [1.22,1.33]	1.00 [0.99,1.01]	1.17 [1.13,1.20]	1.03 [1.01,1.05]
	1.0	2.92 [2.36,3.62]	1.38 [1.19,1.65]	1.00 [0.99,1.00]	1.13 [1.10,1.15]	1.00 [0.99,1.00]
Goofspiel(4)	0.2	3.23 [1.43,7.21]	1.68 [1.24,2.39]	0.93 [0.86,1.02]	1.62 [1.30,2.03]	1.53 [0.99,2.46]
	0.4	4.60 [2.68,8.24]	2.19 [1.31,3.74]	0.94 [0.85,1.03]	1.61 [1.17,2.17]	1.25 [1.00,1.58]
	0.6	7.53 [4.02,16.74]	2.62 [1.93,3.98]	1.04 [1.00,1.08]	1.80 [1.43,2.37]	1.16 [1.01,1.33]
	0.8	8.51 [4.98,14.76]	3.09 [2.01,5.38]	1.01 [0.97,1.04]	1.66 [1.27,2.35]	1.16 [0.99,1.38]
	1.0	9.19 [5.87,16.05]	3.07 [2.03,4.69]	0.97 [0.93,1.01]	1.64 [1.33,2.05]	0.97 [0.94,1.00]
Liar's Dice(1,4)	0.2	5.75 [4.34,7.33]	1.48 [1.12,1.94]	1.04 [0.98,1.12]	1.16 [1.08,1.27]	1.18 [0.99,1.41]
	0.4	4.96 [2.98,7.82]	1.70 [1.31,2.21]	1.04 [1.00,1.08]	1.33 [1.22,1.52]	1.25 [1.01,1.61]
	0.6	5.16 [4.20,6.35]	2.22 [1.44,3.46]	0.98 [0.97,1.00]	1.28 [1.18,1.41]	1.21 [1.05,1.41]
	0.8	3.89 [2.39,6.20]	1.91 [1.45,2.60]	0.99 [0.98,1.01]	1.34 [1.25,1.45]	1.02 [0.99,1.04]
	1.0	2.63 [2.08,3.30]	1.47 [1.22,1.82]	0.99 [0.98,1.00]	1.27 [1.23,1.32]	0.99 [0.98,1.00]

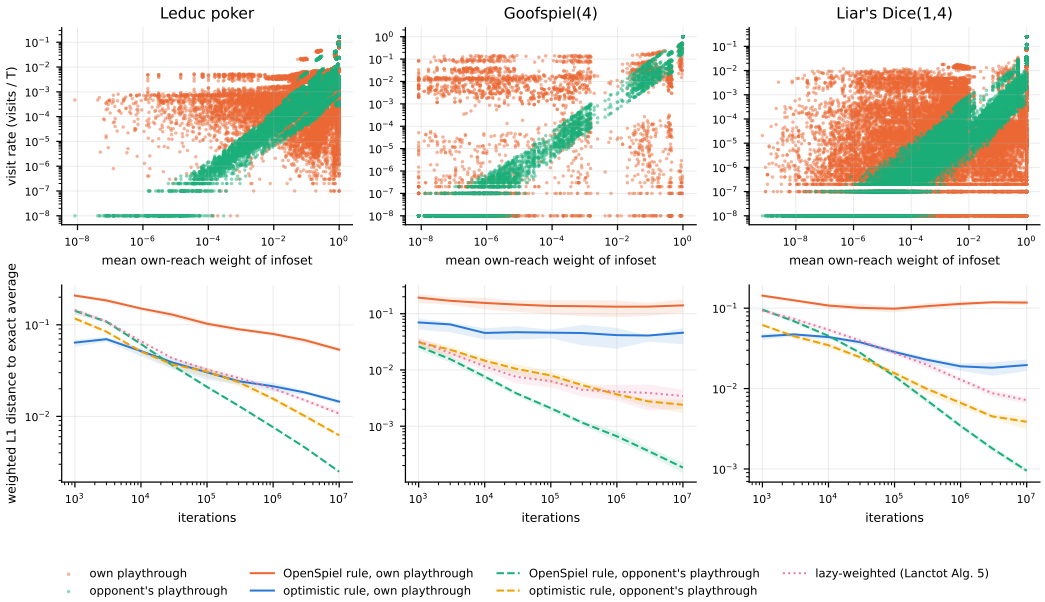

Figure 2: Figure 1 (top row) together with the distance between each rule's average and the exact average, $\sum_I m_I \, \|\bar{\sigma}(I,\cdot) - \bar{\sigma}^{\text{exact}}(I,\cdot)\|_1 / \sum_I m_I$ with $m_I = S^T(I,\cdot)$ summed over actions, against iterations (bottom row), with bands the interquartile range over the twenty seeds.

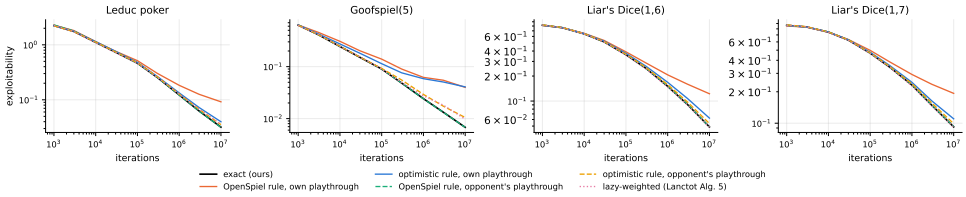

Figure 3: Exploitability of the average returned by each rule, geometric mean over 20 seeds on one shared regret trajectory per seed; four of nine games (all in Figure 4). Solid: accumulated on the player's own playthrough; dashed: on the opponent's.

Table 20: Training time for 10^6 iterations, median of 5 single-thread runs, each rule implemented alone in the same kernel; "none" updates regrets only. The recurrence rule's training and its closing pass are timed separately, and the ratio is their sum over STOCH/OWN.

game	none (s)	STOCH/OWN (s)	recurrence train (s)	closing pass (s)	ratio
Kuhn poker	0.98	0.99	0.99	0.000	1.009
Leduc poker	1.40	1.42	1.45	0.000	1.017
Goofspiel(4)	1.39	1.43	1.43	0.000	1.003
Liar's Dice(1,4)	1.30	1.33	1.34	0.000	1.006
Goofspiel(5)	1.93	1.98	2.02	0.000	1.021
Liar's Dice(1,6)	1.64	1.68	1.72	0.001	1.024
Goofspiel(6)	3.16	3.28	3.42	0.001	1.045
Liar's Dice(1,7)	1.72	1.76	1.79	0.002	1.017
Liar's Dice(1,8)	1.88	1.94	2.01	0.010	1.044

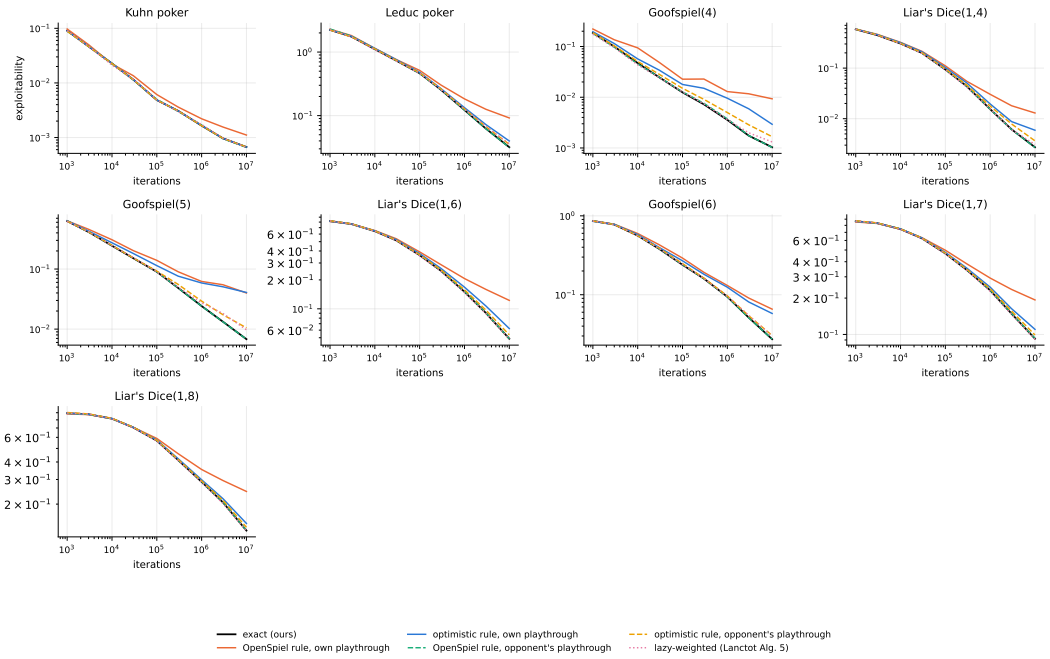

Figure 4: Figure 3 for all games with 20 seeds.

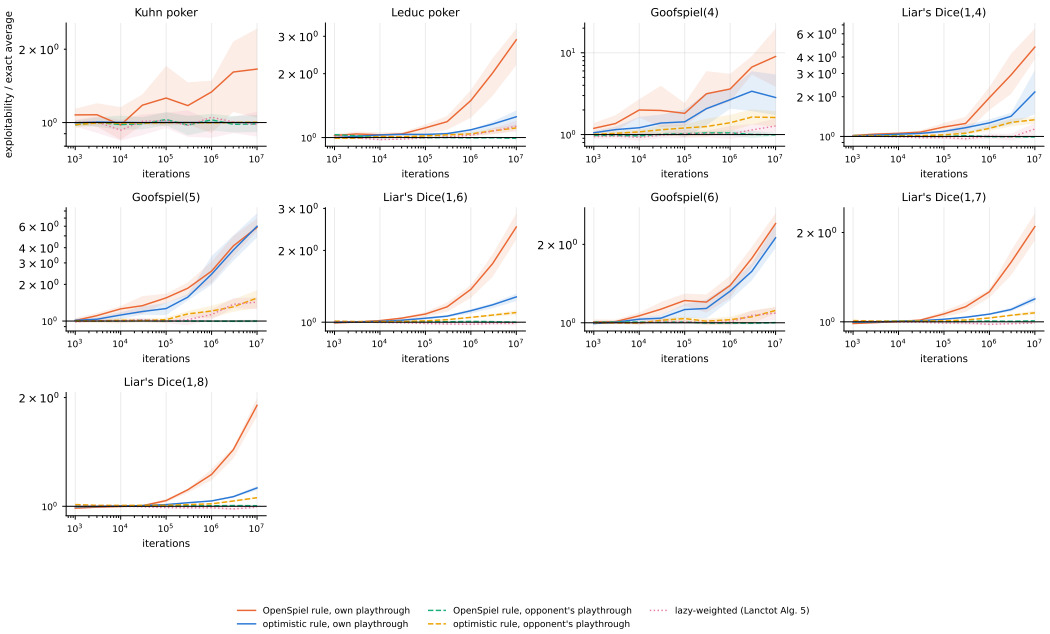

Figure 5: Ratio of each rule's exploitability to the exact average against iterations; line is the geometric mean over 20 seeds, band the interquartile range.