ADAPTIVE LOCAL PERTURBATION COUNTERFACTUAL REGRET MINIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Imperfect-information games can admit Nash equilibria whose behavior at low-probability or unreachable information sets is strategically implausible. We introduce ALP-CFR, a counterfactual-regret-minimization framework that uses information-set-level local perturbations and adapts their magnitude from cumulative reach information. Under-reached descendants generate protection requests that propagate toward ancestors, allowing the method to allocate larger local perturbations where downstream behavior receives too little effective reach. This paper develops the framework, its full-tree regret analysis, and the conditions needed for a sampling-based extension. We establish the scope of the claims carefully: the results concern perturbed-game regret, vanishing local perturbations, and sampling estimators under explicit assumptions; they do not, in general, establish convergence to an extensive-form perfect equilibrium (EFPE). Experiments on imperfect-information benchmark games will evaluate the effect of the protection mechanism on Nash and information-set regret metrics.

1 INTRODUCTION

Nash equilibrium (NE) is the standard solution concept for strategic games, but it is not sufficient to determine behavior at information sets that are unreachable under the equilibrium profile (Nash, 1951). This distinction is important in imperfect-information games: a small deviation by an opponent can move play to a part of the tree whose prescribed action has received little training. Sequential equilibrium (SE) and extensive-form perfect equilibrium (EFPE) refine NE by imposing sequential rationality and by studying limits of completely mixed, perturbed games, respectively (Kreps & Wilson, 1982; Selten, 1975).

Counterfactual regret minimization (CFR) is a successful framework for computing equilibria in large extensive-form games (Zinkevich et al., 2007a;b). Its regret updates are counterfactual: each information set receives a signal even when the current profile assigns it little probability. Nevertheless, the effective training signal can become highly uneven across histories and information sets. A global perturbation can force every action to remain possible, but it also keeps suboptimal actions alive after the strategically important part of the tree has been sufficiently explored. A fixed positive perturbation therefore gives a perturbed-game solution, not an EFPE limit.

This paper studies adaptive local perturbation counterfactual regret minimization (ALP-CFR). Instead of applying one perturbation to the whole game, ALP-CFR assigns a local lower-bound parameter to each information set. The parameter decreases with cumulative effective reach. When a downstream node is under-reached, it creates a protection request that is propagated to its upstream information sets. The upstream perturbation then reflects the protected downstream reach rather than only its own raw reach. The purpose is to allocate more training mass to poorly reached behavior while allowing the perturbation to vanish as the relevant reach accumulates.

The paper makes three contributions. First, it gives a precise full-tree algorithm with an explicit update boundary, protected-set state, and output policy convention. Second, it states conditional regret and sampling results without identifying information-set regret with EFPE. Third, it reports an auditable comparison on Kuhn, GoofSpiel-3, Leduc-3, and Leduc-5, including the external-sampling implementation MC-ALP-CFR where a completed run is available. The comparison separates exploitability, maximum information-set regret, and average information-set regret, and marks partial or unavailable experiments instead of imputing them.

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 main paper is organized as follows. Section 2 introduces the game, reach, counterfactual value, and evaluation notation. Section 3 places the method among CFR, sampling, and refinement-oriented algorithms. Section 4 describes the protected local update and its full-tree and sampling variants. Section 5 presents the controlled experiments and their limitations. Complete recursions, protection-state invariants, proofs, parameters, and additional tables are in Appendices A–C.

2 PRELIMINARIES

2.1 EXTENSIVE-FORM GAMES

We consider a finite extensive-form game $\mathcal{G} = (N, H, Z, A, P, \mathcal{I}, u, \chi)$ with two strategic players, chance, and perfect recall. Here H is the set of histories, Z is the set of terminal histories, $A(h)$ is the legal action set, and $P(h)$ identifies the player or chance at a non-terminal history. The chance distribution $\chi(\cdot \mid h)$ is fixed. The information sets $\mathcal{I}_i$ partition the histories controlled by player i, and all histories in an information set have the same legal actions. Utilities are bounded by $|u_i(z)| \le U$. The experiments use two-player zero-sum instances, including sequentialized simultaneous bidding for GoofSpiel-3.

A behavioral strategy assigns $\sigma_i(I, \cdot)$ in the simplex of actions at every information set. Let $\pi_i^\sigma(h)$ be player i's own reach, let $\pi_\chi(h)$ be the chance reach, and define

$$\pi_{-i,c}^\sigma(h) = \pi_\chi(h) \prod_{j \neq i} \pi_j^\sigma(h), \qquad \pi^\sigma(h) = \pi_i^\sigma(h) \pi_{-i,c}^\sigma(h).$$

The notation distinguishes own reach, opponent-and-chance counterfactual reach, and full reach. For an information set I, we use the sum of the corresponding history quantities over $h \in I$. If full reach is positive, the Bayes belief is $\mu^\sigma(h \mid I) = \pi^\sigma(h) / \sum_{h' \in I} \pi^\sigma(h')$. At an unreachable information set, beliefs are instead defined by a completely mixed limiting sequence in the usual SE construction.

The reach notation is deliberately more detailed than a single probability attached to an information set. A player's own reach excludes that player's future action at the point where a counterfactual value is evaluated, whereas the opponent-and-chance reach includes every factor determining how often the player would face the information set. Full reach includes both. This separation matters here: local perturbations use training reach, regret uses counterfactual reach, and the average-policy accumulator uses own reach. Replacing one by another changes the algorithm, even when the numbers have a similar scale in a small game.

At a chance history, the action probability is part of π_χ and is never modified by regret matching. At a strategic history, the behavioral strategy is attached to the information set containing that history. Thus a strategy assigns the same action distribution to all histories in one information set, while beliefs may differ because prefix reach probabilities differ. Perfect recall is used in the theoretical statements to ensure that the player's own prefix information is compatible across histories in an information set.

2.2 COUNTERFACTUAL VALUE AND REGRET

For $I \in \mathcal{I}_i$ and $a \in A(I)$, the action-level counterfactual value is

$$v_i^\sigma(I, a) = \sum_{h \in I} \pi_{-i,c}^\sigma(h) \sum_{z \in Z(h,a)} \pi_{ha}^\sigma(z) u_i(z),$$

where the continuation reach $\pi_{ha}^\sigma(z)$ excludes the prefix reach and the forced action at I. The value of the strategy at the information set is $v_i^\sigma(I) = \sum_a \sigma_i(I, a) v_i^\sigma(I, a)$, and the instantaneous counterfactual regret is $r_i^\sigma(I, a) = v_i^\sigma(I, a) - v_i^\sigma(I)$. The solver's cumulative training regret is $R_i^T(I, a) = \sum_{t \le T} r_i^{\sigma^t}(I, a)$.

Regret matching forms a base distribution

$$\beta_i^{t+1}(I, a) = \begin{cases} [R_i^t(I, a)]_+ / \sum_b [R_i^t(I, b)]_+, & \text{if the denominator is positive,} \\ 1/|A(I)|, & \text{otherwise,} \end{cases}$$

where $[x]_+ = \max(x, 0)$. The base distribution and the executed strategy are distinct. The evaluator uses the action-level values of the evaluated profile, not the cumulative training regret. For profile ρ, define the normalized information-set regret

$$\mathrm{ISRegret}_i(I; \rho) = \max_a \widetilde{v}_i^{\rho}(I, a) - \sum_b \rho_i(I, b)\widetilde{v}_i^{\rho}(I, b), \quad \widetilde{v}_i^{\rho}(I, a) = \frac{v_i^{\rho}(I, a)}{\pi_{-i,c}^{\rho}(I)},$$

when the denominator is positive, and zero otherwise. We report the maximum and the information-set average of this quantity. For the two-player constant-sum games studied here, exploitability summarizes the worst case over the opponent's strategies and the minimum payoff performance they can expose: it is the average gain that a player's best response can obtain against the evaluated strategy, equivalently the loss exposed by the most favorable deviation. Lower values mean that the worst-case outcome is closer to equilibrium and the strategy is harder to exploit. All paper plots and tables use this normalized exploitability metric.

2.3 Equilibrium concepts and perturbation limits

For a profile σ, NE requires that no player can improve by replacing their complete behavioral strategy while the other player's strategy is held fixed. This global condition can leave arbitrary prescriptions at an unreachable information set. An SE assessment adds beliefs μ and requires sequential rationality at every information set together with consistency: the assessment is the limit of completely mixed assessments whose beliefs are generated by Bayes' rule. EFPE uses a related perturbed-game construction. For lower bounds $\varepsilon_{I,a} > 0$, let

$$\Pi_i^{\varepsilon} = \{\sigma_i : \sigma_i(I, a) \geq \varepsilon_{I,a} \text{ for every } I, a\}.$$

An equilibrium in Π^{ε} solves the perturbed game, and EFPE is defined through a suitable sequence in which the bounds vanish. A strategy with one finite positive lower bound should therefore be called a perturbed-game strategy, not an EFPE without qualification.

ALP-CFR uses local bounds of the form $\delta(I)/|A(I)|$, which can vary with both the information set and the training history. The schedule is motivated by the reach requirements of refinement learning, but local vanishing alone does not supply the global belief-consistency argument in the definition of EFPE. We consequently use "EFPE-oriented" for the mechanism and reserve "EFPE" for a solution concept or an explicitly identified baseline.

2.4 Output policies and equilibrium refinements

The reach-weighted average strategy is

$$\bar{\sigma}_i^T(I, a) = \frac{\sum_{t \leq T} \pi_i^{\sigma^t}(I)\sigma_i^t(I, a)}{\sum_{t \leq T} \pi_i^{\sigma^t}(I)},$$

with the uniform distribution used when the denominator is zero. The last iterate σ^T is a different object. The main full-tree ALP-CFR and MC-ALP-CFR curves use the average policy; EGT and RTCFR+ retain their last-iterate convention, which is stated in the appendix manifest.

For fixed action-wise lower bounds $\varepsilon_{I,a} > 0$, the feasible strategy set is a perturbed game. EFPE concerns a suitable limit of equilibria of such perturbed games as the perturbation vanishes. An ALP-CFR run with a finite, moving local schedule is therefore not itself an EFPE and maximum information-set regret is not a sufficient EFPE certificate. These distinctions determine both the claims and the interpretation of the results. The moving-schedule discussion is written for the ordinary regret-matching reference update; the reported protected vertex update is an explicit implementation path whose convergence requires the additional conditional assumptions stated in Appendix A.

3 Related Work

3.1 CFR and its variants

CFR minimizes counterfactual regret at every information set and gives an average-strategy convergence guarantee in two-player zero-sum games (Zinkevich et al., 2007a;b). CFR+ truncates negative cumulative regret and often improves practical convergence (Bowling et al., 2015). Discounted

CFR applies separate discounting schedules to historical positive and negative regrets (Brown & Sandholm, 2019). Monte Carlo CFR replaces complete traversal by sampled counterfactual estimators, with external sampling enumerating the traverser's actions and sampling opponent and chance actions (Lanctot et al., 2009). These methods primarily target Nash equilibrium; they do not by themselves prescribe how to allocate training effort among low-reach information sets.

The implementation comparison also has a semantic dimension. A full-tree iteration can contain two player updates or a single joint traversal, and a sampling iteration can visit a variable number of histories. These choices affect both solver state and apparent progress at a fixed iteration label. We preserve each verified baseline's update semantics and report logical rounds, player updates, and visited nodes separately. The main figure is not a wall-clock ranking and does not silently equate a joint traversal with two alternating traversals.

3.2 Equilibrium refinements

SE combines sequential rationality with consistency of beliefs at every information set (Kreps & Wilson, 1982; Miltersen & Sørensen, 2006; Gilpin & Sandholm, 2006; Turocy, 2010). EFPE models completely mixed perturbations and studies the limiting equilibrium (Selten, 1975). Sequence-form and mathematical-programming approaches provide strong refinement formulations but can be difficult to scale to large trees (Hansen et al., 2010; Farina & Gatti, 2017; Farina et al., 2018). Behaviorally constrained CFR incorporates perturbations into regret minimization (Farina et al., 2017); its fixed perturbation has a direct perturbed-game interpretation but does not by itself produce the vanishing-perturbation limit.

3.3 Iterative regularization and last-iterate methods

Smoothing, excessive-gap, and online mirror-descent methods provide other first-order routes to refinement-oriented optimization (Hoda et al., 2010; Kroer et al., 2018; 2020; Bernasconi et al., 2024; Rakhlin & Sridharan, 2013). RTCFR+ studies efficient last-iterate convergence in extensive-form games (Meng et al., 2025), while regularized CFR variants study the effect of regularization on solving games (Liu et al., 2023). These methods optimize different objectives and use different output conventions. We consequently compare the verified implementations through explicit metrics rather than calling every regularized method an EFPE solver.

3.4 Position of ALP-CFR

ALP-CFR is complementary to these approaches. Its contribution is a local, reach-adaptive perturbation and a protected-set mechanism that carries a downstream low-reach signal to upstream information sets. The mechanism is compatible with complete and sampled traversals, but the sampling proof requires the actual estimator and sampling distribution to match. The paper therefore reports the full-tree method and the separately validated external-sampling implementation as distinct algorithms, and does not reuse historical SPCFR, LP-CFR compatibility, or unified diagnostic runs as if they were independent published baselines.

4 Method

4.1 Local perturbation and protected reach

At information set I, regret matching first produces $\beta_i^t(I,\cdot)$. ALP-CFR executes the feasible strategy

$$\sigma_i^t(I,a) = (1-\delta_t(I))\beta_i^t(I,a) + \frac{\delta_t(I)}{|A(I)|}. \tag{1}$$

Thus each legal action has probability at least $\delta_t(I)/|A(I)|$. Let $Q_t(I)$ be cumulative effective reach. The base local schedule is

$$\delta_t^{\text{base}}(I) = \frac{1}{a + b\sqrt{Q_t(I)}}, \qquad a, b > 0. \tag{2}$$

The square-root dependence is a schedule choice, not a claim that every vanishing perturbation is theoretically equivalent.

For a history h, the raw cumulative reach is accumulated from the reach contribution at each update boundary. The corresponding information-set quantity is a sum over histories, not an average over the number of nodes:

$$Q_t(I) = \sum_{h \in I} Q_t(h), \qquad q_t(h) = \pi_i^{\sigma^t}(h)\pi_{-i,c}^{\sigma^t}(h).$$

The exact factorization is evaluated by the solver's traversal state. This definition makes explicit why protection is stateful: two information sets with the same logical iteration count can have very different cumulative reach and hence different local perturbations.

The protected state is represented by node-level retained sets but is triggered at the information-set level used by the solver. Let $\mathrm{Pred}(I)$ be the predecessor information sets of I. A request is started when, for some $J \in \mathrm{Pred}(I)$,

$$Q_t(I) < \sqrt{Q_t(J)}.$$

The request retains the decision histories in I and is propagated through every ancestor information set. At an ancestor K, the retained requests are reduced to minimal downstream sets $\mathcal{P}_t(K)$; incomparable branches remain, while a request covered by a deeper request is removed. The implementation computes

$$B_t(K) = \min\left(Q_t(K), \min_{S \in \mathcal{P}_t(K)} \sum_{h \in S} Q_t(h)\right), \qquad L_t = \min_J Q_t(J),$$

where the inner minimum is omitted when $\mathcal{P}_t(K)$ is empty, and uses

$$\widetilde{Q}_t(K) = \max\{B_t(K), L_t\}.$$

Thus the protected quantity is a retained-set minimum with a global information-set reach floor, rather than a sum over all frontier nodes. The executed perturbation replaces Q_t by $\widetilde{Q}_t$ in Eq. equation 2. Obsolete requests are rechecked and retracted without removing protection generated by another descendant. Full pseudocode and frontier invariants are given in Appendix A.

The protected solver uses vertex regret. If $v_i^t(I, a)$ denotes the counterfactual action value, its regret increment is

$$\Delta R_i^t(I, a) = (1 - \delta_t(I))\left(v_i^t(I, a) - \sum_b \beta_i^t(I, b) v_i^t(I, b)\right).$$

The ordinary σ-weighted increment is retained only for the non-vertex reference path; the ALP-CFR and MC-ALP-CFR results reported here use the vertex form.

4.2 One logical round

The canonical full-tree implementation is alternating. A logical round contains two half-rounds, first player 0 and then player 1. The protected state and local perturbations are constructed once at the beginning of the logical round and are frozen for both full-tree traversals. Each half-round then enumerates all legal strategic actions and chance outcomes, computes counterfactual values, updates the selected player's regret and reach accumulators, and updates the reach-weighted average-policy accumulator. The second half-round sees the first player's newly updated policy, but not a new protected schedule. MC-ALP-CFR is separate: its Python implementation recomputes and freezes its sampling schedule at each half-round.

The central loop is summarized below. The subroutines refer to the exact recursions in Appendix A and make explicit which state is shared across the two player updates.

ALP-CFR / MC-ALP-CFR one round

Initialize R, average-policy numerators and denominators, node and information-set Q, and the active protection requests. For full-tree ALP-CFR, build the request closure and freeze $\widetilde{Q}, \delta, \sigma$ once per logical round. For MC-ALP-CFR, repeat that construction at each half-round. For each player i in the fixed update order: (1) call FullTreeUpdate or ExternalSampleUpdate; (2) update R_i, the average strategy, and the sampled or full-tree node counters; (3) recompute cumulative reach; (4) generate, merge, propagate, and

retract protection requests; and (5) recompute regret matching for the next half-round. At a full-tree node, enumerate every action. At an external-sampling node, enumerate the current player's actions and sample opponent and chance actions from the current strategy or chance distribution. Return the reach-weighted average strategy and, when requested, the last iterate.

The procedure preserves the original solver semantics: regret post-processing (including CFR+ or DCFR when used) is applied at its original update boundary, player order is not collapsed, and evaluation never mutates training state or consumes the training random stream.

The update order also fixes the meaning of the average strategy. For the first half-round, both players' current policies are those available at the start of the round. After player 0's regret update, player 1's half-round uses the new player-0 state and player 1's previous state. The average-policy accumulator is updated for the player being traversed, not retroactively for a profile that was never used. This convention is important when comparing a protected alternating run with a historical joint traversal: changing the order changes the sequence of strategies even if the final checkpoint and named iteration count are identical.

4.3 External sampling and work accounting

MC-ALP-CFR uses standard external sampling. At a traverser information set, all current-player actions are enumerated; at opponent and chance nodes, one action is sampled. No outcome-sampling $1/p$ multiplier is inserted into this estimator. The sampling distribution, reach accumulation, and value recursion are matched as described in Appendix A. Full-tree evaluation is performed only at checkpoints.

For full-tree methods, the x-axis is logical iteration and the recorded work includes player updates, traversals, visited nodes, and terminal nodes. For MC-ALP-CFR, the paper also records the raw sampled rounds and defines full-tree-equivalent rounds as

$$\text{equivalent rounds} = \frac{\text{cumulative training-node visits}}{\text{complete game-tree node count}}.$$

Evaluation visits are excluded. This normalization is reported as a separate cost convention; it is not used to claim wall-clock superiority.

The node counter is separated by purpose. Training visits contribute to the sampling budget; evaluator visits are written to the checkpoint report but do not increase the training budget. A full-tree evaluator may inspect all nodes at a checkpoint for both a sampled and a full-tree solver, but that does not turn the sampled update into a full-tree update. This separation is used in the appendix tables and manifest so that a large evaluator run cannot make a sampling curve appear to have received more training.

4.4 Scope of the theory

For a fixed positive local perturbation, ordinary regret matching has the corresponding perturbed-game interpretation. For a moving schedule, the potential argument contains ordinary regret, perturbation, reach-variation, and feasible-set drift terms. Under bounded values, perfect recall, positive chance or sampling probabilities, controlled drift, and divergence of the relevant cumulative reach, the normalized information-set regret is sublinear. For external sampling, conditional unbiasedness of the matched recursion gives a martingale error term, with concentration depending on the conditional variance. These are conditional statements about regret and reach; they do not establish general SE or EFPE convergence. The complete lemmas and proofs are in Appendix B.

5 Experiments and Analysis

5.1 Games, protocols, and baselines

We use the fixed two-player zero-sum, imperfect-information configurations Kuhn, GoofSpiel-3, Leduc-3, and Leduc-5. Kuhn uses three private cards and the standard pass/bet/fold actions. Leduc uses the project's fixed deck and betting-round parameters. GoofSpiel uses simultaneous bidding

with private hands and public prize cards. All current runners load OpenSpiel's official turn-based conversion, so the reported tree sizes and full-tree results use the same sequentialized representation; the conversion preserves the simultaneous-move information structure rather than substituting a different game. The exact game parameters, node counts, payoff convention, and action rules are in Appendix C.

Before long runs, each solver passed the project checks for environment registration, normalized nonnegative policies, finite regret/Q/delta values, checkpoint save-load continuation, and evaluator non-interference. A failed check is not repaired by changing a curve after the fact. The run manifest stores the resolved perturbation parameters, update order, traversal mode, output policy, seed, code identity when recorded, and the final checkpoint status. These checks are especially important for GoofSpiel, where loading a native simultaneous node as an ordinary sequential decision would define a different information structure.

For each checkpoint we record the actual iteration count, training-node visits, complete traversals or sampled trajectories, exploitability, maximum information-set regret, and average information-set regret. The evaluator computes the last two quantities from the evaluated strategy's counterfactual values. It does not substitute cumulative solver regret, Q, or the regret of a different output policy. The same independent full-tree evaluator is used whenever the corresponding result is available.

All full-tree runs use the recorded project checkpoint schedule 1k, 2k, 5k, 10k, 20k, 50k, 100k, 200k, 500k, and 1M logical rounds. The sampling budget is expressed in equivalent rounds. The explicit main-figure whitelist is CFR, CFR+, DCFR, EGT, RTCFR+, EFPE-REG, ALP-CFR, and MC-ALP-CFR. ALP-CFR is only the verified alternating protected full-tree binding; MC-ALP-CFR is only the external-sampling implementation. Historical SPCFR, LP-CFR, compatibility, Unified-Raw, Unified-Vertex, joint diagnostic, and protection debugging runs are excluded from the input data, not merely renamed in the legend. Each plotted curve is listed in `diagnosis/main_figure_manifest.csv`.

5.2 Main comparison

The final recorded values show that the global and local metrics can disagree. For Kuhn at 1M, ALP-CFR has exploitability 5.13×10^{-6}, maximum local regret 2.28×10^{-5}, and average local regret 6.79×10^{-6}; MC-ALP-CFR has exploitability 4.56×10^{-4} and maximum local regret 2.21×10^{-3} at the same equivalent budget. The sampling result is therefore qualitatively consistent with a finite-variance penalty, not evidence of equality with full-tree training.

GoofSpiel-3 is more revealing for local behavior. ALP-CFR reaches exploitability 3.02×10^{-5}, maximum local regret 1.36×10^{-1}, and average local regret 1.56×10^{-2}. The conventional full-tree baselines reach much smaller exploitability, while their maximum local regrets stay near one in the recorded evaluator. This is precisely why a global metric alone does not characterize the refinement-oriented behavior. EGT has a small maximum local regret in this game but a larger exploitability, illustrating that the two axes should not be collapsed into one ranking.

Leduc-3 exposes both the game-size and completeness limitations. DCFR and ALP-CFR have completed 1M full-tree runs; their exploitability values are 2.83×10^{-7} and 1.35×10^{-4}, while their maximum local regrets are 3.94 and 2.00, respectively. CFR, CFR+, and LP-CFR-compatible historical groups are partial and are not included in the main whitelist. Leduc-5 has completed conventional full-tree baselines, but the ALP-CFR and MC-ALP-CFR rows are unavailable in the current inventory. The figure and analysis therefore do not claim a sampling or protected-method result for Leduc-5.

These results do not show that ALP-CFR is uniformly best. They show that the protection mechanism changes the distribution of local training signal and that local-regret behavior and exploitability may move in different directions. The missing and partial rows, mixed historical batches, and different output policies of some formal baselines limit causal comparisons. We retain those limitations instead of treating a short or absent run as a curve.

The metric pattern also clarifies what can and cannot be inferred from the figure. A lower exploitability indicates a smaller best-response deviation, but it does not say how the strategy behaves at an information set that is rarely reached by the evaluated profile. Conversely, a lower maximum local regret can coexist with a comparatively large exploitability when the information sets with the

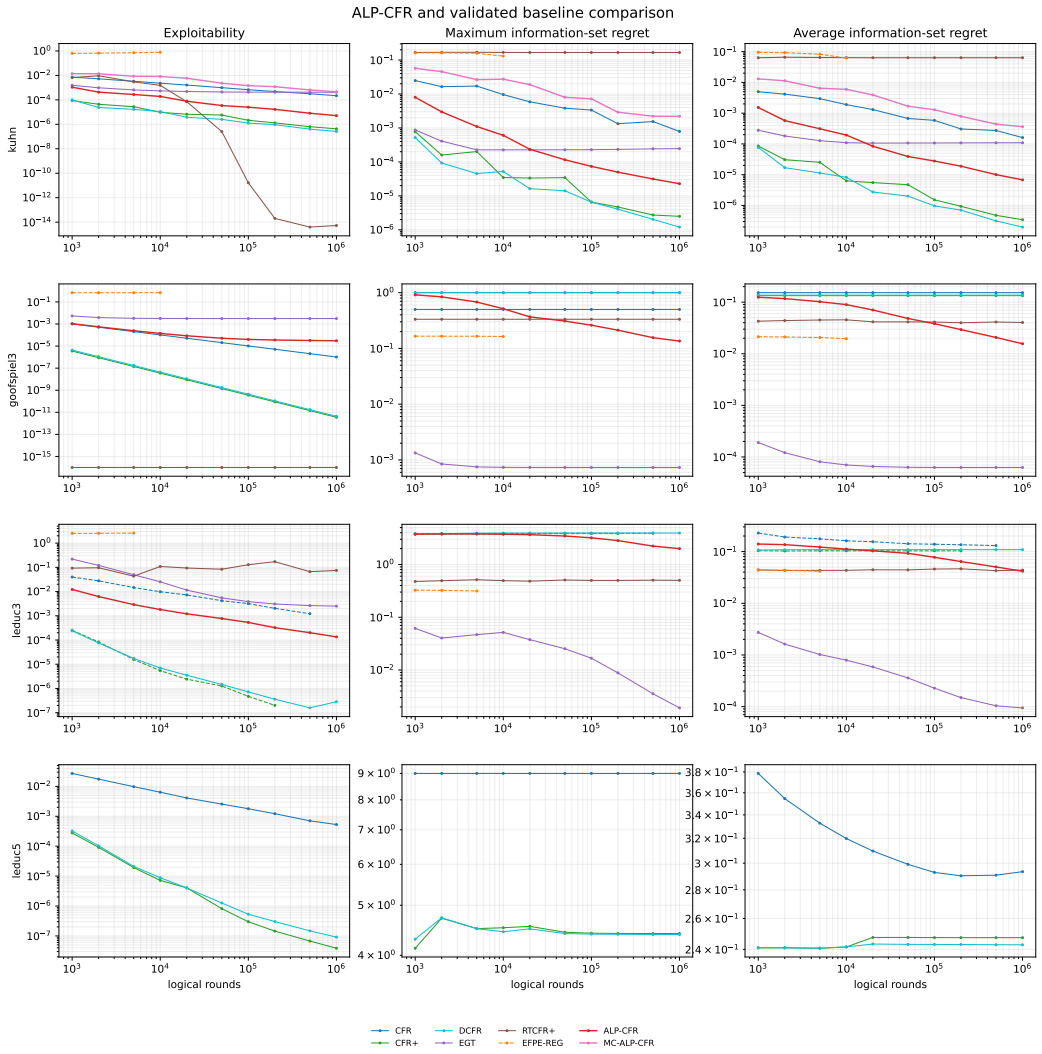


Figure 1: Controlled comparison on the four paper games. Rows are Kuhn, GoofSpiel-3, Leduc-3, and Leduc-5; columns are exploitability, maximum information-set regret, and average information-set regret. ALP-CFR denotes the protected alternating full-tree implementation and MC-ALP-CFR denotes external sampling. Full-tree curves use logical rounds, while MC uses the explicit full-tree-equivalent convention in Section 4.3. Dashed curves are partial; unavailable experiments are omitted and listed in Appendix C. MC shaded bands, when present, are the mean and standard deviation over seeds 0–2.

largest local errors carry little global reach. Average local regret adds a third perspective: it distinguishes a method that improves a small number of difficult information sets from one that changes the typical information set. We therefore discuss the three columns jointly and do not select a single metric as the definition of refinement quality.

The baseline identities further constrain the interpretation. CFR, CFR+, and DCFR are regret-minimization baselines with average-policy output in the full-tree comparison. EGT and RTCFR+ are sequence-form or last-iterate baselines whose output policy is different, so a small local-regret value for one of them is not a direct proof that it solves the same objective as ALP-CFR. EFPE-REG is retained only where its recorded run is valid and is classified by its actual regularized objective rather than by the presence of the word "EFPE" in a source directory. MC-ALP-CFR is treated as a separate estimator: its mean curve and seed variation summarize stochastic training, while its full-tree evaluator does not remove the sampling variance incurred during training.

This distinction is also why the main comparison does not include the project's intermediate names. An internal compatibility wrapper can be useful for debugging equality of update rules, but putting it beside the final algorithm would count an implementation history as an independent scientific baseline. The manifest records those excluded sources for auditability, and the appendix reports missing or partial experiments rather than filling them with a historical curve from another commit or another traversal convention.

5.3 Reading the controlled curves

The detailed comparison protocol and its treatment of partial results are given in Appendix C.

The Kuhn row is the cleanest completed comparison because the full-tree and MC runs both reach their stated budget. CFR+ and DCFR obtain the smallest global deviations among the regret baselines, while ALP-CFR remains within the same order of magnitude but reports a different local-regret profile. EGT's last iterate has a larger exploitability than ALP-CFR while its maximum local regret is smaller. This is not contradictory: the last-iterate policy and the reach-weighted average policy are different evaluated objects, and the two metrics weight deviations differently. The MC mean is worse than the full-tree ALP-CFR curve at 1M equivalent rounds, but the comparison is an estimator comparison rather than a proof that sampling cannot match the full-tree limit.

In GoofSpiel-3, the complete-tree size makes the distinction between logical rounds and work especially visible. The baseline exploitability curves decrease rapidly, yet several baseline maximum information-set regrets remain close to one. ALP-CFR lowers its maximum and average local regrets together, although its exploitability is not the best in the panel. EGT gives a useful counterexample to a single-metric interpretation: its local values are small, but its global deviation is larger. The figure therefore supports the narrower claim that the protected update changes local training behavior, not the stronger claim that it dominates every equilibrium metric.

Leduc-3 and Leduc-5 primarily test scale. Leduc-3 has billions of recorded full-tree node visits at 1M rounds, and only DCFR and the controlled ALP-CFR binding are complete among the C++ full-tree protected-method inputs. The treeplex baselines are reported separately because their update work is not a full-tree node traversal. In Leduc-5, the conventional baselines complete their runs, but protected and sampled outputs are absent. The Leduc-5 row is consequently a baseline reference and not evidence for or against the ALP-CFR mechanism. This asymmetric coverage is why the conclusion is phrased in terms of observed metric patterns and implementation scope rather than a universal algorithm ranking.

Across all rows, the plotted x-coordinate is a training-budget coordinate, not elapsed time. The historical source batches also have different code commits, and the manifest makes that visible. The main figure can therefore serve as a descriptive synthesis of the validated outputs, but a causal statement about one update rule versus another requires the same-commit, parameter-matched controlled matrix. That matrix, its protection diagnostics, and its excluded intermediate groups remain available as project artifacts; they are not silently folded into the paper's final method name.

6 Conclusion

ALP-CFR combines local perturbation with upstream protection of under-reached information sets. The method has a clear full-tree update semantics and a separate external-sampling implementation, but its current theory is conditional and does not prove general EFPE or last-iterate convergence. The controlled results confirm that exploitability and information-set regret measure different aspects of the output strategy. Future work should complete matched large-game sampling runs, establish stronger moving-set conditions, and evaluate refinement quality with independent equilibrium and belief diagnostics.

REFERENCES

REFERENCES

Martino Bernasconi, Alberto Marchesi, and Francesco Trovò. Learning extensive-form perfect equilibria in two-player zero-sum sequential games. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 2152–2160, 2024.

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. Solving imperfect-information games via discounted regret minimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1829–1836, Jul. 2019. doi: 10.1609/aaai.v33i01.33011829.

Gabriele Farina and Nicola Gatti. Extensive-form perfect equilibrium computation in two-player games. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pp. 502–508. AAAI Press, 2017.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Regret minimization in behaviorally-constrained zero-sum games. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 1107–1116, 2017.

Gabriele Farina, Nicola Gatti, and Tuomas Sandholm. Practical exact algorithm for trembling-hand equilibrium refinements in games. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5044–5054, 2018.

Andrew Gilpin and Tuomas Sandholm. Finding equilibria in large sequential games of imperfect information. In *Proceedings of the 7th ACM conference on Electronic commerce*, pp. 160–169, 2006.

Kristoffer Arnsfelt Hansen, Peter Bro Miltersen, and Troels Bjerre Sørensen. The computational complexity of trembling hand perfection and other equilibrium refinements. In *Proceedings of the 3rd International Symposium on Algorithmic Game Theory (SAGT)*, pp. 198–209. Springer, 2010.

Samid Hoda, Andrew Gilpin, Javier Pena, and Tuomas Sandholm. Smoothing techniques for computing nash equilibria of sequential games. *Mathematics of Operations Research*, 35(2):494–512, 2010.

David M Kreps and Robert Wilson. Sequential equilibria. *Econometrica: Journal of the Econometric Society*, pp. 863–894, 1982.

Christian Kroer, Gabriele Farina, and Tuomas Sandholm. Solving large sequential games with the excessive gap technique. *Advances in neural information processing systems*, 31, 2018.

Christian Kroer, Kevin Waugh, Fatma Kılınç-Karzan, and Tuomas Sandholm. Faster algorithms for extensive-form game solving via improved smoothing functions. *Mathematical Programming*, 179(1):385–417, 2020.

Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte carlo sampling for regret minimization in extensive games. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

Mingyang Liu, Asuman E. Ozdaglar, Tiancheng Yu, and Kaiqing Zhang. The power of regularization in solving extensive-form games. In *International Conference on Learning Representations*, 2023.

Linjian Meng, Tianpei Yang, Youzhi Zhang, Zhenxing Ge, Shangdong Yang, Tianyu Ding, Wenbin Li, Bo An, and Yang Gao. Efficient last-iterate convergence in solving extensive-form games. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Peter Bro Miltersen and Troels Bjerre Sørensen. Computing sequential equilibria for two-player games. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2006)*, pp. 107–116, USA, 2006. ACM Press.

John Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems 26 (NeurIPS 2013)*, pp. 3066–3074, 2013.

Reinhard Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory*, 4(1):25–55, 1975.

Theodore L Turocy. Computing sequential equilibria using agent quantal response equilibria. *Economic Theory*, 42:255–269, 2010.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pp. 1729–1736, Red Hook, NY, USA, 2007a. Curran Associates Inc. ISBN 9781605603520.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007b.

A Detailed Algorithms and Protection Mechanism

This appendix fixes the implementation semantics used by the main text. The solver state at an information set contains cumulative regret $R_i(I, a)$, the reach-weighted average-policy numerator and denominator, and cumulative effective reach. At each history the implementation also stores node-level reach and the active protection requests. All quantities are initialized to zero, except the uniform behavioral strategy used before the first regret update. The random seed, game parameters, and resolved solver options are stored in the per-run configuration file.

A.1 Update boundary and full-tree recursion

A project round has two ordered half-rounds, $s = (t, 0)$ and $s = (t, 1)$. For full-tree ALP-CFR, requests are closed under ancestor propagation once at the beginning of the logical round, reduced to their retained sets, and used to compute $\widetilde{Q}_s$, δ_s, and the executed policy for both player traversals. These objects are frozen for the round. MC-ALP-CFR recomputes and freezes its sampling schedule separately at each half-round. At a terminal history the recursion returns the utility vector. At chance, it enumerates all actions with their chance probabilities. At a strategic history it enumerates all actions, computes child values, and combines them with the current behavioral strategy. At an information set belonging to the updating player, the opponent-and-chance reach weights the action values and the regret increment. The selected player's average-policy numerator and denominator are updated once per information set using own reach.

The resulting update is schematically

$$
\begin{aligned}
v_i(I, a) &= \sum_{h \in I} \pi_{-i,c}(h) V_i(h, a), \\
r_i^{\mathrm{ord}}(I, a) &= v_i(I, a) - \sum_b \sigma_i(I, b) v_i(I, b), \\
r_i^{\mathrm{vert}}(I, a) &= (1 - \delta(I)) \left(v_i(I, a) - \sum_b \beta_i(I, b) v_i(I, b) \right), \\
R_i(I, a) &\leftarrow \mathrm{PostProcess}(R_i(I, a) + r_i(I, a)), \\
(S_i, W_i) &\leftarrow (S_i + \pi_i(I)\sigma_i(I, \cdot), W_i + \pi_i(I)).
\end{aligned}
$$

Here $r_i = r_i^{\mathrm{ord}}$ for the ordinary reference path and $r_i = r_i^{\mathrm{vert}}$ for the protected ALP-CFR and MC-ALP-CFR paths. The latter is the update used by the reported protected results. The post-processing is the identity for ordinary CFR, positive truncation for CFR+, and the configured discounting transformation for DCFR. The historical joint binding is not the paper method.

A.2 External-sampling recursion

For a half-round updating player i, the external sampler enumerates all actions at player i's information sets and samples one action at an opponent information set. Chance actions are sampled from χ. The sampled recursion returns the selected child value at sampled nodes and the policy-weighted sum at traverser nodes. It does not insert an outcome- sampling importance ratio. The probability of the sampled branch is instead accounted for by conditional expectation. A node visit adds the traverser's own reach factor to the sampled reach counter. Consequently,

$$
\mathbb{E}[\mathbf{1}\{h \text{ visited}\}\pi_i(h) \mid \mathcal{F}_{s-1}] = \pi_{-i,c}(h)\pi_i(h),
$$

provided the sampling distributions match the policies used by the recursion. All current GoofSpiel-3 runners load OpenSpiel's turn-based representation of the simultaneous bidding game; the reported tree counts therefore refer to that common transformed representation. The conversion does not reinterpret the bidding node as a different imperfect-information game.

A.3 Protection requests and frontiers

After a full-tree schedule boundary, the solver checks the information-set trigger $Q_s(I) < \sqrt{Q_s(J)}$ for each predecessor information set J. The request retains the decision histories in I and is propagated to every ancestor information set, including intermediate sets and the root. At an information

set K, retained request sets are partially ordered by the ancestor relation. The deepest retained set on each branch is kept, along with incomparable branches. Define

$$B_s(K) = \min\left(Q_s(K), \min_{S\in\mathcal{P}_s(K)} \sum_{h\in S} Q_s(h)\right), \qquad L_s = \min_J Q_s(J),$$

with the inner minimum omitted when no request reaches K, and

$$\widetilde{Q}_s(K) = \max\{B_s(K), L_s\}.$$

For MC-ALP-CFR, the Python implementation uses the same information-set trigger together with the additional guard $Q_s(I) < Q_s(J)$. The two implementations are therefore reported as distinct schedule variants rather than being treated as round-by-round identical. The retained sets prevent double counting of covered paths. Obsolete requests are rechecked at every boundary. Retraction is local first and then propagates upward only when no other descendant request remains. Reach accumulated inside the current traversal is not retroactively applied to the current perturbation.

A.4 Reproducibility and policy output

Checkpoint state includes regrets, average-policy accumulators, node and information-set reach, request/frontier state, the iteration counters, and the random-number-generator state for sampling. A checkpoint is resumable only when the game, solver identity, behavior-affecting configuration, seed, and code metadata agree. Evaluations use a separate deterministic evaluator and do not change the training state. The default displayed policy is the reach-weighted average policy for CFR-family methods and MC-ALP-CFR; EGT and RTCFR+ use their recorded last-iterate policy.

B Theoretical Analysis and Proofs

B.1 Feasibility and fixed perturbations

The policy map in the main text immediately gives

$$\sigma_i^t(I,a) \geq \delta_t(I)/|A(I)|.$$

For a fixed positive $\delta(I)$, all iterates lie in a fixed product of simplexes with action-wise lower bounds. The usual counterfactual-regret potential argument therefore applies to this perturbed strategy space. It establishes an approximate equilibrium of the fixed perturbed game when normalized cumulative regret vanishes. It does not establish an equilibrium of the unperturbed game or EFPE.

B.2 Moving local schedules

Let $M_T(I)$ be the number of half-rounds that update I. With bounded counterfactual values, the positive-regret potential satisfies a one-step inequality of the form

$$F_{s+1} - F_s \leq C_U^2/2 + \langle R_s^+, p_s\rangle + \langle R_s^+, d_s\rangle + \langle R_s^+, w_s\rangle,$$

where p_s is the local perturbation contribution, d_s is feasible-set drift, and w_s is reach variation. If

$$\sum_{s\leq T} q_s(I)\delta_s(I) = o(M_T(I)), \qquad \sum_{s\leq T}(\|d_s(I)\|_1 + \|w_s(I)\|_1) = o(M_T(I)),$$

then applying Young's inequality and taking square roots gives

$$R_T(I) \leq C_1\sqrt{M_T(I)} + C_2\sum_{s\leq T} q_s(I)\delta_s(I) + C_3\sum_{s\leq T}(\|d_s(I)\|_1 + \|w_s(I)\|_1).$$

Dividing by $M_T(I)$ yields vanishing normalized regret under the displayed conditions. The moving-set and reach assumptions are essential; ordinary CFR analysis alone does not imply them. The displayed recurrence is written for the ordinary-RM reference update. The reported protected vertex update uses the explicit beta-centered map above; extending this bound to that map would require an additional lemma, so the ordinary bound is not claimed automatically for the vertex implementation.

Table 1: Game scale and information-set statistics.

game	histories_total	terminal_histories	information_sets	maximum_depth	maximum_branching_factor	chance_branching_factor
kuhn	58	30	12	5	3	3
goofspiel3	418	216	114	6	3	3
leduc3	1939	1116	288	11	3	3
leduc5	55361	32760	2760	11	10	10

Table 2: Algorithm identities and configuration conventions.

algorithm	implementation	update	policy	mode	protection	regret	boundary	parameters	seeds
CFR/CFR+/DCFR	custom OpenSpiel C++	joint or alternating	average	full-tree	no	ordinary/method-specific	logical round or method-specific	not applicable	0
LP-CFR	original C++ local perturbation	alternating	average	full-tree	no	ordinary	half-round	epsilon resolved per config	0
ALP-CFR	protected alternating binding	alternating	average	full-tree	B/L floor + trigger	vertex	logical round; freeze once	a=10000,b=1000	0
EGT/RTCFR+/EFPE_REG	treeplex Python	method-specific	last	full-tree treeplex	method-specific	method-specific	method-specific	method-specific	0
MC-ALP-CFR	Python external sampling	two half-rounds	average	sampling	B/L floor + guard	vertex	half-round; freeze each half	epsilon,a,b resolved	0,1,2

B.3 Reach and protection

The protection rule is intended to make the information-set condition $Q_T(I) \to \infty$ hold whenever a relevant ancestor is reachable and the protected-support condition is satisfied. Under positive chance probabilities, finite depth, bounded frontiers, and this support condition, the local schedule tends to zero while retaining positive action probabilities at every finite time. This gives a conditional reach-normalized regret statement, not a universal guarantee: the proof does not replace the required support condition with a theorem about arbitrary games or arbitrary initializations.

B.4 External-sampling estimator

Let $\widehat{v}_i^s(I,a)$ be the value returned by the matched external- sampling recursion. Conditional on $\mathcal{F}_{s-1}$, expansion over the sampled opponent and chance actions gives

$$\mathbb{E}[\widehat{v}_i^s(I,a) \mid \mathcal{F}_{s-1}] = \sum_z p_s(z) g_i(z;I,a) = v_i^s(I,a).$$

Therefore sampled regret is an unbiased increment and its error is a martingale difference. If the sampled values have conditional variance $V_s(I,a)$ and bounded increments $B(I,a)$, Freedman-type concentration gives, with probability at least $1-\alpha$,

$$\widehat{R}_T(I,a) - R_T(I,a) \le \sqrt{2\left(\sum_{s\le T} V_s(I,a)\right)\log(2|A(I)|/\alpha)} + \frac{2B(I,a)}{3}\log(2|A(I)|/\alpha).$$

The bound is conditional on positive sampling support and the exact recursion. Vanishing local perturbations can reduce the minimum sampling probability, so the variance term cannot be treated as a uniform constant without an additional lower-bound or adaptive-clipping assumption. No claim of general MC-ALP-CFR convergence to EFPE follows from this estimator result.

B.5 Boundary of the claims

The proofs assume finite perfect-recall games, bounded utilities, the stated update order, matched evaluator and training definitions, and the support and drift conditions above. They do not prove general EFPE convergence, do not make maximum information-set regret sufficient for EFPE, and do not transfer automatically to the historical compatibility or simultaneous-game bindings.

C Experimental Details and Additional Results

C.1 Game scale and parameter record

The exact game scale and the algorithm parameter export are given in Tables 1 and 2. Displayed decimal values use six significant digits; raw CSV files and checkpoints retain their original precision. The main-figure audit file records the display name, actual solver/config identity, result file, output policy, game, training budget, Git metadata when available, and completeness flag.

Table 3: Final recorded metrics used for the paper comparison. Values are shown with six significant digits; partial and unavailable runs are retained as status information rather than being imputed.

Game	Method	Status	Exploitability	Max IS regret	Avg IS regret
Kuhn	CFR	complete	0.000218950	0.000788540	0.000160622
Kuhn	CFR+	complete	4.31127e-7	2.50055e-6	3.44170e-7
Kuhn	DCFR	complete	2.61816e-7	1.20313e-6	1.97868e-7
Kuhn	ALP-CFR	complete	5.13397e-6	2.27832e-5	6.79446e-6
Kuhn	EGT	complete	0.000421330	0.000245339	0.000109275
Kuhn	RTCFR+	complete	5.32907e-15	0.166667	0.0634495
Kuhn	MC-ALP-CFR	complete	0.000456180	0.00220556	0.000362621
Kuhn	EFPE-REG	partial	0.782437	0.130495	0.0618043
GoofSpiel-3	CFR	complete	1.02564e-6	0.500000	0.152791
GoofSpiel-3	CFR+	complete	3.53810e-12	1.00000	0.133833
GoofSpiel-3	DCFR	complete	4.34615e-12	1.00000	0.136886
GoofSpiel-3	ALP-CFR	complete	3.01965e-5	0.135512	0.0156427
GoofSpiel-3	EGT	complete	0.00311668	0.000732586	6.25780e-5
GoofSpiel-3	RTCFR+	complete	0	0.333333	0.0404051
GoofSpiel-3	MC-ALP-CFR	unavailable	–	–	–
GoofSpiel-3	EFPE-REG	partial	0.691871	0.163737	0.0195775
Leduc-3	CFR	partial	0.00122023	3.90416	0.130626
Leduc-3	CFR+	partial	2.00451e-7	3.85404	0.103458
Leduc-3	DCFR	complete	2.83263e-7	3.94189	0.108920
Leduc-3	ALP-CFR	complete	0.000134844	1.99619	0.0416888
Leduc-3	EGT	complete	0.00249946	0.00191884	9.47439e-5
Leduc-3	RTCFR+	complete	0.0745020	0.500810	0.0438335
Leduc-3	MC-ALP-CFR	unavailable	–	–	–
Leduc-3	EFPE-REG	partial	2.60246	0.314814	0.0415700
Leduc-5	CFR	complete	0.000529058	9.00000	0.293506
Leduc-5	CFR+	complete	3.86557e-8	4.40783	0.247380
Leduc-5	DCFR	complete	9.12356e-8	4.38863	0.242858
Leduc-5	ALP-CFR	unavailable	–	–	–
Leduc-5	MC-ALP-CFR	unavailable	–	–	–

Table 4: Available MC-ALP-CFR sampling statistics; unexported quantities are not inferred.

game	status	sampled_node_visits	equivalent_rounds	mean_sampled_value	sample_variance	effective_sample_size	min_sampling_probability	max_weight
kuhn	complete	5.8e+07	1e+06	not_exported	not_exported	not_exported	not_exported	not_used_by_external_sampling
goofspiel3	unavailable							
leduc3	unavailable							
leduc5	unavailable							

C.2 Final metrics and completeness

Table 3 is the filtered final table used for this paper. It excludes the project's historical SPCFR/LP-CFR/Unified and compatibility paths. A status of partial means the endpoint is below the target budget; unavailable means no validated data file exists. The absence of a row in the main figure is therefore a data-availability decision, not an implicit zero or an interpolated point.

C.3 Sampling and low-reach diagnostics

For MC-ALP-CFR, training nodes and equivalent rounds are retained separately; evaluation nodes are excluded from the equivalent-round counter. The current complete 1M sampling result is Kuhn with seeds 0–2. GoofSpiel-3 has a turn-based conversion validation but no completed 1M MC curve in this paper version. Leduc-3 and Leduc-5 MC rows are likewise unavailable. Sampling statistics and exported fields are summarized in Table 4.

The controlled protected runs export information-set Q, effective Q, delta, request and frontier statistics at 10k, 100k, 500k, and 1M when the run exists. Fixed reach buckets and protection counts are summarized in Table 5. These diagnostics are not used to select a better curve for the main figure.

C.4 Controlled comparison protocol

The controlled interpretation follows a simple rule: only one factor is allowed to change inside a named comparison. For ALP-CFR, the protected alternating binding uses the same local perturbation formula, Q definition, player order, average-policy update, and regret post-processing as

Table 5: Controlled B low-reach Q-bucket diagnostics at 1M.

game	status	q_bucket	count	q_mean	mean_regret	maximum_regret	protected_fraction	mean_perturbation
kuhn	complete	q4	12	196796	6.79446e-06	2.27832e-05	not_exported	not_exported
goofspiel3	complete	q1	16	9.83421e-06	0.0169769	0.0757488	not_exported	not_exported
goofspiel3	complete	q2	24	0.0206307	0.0583858	0.135512	not_exported	not_exported
goofspiel3	complete	q3	16	0.115733	0.00571372	0.0395296	not_exported	not_exported
goofspiel3	complete	q4	58	68965.5	0.00032681	0.00536504	not_exported	not_exported
leduc3	complete	q1	46	0.129712	0.246785	1.99619	not_exported	not_exported
leduc3	complete	q2	30	3.68272	0.0204003	0.227079	not_exported	not_exported
leduc3	complete	q3	10	46.9936	0.000522336	0.00133414	not_exported	not_exported
leduc3	complete	q4	202	19702.5	0.000183385	0.00254378	not_exported	not_exported
leduc5	unavailable							

its unprotected or diagnostic counterparts; the paper figure includes only the canonical protected binding. For sampling, the solver keeps the raw sample round count, cumulative training nodes, and equivalent rounds in separate fields. This avoids presenting a sampled checkpoint as if it had performed a full-tree traversal merely because the evaluator later visited the entire tree.

Every run writes the resolved values of the parameters that affect behavior: the perturbation schedule and its location in the policy map, CFR+/DCFR options, average-policy weighting, player update order, protection mode, sampling policy, exploration parameters, random seed, and output-policy type. The reported seed mean and standard deviation for MC-ALP-CFR are computed at matching equivalent-round checkpoints over seeds 0–2; no smoothing or selection of the best seed is applied. Deterministic full-tree methods are not repeated to manufacture a confidence interval. The checkpoint evaluator is independent of the training RNG and does not change the solver state.

The comparison also records the status of the code and result source. The full-tree historical baselines, the controlled ALP-CFR binding, the corrected EGT batch, and the treeplex baselines are separate result batches, so a plotted curve is not interpreted as a same-commit causal experiment. The manifest exposes this limitation through the commit field and configuration path. The controlled ALP-CFR curve is the one used for the paper name; a historical solver is not promoted to that name solely because its CSV label contains a similar substring.

Finally, the protocol treats partiality as a first-class result state. A curve ending at 200k or 500k is drawn dashed when it is included as a formal baseline, but its endpoint is not compared with a completed 1M endpoint. A method with no validated metrics is omitted from the panel and listed as unavailable in this appendix. This is conservative for the current manuscript, but it makes the displayed differences auditable and prevents the figure from turning resource limits or a missing evaluator export into an algorithmic claim.

C.5 Additional figure inventory and missing experiments

The repository retains raw checkpoint CSVs, solver logs, and diagnostic plots outside the paper source. The figure manifest is the authoritative input mapping for Figure 1; its whitelist contains no historical middle-version method. Missing or partial experiments are: MC-ALP-CFR for GoofSpiel-3, Leduc-3, and Leduc-5; ALP-CFR for Leduc-5; EGT and RTCFR+ for Leduc-5; and incomplete EFPE-REG rows in the reported games. Several historical Leduc-3 full-tree runs are also partial and are excluded from causal claims. These limitations are reported explicitly so that the paper does not convert a finite experiment into a convergence statement.