Survey on Advanced Equilibrium-Finding Techniques in Imperfect Information Extensive Form Games

Vincent Wan
University of Waterloo
Waterloo, Ontario, Canada

ABSTRACT

Equilibrium finding in imperfect information extensive form games (EFGs) has been a cornerstone of artificial intelligence research, driving advancements in strategic reasoning and decision-making under uncertainty. While traditional algorithms like Counterfactual Regret Minimization (CFR) have been successful in solving large-scale poker games, recent developments focus on enhancing efficiency through variance reduction, predictive regret minimization, and meta-learning techniques. This survey reviews three key contributions in this space: improvements to Monte Carlo Counterfactual Regret Minimization (MCCFR) for faster convergence, predictive Blackwell approachability for more efficient regret minimization, and a meta-learning framework designed to accelerate equilibrium computation across distributions of game scenarios. We analyze these methods through three important research papers presented at respected international conferences, discussing their theoretical foundations, strengths and limitations, comparisons, and practical opportunities for further research.

ACM Reference Format:
Vincent Wan. 2025. Survey on Advanced Equilibrium-Finding Techniques in Imperfect Information Extensive Form Games. In . ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 SUMMARY OF PAPERS

1.1 Generalized Sampling and Variance in Counterfactual Regret Minimization [1]

The paper discusses Monte Carlo Counterfactual Regret Minimization (MCCFR), an extension of Counterfactual Regret Minimization (CFR), wherein the counterfactual values are estimated by sampling a subset of the terminal nodes, thus reducing the complexity of the game tree. By using this estimator, the paper establishes that the average regret, which is essential for approximating a Nash equilibrium (NE), is bounded by the estimator's variance, assuming the estimator is both bounded and unbiased. A key contribution of the paper is the introduction of the probing technique, which improves upon standard MCCFR by probing non-sampled terminal nodes, leading to a more accurate estimate of counterfactual values. This results in a tighter bound on the average regret, accelerating convergence toward the NE and providing faster approximations.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1.1.1 *Important Prior Findings.*

Counterfactual Regret Minimization (CFR) is an iterative algorithm used to find an ϵ-Nash equilibrium in two-player zero-sum games, with a time complexity of $O(|H||I_i|/\epsilon^2)$, where H is the set of histories and I_i is the set of information sets for player i. On each iteration t, CFR traverses the game tree and calculates the counterfactual value for player i at information set $I \in I_i$ under the current strategy profile σ^t:

$$v_i(\sigma, I) = \sum_{z \in Z_I} u_i(z)\pi^{\sigma}_{-i}(z[I])\pi^{\sigma}(z[I], z)$$

where $u_i(z)$ is the utility of player i at terminal history z, $\pi^{\sigma}_{-i}(h)$ is the product of all the players' contribution (including chance) to the probability of history h occurring if all players choose actions according to σ, except that of player i, $\pi^{\sigma}(h, h')$ is the probability of h' occurring after h, given that h has occurred, Z_I is the set of terminal histories passing through I, and $z[I]$ is the prefix of z contained in I.

The algorithm then computes the regret for each action $a \in A(I)$ at iteration t, which is computed as:

$$r_i^t(I, a) = v_i(\sigma^t_{(I \to a)}, I) - v_i(\sigma^t, I)$$

where $\sigma^t_{(I \to a)}$ represents the strategy profile at time t except that at I, action a is always taken. This measures how much player i would rather play a at I than play σ^t, hence the name "regret".

The counterfactual regrets are then accumulated over T iterations:

$$R_i^T(I, a) = \sum_{t=1}^{T} r_i^t(I, a)$$

The strategy σ^{T+1} is updated using regret matching:

$$\sigma^{T+1}(I, a) = \frac{R_i^{T,+}(I, a)}{\sum_{b \in A(I)} R_i^{T,+}(I, b)}$$

where $x^+ = \max(x, 0)$, and actions are chosen uniformly at random if the denominator is zero.

The external regret is minimized over time, leading to a strategy profile σ^T that approximates an ϵ-Nash equilibrium:

$$R_i^T = \max_{\sigma'} \sum_{t=1}^{T} \left[u_i(\sigma', \sigma^t_{-i}) - u_i(\sigma^t_i, \sigma^t_{-i}) \right]$$

If $R_i^T / T < \epsilon$, the strategy profile σ^T is an 2ϵ-Nash equilibrium.

Thus, CFR converges to an ϵ-Nash equilibrium in two-player zero-sum games by minimizing counterfactual regrets through iterative updates.

MCCFR is a variant of CFR designed to make solving large games more efficient by sampling a subset of the game tree on each iteration instead of traversing the entire tree. Let Z be the set of terminal histories (end states of the game). Let $\mathbf{Q}$ be a set of subsets

(blocks) of Z, where the union of $\mathbf{Q}$ spans Z. On each iteration, a block $Q \in \mathbf{Q}$ is sampled randomly from $\mathbf{Q}$. Outcome sampling is a specific form of MCCFR, where each block consists of a single terminal history. On each iteration, a block is sampled, and the game is traversed by selecting a random action at each decision point until a terminal history is reached.

The sampled counterfactual value for player i at information set I for action a is given by:

$$\tilde{v}_i(\sigma, I) = \sum_{z \in Z_I \cap Q} u_i(z)\pi^{\sigma}_{-i}(z[I])\pi^{\sigma}(Z[I], z)/q(z)$$

where $q(z)$ is the probability that the terminal history z was sampled. The sampled counterfactual regret for action a at information set I on iteration t is:

$$\tilde{r}^t_i(I, a) = \tilde{v}_i(\sigma^t_{(I\to a)}, I) - \tilde{v}_i(\sigma^t, I).$$

The sampled counterfactual values are unbiased estimates of the true counterfactual values. As fewer actions are sampled, the accuracy of the counterfactual values decreases, which introduces variance. The variance affects both regret updates and the values passed back up the game tree. Despite the introduction in variance, MCCFR still converges to an approximate Nash equilibrium faster than traditional CFR in many games.

Theorem 1.1. *Let $p \in (0, 1]$. When using outcome-sampling MCCFR, with probability $1 - p$, average regret is bounded by*

$$\frac{R^T_i}{T} \le \left(\tilde{\Delta}_i + \frac{2\tilde{\Delta}_i}{\sqrt{p}}\right)\frac{|I_i|\sqrt{|A_i|}}{\sqrt{T}}$$

where $|A_i| = \max_{I \in I_i} |A(I)|$ and $\tilde{\Delta}_i = \frac{\Delta_i}{\delta}$, such that $\Delta_i = \max_{z \in Z} u_i(z) - \min_{z \in Z} u_i(z)$, and $\delta > 0$ is chosen such that $\forall z \in Z$ either $\pi^{\sigma}_{-i}(z) = 0$ or $q(z) \ge \delta > 0$ at every iteration.

This bound shows that the average regret bound is related to the difference between the maximum utility of a terminal node, and the minimum utility of a terminal node. This theorem is important because the goal of MCCFR is to minimize regret, which ultimately leads to approximating a Nash equilibrium. The bound in Theorem 1.1 tells us how quickly this happens and sets expectations for the convergence behavior of the algorithm. To be more specific, Theorem 1.1 indicates that the speed at which regret is reduced is influenced by the bound $\tilde{\Delta}_i$ on the difference between two sampled counterfactual values.

This concludes the important prior findings that they have stated in the paper. After the prior findings, they focus on their new main contributions, which are new theorems and lemmas, alongside a new CFR sampling algorithm: MCCFR with probing.

1.1.2 Main Contributions.

Lemma 1.2. *Let $p \in (0, 1]$ and suppose that there exists a bound $\hat{\Delta}_i$ on the difference between any two estimates, $\hat{v}_i(\sigma_{(I\to a)}, I) - \hat{v}_i(\sigma_{(I\to b)}, I) \le \hat{\Delta}_i$. If strategies are selected according to regret matching on the estimated counterfactual regrets, then with probability at least $1 - p$, the average regret is bounded by*

$$\frac{R^T_i}{T} \le |I_i|\sqrt{|A_i|}\left(\frac{\hat{\Delta}_i}{\sqrt{T}} + \sqrt{\frac{\mathbf{Var}}{pT} + \frac{\mathbf{Cov}}{p} + \frac{\mathbf{E}^2}{p}}\right)$$

where $\mathbf{Var} = \max_{t \in \{1,\cdots,T\}, I \in I_i, a \in A(I)} Var[r^t_i(I, a) - \hat{r}^t_i(I, a)]$ with $\mathbf{Cov}$ and $\mathbf{E}$ similarly defined, and $\hat{v}_i(\sigma, I)$ denotes any estimator of the true counterfactual value $v_i(\sigma, I)$.

Lemma 1.2 generalizes the regret minimization framework by considering not just outcome sampling but also any estimator of the true counterfactual values. It introduces the idea of estimating the regret for action choices using probabilistic bounds that incorporate bias, variance, and covariance of the estimators. This generalization opens up several new possibilities not explored in MCCFR. For example, rather than sampling a block of terminal histories, one could sample a set of information sets and update regrets only at those specific locations. By allowing for any estimator of the counterfactual values, this result broadens the scope of techniques that can be used in MCCFR. It opens up possibilities for using different sampling schemes or methods to improve the efficiency of the algorithm.

Theorem 1.3. *If in addition to the conditions of Lemma 1.2, $\hat{v}_i(\sigma, I)$ is an unbiased estimator of $v_i(\sigma, I)$, then with probability at least $1-p$,*

$$\frac{R^T_i}{T} \le \left(\tilde{\Delta}_i + \frac{\sqrt{\mathbf{Var}}}{\sqrt{p}}\right)\frac{|I_i|\sqrt{|A_i|}}{\sqrt{T}}$$

This theorem focuses on unbiased estimators of the counterfactual values and shows that using unbiased estimators with lower variance leads to faster convergence to a Nash equilibrium, since the bound is smaller on average regret after T iterations. Hence, this result is key for improving the computational efficiency of MCCFR. A careful choice of estimators can significantly reduce the number of iterations required to reach a near-equilibrium state. However, it must be done carefully since if the estimator is computationally expensive, each iteration can take longer, potentially increasing the overall computation time.

1.1.3 A New CFR Sampling Algorithm.

The paper introduces a new variant of MCCFR that reduces variance and improves the time to approximate a Nash Equilibrium. This is done by introducing a new bounded and unbiased estimator that has lower variance than that from regular MCCFR, and can be computed nearly as fast. The key improvement is achieved by probing non-sampled actions using a single Monte Carlo roll-out to obtain more accurate estimates of their counterfactual values, ensuring an unbiased estimate that reduces variance. In regular MCCFR, non-sampled actions are assigned zero counterfactual value, leading to a high variance estimator. The probing method addresses this by replacing these ignored values with more accurate estimates of the true counterfactual value of that action, resulting in tighter regret bounds and faster convergence to the NE. Although the paper focuses on probing a single action per non-sampled action, it suggests that probing multiple actions or using off-policy techniques could further reduce variance and improve performance. This is discussed in the technical report.

Here is how the paper extends MCCFR. Once Q has been sampled, create an additional set of terminal histories, or probes, $B \subseteq Z \setminus Q$, as follows. For each non-terminal history h with $P(h) = i$ reached, and each action $a \in A(h)$ that is not sampled by Q, we generate exactly one terminal history $z = z_{ha} \in B$, where $z \in Z \setminus Q$ is selected on-policy (i.e., with probability $\pi^{\sigma}(ha, z)$). Given both Q

and B, when $Z_I \cap Q \neq \emptyset$, our estimated counterfactual value is defined as

$$\hat{v}_i(\sigma, I) = \frac{1}{q_i(I)} \Big[\sum_{z \in Z_i \cap Q} \pi_i^\sigma(z[I], z) u_i(z) + \sum_{z_{ha} \in Z_i \cap B} \pi_i^\sigma(z_{ha}[I], ha) u_i(z_{ha}) \Big]$$

where

$$q_i(I) = \Pi_{(I', a') \in X_i(I)} P[a' \in Q(I')]$$

is the probability that $Z_I \cap Q \neq \emptyset$ is contributed by sampling player i's actions. Here, $X_i(I)$ is the sequence of information set-action pairs for player i that leads to information set I, and this sequence is unique due to perfect recall. When $Z_I \cap Q = \emptyset$, $\hat{v}_i(\pi; I)$ is defined to be zero.

Proposition 1.4. *If $q_i(I) > 0$ for all $I \in I_i$, then $\hat{v}_i(\sigma, I)$ is a bounded, unbiased estimate of $v_i(\sigma, I)$.*

This proposition is important because, combined with Theorem 1.3's unbiased estimator bound, MCCFR with probing has probabilistic guarantees, ultimately leading to an ϵ-Nash equilibrium. The new algorithm in the paper outlines pseudocode for updating regrets using estimated counterfactual values.

The `Probe` function follows a single trajectory through the game tree, starting from a history h, by following chance probabilities and the current strategy (found using regret matching) until it reaches a terminal history, returning the utility at that point.

The `WalkTree` function is the main component of the algorithm and handles three primary cases:

(1) If the current history h is terminal, it simply returns the utility.
(2) If player i does not act at h, the algorithm follows a single action based on the probability distribution Q.
(3) If player i acts at h, the algorithm samples a set of actions. For each sampled action, the counterfactual value is calculated by recursively traversing the action's trajectory and adjusting the sample probability for future histories. Non-sampled actions, unlike in traditional MCCFR where they are ignored, are handled by the `Probe` function.

For each action, the counterfactual value is estimated as $\hat{v}_i(\sigma_{(I \to a)}, I) = \frac{v[a]}{q}$, where q is the probability of reaching the information set I contributed by player i's actions. After gathering all the action values, the algorithm updates the `regret` for each action.

The `Solve` function is executed for a sufficient number of iterations T to approximate a Nash equilibrium.

This concludes the main contributions of the paper.

1.1.4 Experiments. After the main contributions, they focus on the experimental results when comparing regular MCCFR to their new sampling algorithm in three different domains: Goofspiel, Bluff, and heads-up limit Texas hold'em poker. They first describe the games themselves, including how they're played and the utilities of each outcome. They then state that they use domain knowledge as well as intuition to select the sampling schemes **Q**. Finally, they discuss the test runs. Here is a summary of these runs.

- Empirical variance of the counterfactual estimates was compared between MCCFR and the new algorithm. The new algorithm showed significantly lower variance (0.133 vs. 0.295 for MCCFR) for Goofspiel, validating the benefit of the probing technique.
- Convergence Speed: The new algorithm converged faster than MCCFR in all three domains.
  - Goofspiel: 31% faster.
  - Bluff: 10% faster.
  - Texas Hold'em: 18% faster.

  The improvement was statistically significant In Goofspiel. The new algorithm achieved the same level of exploitability as MCCFR in about half the time.

These results demonstrate that the new sampling algorithm, utilizing probing for variance reduction, improves both convergence speed and estimate accuracy across multiple domains.

1.2 Faster Game Solving via Predictive Blackwell Approachability: Connecting Regret Matching and Mirror Descent [2]

This paper connects Follow-the-Regularized-Leader (FTRL) and Online Mirror Descent (OMD), two well-known Online Linear Optimization (OLO) algorithms, to Regret Matching (RM) and Regret Matching+ (RM+), respectively, through Blackwell approachability, which is a framework for repeated games with vector-valued payoffs. It then extends them with predictive capabilities that estimate future payoffs, leading to a new algorithm called Predictive Regret Matching+ (PRM+), which enhances performance and leads to faster convergence in EFGs. Experimental results show that PRM+ significantly outperforms traditional algorithms such as CFR+, Discounted CFR, and Linear CFR, achieving convergence rates up to two orders of magnitude faster, except for certain poker games, emphasizing that the new algorithm offers an improvement over traditional regret matching methods, thus advancing the state-of-the-art in equilibrium finding techniques in imperfect information EFGs.

1.2.1 Online Linear Optimization, Regret Minimizers, and Predictions. The paper first defines the concepts of OLO, regret minimizers, and predictions. Here are the most important concepts.

An oracle of the OLO problem supports two primary operations: `NEXTSTRATEGY` and `OBSERVELOSS`. The `NEXTSTRATEGY` operation returns a point $x^t \in D \subseteq \mathbb{R}^n$, while the `OBSERVELOSS` operation receives a loss vector ℓ^t that evaluates the previously returned strategy x^t. The loss vector depends on all past strategies, and the strategy depends on all prior losses and strategies. In other words, the future strategies and losses cannot be observed at time t.

The goal is to ensure that the regret at time T, denoted as:

$$R^T(\hat{x}) = \sum_{t=1}^{T} \langle \ell^t, x^t \rangle - \sum_{t=1}^{T} \langle \ell^t, \hat{x} \rangle = \sum_{t=1}^{T} \langle \ell^t, x^t - \hat{x} \rangle$$

grows sublinearly with respect to T, for all $\hat{x} \in D$. Oracles that achieve this are referred to as regret minimizers.

It is important to note that the subsequent loss vector ℓ^t can be predicted. This leads to the concept of predictive oracles, where in the `NEXTSTRATEGY` operation, it receives a prediction $m^t \in \mathbb{R}^n$

of the forthcoming loss vector ℓ^t. There exist algorithms that can guarantee the following bound:

$$R^T = O\left(1 + \sum_{t=1}^{T} \|\ell^t - m^t\|^2\right)$$

However, the most effective known oracles for OLO are FTRL and OMD.

In both algorithms, $\eta > 0$ is a step size parameter, $D \subseteq \mathbb{R}^n$ is a convex and closed set, and $\varphi : D \to \mathbb{R}_{\geq 0}$ is a 1-strongly convex differentiable regularizer. The Bregman divergence in Algorithm 3 (OMD) is denoted as:

$$D_\varphi(x\|c) := \varphi(x) - \varphi(c) - \langle \nabla\varphi(c), x - c\rangle \quad \forall x, c \in D.$$

Here is an important proposition that they made. It is a generalization of a previous proposition where it does not assume that the domain is a simplex, and it does not use quantities that might be unbounded in non-compact domains D.

Proposition 1.5. *The regret accumulated by predictive FTRL and predictive OMD compared to any strategy $\hat{x} \in D$ is bounded as:*

$$R^T(\hat{x}) \leq \frac{\varphi(\hat{x})}{\eta} + \eta \sum_{t=1}^{T} \|\ell^t - m^t\|_*^2 - \frac{1}{c\eta} \sum_{t=1}^{T-1} \|x^{t+1} - x^t\|^2,$$

where $c = 4$ for FTRL and $c = 8$ for OMD, and $\|\cdot\|_*$ denotes the dual of the norm with respect to which φ is 1-strongly convex.

This implies that with an appropriate step size (e.g., $\eta = T^{-1/2}$), predictive FTRL and predictive OMD guarantee:

$$R^T(\hat{x}) = O(T^{1/2}) \quad \text{for all } \hat{x}.$$

Thus, predictive FTRL and predictive OMD are regret minimizers.

1.2.2 Blackwell Approachability. Next, the paper discusses Blackwell Approachability, which extends the idea of playing repeated two-player games to games where payoffs are vectors instead of single scalar values. Here is the game setup.

Game Setup:

(1) Player 1 selects an action x^t from a set X that is compact and convex.
(2) Player 2 selects an action y^t from a set Y that is compact and convex.
(3) Player 1 receives a vector-valued payoff $u(x^t, y^t) \in \mathbb{R}^d$, where u is a biaffine function.

The objective of player 1 is to ensure that the average payoff converges to a specific closed convex target set $S \subseteq \mathbb{R}^d$, no matter what actions Player 2 takes. Formally, this means Player 1 needs to find a sequence $\{x^t\}_{t=1}^T$ such that:

$$\min_{\hat{s} \in S} \left\| \hat{s} - \frac{1}{T} \sum_{t=1}^{T} u(x^t, y^t) \right\|_2 \to 0 \quad \text{as } T \to \infty, \tag{1}$$

for any sequence $\{y^t\}_{t=1}^T$.

A key concept of Blackwell Approachability is the following.

Definition 1.6 (Approchable halfspace, forcing function). Let $(X, Y, u(\cdot,\cdot), S)$ be a Blackwell approachability game as described above, and let $H \subseteq \mathbb{R}^d$ be a halfspace. The halfspace H is said to be forceable if there exists an action $x^* \in X$ such that for all $y \in Y$:

$$u(x^*, y) \in H.$$

In this case, x^* is called a forcing action for H.

Blackwell's approachability theorem states that every halfspace H that contains S is forceable, if and only if (1) is satisfied.

1.2.3 Connecting Blackwell Approachability to OLO. Next, the paper connects Blackwell Approachability to OLO. This is the section where they extend a proof from a previous paper, stating that it is always possible to convert a regret minimizer into an algorithm for a Blackwell Approachability game. They extended it by allowing more flexibility in the choice of the domain of the regret minimizer. This will allow them to connect RM and RM+ to FTRL and OMD respectively.

To do this, they first propose a new algorithm (algorithm 3) which provides a way to play the Blackwell Approachability game such that (1) is satisfied.

The algorithm operates as follows. The regret minimizer selects decisions from the polar cone of C, which determines a normal vector for choosing a halfspace to force. At time t, the algorithm selects a forcing action x_t for the halfspace H_t, induced by the last decision θ_t from the OLO oracle L. The oracle L then incurs a loss of $-u(x_t, y_t)$, where u is the payoff function in the Blackwell approachability game. Here is another important proposition they made, that connects the algorithm and Blackwell Approachability.

Proposition 1.7. *Let $(X, Y, u(\cdot,\cdot), C)$ be a Blackwell Approachability game, where $C \subseteq \mathbb{R}^n$ is a closed convex cone, and every halfspace $H \supseteq C$ is approachable. Define:*

$$K := C^\circ \cap \mathbb{B}_2^n, \quad \text{where } C^\circ = \{x \in \mathbb{R}^n : \langle x, y\rangle \leq 0, \forall y \in C\}$$

is the polar cone of C, and $\mathbb{B}_2^n$ is the unit ball. Let L be an OLO oracle (e.g., FTRL or OMD) with decision domain D satisfying $K \subseteq D \subseteq C^\circ$. Then, at all times T, the distance between the average accumulated payoff from the algorithm and C is bounded by:

$$\min_{\hat{s} \in C} \left\| \hat{s} - \frac{1}{T} \sum_{t=1}^{T} u(x^t, y^t) \right\|_2 \leq \frac{1}{T} \max_{\hat{x} \in K} R_{\mathcal{L}}^T(\hat{x}),$$

where $R_{\mathcal{L}}^T(\hat{x})$ is the regret of $\mathcal{L}$ up to time T relative to always playing $\hat{x} \in K$.

Since K is compact and L is a regret minimizer, $\frac{1}{T} \max_{\hat{x} \in K} R^T(\hat{x}) \to 0$ as $T \to \infty$, ensuring that algorithm 3 satisfies the Blackwell approachability criterion.

1.2.4 Connecting FTRL and OMD with RM and RM+. Next, they show that creating a regret minimizer for a simplex domain $\Delta_n = \{x \in \mathbb{R}_{\geq 0} : \|x\|_1 = 1\}$ can be reduced to constructing an algorithm for a Blackwell Approachability game $\Pi = (\Delta_n, \mathbb{R}^n, u(.,.), \mathbb{R}^n_{\leq 0})$ such that the halfspace $H_a = \{x \in \mathbb{R}^n : \langle x, a\rangle \leq 0\}$ is forceable for all $a \in \mathbb{R}^n_{\leq 0}$. Such a forcing action is given by $g(a) = \frac{a}{||a||_1}$ when $a \neq 0$, otherwise $g(a)$ can be any element in Δ_n. This gives rise to a lemma that bounds the average regret.

Lemma 1.8. *The regret $R^T(\hat{x}) = \frac{1}{T}\sum_{t=1}^{T}\langle \ell^t, x^t - \hat{x}\rangle$ accumulated up to any time T by the decisions $x_1, \ldots, x_T \in \Delta^n$, compared to any $\hat{x} \in \Delta^n$, is related to the distance of the average Blackwell payoff from the target cone $\mathbb{R}^n_{\leq 0}$ as:*

$$\frac{1}{T}R^T(\hat{x}) \leq \min_{\hat{s}\in\mathbb{R}^n_{\leq 0}} \left\| \hat{s} - \frac{1}{T}\sum_{t=1}^{T} u(x^t, \ell^t) \right\|_2 .$$

Thus, a strategy for the Blackwell approachability game Π is a regret-minimizing strategy for the simplex domain Δ^n.

By leveraging algorithm 3 with specific regret minimizers, the following theorems are revealed.

Theorem 1.9. *(FTRL reduces to RM) For all $\eta > 0$, when the algorithm is set up with $D = \mathbb{R}^n_{\geq 0}$ and regret minimizer $\mathcal{L}^{ftrl}_\eta$ to play Π, it produces the same iterates as the RM algorithm. $\mathcal{L}^{ftrl}_\eta$ is the FTRL algorithm instantiated over the conic domain D with $\varphi(x) = \frac{1}{2}||x||_2^2$ and arbitrary step size η.*

Theorem 1.10. *(OMD reduces to RM+) For all $\eta > 0$, when the algorithm is set up with $D = \mathbb{R}^n_{\geq 0}$ and regret minimizer $\mathcal{L}^{omd}_\eta$ to play Π, it produces the same iterates as the OMD algorithm. $\mathcal{L}^{omd}_\eta$ is the OMD algorithm instantiated over the conic domain D with $\varphi(x) = \frac{1}{2}||x||_2^2$ and arbitrary step size η.*

These new theorems are important. Theorem 1.9 establishes that applying FTRL with domain $\mathbb{R}^n_{\geq 0}$ in algorithm 3 produces the same iterates as RM. Similarly, Theorem 1.10 shows that applying OMD with the same domain recovers the RM+ update rule. This means that RM and RM+ can be interpreted as fundamental regret minimization strategies derived from FTRL and OMD within the Blackwell approachability framework, hence offering a theoretical explanation for their success in EFG solving.

1.2.5 Predictive Blackwell Approachability and Predictive RM and RM+. The paper introduces a new algorithm for Blackwell approachability that converges faster to the target set when good predictions of future payoff vectors are available. It does this by applying proposition 1.7. Since the regret minimizer's loss is defined as $\ell^t = -u(x^t, y^t)$, any prediction v^t of $u(x^t, y^t)$ naturally serves as a prediction for the next loss. As long as this prediction is correctly incorporated, proposition 1.7 remains valid. They then provide another proposition that imposes bounds on the convergence speed.

Proposition 1.11. *Let $(X, Y, u(\cdot,\cdot), S)$ be a Blackwell Approachability game, where every halfspace $H \supseteq S$ is approachable. For all T, given predictions v^t of the payoff vectors, there exist algorithms for selecting $x^t \in X$ at each time t such that:*

$$\min_{\hat{s}\in S} \left\| \hat{s} - \frac{1}{T}\sum_{t=1}^{T} u(x^t, y^t) \right\|_2 \leq \frac{1}{\sqrt{T}} \left(1 + \frac{2}{T}\sum_{t=1}^{T} \|u(x^t, y^t) - v^t\|_2^2 \right).$$

This bound shows that the quality of predictions directly influences convergence speed, where better predictions lead to faster convergence. Using this notion, they can now explore the connection between predictive Blackwell Approachability and predictive RM/RM+. Since they have established that FTRL corresponds to RM and OMD to RM+, it follows that predictive FTRL and predictive OMD should align with PRM and PRM+, respectively. This turns out to be true, and the following theorems confirm their correctness and provide regret bounds.

Theorem 1.12 (Correctness of PRM, PRM+). *Let $L^{ftrl^*}_\eta$ and $L^{omd^*}_\eta$ denote the predictive FTRL and predictive OMD algorithms instantiated with $D = \mathbb{R}^n_{\geq 0}$ and regularizer $\varphi(x) = \frac{1}{2}||x||_2^2$, and predictions $v_t = \langle m^t, x^{t-1}\rangle \mathbf{1} - m^t$ for the Blackwell Approachability game $\Pi = (\Delta^n, \mathbb{R}^n, u(.,.), \mathbb{R}^n_{\leq 0})$. For all step sizes $\eta > 0$, when algorithm 3 is set up with $D = \mathbb{R}^n_{\geq 0}$, the regret minimizer $L^{ftrl^*}_\eta$ (resp., $L^{omd^*}_\eta$) to play Π, it produces the same iterates as the PRM (resp., PRM+) algorithm. Furthermore, PRM and PRM+ are regret minimizers for the domain Δ^n, and at all times T satisfy the regret bound*

$$R^T(\hat{x}) \leq \sqrt{2\sum_{t=1}^{T} \|u(x^t, \ell^t) - v^t\|_2^2}.$$

Theorem 1.12 is significant since it provides formal regret guarantees for PRM and PRM+. Specifically, it states the following.

- PRM and PRM+ produce the same iterates as the predictive versions of FTRL and OMD, respectively.
- The regret of PRM and PRM+ depends on the quality of the predictions v^t.
- The best regret bound is obtained by selecting an optimal step size η.
- The bound is sublinear in T. Hence, it follows that $R^T(\hat{x}) = O(T^{\frac{1}{2}})$.

1.2.6 Experiments. After the main contributions, the authors focus on experiments conducted on two-player zero-sum games. They use PRM+ as the regret minimizer instead of RM and apply heuristics to improve performance, resulting in the PCFR+ algorithm. They then compare PCFR+ to other state-of-the-art CFR variants, including CFR+, Discounted CFR, and Linear CFR.

The experiments were conducted on common benchmark games, with results presented for seven games in the main paper and 11 additional games in the appendix. The algorithms were tested for performance across iterations, with the x-axis representing the number of iterations and the y-axis showing the Nash gap and prediction accuracy. For non-predictive algorithms, the prediction was set to zero, while for the predictive algorithm, the prediction was updated using the previous loss.

Key results showed that PCFR+ was significantly faster than other algorithms, such as CFR+, LCFR, and DCFR, especially on games like Battleship and Pursuit-evasion, where it outperformed others by 3-6 orders of magnitude. On Goofspiel, PCFR+ was also faster but not as much. In poker games, PCFR+ was faster than CFR+ but slower than DCFR. In the small matrix game, PRM+ converges very rapidly. Across non-poker games in the appendix, PCFR+ seems to beat the other algorithms by several orders of magnitude.

PCFR+ performs extremely fast in Kuhn poker, outperforming all other algorithms. However, in Leduc poker and the River endgame, the predictions in PCFR+ offer less improvement. In the River endgame, PCFR+ performs similarly to CFR+, and in Leduc poker, it provides a small speedup over CFR+. DCFR is the fastest in these

two poker games. In contrast, DCFR performs worse than CFR+ in non-poker games but is sometimes on par. In the appendix, quadratic averaging in CFR+ helps it perform slightly better than PCFR+ in the River endgame and Leduc poker. Overall, PCFR+ was the fastest method for non-poker games, while DCFR was the fastest for poker games.

PCFR+ converges quickly when predictions are accurate. In games like Battleship and Pursuit-evasion, where predictions become highly accurate quickly, PCFR+ converges very fast. In Goofspiel, with less accurate predictions, PCFR+ is still much faster than other algorithms. In the River endgame, where predictions are less accurate, PCFR+ performs similarly to CFR+, but slower than DCFR. Similar results are observed in the appendix.

These results demonstrate that the new PCFR+ achieves significantly faster convergence compared to previous state-of-the-art algorithms, except in two poker games, thus advancing the state-of-the-art in equilibrium finding techniques in imperfect information EFGs.

1.3 Learning Not to Regret [3]

This paper presents a new approach to regret minimization by extending traditional frameworks to handle sampled games. It introduces two meta-learning algorithms, Neural Online Algorithm (NOA) and Neural Predictive Regret Matching (NRPM), which use recurrent neural networks to accelerate convergence to a Nash equilibrium in games with variable distributions. Existing algorithms typically assume isolated games, overlooking the fact that players often participate in multiple, interconnected games. For example, in poker, the public cards change each round, meaning each game can be seen as a sample from a distribution of possible public card configurations. NOA and NRPM improve upon traditional regret minimization methods by optimizing for environments where the game is sampled from a distribution rather than fixed. Experiments demonstrate that NOA and NRPM outperform RM and PRM in both matrix and sequential games, achieving faster convergence and lower exploitability.

1.3.1 Background. The authors provide a brief overview of the regret minimization framework, noting that while this paper introduces different notation, the fundamental framework remains unchanged from prior work. Essentially, an online algorithm m interacts repeatedly with an unknown environment g, choosing actions from a set A and receiving a reward vector x. The objective of the regret minimization algorithm is to minimize cumulative regret over time.

$$R^T = \sum_{t=1}^{T} r(\sigma^t, x^t).$$

At each timestep $t \leq T$, the algorithm selects a strategy σ^t from the probability simplex $\Delta^{|A|}$ and receives a reward x^t from the environment. The rewards follow an unknown concave function and are bounded. Instantaneous regret is defined as the difference between the reward obtained by a fixed action and the reward obtained under σ^t, given by:

$$r(\sigma^t, x^t) = x^t - \langle \sigma^t, x^t \rangle \mathbf{1}.$$

An algorithm m is considered a regret minimizer if its external regret $||R^T||_\infty$ grows sublinearly in T. This ensures that the average strategy converges to a Nash equilibrium in two-player zero-sum games. Finally, the exploitability of a strategy σ (i.e., its deviation from a Nash equilibrium) is defined as:

$$\text{expl}(\sigma) = \max_{\sigma^*} \min_{x} \langle \sigma^*, x(\sigma^*) \rangle - \min_{x} \langle \sigma, x(\sigma) \rangle.$$

This definition aligns with the standard measure of exploitability in two-player zero-sum games, where an adversary controls the environment's response. Exploitability is used in the experiments as a measure of performance.

1.3.2 Learning Not to Regret. Next, the paper describes the meta-learning framework and introduces two variants of meta-learned algorithms: one that guarantees regret minimization (NOA) and one that does not (NRPM).

The objective is to identify an online algorithm m_θ, parameterized by θ, for a given distribution of games G, such that the expected external regret after T steps is minimized. This can be expressed as:

$$L(\theta) = \mathbb{E}_{g \sim G} \left[\max_{a \in A} \sum_{t=1}^{T} r_a(\sigma_\theta^t, x^t) \right]$$

where σ_θ^t represents the strategy chosen by m_θ at time step t. To achieve this, a recurrent neural network parameterized by θ is trained, capturing temporal dependencies through the hidden state h. The use of an RNN is appropriate as it effectively models the time-dependent nature of strategies and rewards. The authors now propose their new algorithms, NOA and NRPM.

1.3.3 Neural Online Algorithm (NOA). NOA parameterizes the online algorithm m_θ to directly produce the strategy σ_θ^t. At each time step t, m_θ receives the reward x^t and the cumulative regret R^t, and updates its hidden state h^t. The gradient $\frac{\delta \mathcal{L}}{\delta \theta}$ is computed by sampling a batch of tasks and applying backpropagation through the computational graph. The gradient is derived from the final external regret and is propagated through the sequence of regrets $r^1, \ldots, r^T$, strategies $\sigma^1, \ldots, \sigma^T$, and hidden states $h^0, \ldots, h^{T-1}$. Notably, the gradient does not propagate through the rewards $x^0, \ldots, x^{T-1}$ or the cumulative regrets $R^1, \ldots, R^T$ entering the network. Consequently, this means only the hidden states of the neural network can influence the earlier optimization steps. NOA has strong empirical performance but does not possess regret convergence guarantees, because it only maximizes the cumulative rewards.

1.3.4 Neural Predictive Regret Matching (NRPM). NRPM extends the PRM framework discussed in the previous paper. In this framework, the regret prediction is represented by π, a function that returns the predicted regret p^{t+1} at time step $t+1$. This predicted regret p^{t+1} is then used by the PRM algorithm to compute the next strategy σ^{t+1}. For simplicity, the paper employs a basic predictor π, which assumes that the rewards at the next time step will be the same as the current rewards. Below is the updated algorithm for PRM, with the new notations introduced in the paper.

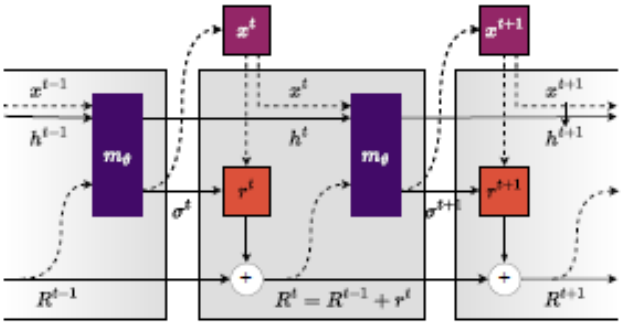

Figure 1: Neural Online Algorithm (NOA) Computational Graph

NRPM parametrizes the prediction function π using a recurrent neural network, denoted as π_θ, where θ represents the network's parameters. This network is trained to minimize the expected external regret. At each time step, π_θ takes as input the current rewards x^t, cumulative regret R^t, and the hidden state h^t, and outputs the predicted regret r^{t+1} for the next time step. The gradient for training π_θ is computed in a manner similar to the method used in NOA, where it is estimated using backpropagation through a batch of tasks. The authors demonstrate that the cumulative regret of the NRPM algorithm grows sub-linearly, confirming that it is indeed a regret minimizer. The formal proof of the theorem below uses the PRM regret bounds proven by the authors from the last paper explored.

Theorem 1.13. *(Correctness of Neural-Predicting) Let $\alpha \geq 0$ and π_θ by a regret predictor with outputs bounded in $[-\alpha, \alpha]^{|A|}$. Then PRM which uses π_θ is a regret minimizer.*

This theorem is crucial since it ensures that the algorithm learns to minimize regret over the specific domain it trains on, while also ensuring convergence guarantees in other domains.

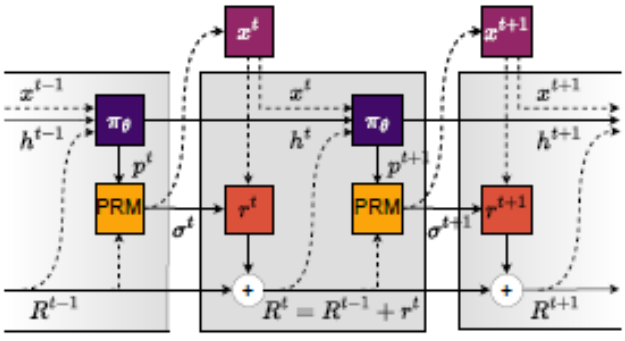

Figure 2: Neural Predictive Regret Matching (NRPM) Computational Graph

1.3.5 Experiments. After the main contributions, the authors conducted many experiments on regret minimization in the context of imperfect information EFGs. For these experiments, they employed a specific neural network architecture: a two-layer LSTM. In the case of NOA, these two layers were followed by a fully connected layer with a softmax activation function. For NPRM, the outputs were scaled by $\alpha \geq 2\Delta_{\max}$. The networks were provided contextual information such as the player's observations. They trained the models by minimizing expected external regret for $T = 64$ iterations over 512 epochs using the Adam optimizer. Other hyperparameters were determined through a grid search. For evaluation, the authors computed the exploitability of the strategies up to $2T = 128$ iterations to assess whether the algorithms could generalize beyond the training horizon T and whether they continued reducing exploitability. Both NOA and NPRM were trained and evaluated, and their performance was compared against traditional RM and PRM.

In their experiments with matrix games, the authors observed that when the game distribution is constant, the convergence of NOA and NRPM is significantly faster compared to RM and PRM. In particular, NOA outperforms NRPM, which the authors hypothesize is due to NRPM's higher functional dependency restrictions and its issues with vanishing gradients when cumulative regret is small, as well as exploding gradients when cumulative regret is large. When the game distribution is not constant, NOA and NRPM still outperform RM and PRM, even after the training horizon T. Both NOA and NRPM converge smoothly toward equilibrium, whereas RM and PRM explore a large portion of the policy space, resulting in slower convergence. The authors note that RM exhibits performance similar to PRM in both matrix and sequential games, explaining that the similarity arises from the fact that in their experiments, regret is minimized against an adversary, rather than a self-play opponent.

In their experiments with sequential games, the authors focused on a subgame of no-limit Texas Hold'em Poker, which consists of approximately 62,000 information states. The game distribution is a uniform distribution of public cards, and they used 1000 iterations of CFR+ for training. To improve performance, they augmented the input to the network by including features corresponding to the player's observations at each state. This included the beliefs of both players and an encoding of the private and public cards, allowing both NOA and NRPM to learn in context. The results showed that both NOA and NRPM closely approximate a Nash equilibrium in fixed games. More notably, in the sampled distribution game setting, NOA and NRPM outperformed RM and PRM significantly, reducing exploitability about 10 times faster. PRM showed similar performance to RM, as in the matrix game setting. To further quantify the improvements, the authors tracked the number of steps required to reach solutions of specified quality. Both NOA and NRPM outperformed RM and PRM across all levels of solution quality.

In terms of computational time, the authors were initially concerned that the reduction in interactions with the environment might increase computational time due to the overhead of calling the neural network. However, despite this concern, NOA and NRPM still outperformed RM and PRM in matrix games. On river poker, which involves a large number of information states and expensive interactions with the environment, NOA and NRPM showed massive improvements. For instance, exploitability reached by NRPM in one minute would take RM approximately 26 minutes and PRM about 34 minutes. The authors also tested whether the algorithms exhibited convergence guarantees in out-of-distribution settings,

akin to evaluating the performance of algorithms on unseen test data. As expected, since NOA lacks convergence guarantees on regret, its performance significantly deteriorated in these settings. In contrast, NRPM was able to maintain performance and continue minimizing regret, outperforming both RM and PRM. Finally, the authors performed further experiments to gain deeper insights, including modifications to NOA and NRPM. These modifications involved summing only the positive parts of the regrets, akin to the RM+ and PRM+ variants of RM and PRM, as well as using the Hedge algorithm instead of RM to produce the strategy σ. As anticipated, these modifications outperformed the corresponding variants of RM and PRM.

These experimental results demonstrate that the new algorithms NOA and NRPM achieve lower exploitability roughly an order of magnitude faster than traditional regret minimization methods, thus advancing the state-of-the-art in equilibrium finding techniques in imperfect information EFGs.

2 OVERVIEW OF OPEN RESEARCH PROBLEMS

Here are some of the recent open research problems in the subarea that are unrelated to the three papers surveyed.

2.0.1 [7] Scalable, Generalizable, and Offline Methods. A recent paper focuses on constructing scalable, generalizable, and offline algorithms for solving large-scale imperfect-information EFGs. The main contributions of the work are as follows.

The authors propose two algorithms that significantly enhance scalability. These methods are evaluated in pursuit-evasion games and demonstrate superior performance over existing baseline algorithms in terms of both speed and scalability.

A new algorithm is developed to improve generalizability. Experimental results show that it consistently outperforms prior algorithms in terms of solution quality and generalizability.

An offline algorithm is presented for solving imperfect information EFGs. It demonstrates superiority over existing offline reinforcement learning algorithms by effectively computing equilibrium strategies using pre-collected datasets.

The paper outlines several open research questions within the broader area of solving imperfect information EFGs. The questions are as follows.

While the proposed algorithms perform well on specific types of games, like team-adversary and pursuit-evasion, developing scalable algorithms that generalize to a broader class of imperfect information EFGs remains a key challenge. This requires methods that can adapt to diverse game structures and dynamics.

Most existing algorithms focus on computing Nash equilibrium. A promising research direction is to explore algorithms that can compute a range of equilibrium strategies based on different user-defined requirements.

Incorporating LLMs can enhance the generalizability of existing frameworks. These models can help create more flexible and robust approaches capable of handling a wider variety of equilibrium concepts and adapting to diverse game-theoretic scenarios.

Although the presented offline algorithm marks an important step, several future directions remain.

- Investigate more advanced offline methods and improved parameter estimation techniques to enhance the effectiveness and efficiency of solving increasingly complex games in the offline setting.
- Apply offline learning approaches to other game models, such as stochastic and Bayesian games, to bridge the gap between theoretical developments and real-world applications.
- Incorporate real-world datasets into the offline learning process to validate and refine algorithms in practical scenarios, ensuring broader applicability.

2.0.2 [8] Designing Algorithms for Deep Reinforcement Learning. A recent paper focuses on designing algorithms suitable for deep reinforcement learning in two-player zero-sum EFGs with imperfect information. One of the key contributions of the paper is QFR, which is a regret minimization algorithm that extends standard counterfactual values with trajectory-based Q-values and employs different learning rates per information set to make learning more adaptable to local game dynamics. To eliminate the need for importance sampling, the authors also proposed a bidilated regularizer, which scales the regularization in each information set based on both players' reach probabilities.

As for open research problems, one of the most significant challenges is to scale QFR to large, complex games, such as dark chess, with a vast state space and strategic complexity. Another remaining open research problem is the development of algorithms using a single global learning rate across all information sets, which would be required for compatibility with neural networks, since neural network parameter updates are not compatible with different learning rates per information set. Resolutions to these problems would be significant steps towards actual usage of reinforcement learning for solving large-scale imperfect information games.

We will now move on to proposals of open research problems in the subarea that are related to the three papers surveyed.

2.1 Generalized Sampling and Variance in Counterfactual Regret Minimization [1]

2.1.1 Tradeoffs in Probing and Potential Solutions. Let K represent the number of probes performed per iteration. Note that when $k = 1$, MCCFR with probing uses the traditional probing approach as described in the paper. An increase in K reduces the variance of the counterfactual value estimate, which results in a stricter probabilistic bound on the average regret, thereby accelerating convergence to a Nash equilibrium over a larger number of iterations. However, this benefit comes with the tradeoff of increased time complexity per iteration, which can slow convergence. Thus, there is a balance between the number of probes and the time complexity per iteration, because while the increased number of probes could slow individual iterations, the total number of iterations might decrease. Therefore, further research is needed to identify the optimal number of probes per action at each information set. Finding the right balance between the number of probes and the time complexity per iteration can further improve convergence rates.

As an example of open research, it could be beneficial to adapt the number of probes dynamically across iterations, adjusting the number of probes for each action and for each information set.

2.1.2 Domain Knowledge To Select Sampling Schemes. When the authors conducted the experiments, they used domain knowledge as well as intuition to select the sampling schemes $\mathbf{Q}$. In the case of heads-up limit Texas hold'em poker, the authors always sample fold and raise actions, while the call action is sampled with a probability of 0.5. Folds terminate the game immediately and are inexpensive to compute, while raises increase the betting amount, expanding the utility magnitudes. Although this use of domain knowledge for determining the sampling schemes is valid, further research can be made to improve this. Specifically, the fixed 0.5 probability for sampling the call action seems arbitrary and may not reflect the true strategic importance of calling in different game states. In many poker scenarios, the decision to call is highly context-dependent, influenced by factors such as pot odds, the cards held, and the opponent's tendencies. A fixed 0.5 probability ignores these factors, potentially leading to an inaccurate representation of call actions in the decision tree. If the call action is sampled too frequently in situations where folding is preferable, it can lead to inefficient regret updates. On the other hand, under-sampling the call action in favorable situations may delay convergence to an optimal strategy.

As an example of open research, instead of using a fixed probability of 0.5, more sophisticated sampling methods could be employed, such as context-dependent strategies that adjust the probability of sampling the call action based on the current game state. For example, one could incorporate factors like pot odds, opponent behavior, and the player's hand strength into the sampling decision.

2.2 Faster Game Solving via Predictive Blackwell Approachability: Connecting Regret Matching and Mirror Descent [2]

2.2.1 Exponential Weighted Moving Average. The authors performed experiments using PRM and PRM+ by setting $m^t = \ell^{t-1}$ for all $t \geq 2$, with $m^1 = 0$. This implies that the initial predicted loss is set to 0, while subsequent predicted losses are set to the loss from the previous timestamp. This approach provides a reasonable approximation of the next loss. However, further research can be made to improve this.

This paper provides an insight for predicting the losses by drawing inspiration from the concept of round-trip time (RTT) estimation in computer networks. Further research is suggested to explore this improved prediction, along with examples of potential games where it could be tested.

The Exponential Weighted Moving Average (EWMA) is a statistical method used to model time series data, particularly in computer networks for estimating the round-trip time (RTT) of packets. The general formula for EWMA is as follows:

$$EWMA^t = \alpha \times r^t + (1 - \alpha) EWMA^{t-1}$$

where α is a user-defined weight, r^t is the observed value at time t, and $EWMA^t$ represents the expected moving average at time t.

Building on this, here is an alternative prediction of the loss at timestamp t.

$$m^t = \begin{cases} 0 & \text{if } t = 1 \\ \alpha \times \ell^{t-1} + (1 - \alpha) \times m^{t-1} & \text{if } t \geq 2 \end{cases}$$

This equation offers an enhanced estimate by balancing recent and historical data. The value of α determines the trade-off between weighting recent losses more heavily or giving more significance to historical losses. A higher α emphasizes recent losses, while a lower α incorporates more historical data. This balance ensures that the prediction is not overly influenced by volatile, short-term changes in loss, smoothing out the impact of such fluctuations.

This approach is an advantage over the original prediction model because by incorporating both recent and historical losses, the prediction method allows for more stable estimates, reducing the impact of sudden, transient changes that may not follow the overall trend. Such an approach is particularly useful in dynamic game-theoretic scenarios, where rapid responses to changing losses are necessary, but short-term volatility should not dominate the prediction.

Further research is needed to identify which types of EFGs would benefit most from this improved loss prediction and to determine the optimal value of α for different games. The likelihood of such games benefiting from this method is high, especially in contexts like poker, where losses can vary significantly depending on the outcome of each hand and betting strategies.

2.2.2 Full Version. Here are some extra details listed from the full version of the paper, as well as their potential ability for further research.

The full paper presents additional experimental results and utilizes PCFR+, CFR+, LCFR, and DCFR. The results indicate that PCFR+ significantly outperforms the other algorithms, except for a version of Leduc poker. This aligns with the experimental findings in the regular paper, where DCFR outperforms CFR+ in poker but not in other domains. In the Small Matrix game, CFR+, LCFR, and DCFR exhibit similar performance, with slower than T^{-1} convergence, while PCFR+ does not exhibit this issue, highlighting that small matrix games pose a challenge for non-predictive methods but not for predictive ones. As an interesting future research direction, the authors suggest investigating the properties of rapidly decreasing prediction errors, potentially providing stability guarantees similar to those offered by FTRL and OMD. This encourages further experimentation and advancement in the field of regret minimization and CFR.

The authors then experimented with linear and quadratic averaging of strategy iterates in PCFR+ and CFR+. They found that CFR+ with quadratic averaging performs similarly to CFR+ with linear averaging. In contrast, PCFR+ with linear averaging outperforms or performs similarly to PCFR+ with quadratic averaging in two games. They also observed that PCFR+ outperforms CFR+ (with both linear and quadratic averaging) in 11 games, but performs worse in two poker games. This indicates that the speedup of PCFR+ is not due to the switch from linear to quadratic averaging. As an area of open research, they could test these findings across more games to verify if the results hold beyond just two games. The conclusion that linear averaging is better for PCFR+ and worse for CFR+ might need more evidence. Additionally, they could test

whether PCFR+ consistently outperforms CFR+ with both averaging strategies in more than 11 games to confirm this relationship. Another area of open research would be to experiment with cubic averaging of strategy iterates and compare its performance against linear and quadratic averaging. Finally, an open research examining the impact of different averaging methods on other algorithms like DCFR, LCFR, and ECFR could provide more insights into how averaging strategies interact with these algorithms.

Finally, the authors proposed an algorithm for Regret Matching in the discounted sense, called Predictive Discounted CFR (PDCFR), which adjusts regrets based on whether they are positive or negative. They used specific coefficients, $\alpha = \frac{3}{2}$ and $\beta = 0$, known from previous work to improve convergence and outperform CFR+ in various settings. Open research could be done to explore the effects of different α and β values in PDCFR, and compare this with other algorithms like PCFR+, CFR+, and DCFR. Additionally, they tested quadratic averaging of both strategy iterates and past loss vectors for predicting future losses. This quadratic average loss prediction was shown to lead to faster convergence than linear averaging from experiments. An open research could be made to test other forms of averaging, such as cubic averaging, to improve predictions and compare the Nash gap convergence rates for PDCFR, PCFR+, CFR+, and DCFR using different loss prediction methods. This could again provide more insights into how averaging strategies interact with these algorithms.

2.3 Learning Not to Regret [3]

2.3.1 Self-Play. The authors planned to extend their results to the self-play settings. If they do so, one key outcome is expected. Prediction-based algorithms, such as PRM, are anticipated to outperform RM in self-play environments due to their ability to make more accurate predictions. This advantage results in a smaller regret bound, leading to faster convergence to minimal exploitability with reduced exploration of the policy space. The improved performance of these algorithms stems from the stability of an opponent's strategy across iterations due to the self-play setting.

2.3.2 Neural Network. Here is an open research problem of performing the experiments over different potential neural network architectures for the NOA and NPRM algorithms to find better configurations that maximize the rate of convergence to a Nash equilibrium. Several key considerations for enhancing performance are outlined.

- Utilize a Long Short-Term Memory (LSTM) network with more than two layers. This increased complexity could reduce the number of iterations required to achieve lower exploitability.
- Explore the use of Recurrent Neural Networks (RNNs) instead of LSTMs. While RNNs lack the capacity to capture long-term dependencies in sequential data, they feature a simpler architecture with recurrent connections. As a result, while the number of iterations may increase, the time required for each iteration will be significantly reduced, potentially leading to faster achievement of lower exploitability.
- Investigate different activation functions. In the paper, the authors employ a fully connected layer with a softmax activation function, which is commonly used for multi-class classification. However, this may not be the most suitable choice for minimizing the expected external regret over T steps. Alternatives such as Tanh or Sigmoid activation functions, commonly used in RNNs and LSTMs, may be more appropriate as they better align with the network's goal of minimizing regret.
- Consider extending the number of steps and epochs, such as using $T = 128$ and 1024 epochs. This adjustment could further reduce the expected value of external regret. However, increasing these values may also lead to overfitting. To mitigate overfitting, techniques such as dropout, where random nodes are assigned a weight of zero, can be employed. Additionally, cross-validation can be used to assess the model's performance by training multiple models on different data subsets and testing them on the remaining data.
- Experiment with different hyperparameters for the Adam optimizer when minimizing the objective function associated with external regret. For example, varying the cosine learning rate decay between 10^{-4} and $3 \cdot 10^{-5}$ could improve performance. Other optimizers such as Adamax, AdamW, Adadelta, or SGD may also be considered, as they provide alternative strategies for adapting first and second-order moment estimates.
- Adjust other relevant hyperparameters. In the paper, the authors varied the size of the LSTM layer, the number of games per batch gradient update, and the regret prediction bound α. Increasing the LSTM layer size may add complexity to the architecture, potentially reducing the expected external regret. Similarly, increasing the number of games in each batch could reduce the number of iterations needed for convergence, though it may also increase the time per iteration. Additionally, modifying the regret prediction bound α could change the network's operation. Other potential hyperparameters include the learning rate (step size) of the gradient descent, which must be balanced to avoid overshooting or slow convergence, and the number of neurons in the hidden layers, which affects the model's complexity and output.

2.3.3 Full Version. Here are some extra details listed from the full version of the paper, as well as their potential ability for further research.

- The Rock-Paper-Scissors (RPS) experiment uses a modified payoff matrix that introduces bias in the fixed variant compared to the original unbiased version. This modification was intended to create a non-uniform equilibrium strategy, differing from the standard initialization in RM and PRM algorithms. Open research of additional experimentation with different probability distributions for the matrix parameters X and Y, such as non-uniform, continuous, and discrete distributions, could benefit the study. Exploring variants like $U(-2, 2)$, exponential, or geometric distributions could provide more insights into convergence behavior and exploitability.
- The focus on only RPS and a uniform matrix game limits the generalizability of the findings. Open research on studying a broader range of matrix and EFGs, such as Goofspiel,

Liar's Dice, and Pursuit-Evasion, could help assess algorithm performance across different scenarios.
- The river poker experiment is based on a simplified subgame of Texas Hold'em with 61,617 information states. Open research on experimenting with different initial pot sizes, total budgets, and sampling techniques for public cards (e.g., biased or clustered sampling) could lead to valuable insights into strategy adaptation.
- The study primarily relies on CFR+ for value approximation, but open research on experimenting with alternative regret minimization algorithms, like DCFR or LCFR, might reveal different trade-offs between convergence speed and policy exploration.
- The integration of NOA and NRPM with regret minimization methods like RM+ and Hedge introduces both improvements and drawbacks. RM+ enhances NPRM but weakens NOA due to limited regret information. Hedge, with a fixed learning rate, slows convergence and reduces NPRM performance. A key limitation of the Hedge algorithm is its reliance on a fixed learning rate β. Open research on experimenting with adaptive learning rates, like time-decayed, square-root, or logarithmic decay functions, could improve stability and convergence efficiency.

3 STRENGTHS AND WEAKNESSES

3.1 Generalized Sampling and Variance in Counterfactual Regret Minimization [1]

Paper [1] is mediocre in both theoretical depth and practical application. While it introduces key concepts such as CFR and RM effectively, and offers significant background on game theory, it remains unexplored in some areas. The paper does a commendable job of explaining foundational game-theoretic concepts like Nash equilibrium, utility functions, strategy profiles, expected utilities, and best responses, which makes it accessible for new students, especially compared to the more complex paper [2]. However, it still presents challenges in understanding, especially for certain bounds that are difficult to prove, making it more challenging to understand than paper [3].

The paper defines CFR in great detail, exploring its properties, including counterfactual values and regrets, and includes diagrams such as game trees to help visualize these concepts. However, some definitions are quite complex and require a thorough background knowledge of CFR to conceptualize. For example, the definition of the estimated counterfactual value $\hat{v}_i(\sigma, I)$ obtained by probing consists of almost half a page to define, and is filled with variables and symbols that are challenging to interpret and follow.

A key theoretical contribution of the paper is the introduction of an estimator for counterfactual values that reduces variance compared to the estimator used in traditional MCCFR. While this method is theoretically justified, the paper could have provided a more in-depth discussion of the practical implications.

In summary, the first paper provides valuable theoretical insight but could benefit from deeper practical exploration. It successfully introduces key concepts and methods, though some aspects, particularly the proof-heavy theoretical sections, may prove challenging for readers new to the topic.

3.1.1 Strengths. The experimental study employs a comparative analysis of MCCFR and the new sampling algorithm across three distinct domains, ensuring a moderate evaluation of performance. The selection of Goofspiel, Bluff, and Texas Hold'em provides a mix of imperfect information games, making the findings generalizable to different strategic settings.

Furthermore, the empirical evaluation includes both variance reduction analysis and convergence speed comparisons, which are critical for assessing the efficiency of the new sampling algorithm. The use of empirical variance measurements to validate the reduction in variance through probing offers strong quantitative support for the new sampling algorithm. Additionally, they performed five runs for each of MCCFR and the new sampling algorithm to record the improvements in exploitability over time. Multiple test runs help account for randomness and variability, improving the reliability of results.

The choice of sampling schemes, guided by domain knowledge and intuition, aligns with their practical considerations for improving algorithm performance. The study successfully justifies its choice of "important" parts of the game tree to focus updates on, which contributes to a meaningful efficiency improvement. The reported statistical significance of the improvements in Goofspiel and Texas Hold'em also strengthens the validity of the findings.

3.1.2 Weaknesses and Areas for Improvement. The reliance on domain knowledge and intuition for selecting sampling schemes introduces potential biases. A more systematic approach, such as an ablation study testing different sampling strategies, would provide stronger justification for these choices. Additionally, it is unclear how sensitive the algorithm's performance is to variations in the sampling schemes. Sensitivity analysis could offer insights into the robustness of the proposed method.

The methodology also lacks a thorough examination of computational costs. While the new sampling algorithm shows improved convergence rates, a direct comparison of runtime and memory usage between MCCFR and the new algorithm would be valuable. Given that computational efficiency is a primary motivation, a deeper analysis of the trade-offs between reduced variance and increased computational complexity is necessary.

There is also limited discussion on the impact of different parameter settings. The study reports results for specific game configurations, but does not explore how the performance scales with larger or more complex instances. A more extensive set of games would clarify whether the improvements hold consistently across different problem sizes.

Lastly, while statistical significance is mentioned for Goofspiel and Texas Hold'em, confidence intervals or hypothesis testing details are not provided. Reporting standard errors or performing statistical tests such as t-tests would enhance the credibility of the results.

3.1.3 Proof Statements. All statements made on the paper have been proved. The proofs are all in the full technical paper. Here is a critical assessment of the proof statements and their detail.

- **Theorem 1.1:** Theorem 1.1 is stated without a proof in the paper, as it has already been proven in prior work. The authors omit the proof details to avoid confusing readers unfamiliar with concepts like RM, CFR, and MCCFR. This decision ensures the paper remains accessible while clearly presenting their findings. Additionally, the bound presented in prior work is slightly tighter than the one in the paper, where the variable $|I_i|$ is replaced by a constant M_i, which is independent of the sampling scheme and satisfies $\sqrt{|I_i|} \leq M_i \leq |I_i|$. Defining this constant is complex, which is why they exclude it here.
- **Lemma 1.2:** Lemma 1.2 is rigorously proven, using prior work in the field. The authors note that its proof is similar to theorem 7 in an earlier paper, allowing readers to refer to that work for additional context.
- **Theorem 1.3:** Theorem 1.3 is rigorously proven, with the authors leveraging lemma 1.2 and proving that the covariance and expected value terms are zero.
- **Proposition 1.4:** Proposition 1.4 is proved by first defining the estimated counterfactual value obtained through probing in its most general form. The authors then demonstrate that the estimated counterfactual value is both bounded and unbiased, leading directly to the proof of proposition 1.4 through a lemma. This proposition highlights the authors' commitment to providing thorough explanations and extending their results by exploring various probing techniques, such as performing multiple probes per action, probing off-policy, and factoring in multiple terminal histories.
- **Further Generalization:** While not necessary for their primary results, the authors add further generalizations to their findings. They show how the bound on theorem 1.3 can be tightened using the game-dependent constant M_i, assuming specific game structures. This is done by partitioning the information sets, defining the M-value for player i, and presenting a new theorem that proposes a tighter bound. The proof follows the structure of lemma 1.2 and theorem 1.3 with only minor changes, demonstrating the authors' extra effort in expanding the paper's scope.

3.2 Faster Game Solving via Predictive Blackwell Approachability: Connecting Regret Matching and Mirror Descent [2]

Paper [2] is significantly harder to read than paper [1] and paper [3], particularly for students new to the topic. It is heavily focused on theory but has mediocre practical insights. It contains numerous proofs that require a solid understanding of prior works, such as FTRL, OMD, and Blackwell Approachability. Many of these proofs reference previous findings or build upon earlier research. However, the paper assumes familiarity with the background literature, and only briefly goes over the concepts that the paper builds upon, which could make it difficult for newcomers to fully grasp the concepts. For example, proposition 1.7 consists of many variables and symbols, as well as mathematical terms such as closed convex cone, unit ball, and polar cone, which could be a challenge to understand for those who aren't familiar with mathematical structures.

The experiments in the second paper rely heavily on findings from prior research, such as quadratic averaging of the strategy iterates that leads to better practical performance, and specific parameter recommendations to ensure the best version of discounted CFR. In terms of experimentation, the authors conducted a very thorough benchmarking. They evaluated many games, including three poker variants and 11 additional games in the appendix, far more than paper [1] and paper [3]. But this extensive experimental setup is overshadowed by the complexity of the theory.

3.2.1 Strengths. The study conducts a very thorough benchmarking by comparing their new PCFR+ with prior state-of-the-art CFR variants such as CFR+, Discounted CFR, and Linear CFR. They compare across seven games in the main paper, and an additional 11 games on the full paper, ensuring very meaningful and broad evaluations of performance. The diverse selection of games, spanning both poker and non-poker domains, strengthens the validity of conclusions. Performance metrics focus on Nash gap reduction and prediction accuracy over iterations, effectively visualized on a log scale to highlight improvements. Additionally, an ablation study isolates the impact of quadratic averaging, confirming that observed performance gains stem from the predictive approach rather than averaging effects alone.

3.2.2 Weaknesses and Areas for Improvement. The study lacks a rigorous theoretical justification for the applied heuristics, such as quadratic averaging and alternating updates, leaving their general effectiveness unclear. Results are primarily based on visual comparisons of log-scale plots without statistical significance tests, making it difficult to assess whether performance differences are meaningful. While PCFR+ excels in non-poker EFGs, it underperforms in poker games compared to DCFR, suggesting possible overfitting to certain game structures. Lastly, the study does not explore the broader implications of PCFR+ for imperfect information games beyond the tested EFGs, leaving its real-world applicability in multi-agent strategic settings an open question.

3.2.3 Proof Statements. All statements made on the paper have been proved. The proofs are all in the full technical paper. Here is a critical assessment of the proof statements and their detail.

- **Proposition 1.5 (FTRL):** Proposition 1.5 for FTRL is proven in great detail using a technical lemma from a previous paper. The authors combine insights from previous works to rigorously prove the proposition. The authors also reference the original works, allowing readers to learn more about the derivation and proof process.
- **Proposition 1.5 (OMD):** Similarly, Proposition 1.5 for OMD is proven with great detail and rigor, utilizing two lemmas. The proof draws from previous works, providing a detailed and thorough explanation.
- **Proposition 1.7:** Proposition 1.7 is proven using a result from previous works.
- **Lemma 1.8:** Lemma 1.8 is proven in great detail, using clever inner product properties and techniques to simplify the proof.
- **Theorem 1.9 and 1.10:** The reductions of FTRL and OMD, to RM and RM+, are proven with clear explanations. The proofs start with the loss from the OLO, then use the choice

of D and regularizer $\psi(x)$ to simplify the update step to the appropriate strategy output in RM and RM+.

- **Proposition 1.11:** Proposition 1.11 is proven using a result from previous works, which shows how a Blackwell approachability game with a non-conic target set can be converted to a conic target set, incurring a factor of 2 in the distance bound. This leads to the appropriate bound and proof conclusion.
- **Theorem 1.12 (Correctness of PRM and PRM+):** The correctness of PRM and PRM+ is proven by breaking down the analysis according to the OLO oracle used. The PRM part of the proof is similar to the regular RM proof, while the PRM+ part uses findings from Theorem 1.10 for a straightforward proof.
- **Regret Bound for PRM and PRM+:** The bound on the regret for PRM and PRM+ is proven using inner product properties and detailed steps to arrive at the conclusion.

3.2.4 Full Version. Here is an extra detail listed from the full version of the paper where the authors explore the limitation of why PCFR+ outperforms CFR+ and DCFR in some games but not others, identifying several key observations.

First, the game size does not reliably predict the effectiveness of PCFR+, as it performs well in medium-sized games like River Endgame and Liar's Dice.

Second, a high ratio of terminal states to decision points seems to enhance predictive methods' performance, as seen in Pursuit Evasion, though this conclusion is based on a limited set of games and may not generalize to more complex scenarios.

Lastly, the presence of private information alone does not dictate the success of predictive methods, as games with different hidden information structures, such as Poker, Liar's Dice, and Battleship, show varying results.

In addition, in the full version of the paper, the authors describe the game instances used in their experiments, explaining the scenarios, player actions, and utilities in both mathematical and plain language. This makes it accessible to readers who are not familiar with EFGs or game theory terminology, allowing newcomers to easily understand the examples and experimental setup.

3.2.5 Bibliographic Remarks. The authors highlight two related works that intersect with their study. First, Gordon's Lagrangian Hedging Framework shares similarities with the predictive approach introduced in the paper. However, the authors of that approach were unaware of Gordon's results at the time of writing. This leaves an open question regarding how predictive methods could be integrated into Gordon's framework to enhance regret minimization and CFR algorithms.

Second, Burch's "optimistic RM+" algorithm, mentioned in his PhD thesis, lacks a formal definition and theoretical analysis. The authors plan to verify its details with Burch to clarify whether his algorithm is the same as PRM+ as defined in the paper.

Several limitations arise from these unexplored areas, along with potential solutions. The absence of an analysis comparing the predictive method to Gordon's framework means valuable theoretical insights may have been overlooked. To address this, future research could conduct comparative studies on Nash gaps across different games and perform proof-based analyses to uncover new bounds in regret minimization. Additionally, the unclear relationship between "optimistic RM+" and PRM+ necessitates further investigation, as valuable theoretical insights may have been overlooked as well. Collaborating with Burch or analyzing his unpublished work could provide a formal definition of "optimistic RM+," enabling a direct comparison with PRM+. If meaningful differences emerge, experiments and theoretical assessments could refine our understanding of its correctness, regret bounds, and potential for faster convergence.

3.3 Learning Not to Regret [3]

3.3.1 Strengths. Paper [3] is light on the theory but focuses heavily on the practicality. It contains minimal proofs, with the only notable one being the correctness of neural predicting, and even that is easily derived using a previous theorem from paper [2] (the PRM regret bound). The rest of the paper consists mainly of defined equations, neural network architecture, diagrams that are straightforward to follow, and an algorithm for PRM+, which is essentially the same as in the second paper, albeit with different syntax and symbols.

The paper is easiest to read compared to the other two papers, due to its detailed and comprehensive introduction to historical discoveries in regret minimization and counterfactual regret algorithms. The introduction is particularly descriptive, providing insights into meta-learning within the predictive regret framework, which enables faster convergence and general guarantees. After the introduction, the paper gives a background on the regret minimization framework, making it accessible to readers with minimal prior knowledge. The paper then seamlessly transitions into a discussion of the meta-learning framework, outlining its objectives, introducing the NOA and NRPM algorithms, evaluating their strengths and weaknesses, and ultimately establishing the correctness of NRPM in a clear and structured manner.

The experimental results are presented with clear, legible graphs, including one that tracks the trajectories of the current strategies over 128 steps, an addition not seen in paper [1] and paper [2]. The paper also evaluates several metrics that were not covered in paper [1] and paper [2], such as trajectories, policy space, steps to reach target exploitability, wall time (computational time reduction), and out-of-distribution convergence. Additionally, the paper describes experiments with alternative strategies, such as assuming only positive regrets (similar to PRM+) and using Hedge instead of RM to produce the strategy σ.

3.3.2 Weaknesses and Areas for Improvement. The experiments in the paper focus on two types of games: matrix games (e.g., rock-paper-scissors) and sequential games (e.g., river poker). However, the paper could have benefitted from testing additional architectures and parameters for the neural network. It also would have been valuable to test more games of varying sizes, beyond just the 62,000 information states of river poker, to better understand the impact of game size on exploitability and convergence.

The paper also lacks a thorough sensitivity analysis of hyperparameters, leaving open questions about how factors such as LSTM size, optimizer choice, and learning rate impact performance. NOA's generalization ability is uncertain, as it lacks theoretical guarantees and struggles with out-of-distribution tasks, suggesting potential

overfitting. The study also omits comparisons against other regret minimization methods, such as Discounted CFR and Linear CFR, limiting the scope of its evaluation. While neural networks improve convergence speed, it remains unclear whether this advantage stems from their ability to capture complex patterns or simply from tuning hyperparameters. Furthermore, despite converging in fewer steps, NOA and NPRM introduce computational overhead due to neural network inference, raising concerns about their practical feasibility in large-scale, real-time decision-making scenarios. Finally, the study focuses on abstract games without testing these methods in real-world strategic applications like finance or cybersecurity, where validation would provide stronger practical insights.

3.3.3 Proof Statements. In the third paper, all statements are supported by proof. The only notable theorem, theorem 1.13, is presented with a clear and straightforward proof that uses paper [2]'s PRM regret bound, ensuring that the logical foundation of the paper is solid.

3.4 All Papers

The graphical presentations in all three papers, as well as their full variants, are well-organized and visually clear. Each figure includes proper descriptions, labels, and legends, which facilitate easy interpretation. Furthermore, the use of vibrant and colorful elements enhances the clarity and readability of the graphs.

4 COMPARISONS

4.0.1 Advancements. Paper [1] introduces an improvement by reducing variance through a Monte Carlo rollout to estimate counterfactual values for non-sampled terminal nodes in MCCFR. This method accelerates convergence to a Nash equilibrium compared to the original MCCFR, though it still faces scalability challenges in more complex games.

Paper [2] builds on the ideas from paper [1] and incorporates predictive elements to extend regret minimization algorithms, such as PRM and PRM+. These predictive algorithms refine strategy updates by incorporating predictions about future regret, offering a more efficient way of minimizing regret.

Finally, paper [3] extends paper [2]'s ideas and introduces a meta-learning approach to regret minimization, focusing on minimizing external regret across a distribution of games. This is done by combining traditional regret minimization with deep learning, resulting in key algorithms, NOA and NRPM, which outperform earlier methods like RM and PRM by achieving near-equilibrium strategies more quickly.

While each paper builds on the previous one, they progressively shift focus from improving regret minimization techniques in large-scale game trees (in paper [1]), to incorporating predictive methods (in paper [2]), and finally leveraging deep learning for efficient convergence (in paper [3]). The first paper is primarily concerned with improving scalability in large games by reducing variance in sampling methods, while the second paper extends these ideas into predictive modeling, which significantly improves convergence rates. The third paper introduces a deep learning approach that enhances both convergence speed and computational efficiency.

4.0.2 Measures of Deviation from Nash Equilibrium. Each paper employs a different measure to quantify deviations from Nash equilibrium in their respective experiments.

Paper [1] evaluates strategy deviation using *exploitability*, defined as

$$e(\sigma) = \frac{b_1(\sigma_2) + b_2(\sigma_1)}{2}$$

where $b_1(\sigma_2)$ and $b_2(\sigma_1)$ represent the best response utilities for players 1 and 2 against the opponent's strategy. This formulation reflects the average loss against a worst-case opponent, with a Nash equilibrium having zero *exploitability*.

Paper [2] introduces the *Nash Gap*, which measures the maximum incentive any player has to deviate from their current strategy.

$$\text{Nash Gap}(\sigma) = \max\left\{ \max_{\sigma_1' \in \Sigma_1} [u_1(\sigma_1', \sigma_2) - u_1(\sigma)], \max_{\sigma_2' \in \Sigma_2} [u_2(\sigma_2', \sigma_1) - u_2(\sigma)] \right\}.$$

A Nash equilibrium has a *Nash Gap* of zero. Unlike the first paper's exploitability measure, this formulation explicitly accounts for the incentive to deviate, making it applicable to both zero-sum and non-zero-sum games.

Paper [3] presents a generalized notion of exploitability, incorporating an adversarial framework:

$$\text{expl}(\sigma) = \max_{\sigma^*} \min_{x} \langle \sigma^*, x(\sigma^*) \rangle - \min_{x} \langle \sigma, x(\sigma) \rangle.$$

$x(\sigma)$ represents the reward vector from the environment based on the strategy σ. This formulation extends the traditional exploitability concept by considering both unilateral and bilateral deviations. It can be rewritten as

$$\text{expl}(\sigma) = \max_{\sigma^*} u(\sigma^*) - u(\sigma), \tag{1}$$

where

$$u(\sigma) = \min_{x} \langle \sigma, x(\sigma) \rangle. \tag{2}$$

Compared to the first paper, this measure does not average over unilateral deviations but instead evaluates the worst-case deviation scenario directly, and unlike the second paper, it explicitly considers adversarial interactions.

4.1 Heuristics

Quadratic averaging of strategy iterates involves weighting strategies x^t quadratically over T iterations, given by $\frac{6}{T(T+1)(2T+1)} \sum_{t=1}^{T} t^2 x^t$. This approach contrasts with linear averaging, where strategies are weighted linearly, $\frac{2}{T(T+1)} \sum_{t=1}^{T} t x^t$, and standard averaging, where strategies are weighted uniformly, $\frac{1}{T} \sum_{t=1}^{T} x^t$.

Upon reviewing paper [2], a notable heuristic identified is the use of quadratic averaging for strategy iterates, which has been demonstrated to enhance practical performance. In comparison, paper [1] and paper [3] apply standard averaging in two-player zero-sum games.

Paper [2] investigates the CFR+ algorithm with quadratic averaging, yielding the following observation that incorporating quadratic averaging significantly enhances CFR+'s efficiency in poker games, particularly in the River endgame and Leduc poker, outperforming PCFR+.

A possible explanation for quadratic averaging's improved empirical performance, particularly in poker, is that it assigns greater weight to more recent strategies, thereby mitigating the influence of earlier, suboptimal strategies. Given that early iterations often involve less refined strategies, this discounting mechanism helps improve strategic adaptation over time. Its advantages are particularly evident in poker, where the game's complexity, stochastic dynamics, and need for adaptive strategy refinement make quadratic averaging a compelling alternative to standard and linear averaging techniques.

5 REFERENCES

Here are the references:

[1] Gibson, R., Lanctot, M., Burch, N., Szafron, D., & Bowling, M. (2012). Generalized sampling and variance in counterfactual regret minimization. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 26, No. 1, pp. 1355-1361).

[2] Farina, G., Kroer, C., & Sandholm, T. (2021, May). Faster game solving via predictive blackwell approachability: Connecting regret matching and mirror descent. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 6, pp. 5363-5371).

[3] Sychrovský, D., Šustr, M., Davoodi, E., Bowling, M., Lanctot, M. and Schmid, M. 2024. Learning Not to Regret. Proceedings of the AAAI Conference on Artificial Intelligence. 38, 14 (Mar. 2024), 15202-15210. DOI:https://doi.org/10.1609/aaai.v38i14.29443.

[4] Lanctot, M., Gibson, R., Burch, N., & Szafron, D. (2012). Generalized Sampling and Variance in Counterfactual Regret Minimization.

[5] Farina, G., Kroer, C., & Sandholm, T. (2020). Faster Game Solving via Predictive Blackwell Approachability: Connecting Regret Matching and Mirror Descent. arXiv preprint arXiv:2007.14358.

[6] Sychrovský, D., Šustr, M., Davoodi, E., Bowling, M., Lanctot, M., & Schmid, M. (2023). Learning not to Regret. arXiv preprint arXiv:2303.01074.

[7] Li, S. (2025). Scalable, generalizable, and offline methods for imperfect-information extensive-form games.

[8] Liu, M. (2025). On Solving Larger Games: Designing New Algorithms Adaptable to Deep Reinforcement Learning (Doctoral dissertation, Massachusetts Institute of Technology).