Alternative Function Approximation Parameterizations for Solving Games: An Analysis of f-Regression Counterfactual Regret Minimization

Ryan D'Orazio[*] [†]
rdorazio@ualberta.ca

Dustin Morrill[*] [†]
morrill@ualberta.ca

James R. Wright[*]
james.wright@ualberta.ca

Michael Bowling[*]
mbowling@ualberta.ca

ABSTRACT

Function approximation is a powerful approach for structuring large decision problems that has facilitated great achievements in the areas of reinforcement learning and game playing. Regression counterfactual regret minimization (RCFR) is a simple algorithm for approximately solving imperfect information games with normalized rectified linear unit (ReLU) parameterized policies. In contrast, the more conventional softmax parameterization is standard in the field of reinforcement learning and yields a regret bound with a better dependence on the number of actions. We derive approximation error-aware regret bounds for (Φ, f)-regret matching, which applies to a general class of link functions and regret objectives. These bounds recover a tighter bound for RCFR and provide a theoretical justification for RCFR implementations with alternative policy parameterizations (f-RCFR), including softmax. We provide exploitability bounds for f-RCFR with the polynomial and exponential link functions in zero-sum imperfect information games and examine empirically how the link function interacts with the severity of the approximation. We find that the previously studied ReLU parameterization performs better when the approximation error is small while the softmax parameterization can perform better when the approximation error is large.

CCS CONCEPTS

• **Theory of computation → Online learning algorithms**; **Regret bounds**; **Exact and approximate computation of equilibria**;

KEYWORDS

Regret minimization; Counterfactual regret minimization; Function approximation; Zero-sum games; Extensive-form games

ACM Reference Format:
Ryan D'Orazio, Dustin Morrill, James R. Wright, and Michael Bowling. 2020. Alternative Function Approximation Parameterizations for Solving Games: An Analysis of f-Regression Counterfactual Regret Minimization. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

[*]Department of Computing Science, University of Alberta. Edmonton, Alberta, Canada.
[†]Equal contributors.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

1 INTRODUCTION

The dominant framework for approximating Nash equilibria in sequential games with imperfect information is **Counterfactual Regret Minimization (CFR)**, which has successfully been used to solve and expertly play human-scale poker games [4, 6, 7, 21]. This framework is built on the idea of decomposing a game into a network of simple regret minimizers [9, 33]. Historically, large games have been abstracted to smaller but strategically similar games through a state-aggregation procedure [11, 16, 32, 33]. The abstract game is solved with CFR and the resulting strategies are translated so they apply to the original game.

Function approximation is a natural generalization of abstraction. In CFR, this amounts to estimating the regrets for each regret minimizer instead of storing them all in a table [5, 19, 22, 28, 31]. Game solving with function approximation can be competitive with domain specific state abstraction [5, 15, 22, 31], and in some cases is able to outperform tabular CFR without abstraction if the players are optimizing against their best responses [20]. Function approximation has facilitated many recent successes in game playing more broadly [24, 25, 29].

Combining regression and regret-minimization with applications to CFR was initially studied by Waugh *et al.* [31], introducing the **Regression Regret-Matching (RRM)** Theorem—giving a sufficient condition for function approximator error to still achieve no external regret. The extension to **Regression Counterfactual Regret Minimization (RCFR)** yields an algorithm that utilizes function approximation in a way similar to reinforcement learning (RL), particularly policy-based RL. Action preferences—cumulative **counterfactual regrets**—are learned and predicted, and these predictions parameterize a stochastic policy.

Conversely, some recent RL algorithms take a regret minimization approach. Regret policy gradient (RPG) [27], exploitability descent (ED) [20], Politex [1], and neural replicator dynamics (NeuRD) [23] either have regret bounds or they are inspired by tabular algorithms with regret bounds.

CFR was originally introduced using **regret matching (RM)** [14] as its component learners. This learning algorithm generates policies by normalizing positive regrets and setting the weight of actions with negative regrets to zero. This truncation of negative regrets is exactly the application of a rectified linear unit (ReLU) function, which is used extensively in the field of machine learning for constructing neural network layers. RCFR, following in CFR's lineage, had only theoretical guarantees with normalized ReLU policies.

However, most RL algorithms for discrete action spaces take a different approach: they exponentiate and normalize the preferences according to the **softmax** function. The **Hedge** or **Exponential Weights** learning algorithm [10] also uses a softmax function to generate policies. It even has a regret bound with a log dependence on the number of actions, rather than a square root dependence, as RM does. This provides us with some motivation for generalizing the RRM and RCFR theory to allow for alternative policy parameterizations.

In fact, RM and Hedge can be unified. Greenwald *et al.* [12] present **(Φ, f)-regret matching**, a general framework for constructing learners for minimizing Φ-regret—a set of regret metrics that include external regret and internal regret when using a policy parameterized by a **link function** f. Generalizing to internal regret has an important connection to correlated equilibria in general sum games [8].

In this paper, we first generalize the RRM Theorem to (Φ, f)-regret matching by extending Greenwald *et al.* framework to the case when the regret inputs to algorithms are approximate. This new approximate (Φ, f)-regret matching framework allows for the use of a broad class of link functions and regret objectives, and provides a simple recipe for generating regret bounds under new choices for both when estimating regrets. Our analysis, both due to improvements previously made by Greenwald *et al.* [12] and more careful application of conventional inequalities, tightens the bound for RRM. The corresponding improvement to the RCFR Theorem [22, 31] is magnified because the bound in this theorem is essentially the RRM bound multiplied by the size of the game. In addition, this framework provides insight into the effectiveness of combining function approximation with regret minimization as the impact of inaccuracy on the bounds vary between link functions and parameter choices.

The approximate (Φ, f)-regret matching framework provides the basis for bounds that apply to RCFR algorithms with alternative link functions, thereby allowing the sound use of alternative policy parameterizations, including softmax. We call this generalization, ***f*-RCFR**. We provide regret and equilibrium approximation bounds for this algorithm with the polynomial and exponential link functions, and we test them in two games commonly used in games research, **Leduc hold'em poker** [26] and **imperfect information goofspiel** [17]. A simple but extensible linear representation is used to isolate the effect of the link function and the degree of approximation on learning performance. We find that the polynomial link function performs better when the approximation error is small while the exponential link function (corresponding to a softmax parameterization) can perform better when the approximation error is large.

This paper is organized as follows. First, we define online decision problems and connect to relevant prior work in this area. We then define approximate regret matching and provide regret bounds for this new class of algorithms. Afterward, we begin our discussion of RCFR and our new generalization by describing extensive-form games and prior work on RCFR. Finally, we present f-RCFR along with exploitability bounds and experiments in Leduc hold'em and goofspiel.

2 ONLINE DECISION PROBLEMS

2.1 Background

We adopt the notation from Greenwald *et al.* [12] to describe an **online decision problem (ODP)**. An ODP consists of a set of possible actions A and set of possible rewards $\mathcal{R}$. In this paper we assume a finite set of actions and bounded $\mathcal{R} \subset \mathbb{R}$ where $\sup_{x \in \mathcal{R}} |x| = U$. The tuple $(A, \mathcal{R})$ fully characterizes the problem and is referred to as a reward system. Furthermore, let Π denote the set of reward functions $r : A \to \mathcal{R}$.

At each round t an agent selects a policy, that is, a distribution over actions $\sigma_t \in \Delta(A)$[1]. The agent samples an action, $a_t \sim \sigma_t$, and subsequently receives a reward function, $r_t \in \Pi$. The agent is able to compute the rewards for actions that were not taken at time t, in contrast to the bandit setting where the agent only observes $r_t(a_t)$.

Crucially, each r_t may be selected arbitrarily from Π. As a consequence, this ODP model is flexible enough to encompass multi-agent, adversarial interactions, and game theoretic equilibrium concepts even though it is described from the perspective of a single agent's decisions.

A learning algorithm in an ODP selects σ_t using information from the history of observations and actions previously taken. We denote this information at time t as history $h \in H_t \doteq A^t \times \Pi^t$, where $H_0 \doteq \{\emptyset\}$. Formally, an online learning algorithm is a sequence of functions $\{L_t\}_{t=1}^{\infty}$, where $L_t : H_{t-1} \to \Delta(A)$.

We denote the **rectified linear unit (ReLU)** function as $x^+ = \max\{x, 0\}$, for all $x \in \mathbb{R}$. Similarly for vectors $x \in \mathbb{R}^N$ we define x^+ to be the componentwise application of the ReLU function.

2.1.1 Action Transformations. To generalize the analysis to different performance metrics, it is useful to define **action transformations**. Action transformations are functions of the form $\phi : A \to \Delta(A)$, mapping each action $a \in A$ to a policy. Let Φ_{ALL} denote the set of all action transformations for the set of actions A. Two important subsets of Φ_{ALL} are the external and internal transformations.

External transformations, Φ_{EXT}, transform all actions to the same action. Formally, if $\delta_a \in \Delta(A)$ is the distribution with full weight on action a, then $\Phi_{EXT} \doteq \{\phi : \exists a \in A \, \forall x \in A \quad \phi(x) = \delta_a\}$. Note that there are $|\Phi_{EXT}| = |A|$-external transformations.

Internal transformations, Φ_{INT}, transform one action to another action. Formally, the internal transformation from action a to action b is defined piecewise as $\phi_{INT}^{(a,b)}(a) = \delta_b$ and $\phi_{INT}^{(a,b)}(x) = \delta_x$ otherwise. Note that there are $|\Phi_{INT}| = |A|^2 - |A| + 1$-internal transformations [12].

We define the policy induced by distribution σ and action transformation ϕ as $[\phi](\sigma) = \sum_{a \in A} \sigma(a)\phi(a)$.

2.1.2 Regret. The regret for not following action transformation ϕ when action a was chosen and reward function r was observed is **ϕ-regret**, $\rho^{\phi}(a, r) = \mathbb{E}_{a' \sim \phi(a)}[r(a')] - r(a)$. For a set of action transformations, Φ, the Φ-regret vector is $\rho^{\Phi}(a, r) = (\rho^{\phi}(a, r))_{\phi \in \Phi}$. The expected ϕ-regret for policy $\sigma \in \Delta(A)$ is $\mathbb{E}_{a \sim \sigma}[\rho^{\phi}(a, r)]$.

For an ODP with observed history h at time t, composed of reward functions $\{r_s\}_{s=1}^{t}$ and actions $\{a_s\}_{s=1}^{t}$ selected by the agent on each round, the cumulative Φ-regret after t-rounds against action

[1] $\Delta(A)$ is the set of all probability distributions over actions in A.

transformations Φ is $R_t^\Phi(h) = \sum_{k=1}^t \rho^\Phi(a_k, r_k)$. For brevity we will omit the h argument, and for convenience we set $R_0^\Phi \doteq 0$.

We seek to bound the expected average maximum Φ-regret, $\mathbb{E}[\frac{1}{t}\max_{\phi\in\Phi} R_t^\phi]$. Choosing Φ to be Φ_{EXT} or Φ_{INT} corresponds to the well studied maximum external regret or maximum internal regret objectives, respectively.

One can also interchange the max and the expectation. In RRM, $\max_{\phi\in\Phi_{EXT}} \mathbb{E}[\frac{1}{t}R_t^\phi]$ is bounded [22, 31]. However, bounds for $\mathbb{E}[\frac{1}{t}\max_{\phi\in\Phi} R_t^\phi]$ imply similar bounds when the expected regret, $\mathbb{E}[\rho_t^\Phi]$, is observed after each round [12, Corollary 18]. The bounds remain the same with the exception of replacing the observed random regrets with their corresponding expected values.

2.2 Approximate Regret-Matching

Given a set of action transformations Φ and a link function $f : \mathbb{R}^{|\Phi|} \to \mathbb{R}_+^{|\Phi|}$, where $\mathbb{R}_+^N$ denotes the N-dimensional positive orthant, we can define a general class of online learning algorithms known as **(Φ, f)-regret-matching** algorithms [12]. A (Φ, f)-regret-matching algorithm at time t chooses $\sigma \in \Delta(A)$ that is a fixed point of

$$M_t(\sigma) \doteq \frac{\sum_{\phi\in\Phi} Y_t^\phi[\phi](\sigma)}{\sum_{\phi\in\Phi} Y_t^\phi},$$

when $R_{t-1}^\Phi \in \mathbb{R}_+^{|\Phi|} \setminus \{0\}$, where $Y_t^\Phi \doteq (Y_t^\phi)_{\phi\in\Phi} \doteq f(R_{t-1}^\Phi)$, and arbitrarily otherwise. Note that M_t is a convex combination of linear operators $\{[\phi]\}_{\phi\in\Phi}$, hence the fixed point always exists by the Brouwer Fixed Point Theorem. If $\Phi = \Phi_{EXT}$ then the fixed point of M_t is a distribution $\sigma \propto Y_t^\Phi$ [13]. Examples of (Φ, f)-regret-matching algorithms include Hart's algorithm [14]—typically called "regret-matching"—and Hedge [10], with link functions $f(x)_i = x_i^+$ and $f(x)_i = e^{\frac{1}{\tau}x_i}$ with temperature parameter $\tau > 0$, respectively.

A useful technique for bounding regret when estimates are used in place of true values is to define an ϵ−Blackwell condition, as was used in the RRM Theorem [31]. The analysis in RRM was specific to $\Phi = \Phi_{EXT}$ and the polynomial link f with $p = 2$. To generalize across different link functions and $\Phi \subseteq \Phi_{ALL}$ we define the (Φ, f, ϵ)-Blackwell condition.

Definition 2.1 ((Φ, f, ϵ)-Blackwell Condition). For a given reward system $(A, \mathcal{R})$, finite set of action transformations $\Phi \subseteq \Phi_{ALL}$, and link function $f : \mathbb{R}^{|\Phi|} \to \mathbb{R}_+^{|\Phi|}$, a learning algorithm satisfies the (Φ, f, ϵ)-Blackwell condition if $f(R_{t-1}^\Phi(h)) \cdot \mathbb{E}_{a\sim L_t(h)}[\rho^\Phi(a, r)] \le \epsilon$.

The Regret Matching Theorem [12] shows that the (Φ, f)-Blackwell condition ($\epsilon = 0$) holds with equality for (Φ, f)-regret-matching algorithms.

We seek to bound the expected average Φ-regret when an algorithm at time t chooses the fixed point of $\tilde{M}_t \doteq \sum_{\phi\in\Phi}\tilde{Y}_t^\phi[\phi]/\sum_{\phi\in\Phi}\tilde{Y}_t^\phi$, when $\tilde{R}_{t-1}^\Phi \in \mathbb{R}_+^{|\Phi|} \setminus \{0\}$ and arbitrarily otherwise, where $\tilde{Y}_t^\Phi \doteq f(\tilde{R}_{t-1}^\Phi)$ and $\tilde{R}_{t-1}^\Phi$ is an estimate of R_{t-1}^Φ, possibly from a function approximator. Such an algorithm is referred to as **approximate (Φ, f)-regret-matching**.

Similarly to the RRM Theorem [22, 31], we show that the ϵ parameter of the (Φ, f, ϵ)-Blackwell condition depends on the link output approximation error, $\left\|Y_t^\Phi - \tilde{Y}_t^\Phi\right\|_1$.

Theorem 2.2. *Given reward system $(A, \mathcal{R})$, a finite set of action transformations $\Phi \subseteq \Phi_{ALL}$, and link function $f : \mathbb{R}^{|\Phi|} \to \mathbb{R}_+^{|\Phi|}$, then an approximate (Φ, f)-regret-matching algorithm, $\{L_t\}_{t=1}^\infty$, satisfies the (Φ, f, ϵ)-Blackwell Condition with $\epsilon \le 2U\left\|Y_t^\Phi - \tilde{Y}_t^\Phi\right\|_1$, where $Y_t^\Phi \doteq f(R_{t-1}^\Phi)$, and $\tilde{Y}_t^\Phi \doteq f(\tilde{R}_{t-1}^\Phi)$.*

Omitted proofs are deferred to the appendix.

For a (Φ, f)-regret-matching algorithm, an approach to bound the expected average Φ-regret is to use the (Φ, f)-Blackwell condition along with a bound on $\mathbb{E}[G(R_t^\Phi)]$ for an appropriate function G [8, 12]. Bounding the regret for an approximate (Φ, f)-regret-matching algorithm will be done similarly, except the bound on ϵ from Theorem 2.2 will be used. Proceeding in this fashion yields the following theorem:

Theorem 2.3. *Given a real-valued reward system $(A, \mathcal{R})$ a finite set $\Phi \subseteq \Phi_{ALL}$ of action transformations. If $\langle G, g, \gamma\rangle$ is a Gordon triple[2], then an approximate (Φ, g)-regret-matching algorithm $\{L_t\}_{t=1}^\infty$ guarantees at all times $t \ge 0$*

$$\mathbb{E}[G(R_t^\Phi)] \le G(0) + t \sup_{a\in A, r\in\Pi} \gamma(\rho^\Phi(a, r)) + 2U\sum_{s=1}^t \left\|g(R_{s-1}^\Phi) - g(\tilde{R}_{s-1}^\Phi)\right\|_1.$$

2.3 Bounds for Specific Link Functions

2.3.1 Polynomial. Given the polynomial link function $f(x)_i = (x_i^+)^{p-1}$ we consider two cases $2 < p < \infty$ and $1 < p \le 2$. For the following results it is useful to denote the maximal activation $\mu(\Phi) = \max_{a\in A}|\{\phi \in \Phi : \phi(a) \ne \delta_a\}|$ [12].

For the case $p > 2$ we have the following bound on the expected average Φ-regret:

Theorem 2.4. *Given an ODP, a finite set of action transformations $\Phi \subseteq \Phi_{ALL}$, and the polynomial link function f with $p > 2$, then an approximate (Φ, f)- regret-matching algorithm guarantees*

$$\mathbb{E}\left[\max_{\phi\in\Phi}\frac{1}{t}R_t^\phi\right] \le$$
$$\frac{1}{t}\sqrt{t(p-1)4U^2(\mu(\Phi))^{2/p} + 2U\sum_{k=1}^t \left\|g(R_{k-1}^\Phi) - g(\tilde{R}_{k-1}^\Phi)\right\|_1},$$

where $g : \mathbb{R}^{|\Phi|} \to \mathbb{R}_+^{|\Phi|}$ and $g(x)_i = 0$ if $x_i \le 0$, $g(x)_i = \frac{2(x_i)^{p-1}}{\|x^+\|_p^{p-2}}$ otherwise.

Similarly for the case $1 < p \le 2$ we have the following.

Theorem 2.5. *Given an ODP, a finite set of action transformations $\Phi \subseteq \Phi_{ALL}$, and the polynomial link function f with $1 < p \le 2$, then an approximate (Φ, f)- regret-matching algorithm guarantees*

$$\mathbb{E}\left[\max_{\phi\in\Phi}\frac{1}{t}R_t^\phi\right] \le \frac{1}{t}\left(t(2U)^p\mu(\Phi) + 2U\sum_{k=1}^t \left\|g(R_{k-1}^\Phi) - g(\tilde{R}_{k-1}^\Phi)\right\|_1\right)^{1/p}$$

where $g : \mathbb{R}^{|\Phi|} \to \mathbb{R}_+^{|\Phi|}$ and $g(x)_i = p(x_i^+)^{p-1}$.

In comparison to the RRM Theorem [22], the above bound is tighter as there is no $\sqrt{|A|}$ term in front of the errors and the $|A|$ term has been replaced by[3] $|A|-1$. These improvements are due to the

[2] A Gordon triple $\langle G, g, \gamma\rangle$ consists of three functions $G : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^n$, and $\gamma : \mathbb{R}^n \to \mathbb{R}$ such that for all $x, y \in \mathbb{R}^n$, $G(x + y) \le G(x) + g(x) \cdot y + \gamma(y)$.

[3] For $\Phi = \Phi_{EXT}$, $\mu(\Phi) = |A|-1$.

tighter bound in Theorem 2.2 and the original Φ-regret analysis [12], respectively. Aside from these differences, the bounds coincide.

2.3.2 Exponential.

THEOREM 2.6. *Given an ODP, a finite set of action transformations $\Phi \subseteq \Phi_{ALL}$, and an exponential link function $f(x)_i = e^{\frac{1}{\tau}x_i}$ with $\tau > 0$, then an approximate (Φ, f)- regret-matching algorithm guarantees*

$$\mathbb{E}\left[\max_{\phi\in\Phi}\frac{1}{t}R_t^{\phi}\right] \leq \frac{1}{t}\left(\tau\ln|\Phi| + 2U\sum_{k=1}^{t}\left\|g(R_{k-1}^{\Phi}) - g(\tilde{R}_{k-1}^{\Phi})\right\|_1\right) + \frac{2U^2}{\tau}$$

where $g : \mathbb{R}^{|\Phi|} \to \mathbb{R}_+^{|\Phi|}$ and $g(x)_i = e^{\frac{1}{\tau}x_i}/\sum_j e^{\frac{1}{\tau}x_j}$.

The Hedge algorithm corresponds to the exponential link function $f(x)_i = e^{\frac{1}{\tau}x_i}$ when $\Phi = \Phi_{EXT}$, so Theorem 2.6 provides a bound on a regression Hedge algorithm. Note that in this case, the approximation error term is not inside a root function as it is under the polynomial link function. This seems to imply that at the level of link outputs, polynomial link functions have a better dependence on the approximation errors. However, g in the exponential link function bound is normalized to the simplex while the polynomial link functions can take on larger values. So which link function has a better dependence on the approximation errors depends on the magnitude of the cumulative regrets, which depends on the environment and the algorithm's empirical performance.

3 EXTENSIVE-FORM GAMES

3.1 Background

A **zero-sum extensive-form game (EFG)** is a tuple

$$(\mathcal{H}, \mathcal{A}, A, p, \sigma_c, \mathcal{S}, r_1).$$

$\mathcal{H}$ is the set of valid action sequences and chance outcomes called **histories**, where an action is an element of $\mathcal{A}$, and the set of actions available at each history is determined by $A : \mathcal{H} \to \mathcal{A}$. The player to act (including the chance "player", c) at each non-terminal history is determined by $p : \mathcal{H} \setminus \mathcal{Z} \to \{1, 2, c\}$, where terminal histories are those with no valid actions, $\mathcal{Z} \doteq \{h | h \in \mathcal{H}, A(h) = \emptyset\}$. σ_c is a fixed stochastic policy assigned to the chance player that determines the likelihood of random outcomes, like those from die rolls or draws from a shuffled deck of cards. $\mathcal{S} \doteq \mathcal{S}_1 \cup \mathcal{S}_2$ is the information partition and it describes which histories players can distinguish between. The set of histories where player $i \in \{1, 2\}$ acts, $\mathcal{H}_i \doteq \{h | p(h) = i\}$, are partitioned into a set of **information states**, $\mathcal{S}_i$, where for each information state $s \in \mathcal{S}_i$, $s \subseteq \mathcal{H}_i$, is a set of histories indistinguishable to i. Since $A(h) = A(h')$ if $h, h' \in s \in \mathcal{S}$, we can denote the actions at s as $A(s)$. We require **perfect recall** so that for all histories in an information state, the sequence of information states admitted by the preceding histories must be identical. $r_1 : \mathcal{Z} \to \mathbb{R}$ is a **reward** or **utility** function for player 1. The game is zero sum because player 2's utility function $r_2 \doteq -r_1$.

Player i's **policy** or **behavioral strategy**, $\sigma_i \in \Sigma_i$ defines a probability distribution over valid actions at each of i's information states, and a **joint policy** or **strategy profile** is an assignment of policies for each player, $\sigma \doteq (\sigma_1, \sigma_2)$. We use $\eta^{\sigma}(z)$ to denote the probability of reaching terminal history $z \in \mathcal{Z}$ under profile σ from the beginning of the game and $\eta^{\sigma}(h, z)$ the same except starting from history $h \in \mathcal{H}$. We subscript η by the player to denote that player's contribution to these probabilities $\eta^{\sigma}(z) = \eta_i^{\sigma}(z)\eta_{-i}^{\sigma}(z)$. The expected value to player i under profile σ is $r_i(\sigma) = r_i(\sigma_1, \sigma_2) = \sum_{z\in\mathcal{Z}} \eta^{\sigma}(z) r_i(z)$.

A **best response** for player i to another player's strategy, σ_{-i}, is a policy that achieves the maximum reward against σ_{-i}, $r_i^*(\sigma_{-i}) = \max_{\sigma_i\in\Sigma_i} r_i((\sigma_i, \sigma_{-i}))$. A profile, σ, is an **ε-Nash equilibrium** if neither player can unilaterally deviate from their assigned policy and gain more than ε. That is, if $r_i(\sigma) + \varepsilon_i \geq r_i^*(\sigma_{-i})$, for each player $i \in \{1, 2\}$, then σ is a $\max\{\varepsilon_1, \varepsilon_2\}$-equilibrium, and the smallest approximation error is achieved when $r_i(\sigma)+\varepsilon_i = r_i^*(\sigma_{-i})$. Therefore, in a zero-sum game, all strategies that are part of ε-Nash equilibria are at most ε-utility away from being minimax optimal.

Since the game is zero-sum, the average of best response values is equal to $(\varepsilon_1+\varepsilon_2)/2$. This is the **exploitability** of the profile σ. We use profile exploitability to measure equilibrium approximation error.

The exploitability of a profile σ is related to $R^{\Phi_{EXT}}$ (abbreviated to R^{EXT}) in a fundamental way. First consider the induced normal form of an EFG, where actions taken by a player consists of specifying an action at each information state. That is, from an ODP perspective, the set of actions available to player i (the learning algorithm) is $\tilde{A} = \prod_{s\in\mathcal{S}_i} A(s)$. We can then define the expected regret at time t for player i with respect to action $a' \in \tilde{A}$ when selecting a policy $\sigma \in \Delta(\tilde{A})$ as the difference

$$\rho_{i,a'}^{t} \doteq \mathbb{E}_{a\sim\sigma}\left[r_i(a', \sigma_{-i}) - r(a, \sigma_{-i})\right].$$

We can then define the cumulative external regret of player i at time t as $R_{i,t}^{\mathrm{EXT}} \doteq \max_{a^*\in\tilde{A}} \sum_{k=1}^{t} \rho_{i,a^*}^{k}$. Note that this is an instance of an ODP problem where the sequence of reward functions for player i is induced by the opponent's sequence of policies. Furthermore, the external regret defined here is with respect to player i's expected reward (*i.e.*, interchanging the expectation and maximum in the previously described Φ-regret objective). The connection between $R_{i,t}^{\mathrm{EXT}}$ and Nash equilibria then follows from the well-known folk theorem.

THEOREM 3.1. *If two ODPs are enmeshed so that the rewards of the learners always sum to zero and the action of one learner influences the reward function of the other, then they represent a repeated zero-sum game. If neither learner, $i \in \{1, 2\}$, suffers more than ε_i external regret after t-rounds, $\frac{1}{t}R_{i,t}^{EXT} \leq \varepsilon_i$, then the profile formed from their average policies, $\bar{\sigma}_{i,t} = \frac{1}{t}\sum_{k=1}^{t} \sigma_{i,k}$, is an $(\varepsilon_1 + \varepsilon_2)$-Nash equilibrium.*

See, for example, Blum and Mansour [3] for a proof.

3.2 Counterfactual Regret Minimization

The idea of **counterfactual regret minimization (CFR)** [33] is that we can decompose an EFG into multiple ODPs, one at each information state. We define the reward for action $a \in A(s)$ in the ODP at $s \in \mathcal{S}_i$ as the **counterfactual value** of playing a, which is the expected value of playing a assuming that player i plays to reach s. Formally,

$$v_i^{\sigma}(s, a) = \sum_{h\in s, z\in\mathcal{Z}} \eta_i^{\sigma}(ha, z)\eta_{-i}^{\sigma}(z) r_i(z),$$

where $ha \in \mathcal{H}$ is the history that results from taking action a at history h, and $\eta_i^{\sigma}(h, z) = 0$ whenever z is unreachable from ha.

Accordingly, the regret, also referred to as instantaneous regret, of the ODP learner at $s \in \mathcal{S}_i$ for not committing to $a \in A(s)$ is

$$\rho_i^{\sigma}(s,a) = v_i^{\sigma}(s,a) - \sum_{a' \in A(s)} \sigma_i(s,a') v_i^{\sigma}(s,a').$$

We denote the cumulative counterfactual regret of an information state s and action a as $R_{i,t}(s,a) = \sum_{k=1}^{t} \rho_i^{\sigma^k}(s,a)$, where we denote the profile at time k as $\sigma^k \doteq (\sigma_{1,k}, \sigma_{2,k})$, and that of s alone as $R_{i,t}(s) = \max_{a \in A(s)} R_{i,t}(s,a)$.

Zinkevich *et al.* [33] showed:

Theorem 3.2 (CFR). *For both players, $i \in \{1,2\}$, the regret of i's policies constructed from their ODP learners after t iterations of CFR is $\frac{1}{t} R_{i,t}^{EXT} \leq \varepsilon_{i,t}$ where $\varepsilon_{i,t} = \frac{1}{t}\sum_{s \in \mathcal{S}_i} \left(R_{i,t}(s)\right)^+$. Furthermore, the profile of average sequence weight policies, $\bar{\sigma}^t \doteq (\bar{\sigma}_{1,t}, \bar{\sigma}_{2,t})$, is an $(\varepsilon_{1,t} + \varepsilon_{2,t})$-Nash equilibrium, where*

$$\bar{\sigma}_{i,t}(s) \propto \sum_{k=1}^{t} \sum_{h \in s, a \in A(h)} \eta_i^{\sigma^k}(h) \sigma_{i,k}(s,a).$$

See Farina *et al.* [9] for the sketch of an alternative proof using the regret circuits framework that is perhaps more intuitive than the proof in the original work.

3.3 f-RCFR

Games that humans are interested in playing, or those that model problems of practical importance, typically have an immense number of information states or actions. But such games often contain structure that can be recovered by endowing information state-action pairs (sequences) with a **feature representation**, $\varphi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d, d > 0$. A function approximator, $y : \mathbb{R}^d \to \mathbb{R}$, could then make use of shared properties between sequences to allow more efficient learning. RCFR [31] uses a function approximator to predict cumulative counterfactual regrets at each information state and generates policies with a normalized ReLU transformation.

Thanks to our new analysis of approximate regret matching, we now know that any link function that admits a no-Φ_{EXT}-regret regret matching algorithm also has an approximate version. Rather than restricting ourselves to the polynomial link function with parameter $p = 2$, we can consider alternate parameter choices or alternative link functions, like the exponential function. So instead of a normalized ReLU policy, we employ a policy generated by the external regret fixed point of link function $f : \mathbb{R}^{|\mathcal{A}|} \to \mathbb{R}_+^{|\mathcal{A}|}$ with respect to approximate regrets predicted by a functional regret estimator, $\tilde{R}(s) = \left(y\left(\varphi(s,a)\right)\right)_{a \in A(s)}$, for all $s \in \mathcal{S}$. More formally, the f-RCFR policy for player i given functional regret estimator $\tilde{R}$ is $\sigma(s) \propto f(\tilde{R}(s))$ when $\tilde{R}(s) \in \mathbb{R}_+^{|A(s)|} \setminus \{0\}$ and arbitrarily otherwise, for all $s \in \mathcal{S}_i$. Since the input to any link function in an approximate regret matching algorithm is simply an estimate of the counterfactual regret, we can reuse all of the techniques previously developed for RCFR-like methods to train regret estimators [5, 19, 22, 28, 31].

Using Theorem 2.3 and the CFR Theorem 3.2, we can derive an improved regret bound with the polynomial link and a new bound with the exponential link.

Corollary 3.3 (polynomial ($p > 2$)). *Given the polynomial link function f with $p > 2$, let $\sigma_{i,k}(s) \propto f(\tilde{R}_k(s))$ be the policy that f-RCFR assigns to player i at iteration k in information state $s \in \mathcal{S}_i$ and denote the cumulative approximation error in s as $\epsilon_i(s) = \sum_{k=1}^{t} \left\| g(R_{k-1}(s)) - g\left(\tilde{R}_{k-1}(s)\right)\right\|_1$, where $g : \mathbb{R}^{|A(s)|} \to \mathbb{R}_+^{|A(s)|}$ and $g(x)_i = 0$ if $x_i \leq 0$, $g(x)_i = \frac{2(x_i)^{p-1}}{\|x^+\|_p^{p-2}}$ otherwise. Then after t-iterations, f-RCFR guarantees, for both players, $i \in \{1,2\}$, $\frac{1}{t}R_{i,t}^{EXT} \leq \varepsilon_{i,t}$, where*

$$\varepsilon_{i,t} = \frac{1}{t} \sum_{s \in \mathcal{S}_i} \sqrt{t(p-1)4U^2(|A(s)|-1)^{2/p} + 2U\epsilon_i(s)}.$$

Noticing that $|A(s)| \leq |\mathcal{A}|$ and letting $\epsilon_i^* = \max_{s \in \mathcal{S}_i} \epsilon_i(s)$, we have

$$\varepsilon_{i,t} \leq \frac{1}{t}|\mathcal{S}_i| \sqrt{t(p-1)4U^2(|\mathcal{A}|-1)^{2/p} + 2U\epsilon_i^*}.$$

Furthermore, the profile of average sequence weight policies, $\bar{\sigma}^t$, is an $(\varepsilon_{1,t} + \varepsilon_{2,t})$-Nash equilibrium.

Proof. This result follows directly from Theorem 3.2. The counterfactual regret, $R_{i,t}(s)$, at each information state corresponds to Φ_{EXT} regret for an online ODP with $\mu(\Phi_{EXT}) = |A(s)| - 1$. Therefore, playing an approximate (Φ_{EXT}, f)-regret matching algorithm at each state with a polynomial link funciton with $p > 2$ results in the regret bound presented in Theorem 2.4 for each state specific ODP. Although Theorem 2.4 is stated with respect to random regrets and counterfactual regret is an expected regret, the analysis of Greenwald *et al.* [12, Corollary 18] allows us to trivially extend our bounds from Section 2.3 to this case. The result then follows trivially from Theorem 3.2. □

The proofs for the polynomial link with $p \leq 2$ and the exponential link are very similar and omitted for brevity.

Corollary 3.4 (polynomial ($p \leq 2$)). *Given the polynomial link function f with $p \leq 2$, let $\sigma_{i,k}(s) \propto f(\tilde{R}_k(s))$ be the policy that f-RCFR assigns to player i at iteration k in information state $s \in \mathcal{S}_i$ and denote the cumulative approximation error in s as $\epsilon_i(s) = \sum_{k=1}^{t} \left\| g(R_{k-1}(s)) - g\left(\tilde{R}_{k-1}(s)\right)\right\|_1$, where $g : \mathbb{R}^N \to \mathbb{R}_+^N$, and $g(x)_i = p(x_i^+)^{p-1}$. Then after t-iterations, f-RCFR guarantees, for both players, $i \in \{1,2\}$, $\frac{1}{t}R_{i,t}^{EXT} \leq \varepsilon_{i,t}$, where*

$$\varepsilon_{i,t} = \frac{1}{t} \sum_{s \in \mathcal{S}_i} \left(t(2U)^p(|A(s)|-1) + 2U\epsilon_i(s)\right)^{1/p}.$$

Noticing that $|A(s)| \leq |\mathcal{A}|$ and letting $\epsilon_i^* = \max_{s \in \mathcal{S}_i} \epsilon_i(s)$, we have

$$\varepsilon_{i,t} \leq \frac{1}{t}|\mathcal{S}_i| \left(t(2U)^p(|\mathcal{A}|-1) + 2U\epsilon_i^*\right)^{1/p}.$$

Furthermore, the profile of average sequence weight policies, $\bar{\sigma}^t$, is an $(\varepsilon_{1,t} + \varepsilon_{2,t})$-Nash equilibrium.

The above theorem provides a tighter bound for RCFR ($p = 2$) than what exists in the literature. The improvement is a direct consequence of the tighter bound for RRM presented in Theorem 2.5 in Section 2.3. Given the application of the RRM Theorem by Brown *et al.* [5], these results should lead to a tighter bound when a function approximator learns from sampled counterfactual regret targets.

Corollary 3.5 (exponential). *Given the exponential link function f with $\tau > 0$, let $\sigma_{i,k} \propto f(\tilde{R}_k(s))$ be the policy that f-RCFR assigns to player i at iteration k given functional regret estimator $y_k : \mathbb{R}^d \to \mathbb{R}$ and denote the cumulative approximation error in s*

as $\epsilon_i(s) = \sum_{k=1}^{t} \left\| g(R_{k-1}(s)) - g\left(\tilde{R}_{k-1}(s)\right) \right\|_1$, *where* $g : \mathbb{R}^N \to \mathbb{R}_+^N$, *and* $g(x)_i = e^{\frac{1}{\tau} x_i} / \sum_j e^{\frac{1}{\tau} x_j}$. *Then after t-iterations, f-RCFR guarantees, for both players,* $i \in \{1, 2\}$, $\frac{1}{t} R_{i,t}^{EXT} \leq \varepsilon_{i,t}$, *where*

$$\varepsilon_{i,t} = \frac{1}{t} \sum_{s \in \mathcal{S}_i} \left(\frac{1}{t} \left(\tau \ln |A(s)| + 2U\epsilon_i(s)\right) + \frac{2U^2}{\tau} \right).$$

Noticing that $|A(s)| \leq |\mathcal{A}|$ *and letting* $\epsilon_i^* = \max_{s \in \mathcal{S}_i} \epsilon_i(s)$, *we have*

$$\varepsilon_{i,t} \leq \left(\frac{1}{t} \left(\tau \ln |\mathcal{A}| + 2U\epsilon_i^*\right) + \frac{2U^2}{\tau} \right).$$

Furthermore, the profile of average sequence weight policies, $\bar{\sigma}^t$, *is an* $(\varepsilon_{1,t} + \varepsilon_{2,t})$*-Nash equilibrium.*

This bound shares the same advantage with respect to the action set size dependence over the polynomial RCFR bounds as the bound of Theorem 2.6 has over the bounds of Theorems 2.5 and 3.4.

With the exponential link function, f-RCFR is approximately Hedge applied to each information state with function approximation. To make a connection with the field of reinforcement learning, we can compare f-RCFR with two recently developed algorithms that also generalize Hedge to sequential decision problems and utilize function approximation: Politex [1] and neural replicator dynamics (NeuRD) [23].

In contrast to f-RCFR, Politex trains models to predict cumulative action values. An action value is proportional to a counterfactual value where the constant depends on the policies of the other players and chance [27, 33]. If Politex instead trains on counterfactual regrets, then we arrive at an f-RCFR instance with a softmax parameterization and a regret estimator updated in a two-step process: construct an instantaneous regret estimator and combine it with the previous estimator to predict cumulative regrets. In fact, our implementation of f-RCFR for the experiments that follow uses the same two-step update procedure.

Instead of training a model of instantaneous regrets, NeuRD performs a gradient descent step on the squared loss between the current policy logits and a target constructed by adding the logits to the instantaneous regret after each iteration. We can see this as a "bootstrap" regret target, as described by Morrill [22], where the policy logits are approximate. NeuRD is therefore an instance of f-RCFR with a softmax parameterization and a regret estimator trained on bootstrap regret targets.

4 EXPERIMENTS

To examine the impact of the link function, choices for their parameters, and the interaction between link function and function approximation, we test f-RCFR in two games commonly used as research testbeds, Leduc hold'em poker [26] and imperfect information goofspiel [17] with linear function approximation.

4.1 Algorithm Implementation

Our regret estimators are independent linear function approximators for each player, $i \in \{1, 2\}$, and action $a \in \bigcup_{s \in \mathcal{S}_i} A(s)$. Our features are built on tug-of-war hashing features [2].

We randomly partition the information states that share the same action into m-buckets and repeat this n-times to generate n-sparse indicator features of length m. The sign of each feature is randomly flipped to -1 independently to reduce bias introduced by collisions. The expected sign associated with all other information states that share a non-zero entry in their feature vector is, by design, zero. We use the number of partitions, n, to control the severity of approximation in our experiments.

We do ridge regression on counterfactual regret targets to train our regret estimators. After the first iteration, we simply add this new vector of weights to our previous weights. Since the counterfactual regrets are computed for each information state-action sequence on every iteration, the same feature matrix is used during training after each iteration. Therefore, the ridge regression solution is a linear function of the targets and the sum of the optimal weights for predicting counterfactual regret yields the ridge regression solution weights for predicting the sum. Beyond training the weights at the end of each iteration, the regrets do not need to be saved or reprocessed.

Since we are most interested in comparing the performance of f-RCFR with different link functions and parameters, we track the average policies for each instance exactly in a table. While this is less practical than other approaches, such as learning the average policies from data, it removes another variable from the analysis and allows us to examine the impact of different link functions in relative isolation. Equivalently, we could have saved copies of the regret estimator weights across all iterations and computed the average policy on demand, similarly to Steinberger [28].

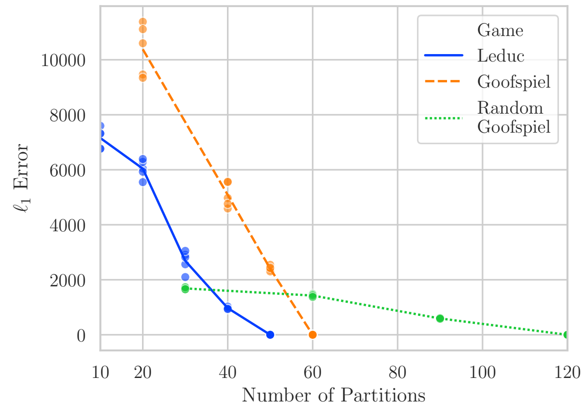

Figure 1: The cumulative counterfactual regret estimation error accumulated over time and information states for select f-RCFR instances in Leduc hold'em poker, goofspiel, and random goofspiel. For each game and setting of the number of partitions, we select the link function and the parameter with the smallest average exploitability over 5-runs at 100K-iterations. The solid lines connect the average error across iterations and dots show the errors of individual runs.

4.2 Games

In Leduc hold'em poker [26], the deck consists of 6 cards, two suits each with 3 ranks (*e.g.*, king, queen, and ace), and played with two players. At the start of the game each player antes 1 chip and receives one private card. Betting is restricted to two rounds with a maximum of two raises each round, and bets are limited to 2 and 4

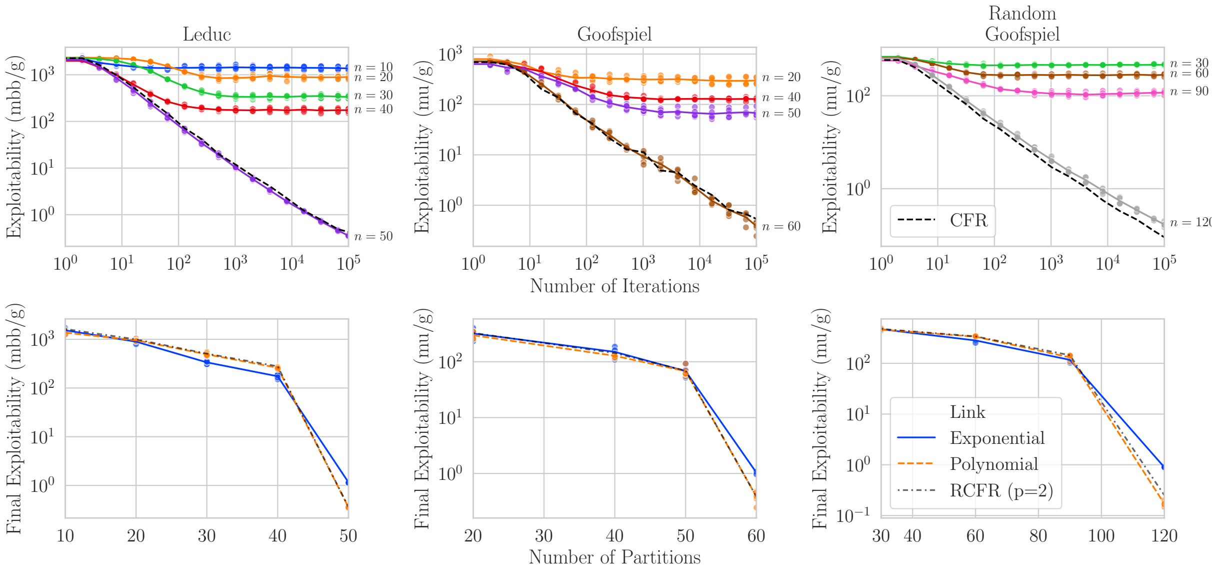

Figure 2: (top) The exploitability of the average strategy profile of tabular CFR and f-RCFR instances during the first 100K-iterations in Leduc hold'em (top left), goofspiel (top center), and random goofspiel (top right). For each setting of the number of partitions, we show the performance of the f-RCFR instance with the link function and parameter that achieves the lowest average final exploitability over 5-runs. The mean exploitability and the individual runs are plotted for the chosen instances as lines and dots respectively. (bottom) The final average exploitability after 100K-iterations for the best exponential and polynomial link function instances in Leduc hold'em (left), goofspiel (center), and random goofspiel (right).

chips. Before the second round of betting a public card is revealed from the deck. Provided no one folds, the player with a private card matching the public card wins, if no players match, the winnings go to the player with the private card of highest rank. This game has 936 states.

Goofspiel is played with two players and a deck with three suits. Each suit consists of N cards of different rank. Two of the suits form the hands of the players. The third is used as a deck of point cards. At each round a card is revealed from the point deck and players simultaneously bid by playing a card from their hand. The player with the highest bid (*i.e.* highest rank) receives points equal to the rank of the revealed card. The player with the most points when the point deck runs out is the winner and receives a utility of +1. The loser receives a utility of -1. We use an imperfect information variant of goofspiel where the bidding cards are not revealed [17]. We use two variants of goofspiel: one with a shuffled point deck and four ranks that we call "random goofspiel" and a second with a sorted point deck in decreasing order but five ranks that we call "goofspiel". Goofspiel is roughly twice as large as Leduc hold'em at 2124-information states, while random goofspiel is larger still at 3608-information states.

Our experiments use the *OpenSpiel* [18] implementations of these games.

Convergence to a Nash equilibrium in each game is measured by the exploitability of the average strategy profile after each f-RCFR iteration. Exploitability in Leduc hold'em is measured in milli-big blinds. Exploitability in goofspiel and random goofspiel is measured in milli-utils.

4.3 Parameters

From Theorems 3.2 and 3.1, any network of external regret minimizers (one at each information state) can be combined to produce an average strategy profile with bounded exploitability. Therefore, the bounds presented in Section 2.3 provide an exploitability bound for f-RCFR algorithms where f is a polynomial or exponential link function, and estimates of counterfactual regrets are used at each information state in place of true values (Corollaries 3.3, 3.4, and 3.5).

Most notably, the appearance of function approximator error within the regret bounds in Section 2.3 appear in different forms depending on the link function f. For the polynomial link function, the bounds vary with the p parameter and similarly the exponential link with the τ parameter. We tested the polynomial link function with $p \in \{1.1, 1.5, 2, 2.5, 3\}$ to test values around the common choice ($p = 2$). The exponential link function was tested with $\tau \in \{0.01, 0.05, 0.1, 0.5, 1\}$ in Leduc hold'em and random goofspiel, and $\tau \in \{0.1, 0.5, 1, 5, 10\}$ in goofspiel.

To examine the relationships between a link function, link function specific parameters, and function approximator error, we examine the empirical exploitability of f-RCFR with different levels of approximation. The degree of approximation is adjusted via the quality of features. In particular, we vary the number of partitions, n. Increasing n increases discriminative power and reduces approximation error (Figure 1).

The number of buckets in each partition is fixed at $m = 10$. If the number of information states that share an action is not evenly divisible by ten, a subset of the buckets are assigned one more information state than the others. Thus, adding a partition adds

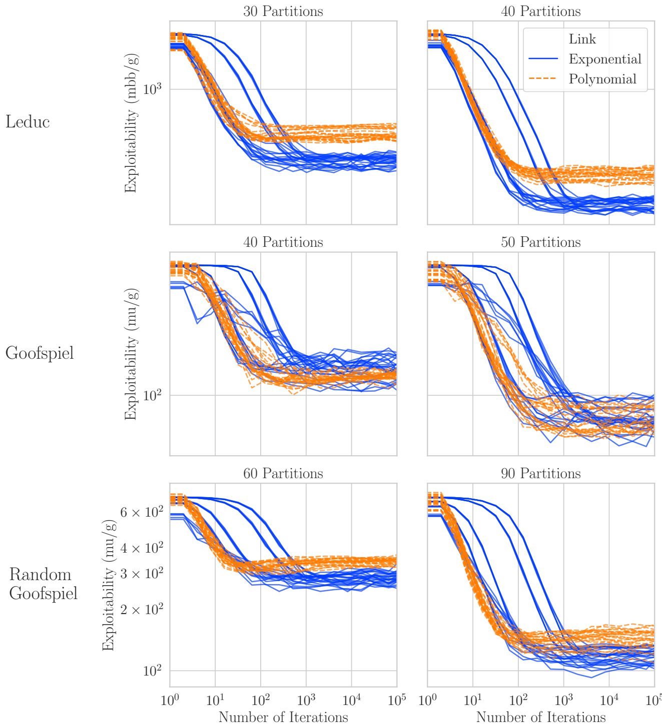

Figure 3: Exploitability of the average strategy profile for all configurations and runs with the exponential and polynomial link functions. The exponential link function achieves a lower exploitability than the polynomial link function when a moderate number of partitions (30 or 40) are used in Leduc hold'em (top). The same occurs in random goofspiel with 60 or 90-partitions (bottom). Both link functions perform similarly in goofspiel with 40 or 50-partitions (center).

ten features. Only one feature per partition is non-zero for any given information set, so the prediction cost grows linearly with the number of partitions. The ridge regression update cost however, grows quadratically with the total number of features.

4.4 Results and Analysis

Figure 2 shows the average exploitability of the best link function and hyper-parameter configuration during learning (top) and after 100k-iterations (bottom). The best parameterization was selected according to the average final exploitability after 100K-iterations over 5-runs. Notice that the exploitability of the average strategy profile decreases as the number of partitions increases, as predicted by the f-RCFR exploitability bounds given the decrease in the prediction error associated with increasing the number of partitions (Figure 1).

With 30 and 40-partitions in Leduc hold'em, and 60 and 90 in random goofspiel, the best instance with an exponential link function outperforms all of those with polynomial link functions, including RCFR (polynomial link with $p = 2$) (Figure 3, top and bottom). These feature parameters correspond to a moderate amount of function approximation error. In addition, this performance difference was observed across all configurations of the exponential and polynomial link in Leduc hold'em. *i.e.*, all of the instances with the exponential link function plateau to a final average exploitability lower than that of all those with polynomial link functions.

The exponential link function does not outperform the polynomial link function in goofspiel or when the number of partitions is large, however (Figure 3, center and Figure 2, bottom). Thus, the relative performance of different link functions is dependent on the game and the degree of function approximation error.

Among the different choices of p for the polynomial link function, $p = 2$ (RCFR) performs well with respect to the other polynomial instances across all partition numbers and in all three games (Figure 2 (bottom)). It is outperformed only by $p = 1.1$ and $p = 1.5$ in random goofspiel with many partitions, $n = 90$ and $n = 120$ respectively.

5 CONCLUSIONS

In this paper, we generalize the RRM Theorem in two dimensions—the link function, including the polynomial and exponential link functions—and regret metrics, including external and internal regret. The generalization to different link functions allows us to construct regret bounds for a general f-RCFR algorithm. The f-RCFR algorithm can approximate Nash equilibria in zero-sum games with imperfect information using alternative functional policy parameterizations beyond the previously studied normalized ReLU parameterization.

We then examine the performance of f-RCFR with the polynomial and exponential link functions under different hyper-parameter choices and different levels of function approximation error in Leduc hold'em poker and imperfect information goofspiel. f-RCFR with the polynomial link function and $p = 2$ often achieves an exploitability competitive with or lower than other choices, but the exponential link function can outperform all polynomial parameters when the functional regret estimator has a moderate degree of approximation.

This work focuses primarily on the benefits of alternatives to the ReLU policy parameterization. However, extending the RRM Theorem to a more general class of regret metrics that includes internal regret also suggests future directions, particularly the approximation of correlated equilibria [8] or extensive-form correlated equilibria [30] with function approximation.

NeuRD [23] and Politex [1] demonstrate that benefits can be gained by adapting a regret-minimizing method to the function approximation case in RL settings. These algorithms are also particular ways of implementing approximate Hedge, utilizing softmax policies. Since ReLU policies outperform softmax policies in some cases, it would be worthwhile to investigate their performance in RL applications.

ACKNOWLEDGMENTS

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Alberta Machine Intelligence Institute (Amii), and Alberta Treasury Branch (ATB). Computing resources were provided by WestGrid and Compute Canada.

REFERENCES

[1] Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellert Weisz. 2019. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*. 3692–3702.

[2] Marc Bellemare, Joel Veness, and Michael Bowling. 2012. Sketch-based linear value function approximation. In *Advances in Neural Information Processing Systems*. 2213–2221.

[3] A. Blum and Y. Mansour. 2007. Learning, Regret Minimization, and Equilibria. In *Algorithmic Game Theory*. Cambridge University Press, Chapter 4.

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

[5] Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. 2019. Deep Counterfactual Regret Minimization. In *Proceedings of the 36th International Conference on Machine Learning (ICML-19)*. 793–802.

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

[7] Noam Brown and Tuomas Sandholm. 2019. Superhuman AI for multiplayer poker. *Science* 365, 6456 (2019), 885–890.

[8] Nicolo Cesa-Bianchi and Gabor Lugosi. 2006. *Prediction, learning, and games*. Cambridge university press.

[9] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. 2019. Regret Circuits: Composability of Regret Minimizers. In *International Conference on Machine Learning*. 1863–1872.

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

[11] Sam Ganzfried and Tuomas Sandholm. 2013. Action translation in extensive-form games with large action spaces: Axioms, paradoxes, and the pseudo-harmonic mapping. In *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*.

[12] Amy Greenwald, Zheng Li, and Casey Marks. 2006. Bounds for Regret-Matching Algorithms.. In *ISAIM*.

[13] Amy Greenwald, Zheng Li, and Casey Marks. 2006. *Bounds for Regret-Matching Algorithms*. Technical Report CS-06-10. Brown University, Department of Computer Science.

[14] S. Hart and A. Mas-Colell. 2000. A Simple Adaptive Procedure Leading to Correlated Equilibrium. *Econometrica* 68, 5 (2000), 1127–1150.

[15] Johannes Heinrich and David Silver. 2016. Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121* (2016).

[16] Michael Johanson, Neil Burch, Richard Valenzano, and Michael Bowling. 2013. Evaluating state-space abstractions in extensive-form games. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 271–278.

[17] Marc Lanctot. 2013. *Monte Carlo Sampling and Regret Minimization for Equilibrium Computation and Decision-Making in Large Extensive Form Games*. Ph.D. Dissertation. Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada.

[18] Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. 2019. OpenSpiel: A Framework for Reinforcement Learning in Games. *CoRR* abs/1908.09453 (2019). arXiv:cs.LG/1908.09453 http://arxiv.org/abs/1908.09453

[19] Hui Li, Kailiang Hu, Zhibang Ge, Tao Jiang, Yuan Qi, and Le Song. 2018. Double neural counterfactual regret minimization. *arXiv preprint arXiv:1812.10607* (2018).

[20] Edward Lockhart, Marc Lanctot, Julien Pérolat, Jean-Baptiste Lespiau, Dustin Morrill, Finbarr Timbers, and Karl Tuyls. 2019. Computing Approximate Equilibria in Sequential Adversarial Games by Exploitability Descent. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 464–470. https://doi.org/10.24963/ijcai.2019/66

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

[22] Dustin Morrill. 2016. *Using Regret Estimation to Solve Games Compactly*. Master's thesis. University of Alberta.

[23] Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Remi Munos, Julien Perolat, Marc Lanctot, Audrunas Gruslys, Jean-Baptiste Lespiau, and Karl Tuyls. 2019. Neural Replicator Dynamics. *arXiv preprint arXiv:1906.00190* (2019).

[24] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 7587 (2016), 484–489.

[25] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* 362, 6419 (2018), 1140–1144.

[26] Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. 2005. BayesâĂŹ bluff: Opponent modelling in poker. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI*. 550–558.

[27] Sriram Srinivasan, Marc Lanctot, Vinicius Zambaldi, Julien Perolat, Karl Tuyls, Remi Munos, and Michael Bowling. 2018. Actor-Critic Policy Optimization in Partially Observable Multiagent Environments. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.). Curran Associates, Inc., 3422–3435.

[28] Eric Steinberger. 2019. Single Deep Counterfactual Regret Minimization. *arXiv preprint arXiv:1901.07621* (2019).

[29] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* (2019). https://doi.org/10.1038/s41586-019-1724-z

[30] Bernhard Von Stengel and Françoise Forges. 2008. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research* 33, 4 (2008), 1002–1022.

[31] Kevin Waugh, Dustin Morrill, James Andrew Bagnell, and Michael Bowling. 2015. Solving games with functional regret estimation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[32] Kevin Waugh, David Schnizlein, Michael Bowling, and Duane Szafron. 2009. Abstraction pathologies in extensive games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 781–788.

[33] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. 2008. Regret minimization in games with incomplete information. In *Advances in neural information processing systems*. 1729–1736.