Regret Minimization with Function Approximation in Extensive-Form Games

by

Ryan D'Orazio

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Statistical Machine Learning

Department of Computing Science

University of Alberta

Abstract

Computing a Nash equilibrium in zero-sum games, or more generally saddle point optimization, is a fundamental problem in game theory and machine learning, with applications spanning across a wide variety of domains, from generative modeling and computer vision to super-human AI in imperfect information games like poker. Despite the broad application of Nash equilibria, traditional methods from optimization and machine learning are not directly applicable. However, in zero-sum games an effective and simple method exists – self-play with online learning. In this setup, an equilibrium is computed by pitting two algorithms against each other to play out a game repeatedly. Online learning with self-play via Counterfactual Regret Minimization (CFR) is the leading approach for saddle point computation in large games with sequential decision making and imperfect information. For very large games, CFR can be scaled in various dimensions such as sampling, subgame decomposition, and function approximation. Despite the growing interests in scaling algorithms with function approximation in areas such as reinforcement learning, current theoretical guarantees for CFR and function approximation are minimal. In this thesis we extend theoretical results for CFR when using function approximation, and complement these worst-case guarantees with experiments on several common benchmark games with sequential decision making and imperfect information.

The thesis is outlined as follows. First, relevant background is given by defining external regret – a quantity to evaluate online learning algorithms.

Then the connection between regret, self-play, and general concave-convex saddle point problems is given. A generalization of external regret in the specific online decision problem is also reviewed. The main theoretical contributions are then presented, generalizing previous work to different types of regret in the online decision problem and with different algorithms. The new theoretical guarantees with function approximation give rise to two new families of algorithms, presented as f-RCFR and f-RCFR+, combining function approximation and CFR like algorithms. Both f-RCFR and f-RCFR+ algorithms are then compared across different games, with f-RCFR+ demonstrating superior performance and better management of function approximator capacity.

Preface

Parts of this thesis was published as a conference paper with shared first co-authorship with Dustin Morrill in the proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, 2020 [14]. My contributions in the publication include the theory presented in Sections 3.1 and 3.2, and Figures 5.1, 5.2, and 5.3. This thesis extends the results presented in the paper with new theoretical results, and experiments. The new theoretical results are presented in Chapter 3 (Section 3.2.1 and afterwards). The new experiments include those for the f-RCFR+ algorithm (Section 4.4), and experiments with sampling (Section 5.4.2). These experiments would not be possible without the OpenSpiel package [38] and Dustin Morill's experiment contributions that were used in the publication.

Outside the scope of this thesis I also performed research within the area of uncertainty in machine learning. In particular, computing simultaneous predictions intervals with applications to survival analysis [54].

Don't worry about the overall importance of the problem;
work on it if it looks interesting.
I think there's a sufficient correlation between interest and importance.

– David Harold Blackwell

It's easier to resist at the beginning than at the end.

– Leonardo da Vinci

Acknowledgements

There are many deserving of acknowledgement, and I cannot fully express in words my gratitude for their support. I would first like to thank my advisors, James R. Wright and Matthew E. Taylor, for their mentorship as well as their support throughout my degree; both have pushed me to take my research the extra step and have been invaluable to my development as a researcher. I am also indebted to my peers at the University of Alberta, Samuel Sokota, Dustin Morrill, Khurram Javed, Joshua Teitz, Daniel Chui, and Ifaz Kabir, among others, for the many insightful conversations about research, life, and other topics; all for which nurtured my development as a researcher and as a person. I would also like to thank all the members of the online learning reading group, especially Zaheen Farraz Ahmad, for introducing me to online learning. I am also thankful to several experienced researchers and faculty, Marc Lanctot, Csaba Szepesvári, Alexander Melnikov, Russell Greiner, and Martha White, for countless discussions on technical topics and for their unique perspectives on research as well as interesting problems. I would also like to acknowledge the generous financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC), for the Alexander Graham Bell Canada Graduate Scholarship - Master's (NSERC CGS-M), and the Faculty of Graduate Studies and Research, for the Walter H. Johns Graduate Fellowship.

I would like to also thank my family, my parents and brother for their never-ending support. I am also grateful to Joey Caucci, David Tagliamonti, and Gianfranco Tarsitano, for their support. Finally, I would like to thank Veronica Piccone for supporting me and my interests over the years. Her unwavering support and kindness has been instrumental to my success as a student and researcher.

Contents

1 Introduction **1**

2 Background Material **6**
2.1 External Regret . . . 7
2.1.1 External Regret and Zero-Sum Games . . . 7
2.1.2 Computing a Nash Equilibrium . . . 8
2.2 Online Learning on the Simplex . . . 11
2.2.1 Beyond External Regret . . . 12

3 Approximate Regret-Matching **15**
3.1 General Bounds . . . 15
3.2 Bounds for Specific Link Functions . . . 20
3.2.1 Lipschitz Conditions . . . 23
3.3 Approximate Regret-Matching+ . . . 25
3.3.1 Related Work . . . 29
3.4 Interchanging Expectation and Max . . . 30
3.5 Future Work . . . 31

4 Approximate Regret-Matching in Extensive-Form Games **33**
4.1 Background . . . 33
4.2 Counterfactual Regret Minimization . . . 38
4.3 f-RCFR . . . 40
4.4 f-RCFR+ . . . 44

5 Experiments **46**
5.1 Algorithm Implementation . . . 46
5.2 Games . . . 50
5.3 Parameters . . . 51
5.4 Results and Analysis . . . 52
5.4.1 f-RCFR+ . . . 53
5.4.2 External Sampling . . . 57

6 Conclusion **61**

References **63**

Appendix A Appendix **69**
A.1 Existing Results . . . 69
A.2 Proofs . . . 70
A.3 Statistical Significance Tests . . . 70
A.3.1 Figure 5.2 . . . 71
A.3.2 Figure 5.4 . . . 73
A.3.3 Figure 5.5 . . . 74

List of Tables

A.1 Statistical tests for top of Figure 5.2 in Leduc. 71
A.2 Statistical tests for top of Figure 5.2 in goofspiel. 71
A.3 Statistical tests for top of Figure 5.2 in random goofspiel. . . . 72
A.4 Statistical tests for bottom of Figure 5.2 in Leduc. 72
A.5 Statistical tests for bottom of Figure 5.2 in goofspiel. 73
A.6 Statistical tests for bottom of Figure 5.2 in random goofspiel. . 73
A.7 Statistical tests for Figure 5.4 in Leduc. 73
A.8 Statistical tests for Figure 5.4 in goofspiel. 74
A.9 Statistical tests for Figure 5.4 in random goofspiel. 74
A.10 Statistical tests for Figure 5.5 in Leduc. 74
A.11 Statistical tests for Figure 5.5 in goofspiel. 74
A.12 Statistical tests for Figure 5.5 in random goofspiel. 75

List of Figures

5.1 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. 47

5.2 (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). 48

5.3 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). 49

5.4 Exploitability of the average strategy profile for all instances of the polynomial link function and all runs. For each game, Leduc (top), goofspiel (center), and random goofspiel (bottom), f-RCFR+ outperforms f-RCFR with much lower exploitability. 54

5.5 Exploitability of the average strategy profile for all instances of the exponential link function and all runs. For each game, Leduc (top), goofspiel (center), and random goofspiel (bottom), f-RCFR+ outperforms f-RCFR when there is large approximation error (columns 1-3). 55

5.6 Cumulative function approximation error for f-RCFR+ (dotted lines) and f-RCFR (solid lines) with different different partitions and exponential link function (left) and polynomial link function (right). For each game, Leduc (top), goofspiel (center), random goofspiel (bottom), f-RCFR+ accumulates much lower approximation error than f-RCFR. 56
5.7 Exploitability of the average profile in the game of leduc for f-RCFR (solid lines) and f-RCFR+ (dotted lines) with external sampling and the the polynomial link function. Except for partition size 50 and when $p = 1.1$, f-RCFR+ achieves lower exploitability. For reference, exploitabiliy of tabular CFR and tabular CFR(RM+) are included. 58
5.8 Exploitability of the average profile in the game of leduc for f-RCFR (solid lines) and f-RCFR+ (dotted lines) with external sampling and the exponential link function. Except for low function approximation error (50 partitions), f-RCFR+ achieves lower exploitability. For reference, exploitabiliy of tabular CFR and tabular CFR(RM+) are included. 59
5.9 Cumulative function approximation error for f-RCFR (solid lines) and f-RCFR+ (dotted lines) with external sampling and the exponential link function (left) and the polynomial link function (right). For all instances, f-RCFR+ accumulates lower function approximation error. 60

Chapter 1

Introduction

A fundamental problem with growing interests and applications to machine learning is saddle point optimization, or min max optimization, or computing a Nash equilibrium in a two-player zero-sum game.[1] Some examples of applications include learning generative models [21], reinforcement learning [13, 64], computer vision [12], training neural networks for supervised learning [49], and learning to play poker at an expert level [4, 7, 41]. Fundamentally, the saddle point problem entails finding an equilibrium in a zero-sum game between the minimization and the maximization player. In practice we seek efficient algorithms that will find a good approximate equilibrium, a pair of strategies, one for each player, that is "close" to an equilibrium. Efficient algorithms will converge to a neighbourhood of an equilibrium with minimal iterations, small memory requirements, and typically only requiring feedback that is easily attainable (*e.g.* using a first-order oracle that only provides gradient or subgradient feedback).

In addition to saddle point optimization, the relatively new field of online convex optimization has caught the interest of many researchers in theoretical machine learning. In online optimization a learning algorithm's performance is measured by it's external regret, the difference in accrued rewards (or costs) versus the best fixed decision in hindsight, for any possible sequence of reward functions (or loss functions). For concave-convex games, algorithms suitable for online convex optimization [11, 28, 47], can be used to efficiently find an

[1] In the sequel we will often refer to two-player zero-sum games as zero-sum.

equilibrium via self-play. Online algorithms are typically simple to implement, and only require first-order information.

Outside of machine learning, connections between equilibria and online learning have been known for decades [3, 11, 26, 27]. In Blackwell's seminal work, zero-sum games are extended to those with vector payoffs, one player seeks to force an average payoff vector to approach a closed convex set regardless of the sequence played by the other player [3]. The beauty of Blackwell's work is a constructive proof, providing a simple algorithm to force approachability of a target set. When applied to traditional matrix games, Blackwell's algorithm yields the popular *regret-matching* algorithm, a parameter-free no-regret online learning algorithm; thus capable of computing a Nash equilibrium in zero-sum games [11, 27].

The simple procedure of playing out a game with two online learning algorithms to find a saddle point is not just theoretically sound but is the leading approach in computing a Nash equilibrium in large sequential games. In particular, 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, 7, 8, 41]. In addition to the online learning approach to solving games, the CFR framework reduces online learning over a large strategy space to many simple online learning problems, one for each decision point in the game [17, 69]. In principle, the CFR framework justifies running any no-regret algorithm at each decision point, however, CFR is often used with the regret-matching algorithm or variants thereof.

For very large games, one cannot store information for each state, rendering CFR intractable. Historically, scaling CFR to large games has been done with abstraction, similar states are grouped together to form a smaller but strategically similar game [20, 31, 66, 69]. The abstract game is solved with CFR and the resulting strategies are translated back to original game.

Function approximation is a natural generalization of abstraction. In CFR, this amounts to estimating the information required for each online algorithm instead of storing them all in a table [6, 39, 42, 57, 65]. Game solving with

function approximation can be competitive with domain specific state abstraction [6, 29, 42, 65], and in some cases is able to outperform tabular CFR without abstraction if the players are optimizing against their best responses [40]. Function approximation has facilitated many recent successes in game playing more broadly [51, 52, 61].

Combining function approximation and regret-minimization with applications to CFR was initially studied by Waugh *et al.* [65], introducing the *Regression Regret-Matching (RRM)* Theorem—giving a sufficient condition for function approximator error to still achieve no regret when combining approximation with regret-matching. A direct application of RRM is a regret-bound for CFR with regret-matching and function approximation, amounting to the *Regression Counterfactual Regret Minimization (RCFR)* algorithm. Intuitively, in a given iteration RCFR prescribes at each state a distribution over actions proportional to a rectified linear unit (ReLU) function applied to approximate action preferences. However, most reinforcement learning (RL) algorithms for discrete action spaces take a different approach: they exponentiate and normalize the preferences according to the softmax function. Coupling CFR with the softmax function amounts to using the *Hedge* or *Exponential Weights* learning algorithm at each decision point to generate policies [19].

In fact, RM and Hedge can be unified. Greenwald *et al.* [23] present (Φ, f)-*regret matching*, a general framework for constructing algorithms to minimize Φ-regret—a generalization of external regret when using a policy parameterized by a *link function* f.

In this thesis, we generalize generalize the RRM Theorem to (Φ, f)-regret matching by extending the Greenwald *et al.* framework to the case when the inputs to the algorithms are approximate, and to a new class of algorithms, (Φ, f)-regret matching+. This new class generalizes the regret-matching+ algorithm; an algorithm commonly observed to accelerated learning in games and played a major in solving human-scale games [4, 59, 60]. This new approximate (Φ, f)-regret matching and (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

approximations are used. Our analysis, both due to improvements previously made by Greenwald *et al.* [23] and more careful application of conventional inequalities, tightens the bound for RRM. The corresponding improvement to the RCFR Theorem [42, 65] 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 varies between link functions and parameter choices.

The approximate (Φ, f)-regret matching and (Φ, f)-regret matching+ frameworks provide 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* and *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* [55] and *imperfect information goofspiel* [36]. 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. Moreover, we find that f-RCFR+ almost always outperforms f-RCFR.

This thesis is organized as follows. In Chapter 2 we formally define a Nash equilibrium in the context of two-player zero-sum games and the connection with online learning. We then define online decision problems and present the Φ-regret matching framework developed by Greenwald, Li, and Marks. In Chapter 3 the framework is extended to approximate regret matching and approximated regret matching+, with regret bounds for these new classes of algorithms. In Chapter 4 we discuss extensive-form games, the CFR framework, and present f-RCFR and f-RCFR+ algorithms along with worst-case guarantees. Afterward, in Chapter 5 we present experiments and results, comparing the f-RCFR and f-RCFR+ algorithms for various parameter choices

and game environments.

Chapter 2

Background Material

Online learning has a rich history, attracting the interests of both the AI and game theory community [50]. Learning "online" consists of making decisions only from past observations. Performance of following such a decision rule is measured by comparing the accrued rewards (costs) with some baseline sequence of decisions. Crucially, in this work we do not make explicit assumptions on the sequence of reward (loss) functions that the agent will observe. More precisely, the reward function may change from one time step to the next, thus allowing for arbitrary changes in the reward function, possibly controlled by an adversary. Indeed, if the losses are chosen by a worst-case adversary then we have a two player game, the decision-maker against the environment. Additionally, we will assume that an agent is maximizing rewards instead of minimizing losses to closely follow the extensive-form games literature. Much of the online learning literature, however, is embedded within the field of online optimization, for which the convention of minimizing losses is used. Thus results can be carried over from one community to another with the appropriate change of sign.

In this chapter we formally define regret, a metric which we use to evaluate online algorithms. Then we define zero-sum games and present the well-known folk-theorem, justifying self-play of online algorithms for solving games. We then close the chapter with a more general definition of regret, Φ-regret.

2.1 External Regret

The online learning problem can be viewed from the perspective of an agent interacting with an environment. Consider an agent that is tasked with making repeated decisions x^t from some convex compact set $\mathcal{X}$, for time steps $t \in \{1, \cdots, T\}$.[1] At a given time step t the reward is given by some concave continuous function $f^t : \mathcal{X} \to \mathbb{R}$, which is revealed to the agent after x^t is chosen. We then evaluate the decisions $\{x^t\}_{t \leq T}$ by comparing the accrued rewards with a simple baseline strategy, the best *fixed* decision in hindsight

$$R_{\mathcal{X}}^T \coloneqq \max_{x \in \mathcal{X}} \sum_{t=1}^{T} f^t(x) - \sum_{t=1}^{T} f^t(x^t). \tag{2.1}$$

The difference in reward $R_{\mathcal{X}}^T$ is referred to as *external regret*. A desired property of decision rules or learning algorithms is *no-regret*.

Definition 1. *A decision rule or learning algorithm is said to be no-regret or Hannan Consistent if*

$$\lim_{T \to \infty} \frac{R_{\mathcal{X}}^T}{T} \leq 0.$$

Equivalently, the regret must grow at a sublinear rate $R_{\mathcal{X}}^T \in o(T)$. Notice the above definition does not make assumptions on the sequence of reward functions $\{f^t\}_{t \leq T}$. In the optimization literature, the above setup is referred to as online convex optimization [28, 68]. It is interesting to note that designing no-regret algorithms for linear functions is enough to achieve no-regret in the more general concave setting if all $\{-f^t\}_{t \leq T}$ are subdifferentiable [28].[2]

2.1.1 External Regret and Zero-Sum Games

Without explicit assumptions on the sequence of reward functions $\{f^t\}_{t \leq T}$, the online problem allows for modeling worst-case adversarial environments.

[1] In the Euclidean space $\mathbb{R}^d$, compactness is equivalent to closed and bounded.

[2] The function f is subdifferentialble at x if there exists g such that for all y in the domain $f(y) \geq f(x) + \langle g, y - x \rangle$.

Therefore, it should not be surprising that there exists a strong connection between online learning and games. We consider the connection between external regret and a simple adversarial environment — two-player zero-sum games.

Consider a two player game consisting of a row and column player whose decisions lie in the convex compact sets $\mathcal{X}$ and $\mathcal{Y}$, respectively. We consider the concave-convex continuous payoff function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. That is, $f(\cdot, y)$ is concave for each y and $f(x, \cdot)$ is convex for each x. The game is zero-sum because the row player is trying to maximize their payoff given by f while the column player is doing the same with payoff given by $-f$. An important solution concept for this zero-sum game is a Nash equilibrium [43], or equivalently a saddle point of f.

Definition 2. *A Nash equilibrium of the two-player zero-sum game given by the concave-convex payoff function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a pair $(\bar{x}, \bar{y}) \in \mathcal{X} \times \mathcal{Y}$ such that*

$$\forall (x, y) \in \mathcal{X} \times \mathcal{Y} \quad f(x, \bar{y}) \leq f(\bar{x}, \bar{y}) \leq f(\bar{x}, y).$$

That is, neither the row player nor the column player can unilaterally improve their payoff by deviating.

The saddle point interpretation is interesting as it corresponds to the minimax value of the game [53]. The Nash equilibrium $(\bar{x}, \bar{y})$ is minimax optimal

$$\max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} f(x, y) = \min_{y \in \mathcal{Y}} \max_{x \in \mathcal{X}} f(x, y) = f(\bar{x}, \bar{y}) = v.$$

The value v corresponds to the value of the game, and if the row player plays their part of a Nash equilibrium they are guaranteed a payoff of at least v, that is $\forall y \in \mathcal{Y} \quad f(\bar{x}, y) \geq v$, and similarly for the column player. In fact, the set of max min strategies for both players coincide with the set of Nash equilibria. Therefore, playing a Nash equilibrium is *safe* and the game is considered solved if such an equilibrium is found in a two-player zero-sum game.

2.1.2 Computing a Nash Equilibrium

The safety guarantee of a Nash equilibrium in a two-player zero-sum game thus motivates computing a strategy $\bar{x}$ that is *close* to a Nash equilibrium.

In particular, offline computation of $\bar{x}$ can later be used against arbitrary opponents. An equilibrium can be approximated as closely as needed in offline self-play, where both players are using no-regret learning algorithms to play the game.

First we define an approximate solution to a game, an ϵ-Nash equilibrium.

Definition 3. *An ϵ-Nash equilibrium of the two-player zero-sum game given by the concave-convex function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a pair $(\bar{x}, \bar{y}) \in \mathcal{X} \times \mathcal{Y}$ such that, for $\epsilon \geq 0$*

$$\forall (x, y) \in \mathcal{X} \times \mathcal{Y} \quad f(x, \bar{y}) - \epsilon \leq f(\bar{x}, \bar{y}) \leq f(\bar{x}, y) + \epsilon.$$

Similar definitions of an approximate equilibrium also exist, such as the following:

$$\max_{x \in \mathcal{X}} f(x, \bar{y}) - \min_{y \in \mathcal{Y}} f(\bar{x}, y) \leq \epsilon', \tag{2.2}$$

$$\frac{\max_{x \in \mathcal{X}} f(x, \bar{y}) - \min_{y \in \mathcal{Y}} f(\bar{x}, y)}{2} \leq \frac{\epsilon'}{2}. \tag{2.3}$$

The value on the left-hand of condition (2.2) is typically referred to as the saddle point residual or gap in the optimization literature. Similarly, the value in condition (2.3) is referred to as the exploitability of $(\bar{x}, \bar{y})$ and is simply the average best response value for both players. We have that an ϵ-Nash equilibrium has a saddle point residual of at most 2ϵ and an exploitability of at most ϵ. Conversely, a saddle point residual of 2ϵ implies an exploitability of ϵ and a 2ϵ Nash equilibrium. Clearly, with smaller ϵ we attain a *better* approximation of the equilibrium and better guarantees against worst-case opponents.

If both the column and row players repeatedly play the game given by f with decisions $\{(x^t, y^t)\}_{t \leq T}$ while observing the reward functions $f^t(\cdot) = f(\cdot, y^t)$ and $\tilde{f}^t(\cdot) = -f(x^t, \cdot)$, respectively, we then have the following folk-theorem giving a worst-case bound on ϵ for the ϵ-Nash equilibrium formed by the average decisions.

Theorem 1. *Given a two-player zero-sum game with concave-convex payoff function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, if both players pick decisions $(x^t, y^t) \in \mathcal{X} \times \mathcal{Y}$ at time $t \in \{1, \cdots, T\}$ then the average strategies $(\bar{x}, \bar{y})$ form an ϵ- Nash equilibrium with $\epsilon \leq \frac{R_{\mathcal{X}}^T + R_{\mathcal{Y}}^T}{T}$. Where $\bar{x} = \frac{1}{T}\sum_{t=1}^T x^t$, and $\bar{y} = \frac{1}{T}\sum_{t=1}^T y^t$.*

Proof. Let $f^t(\cdot) = f(\cdot, y^t)$ and $\tilde{f}^t(\cdot) = -f(x^t, \cdot)$.

$$\max_{x \in \mathcal{X}} f(x, \bar{y}) - f(\bar{x}, \bar{y}) \leq \max_{x \in \mathcal{X}} \sum_{t=1}^T \frac{1}{T} f(x, y^t) - f(\bar{x}, \bar{y}) \tag{2.4}$$

$$= \max_{x \in \mathcal{X}} \sum_{t=1}^T \frac{1}{T} \left(f(x, y^t) - f(x^t, y^t) + f(x^t, y^t) \right) - f(\bar{x}, \bar{y}) \tag{2.5}$$

$$= \max_{x \in \mathcal{X}} \sum_{t=1}^T \frac{1}{T} \left(f^t(x) - f^t(x^t) \right) + \sum_{t=1}^T \frac{1}{T} f(x^t, y^t) - f(\bar{x}, \bar{y}) \tag{2.6}$$

$$= \frac{R_{\mathcal{X}}^T}{T} + \sum_{t=1}^T \frac{1}{T} f(x^t, y^t) - f(\bar{x}, \bar{y}) \tag{2.7}$$

$$\leq \frac{R_{\mathcal{X}}^T}{T} + \sum_{t=1}^T \frac{1}{T} \left(f(x^t, y^t) - f(x^t, \bar{y}) \right) = \frac{R_{\mathcal{X}}^T}{T} + \sum_{t=1}^T \frac{1}{T} \left(\tilde{f}^t(\bar{y}) - \tilde{f}^t(y^t) \right) \tag{2.8}$$

$$\leq \frac{R_{\mathcal{X}}^T + R_{\mathcal{Y}}^T}{T} \tag{2.9}$$

The first and second inequality follow by convexity. Similarly we have for the column player, $\min_{y \in \mathcal{Y}} f(\bar{x}, \bar{y}) - f(\bar{x}, y) \leq \frac{R_{\mathcal{X}}^T + R_{\mathcal{Y}}^T}{T}$.

□

The above proof can be also be modified to bound the saddle point residual by $\frac{R_{\mathcal{X}}^T + R_{\mathcal{Y}}^T}{T}$, which in turn would yield the same result, see Farina, Kroer, and Sandholm [15] for example.

Interestingly, this analysis is decentralized and does not directly deal with the interaction of both learners. The $\Omega(\sqrt{T})$ lower bound for online convex optimization [28], therefore, incorrectly implies a $\Omega(\sqrt{T})$ lower bound on $R_{\mathcal{X}}^T + R_{\mathcal{Y}}^T$. With more careful analysis it is possible to achieve a constant bound if the algorithms in self-play satisfy a condition known as *regret bounded by variation in utilities* (RVU) [16, 58].[3]

[3]The improved regret bounds assume a smooth game, see Section 5.2.3 in Bubeck et al. for a formal definition of smoothness in games [9]. Acceleration of online learning methods

2.2 Online Learning on the Simplex

When decisions live in the simplex $\mathcal{X} = \Delta(A)$, where $\Delta(A)$ is the set of all probability distributions over finite actions A, and reward functions are linear, we recover a well-studied online learning problem.[4] We refer to such a problem as an online decision problem (ODP) and adopt the notation from Greenwald *et al.* [23]. In other works, this problem setting is also referred to as prediction with expert advice [11].

Formally, an ODP consists of a finite set of possible actions A and a bounded set of possible rewards $\mathcal{R} \subset \mathbb{R}$ 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 $r : A \to \mathcal{R}$.

Similar to the general online learning problem, at each round t an agent selects a policy, that is, a distribution over actions $x^t \in \Delta(A)$. The agent then samples an action, $a^t \sim x^t$, and subsequently receives a reward function, $r^t \in \Pi$. Knowledge of the selected reward function allows the agent 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, thus allowing for modeling multi-agent, adversarial interactions, and game theoretic equilibrium concepts even though it is described from the perspective of a single agent's decisions. For example, if we formulate a game with the payoff function $f(x, y) = \langle x, By \rangle$ where the row player makes decisions $x \in \Delta(A_x)$ and the column player $y \in \Delta(A_y)$, then we recover the traditional normal-form zero-sum game. From Theorem 1, if the row and column player observe $f^t(\cdot) = \langle \cdot, By^t \rangle$ and $\tilde{f}^t(\cdot) = -\langle B^\top x^t, \cdot \rangle$ at the end of each round, then an ϵ-Nash equilibrium can be computed with self-play.

Recall that the algorithms employed in self-play are *online*, depending only

for saddle point computation predates the RVU condition [48]. The RVU condition was introduced by Syrgkanis, Agarwal, Luo, and Schapire [58] and allows for mixing and matching different algorithms. Accelerated convergence is guaranteed so long as each learner satisfies the RVU condition.

[4] Observe that this setting entails a compact decision set with a continuous reward function.

on previous experience of play to make their next decision. We denote this experience at time t as a history $h \in H^t \coloneqq A^t \times \Pi^t$, where $H^0 \coloneqq \{\emptyset\}$. We can then formalize the notion of an online algorithm in the ODP setting with the following definition.

Definition 4. *An online learning algorithm in an ODP is a sequence of functions* $\{L_t\}_{t=1}^{\infty}$*, where* $L_t : H_{t-1} \to \Delta(A)$.

2.2.1 Beyond External Regret

From the definition of external regret (2.1) we have for an ODP

$$R_{\mathcal{X}}^T = R_{\Delta(A)}^T = \max_{a \in A} \sum_{t=1}^{T} r^t(a) - \langle r^t, x^t \rangle. \tag{2.10}$$

We overload notation and write the expected reward of policy x with reward function r as the standard inner product $\langle r, x \rangle$, where r is the associated reward vector $\{r(a)\}_{a \in A}$ and x the appropriate vector of probabilities over actions. The best policy in hindsight will be any that has support that is a subset of the set of optimal actions given the observed reward functions – $\operatorname{Argmax}_{a \in A} \sum_{t=1}^{T} r^t(a)$. Therefore, it is sufficient to track the performance of the best action to measure the regret.

In this work we talk about two generalizations of external regret. The first is with respect to the baseline action, instead of comparing with the best action in hindsight we may consider the best sequence of policies that are derived from the policies chosen by the agent $\{x^t\}_{t \leq T}$. The second generalization differs between the expected accrued rewards $\sum_t \langle r^t, x^t \rangle$ and those collected by the sampled actions $\sum_t r^t(a^t)$, where $a^t \sim x^t$. The work by Greenwald, Li, and Marks [23] refer to the former as distribution regret and the latter as action regret. Generalizations to other types of regret allow for a broader application of self-play, including efficient computation of a correlated equilibrium [11]. Both generalizations of external regret are nicely included within the Φ-regret framework for which we make use of and reintroduce below [23].

Action Transformations

To generalize regret to different baselines, 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. The transformations induce alternative sequences of decisions with a baseline expected reward $\langle r^t, \phi(a^t) \rangle$, when $a^t \sim x^t$.

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 $\Phi_{EXT} \coloneqq \{\phi : \exists a \in A \, \forall a' \in A \quad \phi(a') = \delta_a\}$. 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

$$\phi_{INT}^{(a,b)}(a') = \begin{cases} \delta_b & \text{if } a' = a, \\ \delta_{a'} & \text{otherwise.} \end{cases}$$

Note that there are $|\Phi_{INT}| = |A|^2 - |A| + 1$ internal transformations [23].

We define the policy induced by a policy x and action transformation ϕ as $[\phi](x) = \sum_{a \in A} x(a)\phi(a)$. Note that $[\phi](x)$ is the expected policy given by ϕ, $\mathbb{E}_{a \sim x}[\phi(a)]$.

Φ-Regret

The regret for not following action transformation ϕ when action a was chosen and reward function r was observed is the instantaneous ϕ-regret,

$$\hat{\rho}_\phi(a, r) = \mathbb{E}_{a' \sim \phi(a)}[r(a')] - r(a) = \langle r, \phi(a) \rangle - r(a). \tag{2.11}$$

Similarly, on expectation for policy x we have the expected instantaneous ϕ-regret,

$$\rho_\phi(x, r) = \mathbb{E}_{a \sim x}[\hat{\rho}_\phi(a, r)] = \langle r, [\phi](x) \rangle - \langle r, x \rangle. \tag{2.12}$$

For a set of action transformations, Φ, the instantaneous Φ-regret vectors are $\hat{\rho}_\Phi(a, r) = (\hat{\rho}_\phi(a, r))_{\phi\in\Phi}$, and $\rho_\Phi(x, r) = (\rho_\phi(x, r))_{\phi\in\Phi}$ for action a and policy x respectively.

For an ODP with observed history h at time t, composed of reward functions $\{r^t\}_{t=1}^T$ and actions $\{a^t\}_{t=1}^T$ selected by the agent on each round, the Φ-regret after T-rounds against action transformations Φ is $\hat{R}_\Phi^T(h) = \sum_{t=1}^T \hat{\rho}_\Phi(a^t, r^t)$. For brevity we will omit the h argument, and for convenience we set $\hat{R}_\Phi^0 \coloneqq 0$. Note that $\hat{R}$ is a random vector due to the sampled actions. Similarly, we have for the chosen policies $\{x^t\}_{t=1}^T$ the expected Φ-regret $R_\Phi^T = \sum_{t=1}^T \rho_\Phi(x^t, r^t)$.

We seek to bound the expected average maximum Φ-regret,

$$\mathbb{E}\left[\frac{1}{T}\max_{\phi\in\Phi}\hat{R}_\phi^T\right], \tag{2.13}$$

as well as the average expected regret by interchanging the max and the expectation

$$\frac{1}{T}\max_{\phi\in\Phi}R_\phi^T. \tag{2.14}$$

Choosing Φ to be Φ_{EXT} or Φ_{INT} corresponds to the well studied maximum external regret or maximum internal regret objectives, respectively. More precisely, if $\Phi = \Phi_{EXT}$ then $\rho_\Phi(x, r) = (\langle r, [\phi](x)\rangle - \langle r, x\rangle)_{\phi\in\Phi} = (r(a) - \langle r, x\rangle)_{a\in A}$, therefore, $\max_{\phi\in\Phi} R_\phi^T$ amounts to the same external regret objective $R_{\Delta(A)}^T$ defined in (2.10).

Connections to Other Equilibria Outside of two-player zero-sum games, self-play via regret-minimization can be used for computing other types of equilibria. More specifically, if all players are minimizing external regret then an ϵ-coarse correlated equilibrium can be computed with the average decisions or sampled actions. Similarly, if all players are minimizing internal regret then an ϵ-correlated equilibrium can be computed. We refer the reader to Cesa-Bianchi and Lugosi for an extensive review of online learning in connection with computing various eqiilibria [11].

Chapter 3

Approximate Regret-Matching

Recall that no-regret algorithms can be used in self-play to find an ϵ-Nash equilibrium. However, in games with multiple decision points, the regret for both players $R^T_{\mathcal{X}}$, $R^T_{\mathcal{Y}}$, may be decomposed as a sum of regrets, one for each decision point. Instead of reasoning about regret over the policy space for a player in the whole game we can minimize regret *locally* over the simplex at each decision point. For games with many decision points, it may not be feasible to store the information required to be no-regret, typically on the order of the number of decision points. In this work we consider the case of minimizing regret when approximations are used instead of true values, ultimately showing how regret may be bounded in terms of approximation error. In this chapter we focus on the single state/decision point case, where Φ-regret is minimized with approximations. The analysis closely follows the work of Greenwald, Li, and Marks [23]; however, the main contribution comprises extensions of their work when algorithms use approximations, and an extension of their framework to a new class of algorithms inspired by a successful variant of regret-matching, regret-matching+.

3.1 General Bounds

For a finite set of action transformations Φ, we seek no-regret algorithms where the average regret is measured by the objectives 2.13, and 2.14. For the remainder of the chapter, emphasis will be placed on the former objective, however, this chapter will conclude with discussion on transferring the results to the lat-

ter case where regret is defined with expected reward from the chosen policies and not the sampled actions.

Given a finite 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 [23]. A (Φ, f)-regret-matching algorithm at time t chooses $x \in \Delta(A)$ that is a fixed point of

$$M_t(x) \coloneqq \frac{\sum_{\phi \in \Phi} Y_\phi^t[\phi](x)}{\sum_{\phi \in \Phi} Y_\phi^t},$$

when $\hat{R}_\Phi^{t-1} \in \mathbb{R}_+^{|\Phi|} \setminus \{0\}$, where $Y_\Phi^t \coloneqq (Y_\phi^t)_{\phi \in \Phi} \coloneqq f(\hat{R}_\Phi^{t-1})$, 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 $x \propto Y_\Phi^t$ [24]. Examples of (Φ, f)-regret-matching algorithms include Hart's algorithm [27]—typically called "regret-matching"—and Hedge [19], with link functions $f(x)_i = x_i^+$ and $f(x)_i = e^{\frac{1}{\tau} x_i}$ with temperature parameter $\tau > 0$, respectively. When the decision set is the simplex, Hedge possess a better regret bound, however, Regret-matching and its variants are have typically shown to perform better in practice. In the next chapter, we will see that with approximations the performance gap between hedge and regret-matching is less clear, thus motivating studying different link functions.

A useful technique for bounding regret when estimates are used in place of true values is to define an $\epsilon-$Blackwell condition, as was used in the Regression Regret-Matching Theorem (RRM) [65]. 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 5 ((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*

$$\langle f(\hat{R}_\Phi^{t-1}(h)), \mathbb{E}_{a \sim L_t(h)}[\rho_\Phi(a, r)] \rangle \leq \epsilon. \tag{3.1}$$

The Regret Matching Theorem [23] 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

$$\tilde{M}_t(x) \coloneqq \frac{\sum_{\phi \in \Phi} \tilde{Y}_\phi^t[\phi](x)}{\sum_{\phi \in \Phi} \tilde{Y}_\pi^t}, \tag{3.2}$$

when $\tilde{R}_\Phi^{t-1} \in \mathbb{R}_+^{|\Phi|} \setminus \{0\}$ and arbitrarily otherwise, where $\tilde{Y}_\Phi^t \coloneqq f(\tilde{R}_\Phi^{t-1})$ and $\tilde{R}_\Phi^{t-1}$ is an estimate of $\hat{R}_\Phi^{t-1}$, possibly from a function approximator. Such an algorithm is referred to as approximate (Φ, f)-regret-matching.

Similarly to the RRM Theorem [42, 65], we show that the ϵ parameter of the (Φ, f, ϵ)-Blackwell condition depends on the link output approximation error, $\left\| Y_\Phi^t - \tilde{Y}_\Phi^t \right\|_1$.

Theorem 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 $\epsilon \leq 2U \left\| Y_\Phi^t - \tilde{Y}_\Phi^t \right\|_1$, where $Y_\Phi^t \coloneqq f(\hat{R}_\Phi^{t-1})$, and $\tilde{Y}_\Phi^t \coloneqq f(\tilde{R}_\Phi^{t-1})$.*

Proof. We denote $r = (r'(a))_{a \in A}$ as the reward vector for an arbitrary reward function $r' : A \to \mathbb{R}$. Since by construction this algorithm chooses L_t at each timestep t to be the fixed point of $\tilde{M}_t$, all that remains to be shown is that this algorithm satisfies the (Φ, f, ϵ)-Blackwell condition with $\epsilon \leq 2U \left\| Y_\Phi^t - \tilde{Y}_\Phi^t \right\|_1, t > 0$.

By expanding the value of interest in the (Φ, f)-Blackwell condition and applying elementary upper bounds, we arrive at the desired bound. For simplicity, we omit timestep indices and set $L \coloneqq L_t(h)$.

First, suppose $\sum_{\phi\in\Phi}\tilde{Y}_\Phi^t \neq 0$:

$$\begin{aligned}
\langle Y_\Phi^t, \mathbb{E}_{a\sim L}[\rho_\Phi(a,r)]\rangle &= \sum_{\phi\in\Phi} Y_\phi^t \left(\langle r, [\phi](L)\rangle - \langle r, L\rangle\right) \\
&= \langle r, \sum_{\phi\in\Phi} Y_\phi^t \left([\phi]L - L\right)\rangle \\
&= \langle r, \sum_{\phi\in\Phi} \left(\tilde{Y}_\Phi^t - \tilde{Y}_\Phi^t + Y_\phi^t\right)\left([\phi](L) - L\right)\rangle \\
&= \langle r, \left(\sum_{\phi\in\Phi}\tilde{Y}_\Phi^t\right)\left(\tilde{M}L - L\right) + \sum_{\phi\in\Phi}\left(Y_\phi^t - \tilde{Y}_\phi^t\right)\left([\phi](L) - L\right)\rangle \\
&= \langle r, \sum_{\phi\in\Phi}\left(Y_\phi^t - \tilde{Y}_\phi^t\right)\left([\phi](L) - L\right)\rangle \qquad (3.3) \\
&\leq \|r\|_\infty \left\|\sum_{\phi\in\Phi}\left(Y_\phi^t - \tilde{Y}_\phi^t\right)\left([\phi](L) - L\right)\right\|_1 \\
&\leq \|r\|_\infty \sum_{\phi\in\Phi} |Y_\phi^t - \tilde{Y}_\Phi^t| \left(\|[\phi](L)\|_1 + \|L\|_1\right) \\
&\leq \|r\|_\infty \sum_{\phi\in\Phi} |Y_\phi^t - \tilde{Y}_\Phi^t|(1+1) \\
&\leq 2U \left\|Y_\phi^t - \tilde{Y}_\Phi^t\right\|_1 .
\end{aligned}$$

If $\sum_{\phi\in\Phi}\tilde{Y}_\Phi^t = 0$ it is easy to see the inequality still holds.

Therefore, $\{L_t\}_{t=1}^\infty$ satisfies the (Φ, f, ϵ)-Blackwell condition with $\epsilon \leq 2U\left\|Y_\phi^t - \tilde{Y}_\Phi^t\right\|_1$, as required to complete the argument.

□

In the special case that $\sum_{\phi\in\Phi} Y_\phi^t = \sum_{\phi\in\Phi}\tilde{Y}_\phi^t = c$, for $c > 0$, a refinement of theorem (2) can be made, removing the constant of two.

Corollary 1. *Given reward system (A,$\mathcal{R}$), a finite set of action transformations $\Phi \subseteq \Phi_{ALL}$, and link function $f : \mathbb{R}^{|\Phi|} \to \mathbb{R}_+^{|\Phi|}$, if $\sum_{\phi\in\Phi} Y_\Phi^t = \sum_{\phi\in\Phi}\tilde{Y}_\Phi^t = c$ for $c > 0$, then an approximate (Φ, f)-regret-matching algorithm, $\{L_t\}_{t=1}^\infty$, satisfies the (Φ, f, ϵ)-Blackwell Condition with $\epsilon \leq U\left\|Y_\Phi^t - \tilde{Y}_\Phi^t\right\|_1$, where $Y_\Phi^t \coloneqq f(\hat{R}_\Phi^{t-1})$, and $\tilde{Y}_\Phi^t \coloneqq f(\tilde{R}_\Phi^{t-1})$.*

Proof. If $\sum_{\phi\in\Phi} Y_\phi^t = \sum_{\phi\in\Phi}\tilde{Y}_\phi^t = c$ and $L \in \Delta(A)$, then $\sum_{\phi\in\Phi} Y_\phi^t L = \sum_{\phi\in\Phi}\tilde{Y}_\Phi^t L =$

cL. Therefore, in line (3.3) of Theorem 2 we have

$$\begin{aligned}\langle r, \sum_{\phi\in\Phi}(Y_\phi^t - \tilde{Y}_\phi^t)([\phi](L) - L)\rangle &= \langle r, \sum_{\phi\in\Phi}(Y_\phi^t - \tilde{Y}_\phi^t)[\phi](L)\rangle + \langle r, cL - cL\rangle \\ &= \langle r, \sum_{\phi\in\Phi}(Y_\phi^t - \tilde{Y}_\phi^t)([\phi](L))\rangle.\end{aligned}$$

Proceeding in the same steps as theorem (2) yields the result. □

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 $\mathbb{E}[G(\hat{R}_\Phi^t)]$ for an appropriate function G [11, 23]. Bounding the regret for an approximate (Φ, f)-regret-matching algorithm will be done similarly, except the bound on ϵ from Theorem 2 will be used. Proceeding in this fashion yields the following theorem:

Theorem 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[1], then an approximate (Φ, g)-regret-matching algorithm $\{L_t\}_{t=1}^\infty$ guarantees at all times $t \geq 0$*

$$\mathbb{E}[G(\hat{R}_\Phi^t)] \leq G(0) + t \sup_{a\in A, r\in\Pi} \gamma(\rho^\Phi(a, r)) + 2U \sum_{s=1}^{t} \mathbb{E}\left[\left\| g(\hat{R}_\Phi^{s-1}) - g(\tilde{R}_\Phi^{s-1})\right\|_1\right]. \tag{3.4}$$

Proof. The proof is similar to [23, Corollary 7] except that the learning algorithm is playing the approximate fixed point with respect to the link function g. From Theorem 1 we have $\langle g(\hat{R}_\Phi^{t-1}(h)), \mathbb{E}_{a\sim L_t(h)}[\rho_\Phi(a, r)]\rangle \leq 2U \left\| g(\hat{R}_\Phi^{t-1}) - g(\tilde{R}_\Phi^{t-1})\right\|_1$. Noticing that

$$\mathbb{E}_{a\sim L_t(h)}[\rho_\Phi(a, r)] = \mathbb{E}[\rho_\Phi(a, r)|\hat{R}_\Phi^{t-1}]$$

and taking $x^t = \rho_\Phi(a, r), X^t = \hat{R}_\Phi^t$ we have

$$\begin{aligned}&\langle g(X^{t-1}), \mathbb{E}[x^t|X^{t-1}]\rangle + \mathbb{E}[\gamma(x^t)|X^{t-1}] \leq \\ &2U \left\| g(\hat{R}_\Phi) - g(\tilde{R}_\Phi^{t-1})\right\|_1 + \sup_{a\in A, r\in\Pi} \gamma(\rho^\Phi(a, r)).\end{aligned}$$

The result directly follows from Theorem 12 by taking

$$C(\tau) = 2U \left\| g(\hat{R}_\Phi^{\tau-1}) - g(\tilde{R}_\Phi^{\tau-1})\right\|_1 + \sup_{a\in A, r\in\Pi} \gamma(\rho^\Phi(a, r)).$$

. □

[1] 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 $x, y \in \mathbb{R}^n$, $G(x + y) \leq G(x) + g(x) \cdot y + \gamma(y)$.

Remark 1. *If the function g in the approximation error term (3.4) satisfies the condition of Corollary 1 then one attains a tighter bound by removing the constant of two in $C(\tau)$.*

3.2 Bounds for Specific Link Functions

In this section we give regret bounds for approximate (Φ, f)-regret-matching algorithms when f is the polynomial and exponential link function.

Polynomial Link Function

Given the polynomial link function $f(x)_i = (x_i^+)^{p-1}$ we consider two cases $2 < p < \infty$ and $1 < p \leq 2$. For the following results it is useful to denote the maximal activation $\mu(\Phi) = \max_{a \in A} |\{\phi \in \Phi : \phi(a) \neq \delta_a\}|$ [23].

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

Theorem 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*

$$\mathbb{E}\left[\max_{\phi \in \Phi} \frac{1}{t} \hat{R}_\Phi^t\right] \leq$$

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

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

Proof. The proof follows closely to [23, Theorem 9]. Taking $G(x) = \|x^+\|_p^2$ and $\gamma(x) = (p-1)\|x\|_p^2$ then $\langle G, g, \gamma \rangle$ is a Gordon triple [23]. Given the above Gordon triple we have

$$\left(\mathbb{E}\left[\max_{\phi\in\Phi}\hat{R}_\Phi^t\right]\right)^2 \leq \mathbb{E}\left[\left\|(\hat{R}_\Phi^t)^+\right\|\right]_p^2 \tag{3.5}$$

$$= \mathbb{E}[G(\hat{R}_\Phi^t)] \tag{3.6}$$

$$\leq G(0) + t\sup_{a\in A, r\in\Pi}\gamma(\rho^\Phi(a,r)) + 2U\sum_{s=1}^{t}\mathbb{E}\left[\left\|g(\hat{R}_\Phi^{s-1}) - g(\tilde{R}_\Phi^{s-1})\right\|_1\right] \tag{3.7}$$

$$\leq G(0) + t(p-1)4U^2(\mu(\Phi))^{2/p} + 2U\sum_{k=1}^{t}\mathbb{E}\left[\left\|g(\hat{R}_\Phi^{k-1}) - g(\tilde{R}_\Phi^{k-1})\right\|_1\right] \tag{3.8}$$

The first inequality is from Lemma 1. The second inequality follows from Corollary 9 and Theorem 3. The third inequality is an application of Lemma 2. The result then immediately follows. □

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

Theorem 5. *Given an ODP, a finite set of action transformations $\Phi \subseteq \Phi_{ALL}$, and the polynomial link function f with $1 < p \leq 2$, then an approximate (Φ, f)-regret-matching algorithm guarantees*

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

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

Proof. The proof follows closely to [23, Theorem 11]. Taking $G(x) = \|x^+\|_p^p$ and $\gamma(x) = (p-1)\|x\|_p^p$ then $\langle G, g, \gamma\rangle$ is a Gordon triple [23]. Given the above Gordon triple we have

$$\left(\mathbb{E}\left[\max_{\phi\in\Phi}\hat{R}_\Phi^t\right]\right)^p \leq \mathbb{E}\left[\left\|(\hat{R}_\Phi^t)^+\right\|\right]_p^p \tag{3.9}$$

$$= \mathbb{E}\left[G(\hat{R}_\Phi^t)\right] \tag{3.10}$$

$$\leq G(0) + t\sup_{a\in A, r\in\Pi}\gamma(\rho^\Phi(a,r)) + 2U\sum_{s=1}^{t}\mathbb{E}\left[\left\|g(\hat{R}_\Phi^{s-1}) - g(\tilde{R}_\Phi^{s-1})\right\|_1\right] \tag{3.11}$$

$$\leq G(0) + t(2U)^p(\mu(\Phi)) + 2U\sum_{k=1}^{t}\mathbb{E}\left[\left\|g(\hat{R}_\Phi^{k-1}) - g(\tilde{R}_\Phi^{k-1})\right\|_1\right] \tag{3.12}$$

The first inequality is from Lemma 1. The second inequality follows from Corollary 9 and Theorem 3. The third inequality is an application of Lemma 2. The result then immediately follows. □

In comparison to the RRM Theorem [42], 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[2] $|A|-1$. These improvements are due to the tighter bound in Theorem 2 and the original Φ-regret analysis [23], respectively. Aside from these differences, the bounds coincide.

Exponential Link Function

Theorem 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*

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

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

Proof. The proof follows closely to [23, Theorem 13]. Taking $G(x) = \tau\ln\left(\sum_i e^{\frac{1}{\tau}x_i}\right)$ and $\gamma(x) = \frac{1}{2\tau}\|x\|^2_\infty$ then $\langle G, g, \gamma\rangle$ is a Gordon triple [23]. Given the above Gordon triple we have

$$\mathbb{E}\left[\max_{\phi\in\Phi}\frac{1}{\tau}\hat{R}^t_\Phi\right] = \mathbb{E}\left[\ln\, e^{\max_{\phi\in\Phi}\frac{1}{\tau}\hat{R}^t_\Phi}\right] \tag{3.13}$$

$$= \mathbb{E}\left[\ln\max_{\phi\in\Phi} e^{\frac{1}{\tau}\hat{R}^t_\Phi}\right] \tag{3.14}$$

$$\leq \mathbb{E}\left[\ln\sum_{\phi\in\Phi} e^{\frac{1}{\tau}\hat{R}^t_\Phi}\right] \tag{3.15}$$

$$= \frac{1}{\tau}\mathbb{E}[G(\hat{R}^t_\Phi)] \tag{3.16}$$

$$\leq \frac{1}{\tau}\left(G(0) + t\sup_{a\in A, r\in\Pi}\gamma(\rho^\Phi(a,r)) + U\sum_{s=1}^{t}\mathbb{E}\left[\left\|g(\hat{R}^{s-1}_\Phi) - g(\tilde{R}^{s-1}_\Phi)\right\|_1\right]\right) \tag{3.17}$$

$$\leq \frac{1}{\tau}\left(G(0) + t\frac{2U^2}{\tau} + U\sum_{s=1}^{t}\mathbb{E}\left[\left\|g(\hat{R}^{s-1}_\Phi) - g(\tilde{R}^{s-1}_\Phi)\right\|_1\right]\right) \tag{3.18}$$

The second inequality follows from Corollary 9, Theorem 3, and by Remark 1, since $\sum_{\phi\in\Phi} Y^t_\phi = \sum_{\phi\in\Phi}\tilde{Y}^t_\phi = 1$. The result then immediately follows. □

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

The Hedge algorithm corresponds to the exponential link function $f(x)_i = e^{\frac{1}{\tau}x_i}$ when $\Phi = \Phi_{EXT}$, so Theorem 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. 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.

3.2.1 Lipschitz Conditions

The bounds for approximate regret-matching include an error term

$$\sum_{s=1}^{t} \left\| g(\hat{R}_\Phi^{s-1}) - g(\tilde{R}_\Phi^{s-1}) \right\|_1.$$

Given that the objective of a function approximator is typically to minimize a loss of the form $\left\| \hat{R}_\Phi^t - \tilde{R}_\Phi^t \right\|$, it is attractive to include this term directly in the bound. If the function g is Lipschitz continuous then replacing the error with an appropriate prediction error is possible.

Definition 6. *A function $f : \mathbb{R}^d \to \mathbb{R}^m$ is Lipschitz continuous with constant L and with respect to the norm $\|\cdot\|$, $f \in \mathcal{F}(L, \|\cdot\|)$, if*

$$\|f(x) - f(y)\|_* \leq L \|x - y\|, \quad \forall x, y \in \mathbb{R}^d. \tag{3.19}$$

Where $\|\cdot\|_*$ is the dual norm, $\|x\|_* = \sup\{\langle x, y\rangle | \, \|y\| \leq 1\}$. Recall that the norms $\|\cdot\|_p$, $\|\cdot\|_q$ are dual to each other if $\frac{1}{p} + \frac{1}{q} = 1$ [5].

Noticing that in most cases the function g in the error bound (3.4) is the gradient mapping of some potential function G, allows for the application of a well known result, giving a necessary and sufficient condition for Lipschitz continuity for the function of interest g.

Theorem 7 (Theorem 2.1.5 [44])**.** *If G is continuously differentiable then the inclusion of $\nabla G \in \mathcal{F}(L, \|\cdot\|)$ is equivalent to the following condition*

$$0 \leq G(y) - G(x) - \langle \nabla G(x), y - x \rangle \leq \frac{L}{2} \|x - y\|^2 .$$

As a consequence we have the following results.

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

$$\mathbb{E}\left[\max_{\phi \in \Phi} \frac{1}{t} \hat{R}_\Phi^t\right] \leq$$
$$\frac{1}{t}\sqrt{t(p-1)4U^2(\mu(\Phi))^{2/p} + 4|\Phi|^{\frac{1}{p}} U \sum_{k=1}^{t} \mathbb{E}\left[\left\|\hat{R}_\Phi^{k-1} - \tilde{R}_\Phi^{k-1}\right\|_p\right]}.$$

Proof. For $p > 2$, if $G(x) = \|x^+\|_p^2$, g is taken to be the same as in Theorem 4, and $\gamma(x) = (p-1)\|x\|_p^2$, then $\langle G, g, \gamma \rangle$ is a Gordon triple [23]. Therefore, the following condition holds

$$\begin{aligned} \forall x, y \in \mathbb{R}^{|\Phi|} \quad G(y) &\leq G(x) + \langle g(x), y - x \rangle + \gamma(y - x) \\ &= G(x) + \langle g(x), y - x \rangle + (p-1)\|y - x\|_p^2 . \end{aligned}$$

Greenwald, Li, and Marks [24] showed that g is indeed the gradient mapping of G, $g = \nabla G$, and that G is continuously differentiable. G is convex, then by Theorem 7, $g \in \mathcal{F}(2(p-1), ||\cdot||_p)$. From Hölder's inequality we have $\|x\|_1 \leq |\Phi|^{\frac{q-1}{q}} \|x\|_q$.[3]. Applying the Lipschitz condition and picking q such that $\|\cdot\|_q$ is dual to $\|\cdot\|_p$ ($q = {}^p/_{p-1}$) yields the result.

For the case $p = 2$, take $g : \mathbb{R}^{|\Phi|} \to \mathbb{R}_+^{|\Phi|}$ and $g(x)_i = 2(x_i^+)$. We then have

$$\begin{aligned} \|g(x) - g(y)\|_1 &= 2\sum_{i=1}^{|\Phi|} |x_i^+ - y_i^+| \\ &\leq 2\sum_{i=1}^{|\Phi|} |x_i - y_i| = 2\|x - y\|_1 \\ &\leq 2\sqrt{|\Phi|}\,\|x - y\|_2 . \end{aligned}$$

Using the bound from Theorem 5 with p=2 and applying the above inequality gives the result. □

[3] Apply Hölder's inequality with the vector x and $\mathbf{1}$, the vector of all ones and the same dimension as x.

Corollary 3. *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*

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

Proof. Taking $G(x) = \tau\ln\left(\sum_i e^{\frac{1}{\tau}x_i}\right)$, $g(x)_i = e^{\frac{1}{\tau}x_i}/\sum_j e^{\frac{1}{\tau}x_j}$, and $\gamma(x) = \frac{1}{2\tau}\|x\|_\infty^2$, then $\langle G, g, \gamma\rangle$ is a Gordon triple [23]. Observing that G is convex and differentiable [44], and that $g = \nabla G$ [24], then by the definition of a gordon triple we have

$$\begin{aligned}\forall x, y \in \mathbb{R}^{|\Phi|} \quad G(y) &\leq G(x) + \langle g(x), y - x\rangle + \gamma(y - x) \\ &= G(x) + \langle g(x), y - x\rangle + \frac{1}{2\tau}\|y - x\|_\infty^2.\end{aligned}$$

Therefore, by Theorem 7 $g \in \mathcal{F}(1/\tau, \|\cdot\|_\infty)$. Using the well-known fact that the norms $\|\cdot\|_1$ and $\|\cdot\|_\infty$ are dual to each other yields the result. □

The case of $p < 2$ is omitted and left for future-analysis. Leveraging the Gordon triple used in Theorem 5 along with the techniques used by Nesterov [44, Theorem 2.1.5], it may be possible to attain results similar to Corollary 2.

3.3 Approximate Regret-Matching+

In practice, the simple modification of regret-matching algorithms, entailing "forgetting" negative values in the stored regret vector R_Φ^t, has proven to be an important modification in the context of computing Nash equilibria in large games; playing a major role in solving heads-up limit Texas Hold'em [4, 60], and achieving expert-level play in no limit Hold'em [41]. This modified algorithm is typically referred to as regret-matching+ (pronounced regret-matching-plus), initially discovered by Tammelin [59], and later shown to be no-regret by Tammelin, Burch, Johanson, and Bowling [60].

Regret-matching+ was designed for minimizing the regret objective (2.13) with $\Phi = \Phi_{EXT}$, and using the polynomial link function with $p = 2$. The

algorithm removes the negative regret values by performing the update

$$Q_\Phi^t = \left(Q_\Phi^{t-1} + \rho_\Phi(x^t, r^t)\right)^+, \tag{3.20}$$

where $Q_\Phi^0 = 0$, and with policy $x^t \propto (Q_\Phi^{t-1})^+$. We then have the following:

$$\forall \phi \in \Phi \, \forall r^t \in \Pi \quad Q_\phi^{t-1} + \rho_\phi(x^t, r^t) \leq Q_\phi^t.$$

Applying this inequality iteratively, we have $R_\Phi^t \leq Q_\Phi^t$ where the inequality is satisfied for each component of the vectors. Similarly, for regret defined by the sampled actions we have $\hat{R}_\Phi^t \leq \hat{Q}_\Phi^t$ when $\hat{Q}_\Phi^t = \left(\hat{Q}_\Phi^{t-1} + \hat{\rho}_\Phi(a^t, r^t)\right)^+$. Consequently, the modified objectives

$$\mathbb{E}\left[\frac{1}{T}\max_{\phi\in\Phi} \hat{Q}_\phi^T\right], \tag{3.21}$$

and

$$\frac{1}{T}\max_{\phi\in\Phi} Q_\phi^T, \tag{3.22}$$

upper bound the objectives (2.13), (2.14), respectively. Regret bounds for these new objectives therefore imply regret bounds for the original objetives. Similarly to the previous sections, we present the results for objective (3.21). Extensions to objective (3.22) are discussed in the next section.

Regret-matching+ was introduced with update rule (3.20) when $\Phi = \Phi_{EXT}$ and $p = 2$, thus, analysis for different link functions, action transformations, and regret objective (2.13), is to the best of our knowledge novel. Given the results from Section 3.2, bounds for the objectives (3.21), and (3.22), are readily attained for the polynomial link-function via a modified definition of a Gordon triple for which the Gordon triples used in Theorems 4, and 5, satisfy.

First we need to consider a slightly modified version of the (Φ, f, ϵ)-Blackwell condition, where the inner product of $\langle f(\hat{R}_\Phi^{t-1}(h)), \mathbb{E}_{a\sim L_t(h)}[\rho_\Phi(a, r)]\rangle$ is replaced with $\langle f(\hat{Q}_\Phi^{t-1}(h)), \mathbb{E}_{a\sim L_t(h)}[\rho_\Phi(a, r)]\rangle$. For the remainder of this section, the (Φ, f, ϵ)-Blackwell condition will refer to this modified version using $\hat{Q}_\Phi^{t-1}$ instead of $\hat{R}_\Phi^{t-1}$. Furthermore, an algorithm that plays the fixed point using $\hat{Q}_\Phi^{t-1}$ instead of $\hat{R}_\Phi^{t-1}$, the policy x^t is chosen such that $x^t = \tilde{M}_t(x^t)$ where

$Y_\Phi^t \coloneqq (Y_\phi^t)_{\phi\in\Phi} \coloneqq f(\hat{Q}_\Phi^{t-1})$, will be referred to as an approximate (Φ, f)- regret-matching+ algorithm.

Before stating the results we need a result similar to Theorem 3 for approximate (Φ, f)- regret-matching+ algorithms. In addition, we need a different definition than a Gordon triple, which we coin as a positive invariant Gordon triple.

Definition 7. *A positive invariant 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 $x, y \in \mathbb{R}^n$, $G((x+y)^+) \le G(x) + \langle g(x), y\rangle + \gamma(y)$.*

Equipped with this new definition, we can arrive at a result similar to Theorem 3 for approximate (Φ, g)-regret-matching+ algorithms.

Theorem 8. *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 positive invariant Gordon triple, then an approximate (Φ, g)-regret-matching+ algorithm $\{L^t\}_{t=1}^\infty$ guarantees at all times $t \ge 0$*

$$\mathbb{E}[G(\hat{Q}_\Phi^t)] \le G(0) + t \sup_{a\in A, r\in\Pi} \gamma(\rho^\Phi(a,r)) + 2U \sum_{s=1}^{t} \mathbb{E}\left[\left\| g(\hat{Q}_\Phi^{s-1}) - g(\tilde{Q}_\Phi^{s-1})\right\|_1\right]. \tag{3.23}$$

Proof. The proof is similar to [23, Corollary 7] except that the learning algorithm is playing the approximate fixed point with respect to the link function g and using the values $(Y_\phi^t)_{\phi\in\Phi} = g(\tilde{Q}_\Phi^{t-1})$. Observe that Theorem 2 holds when applying the link function g to the vectors $\hat{Q}_\Phi^{t-1}$, and $\tilde{Q}_\Phi^{t-1}$, therefore, we have $g(\hat{Q}_\Phi^{t-1}(h)) \cdot \mathbb{E}_{a\sim L_t(h)}[\rho^\Phi(a,r)] \le 2U \left\| g(\hat{Q}_\Phi^{t-1}) - g(\tilde{Q}_\Phi^{t-1})\right\|_1$.

Noticing that

$$\mathbb{E}_{a\sim L_t(h)}[\rho_\Phi(a,r)] = \mathbb{E}[\rho_\Phi(a,r)|\hat{Q}_\Phi^{t-1}]$$

and taking $x^t = \rho_\Phi(a,r), X^t = \hat{Q}_\Phi^t$ we have

$$\begin{aligned}&\langle g(X^{t-1}), \mathbb{E}[x^t|X^{t-1}]\rangle + \mathbb{E}[\gamma(x^t)|X^{t-1}] \le \\ &2U \left\| g(\hat{Q}_\Phi^{t-1}) - g(\tilde{Q}_\Phi^{t-1})\right\|_1 + \sup_{a\in A, r\in\Pi} \gamma(\rho^\Phi(a,r)).\end{aligned}$$

Unlike Theorem 12 [23, Theorem 6], we let $X^t = (X^{t-1} + x^t)^+$. However, since $\langle G, g, \gamma \rangle$ is a positive invariant Gordon triple, we have

$$\begin{aligned} G(X^t) &= G((X^{t-1} + x^t)^+) \\ &\leq G(X^{t-1}) + \langle g(X^{t-1}), x^t \rangle + \gamma(x^t). \end{aligned}$$

Following the same steps as Greenwald, Li, and Marks [23] and taking

$$C(\tau) = 2U \left\| g(\hat{R}_\Phi^{\tau-1}) - g(\tilde{R}_\Phi^{\tau-1}) \right\|_1 + \sup_{a \in A, r \in \Pi} \gamma(\rho^\Phi(a, r)),$$

gives the desired result. □

Theorem 9. *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*

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

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

Proof. The proof follows closely to [23, Theorem 9]. Taking $G(x) = \|x^+\|_p^2$ and $\gamma(x) = (p-1)\|x\|_p^2$ then $\langle G, g, \gamma \rangle$ is a Gordon triple [23]. In addition, since $G(x^+) = G(x)$ we have that $\langle G, g, \gamma \rangle$ is a positive invariant Gordon triple.

$$\left(\mathbb{E}\left[\max_{\phi \in \Phi} \hat{Q}_\Phi^t\right]\right)^2 \leq \mathbb{E}\left[\left\| (\hat{Q}_\Phi^t)^+ \right\|\right]_p^2 \tag{3.24}$$
$$= \mathbb{E}[G(\hat{Q}_\Phi^t)] \tag{3.25}$$
$$\leq G(0) + t \sup_{a \in A, r \in \Pi} \gamma(\rho^\Phi(a, r)) + 2U \sum_{s=1}^{t} \mathbb{E}\left[\left\| g(\hat{Q}_\Phi^{s-1}) - g(\tilde{Q}_\Phi^{s-1}) \right\|_1\right] \tag{3.26}$$
$$\leq G(0) + t(p-1)4U^2(\mu(\Phi))^{2/p} + 2U \sum_{k=1}^{t} \mathbb{E}\left[\left\| g(\hat{Q}_\Phi^{k-1}) - g(\tilde{Q}_\Phi^{k-1}) \right\|_1\right] \tag{3.27}$$

The first inequality is from Lemma 1. The second inequality follows from Corollary 1 and Theorem 8. The third inequality is an application of Lemma 2. The result then immediately follows. □

Similarly for the case $1 < p \leq 2$ we have a similar result.

Theorem 10. *Given an ODP, a finite set of action transformations $\Phi \subseteq \Phi_{ALL}$, and the polynomial link function f with $1 < p \leq 2$, then an approximate (Φ, f)- regret-matching algorithm guarantees*

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

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

Proof. The proof follows closely to [23, Theorem 11]. Taking $G(x) = \|x^+\|^p_p$ and $\gamma(x) = (p-1)\|x\|^p_p$ then $\langle G, g, \gamma\rangle$ is a Gordon triple [23]. In addition, since $G(x^+) = G(x)$ we have that $\langle G, g, \gamma\rangle$ is a positive invariant Gordon triple.

$$\left(\mathbb{E}\left[\max_{\phi\in\Phi}\hat{Q}^t_\Phi\right]\right)^p \leq \mathbb{E}\left[\left\|(\hat{Q}^t_\Phi)^+\right\|\right]^p_p \tag{3.28}$$

$$= \mathbb{E}\left[G(\hat{Q}^t_\Phi)\right] \tag{3.29}$$

$$\leq G(0) + t\sup_{a\in A, r\in\Pi}\gamma(\rho_\Phi(a,r)) + 2U\sum_{s=1}^{t}\mathbb{E}\left[\left\|g(\hat{Q}^{s-1}_\Phi) - g(\tilde{Q}^{s-1}_\Phi)\right\|_1\right] \tag{3.30}$$

$$\leq G(0) + t(2U)^p(\mu(\Phi)) + 2U\sum_{k=1}^{t}\mathbb{E}\left[\left\|g(\hat{Q}^{k-1}_\Phi) - g(\tilde{Q}^{k-1}_\Phi)\right\|_1\right] \tag{3.31}$$

The first inequality is from Lemma 1. The second inequality follows from Corollary 1 and Theorem 8. The third inequality is an application of Lemma 2. The result then immediately follows. □

Similar to Section 3.2.1 we may use the Lipschitz continuity of the function g to replace the errors to be of the form $\left\|\hat{Q}^t_\Phi - Q^t_\Phi\right\|_1$ (when $p \geq 2$). Analysis of the exponential link function is left as future work. Despite no current regret bound, experiments in later sections imply that approximate (Φ, f)-regret-matching+ algorithm with an exponential link function may posses a useful regret bounds.

3.3.1 Related Work

An analysis of combining regression with regret-matching+ was first done by Morrill [42]. In particular, a bound similar to Theorem 10 was derived for

$\frac{1}{T}\max_{\phi\in\Phi} Q_\phi^T$. The approach involved the analysis of the Gordon triple in Theorem 5, however, $G(Q_\Phi^t + q)$ was considered, with $q = Q_\Phi^{t+1} - Q_\Phi^t$ instead of $q = \rho_\Phi(x, r)$. Unfortunately, in an attempt to bound the inner product $\langle g(Q_\Phi^t), q\rangle$, which would provide an alternative proof to Theorem 9 (when $p = 2, \Phi = \Phi_{EXT}$), not all cases of q were considered in the proof, rendering the proof incomplete [42, Theorem 3.0.10].[4]

Generalizing regret-matching+ to Φ-regret is novel, however, this extension is not surprising by noticing that when $p = 2$, the algorithms presented thus far are closely related to Blackewell's algorithm in his approachability theorem [3]. Therefore, a recent alternative proof by Kroer [33] for regret-matching+ via Blackwell approachability can easily be extended to the Φ-regret case.[5]

3.4 Interchanging Expectation and Max

Extending the previous results to the objective $\frac{1}{T}\max_{\phi\in\Phi} R_\phi^T$ can be done by modifying the algorithms to use the expected regrets R_Φ^T in the fixed point calculation, $M_t(x) = x$, instead of the random values $\hat{R}_\Phi^T$ [23]. To consider the non-sampling case, first observe that we bound the expected regret (2.13) for a given algorithm by bounding the evolution of a potential function G. Equipped with an upper bound on $\mathbb{E}[G(\hat{R}_\Phi^T)]$, we can achieve an appropriate bound on the regret. When considering the non-sampling case the same trick applies, once we know a bound on $G(R_\Phi^T)$ we can achieve a regret bound on $\frac{1}{T}\max_{\phi\in\Phi} R_\phi^T$. More specifically, for a given algorithm we will consider a specific Gordon triple $\langle G, g, \gamma\rangle$. We then have

$$G(R_\Phi^T) = G(R_\Phi^{T-1} + \rho_\Phi(x^T, r)) \leq G(R_\Phi^{T-1}) + \langle g(R_\Phi^{T-1}), \rho_\Phi(x^T, r)\rangle + \gamma(\rho_\Phi(x^T, r)).$$

To achieve a favorable growth rate on G we would like the linear inner product term to be small. Therefore, it is reasonable to redefine the (Φ, f, ϵ)-Blackwell condition to be

$$\langle f(R_\Phi^{T-1}), \rho_\Phi(x^T, r)\rangle \leq \epsilon.$$

[4] Thanks to discussions with Dustin Morrill, a counterexample to Theorem 3.0.10 by Morrill [42] can be found.

[5] As mentioned in the lecture note by Kroer [33], it should be noted that this new alternative proof was developed in conjunction with Gabriele Farina.

This modified condition amounts to the same condition as presented by Cesa-Bianchi and Lugosi when $\epsilon = 0$ [11]. Following similar steps as Theorem 2 and Greenwald, Li, and Marks [23], it can be shown that an approximate (Φ, f)-regret matching algorithm using expected values satisfies

$$\langle f(\tilde{R}_\Phi^{T-1}), \rho_\Phi(x^T, r)\rangle = 0.$$

Therefore,

$$\begin{aligned}\langle f(R_\Phi^{T-1}), \rho_\Phi(x^T, r)\rangle &= \langle f(R_\Phi^{T-1}) - f(\tilde{R}_\Phi^{T-1}), \rho_\Phi(x^T, r)\rangle + \langle f(\tilde{R}_\Phi^{T-1}), \rho_\Phi(x^T, r)\rangle \\ &\leq \left\|\rho_\Phi(x^T, r)\right\|_\infty \left\|f(R_\Phi^{T-1}) - f(\tilde{R}_\Phi^{T-1})\right\|_1 \\ &\leq 2U \left\|f(R_\Phi^{T-1}) - f(\tilde{R}_\Phi^{T-1})\right\|_1.\end{aligned}$$

Immediately, we can recover a new version of Theorem 3. Consequently, all the results follow with similar steps, ultimately providing bounds with R_Φ^t in place of $\hat{R}_\Phi^t$, and $\mathbb{E}\left[\left\|\hat{R}_\Phi^t - \tilde{R}_\Phi^t\right\|_1\right]$ replaced with $\left\|R_\Phi^t - \tilde{R}_\Phi^t\right\|_1$. Moreover, we can apply a similar reasoning to extend the bounds for all the approximate (Φ, f)-regret matching+ algorithms presented in Section 3.3 to the non-sampling case given a positive invariant Gordon triple.

3.5 Future Work

This chapter has discussed extensions of previous work by Waugh, Morrill, Bagnell, and Bowling and Morrill, considering approximation in the more general Φ-regret setting introduced by [23]. The extensions presented herein pose two interesting fundamental questions. First, Section 3.3 extends all the previous results to approximate (Φ, f)-regret-matching+ for the case $f(x) = (x^+)^{p-1}, p > 1$, this gives a new set of algorithms for the purpose of minimizing internal regret and therefore computing a correlated equilibrium in a normal form game [11]. Given the large success of regret-matching+ ($\Phi = \Phi_{EXT}, p = 2$), it would be interesting to know if this experimental success carries over to the internal regret case and for $p \neq 2$. Second, with the unattractive $|A|^3$ time complexity of the fixed point computation for internal regret, can we modify the (Φ, f)-regret-matching+ algorithm in a way similar to Greenwald, Li,

and Schudy [25] to improve computational efficiency and maintain empirical performance?

Chapter 4

Approximate Regret-Matching in Extensive-Form Games

In this chapter we discuss regret minimization with function approximation in extensive-from games, for the purpose of computing an ϵ-Nash equilibrium in a two-player zero-sum game. Recall that a Nash equilibrium in this context consists of a policy that is not exploitable, and will guarantee a minimal reward regardless of how the other player plays. Extensive-from games model sequential decision making in stochastic environments with multiple agents and imperfect information. This includes for example a finite partially observable Markov decision process (POMDP) when conditioning on the entire history of actions and observations.

4.1 Background

Here we define a two-player zero-sum extensive-form game[1] as well as three-important formulations, the normal form, behavioral form, and the sequence form.

Informally, an extensive-from game is a turn based game modelled as a tree. Nodes are states and branches are actions; at each node in the tree one player selects a branch to transition to the next state. Stochastic transitions are represented by a *chance* "player," c. The leaves of the tree are the terminal

[1]We limit our setup to two-player zero-sum games. The regret guarantee applies more generally with N-players, however, the connection between regret and a Nash equilibrium (Theorem 1) is lost.

states of the game, where rewards are then distributed to each player.

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

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

The collection of nodes is the set of histories $\mathcal{H}$. Each node $h \in \mathcal{H}$ is equivalently represented as a history of actions, the sequence of actions taken (including chance) to traverse the tree and arrive at node h. The actions available to all players in the game, including chance outcomes, is given by the set $\mathcal{A}$. For convenience, we define the action function $A : \mathcal{H} \to \mathcal{A}$, where $A(h)$ gives the set of actions available at each history h. The player to act 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, $\mathcal{Z} \coloneqq \{h | h \in \mathcal{H}, A(h) = \emptyset\}$. x_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.

Imperfect information is modelled by a partition $\mathcal{S}$ of the histories belonging to all players except chance. When a player acts at a history they may not be able to observe all the previous actions by the other players. For example, in poker a player cannot observe the cards dealt to the other players, an action made by chance at the beginning of the game. A collection of histories that are indistinguishable to a player form an information state $s \in \mathcal{S}$. We further denote the collection of information states belonging to player i as $\mathcal{S}_i$. An information state $s \in \mathcal{S}_i$ is a subset of the histories belonging to player i, $s \subseteq \{h : h \in \mathcal{H}, p(h) = i\}$. We must have $A(h) = A(h')$ if $h, h' \in s \in \mathcal{S}$, otherwise the player can distinguish histories in the same information state. We can therefore denote the actions at s as $A(s)$. Additionally, we assume *perfect recall* so that for all histories in an information state, the sequence of information states admitted by the preceding histories must be identical. The reward or utility function for player 1 is $r_1 : \mathcal{Z} \to \mathbb{R}$, the game is zero-sum because player 2's utility function is set to $r_2 \coloneqq -r_1$.

Recall that in our definition of a Nash equilibrium (Definition 2), we require a two-argument function f along with a set of policies $\mathcal{X}$ for the row player

and $\mathcal{Y}$ for the column player. The corresponding Nash equilibrium for an extensive-form game will be a saddle point with respected to a function f that gives the expected reward for the first player (row-player). However, we must define the set of policies for each player. How these sets are constructed will have a large impact on the performance of algorithms at both a theoretical and empirical level. Below we introduce three different representations of a policy in an extensive-form game.

Normal Form

The most classical formulation of an action and a policy for a player is that of the normal form.[2] Where agents' actions are specified at a macro level, an action a for player i prescribes an action at each state $s_i \in \mathcal{S}_i$. The set of actions for player i is then the cartesian product $\tilde{\mathcal{A}}_i = \prod_{s \in \mathcal{S}_i} A(s)$, policies or the set of mixed strategies is simply the simplex $\Delta(\tilde{\mathcal{A}}_i)$. Finding a Nash equilibrium can then be written as a bilinear saddle point problem

$$\min_{x \in \Delta(\tilde{\mathcal{A}}_1)} \max_{y \in \Delta(\tilde{\mathcal{A}}_2)} \langle x, By \rangle,$$

where $B \in \mathbb{R}^{|\tilde{\mathcal{A}}_1| \times |\tilde{\mathcal{A}}_2|}$ is a matrix with entry B_{ij} as the expected reward for the row player under chance when the row and column player play actions $i \in \tilde{\mathcal{A}}_1$ and $j \in \tilde{\mathcal{A}}_2$ (*i.e.*, a deterministic policy), respectively. We can use online learning and self-play to compute an ϵ-Nash equilibrium, however, this problem is exponential in the number of states. The number of rows or columns in B is on the order of $|\mathcal{A}|^{|\mathcal{S}_i|}$.

Behavioral Form

A more convenient representation of a policy is the behavioral form. In the behavioral form, player i's **policy** or **behavioral strategy**, $x_i \in \Sigma_i$ defines a probability distribution over valid actions at each of i's information states. Given a policy x_i we denote the probability of selecting action a in information state s as $x_i(s, a)$. Sometimes we will refer to the policy at state s, $x_i(s) = (x_i(s, a))_{a \in A} \in \Delta(A(s))$. A **joint policy** or **strategy profile** is an assignment

[2]The normal form is also referred to as the strategic form in the literature.

of policies for each player, $x \coloneqq (x_1, x_2)$. We also make use of the notation x_{-i} to index the strategies of all players except for i and chance in the profile x. Unlike the strategic form, a policy randomizes over actions at each information state independently. In the case of games with perfect recall, the strategic and behavioral form are equivalent, for each behavioral policy can be represented by an equivalent policy in the strategic form and vice-versa [35].[3]

Given that we will mostly deal with the behavioural form, extra notation is needed for computing the expected rewards and distribution over histories for a given profile x. We use $\eta^x(z)$ to denote the probability of reaching terminal history $z \in \mathcal{Z}$ under profile x from the beginning of the game and $\eta^x(h, z)$ 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 $\eta^x(z) = \eta_i^x(z)\eta_{-i}^x(z) = \eta_1^x(z)\eta_2^x(z)\eta_c^x(z)$. Since chance's contribution, $\eta_c^x(z)$, does not depend on the profile x, we may simply write $\eta_c(z)$. The expected value to player i under profile x is $r_i(x) = r_i(x_1, x_2) = \sum_{z \in \mathcal{Z}} \eta^x(z) r_i(z)$.

The Nash equilibrium in the behavioral form can then be expressed as a solution $(\bar{x}_1, \bar{x}_2)$ to the following saddle point problem

$$\forall (x_1, x_2) \in \Sigma_1 \times \Sigma_2 \quad r_1(x_1, \bar{x}_2) \leq r_1(\bar{x}_1, \bar{x}_2) \leq r_1(\bar{x}_1, x_2). \tag{4.1}$$

A **best response** for player i to another player's strategy, x_{-i}, is a policy that achieves the maximum reward against x_{-i}, $r_i^*(x_{-i}) = \max_{x_i \in \Sigma_i} r_i\left((x_i, x_{-i})\right)$. Recall that a profile, x, is an ε-Nash equilibrium if neither player can unilaterally deviate from their assigned policy and gain more than ε (see Definition 3). Will make use of the alternative metric, exploitability of x (see inequality 2.3), the average of the best response values $(r_1^*(x_{-1}) + r_2^*(x_{-2}))/2$.

Sequence Form

Despite the convenient representation of the behavioral form, the expected reward r_1 is not concave in the policy of the row player nor is it convex in

[3]Two policies are equivalent if they induce the same distribution over outcomes. Interestingly, in general the two forms are not comparable. There are games where behavioral strategies are more expressive and conversely there exists games where the the strategic form is more expressive.

the policy of the column player; as one must multiply probabilities of selecting actions along the terminal history z to compute $\eta^x(z)$. Therefore we cannot directly apply Theorem 1. Fortunately, we can consider the sequence form representation, with the same size the behavioral form (on the order of $|\mathcal{S}_i||\mathcal{A}|$), and recover a bilinear saddle point for the Nash equilibrium [32, 62].

A sequence form policy $\hat{x}_i \in \hat{\Sigma}_i$ for player i is a vector indexed by information state and action pairs, also referred to as sequences, $(s, a), s \in \mathcal{S}_i, a \in A(s)$, including the empty sequence $\varnothing$. $\hat{x}_i(s, a)$ is the probability that player i plays the sequence of actions to reach s and then play a at s.[4]

To recover a valid behavioral policy we must impose some restrictions on the vector $\hat{x}_i$. First, it is useful to refer to the unique *parent* pair $par(s) \in \mathcal{S}_i \times \mathcal{A}$ associated with s, where $par(s)$ is the previous information state and action visited and taken by player i before reaching s. If there are no previous actions before s then $par(s) = \varnothing$. $\varnothing$ refers to the empty sequence at the root of the game. The set of sequence form policies $\hat{\Sigma}_i$ for player i is then defined with the following constraints:

$$\begin{aligned}
&\hat{x}_i \geq 0 \\
&\hat{x}_i(\varnothing) = 1 \\
&\hat{x}_i(par(s)) = \sum_{a \in A(s)} \hat{x}_i(s, a) \quad \forall s \in \mathcal{S}_i.
\end{aligned}$$

The first condition ensures there are no negative entries in the vector. The last condition can be interpreted as a flow of probability, the probability flowing into a state must equal the probability flowing out through the actions available at the state. The last two conditions can also be written as a solution to a linear system $E\hat{x}_i = e$, see Nisan, Roughgarden, Tardos, and Vazirani [45] for details. For a sequence form policy $\hat{x}_i$, we can recover an equivalent behavioral form x_i, where $x_i(s, a) \propto \hat{x}_i(s, a)$.

An equivalent representation of the sequence form, and perhaps more intuitive is that of a *treeplex* [30, 34]. The treeplex representation can be thought of as a tree of simplicies. For a given player there is a scaled simplex $\tilde{\Delta}(s)$ at

[4]Due to perfect recall we have that the sequence of actions played by player i leading to $s \in \mathcal{S}_i$ is well-defined. There cannot be two distinct sequences of actions leading to s.

each $s \in \mathcal{S}_i$, where $\tilde{\Delta}(s) = \hat{x}_i(par(s))\Delta(A(s))$. Notice here that a choice in the scaled simplex entails picking how to randomize over actions at $A(s)$. Indeed, the treeplex view can be thought of as a top-down perspective for building the sequence form policy $\hat{x}_i$ using the associated behavioral form policy x_i.

To form a bilinear saddle point problem we need a payoff matrix $\hat{B}$ that will preserve the normal form and behavioral form Nash equlibria. To this end, we define the set of reachable terminal histories for the sequences, (s, a) and (s', a'), for player 1 and 2 respectively as $Z((s, a), (s', a')) \subseteq \mathcal{Z}$. $Z((s, a), (s', a')) = \varnothing$ when the state and action pairs, which uniquely defines a sequence of actions for each player, cannot end in a terminal history when played against one another.[5] The payoff matrix $\hat{B} \in \mathbb{R}^{dim(\hat{\Sigma}_1) \times dim(\hat{\Sigma}_2)}$ is then given by the following entries $\hat{B}_{(s,a),(s',a')} = \sum_{z \in Z((s,a),(s',a'))} \eta_c(z) r_1(z)$. Note that $\hat{B}$ is sparse, as per convention, we set $\hat{B}_{(s,a),(s',a')} = 0$ if $Z((s, a), (s', a')) = \varnothing$.

The Nash equilibrium can then be computed in the sequence form by finding a solution $(\hat{x}_1, \hat{x}_2)$ to the following saddle point problem

$$\forall (x_1, x_2) \in \hat{\Sigma}_1 \times \hat{\Sigma}_2 \quad \langle x_1, \hat{B}\hat{x}_2 \rangle \leq \langle \hat{x}_1, \hat{B}\hat{x}_2 \rangle \leq \langle \hat{x}_1, \hat{B}x_2 \rangle. \tag{4.2}$$

The solution $(\hat{x}_1, \hat{x}_2)$ will correspond to a solution of (4.1) for the corresponding equivalent behavioral policies. With the saddle point in the sequence form we can now apply no-regret online learning to compute and $\epsilon-$Nash equilibrium using the self-play setup of Theorem 1.

4.2 Counterfactual Regret Minimization

The sequence form allows one to compactly pose a Nash equilibrium as a bilinear saddle point problem (4.2). However, the sequence form is not as convenient as the behavioral form. Fortunately, *counterfactual regret minimization (CFR)* [69] allows for bounding the sequence form regret $R^T_{\hat{\Sigma}_i}$, by the regret over multiple ODPs, one at each information state. Effectively, allow-

[5] Note that it is possible for both players to specify a sequence of actions that can lead to a terminal history but yet are not compatible with each other.

ing one to only deal with behavioral policies and yet minimize regret over the sequence form.

Recall that an ODP requires a set of actions and a set of linear reward functions (see Section 2.2 for details). As in Theorem 1, we construct specific reward functions using the policies picked by both players. At the information state $s \in \mathcal{S}_i$, we define the ODP at s with actions $A(s)$. For the policy profile x selected, we define the reward for $a \in A(s)$ at at s as the counterfactual value of playing a, $v_i^x(s,a)$. The counterfactual value of a is the expected value of playing a assuming that player i plays to reach s and both players play out the rest of the game using x. Formally,

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

where $ha \in \mathcal{H}$ is the history that results from taking action a at history h, and $\eta_i^x(h,z) = 0$ whenever z is unreachable from ha.[6]

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

$$\rho_i^x(s,a) = v_i^x(s,a) - \sum_{a' \in A(s)} x_i(s,a')v_i^x(s,a'). \tag{4.3}$$

The associated instantaneous regret vector is then denoted as $\rho_i^x(s) = (\rho_i^x(s,a))_{a \in A(s)}$. We denote the cumulative counterfactual regret for action a and information state s as $R_i^T(s,a) = \sum_{t=1}^T \rho_i^{x^t}(s,a)$, where $x^t \coloneqq (x_1^t, x_2^t)$ is the profile at time t. The regret vector at s is then $R_i^T(s) = (R_i^T(s,a))_{a \in A(s)}$.

Remark 2. *For convenience we have introduced new notation for the instantaneous regret at state s (4.3), however, this is none other than the expected instantaneous ϕ-regret, $\rho_\phi(x,r)$(2.12), where x is the behavioral policy at state s, $\phi \in \Phi_{EXT}$, and the reward function is $r^t(\cdot) = v_i^{x^t}(s,\cdot)$.*

Zinkevich *et al.* [69] showed how the external regret over the sequence form $R_{\hat{\Sigma}_i}^T$ is upper bounded by the external regrets over each state

[6]Note that if x_{-i} does not allow for player i to reach s then the counterfactual value is 0 for all actions $a \in A(s)$.

$\max_{a \in A(s)} R_i^T(s,a)$.[7]

Theorem 11 (CFR)**.** *For both players, $i \in \{1,2\}$, the regret of i's policies constructed from their ODP learners after T iterations of CFR*

$$R^T_{\hat{\Sigma}_i} \leq \sum_{s \in \mathcal{S}_i} \left(\max_{a \in A(s)} R_i^T(s,a) \right)^+ .$$

Furthermore, the behavioral policy constructed from the average sequence form policy, $\bar{x} \coloneqq (\bar{x}_1, \bar{x}_2)$, is an $\left(R^T_{\hat{\Sigma}_1} + R^T_{\hat{\Sigma}_2}\right)/T$-Nash equilibrium, where

$$\bar{x}_i(s,a) \propto \sum_{t=1}^{T} \sum_{h \in s, a \in A(h)} \eta_i^{x^t}(h) x_i^t(s,a).$$

See Farina *et al.* [17] for the sketch of an alternative proof using the regret circuits framework that is perhaps more intuitive than the proof in the original work.[8]

With Theorem 11 and Remark 2, we can compute a ϵ-Nash equilibrium with any of the approximate regret-matching algorithms from Chapter 3, using the polynomial or exponential link function, and their *plus* variants.

4.3 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 [65] uses a function approximator to predict cumulative

[7] The regret bound shown by Zinkevich, Johanson, Bowling, and Piccione is quite loose, see for example Farina, Kroer, and Sandholm [15] for a tighter bound. Furthermore, the concept of counterfactual regret applies more generally to sequential decision processes (not just extensive-from games), and with possible additional convex losses/concave rewards at the decision nodes [15].

[8] The alternative proof given by Farina, Kroer, and Sandholm provides new insights on the generality of a CFR like approach to online optimization. In fact, it is shown that CFR is a specific instance of regret *decomposition*. Regret decomposition applies more generally, it is applicable when the decision set $\mathcal{X}$ can be constructed by simple convex preserving operations.

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 $x(s) \propto f(\tilde{R}(s))$ 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 [6, 39, 42, 57, 65].

Using Theorem 3 and the CFR Theorem 11, we can derive an improved regret bound with the polynomial link and a new bound with the exponential link.

Corollary 4 (polynomial ($p > 2$))**.** *Given the polynomial link function f with $p > 2$, let $x_i^t(s) \propto f(\tilde{R}_i^{t-1}(s))$ be the policy that f-RCFR assigns to player i at iteration t in information state $s \in \mathcal{S}_i$ and denote the cumulative approximation error in s as $\epsilon_i(s) = \sum_{t=1}^T \left\| g(R_i^{t-1}(s)) - g\left(\tilde{R}_i^{t-1}(s)\right)\right\|_1$, 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, $i \in \{1, 2\}$,*

$$R_{\hat{\Sigma}_i}^T \leq \sum_{s\in\mathcal{S}_i} \sqrt{T(p-1)4U^2(|A(s)|-1)^{2/p} + 2U\epsilon_i(s)}.$$

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

$$R_{\hat{\Sigma}_i}^T \leq |\mathcal{S}_i|\sqrt{T(p-1)4U^2(|\mathcal{A}|-1)^{2/p} + 2U\epsilon_i^*}.$$

Proof. This result follows directly from Theorem 11 and Remark 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 function with $p > 2$ results in the regret bound presented in Theorem 4 for each state specific ODP. Although Theorem 4 is stated with respect to random regrets and counterfactual regret is an expected regret, the analysis of Greenwald *et al.* [23, Corollary 18] allows us to trivially extend our bounds from Section 3.2 to this case (see Section 3.4 for more details). The result then follows trivially from Theorem 11. □

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

Corollary 5 (polynomial $(1 < p \leq 2)$)**.** *Given the polynomial link function f with $p \leq 2$, let $x_i^t(s) \propto f(\tilde{R}_i^{t-1}(s))$ be the policy that f-RCFR assigns to player i at iteration t in information state $s \in \mathcal{S}_i$ and denote the cumulative approximation error in s as $\epsilon_i(s) = \sum_{t=1}^{T} \left\| g(R_i^{t-1}(s)) - g\left(\tilde{R}_i^{t-1}(s)\right) \right\|_1$, 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, $i \in \{1, 2\}$,*

$$R_{\hat{\Sigma}_i}^T \leq \sum_{s \in \mathcal{S}_i} \left(T(2U)^p(|A(s)| - 1) + 2U\epsilon_i(s)\right)^{1/p}.$$

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

$$R_{\hat{\Sigma}_i}^T \leq |\mathcal{S}_i| \left(T(2U)^p(|\mathcal{A}| - 1) + 2U\epsilon_i^*\right)^{1/p}.$$

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 5 in Section 3.2. Given the application of the RRM Theorem by Brown *et al.* [6], and the recent stochastic regret minimization results from Farina, Kroer, and Sandholm [18], our results should lead to a tighter bound when a function approximator learns from sampled counterfactual regret targets.

Corollary 6 (exponential)**.** *Given the exponential link function f with $\tau > 0$, let $x_i^t \propto f(\tilde{R}_i^{t-1}(s))$ be the policy that f-RCFR assigns to player i at iteration t and denote the cumulative approximation error in s as $\epsilon_i(s) =$*

$\sum_{t=1}^{T} \left\| g(R_i^{t-1}(s)) - g\left(\tilde{R}_i^{t-1}(s)\right) \right\|_1$, *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,* $i \in \{1, 2\}$,

$$R_{\hat{\Sigma}_i}^T \leq \sum_{s \in \mathcal{S}_i} \left(\tau \ln|A(s)| + 2U\epsilon_i(s) + \frac{T2U^2}{\tau} \right).$$

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

$$R_{\hat{\Sigma}_i}^T \leq \left(\tau \ln|\mathcal{A}| + 2U\epsilon_i^* + \frac{T2U^2}{\tau} \right).$$

Furthermore, the profile of average sequence weight policies, $\bar{x}^t$, *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 6 has over the bounds of Theorems 4 and 5.

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) [46].

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 [56, 69]. 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 [42], 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.

4.4 f-RCFR+

Given our extension to (Φ, f)-regret-matching+ (Theorems 10 and 9), we can derive bounds for an f-RCFR+ algorithm. f-RCFR+ is the same as f-RCFR except uses an approximate (Φ, f)-regret-matching+ algorithm at each state, $x_i^t(s) \propto f(\tilde{Q}_i^{t-1}(s))$. Where $\tilde{Q}_i^t(s)$ is an approximation of $Q_i^t(s)$, the regret vector attained with the following update rule

$$Q_i^t(s) = (Q_i^{t-1}(s) + \rho_i^{x^t}(s))^+,$$

$Q_i^t(s)$ is sometimes referred to as the "Q-regret" vector at state s. Recalling the upper bound $R_i^T(s) \leq Q_i^T(s)$ and following similar reasoning to Corollary 4 we attain the following results.

Corollary 7. *Given the polynomial link function f with $p > 2$, let $x_i^t(s) \propto f(\tilde{Q}_i^{t-1}(s))$ be the policy that f-RCFR assigns to player i at iteration t in information state $s \in \mathcal{S}_i$ and denote the cumulative approximation error in s as $\epsilon_i(s) = \sum_{t=1}^T \left\| g(Q^{t-1}(s)) - g\left(\tilde{Q}_i^{t-1}(s)\right) \right\|_1$, 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, $i \in \{1, 2\}$,*

$$R_{\hat{\Sigma}_i}^T \leq \sum_{s \in \mathcal{S}_i} \sqrt{T(p-1)4U^2(|A(s)| - 1)^{2/p} + 2U\epsilon_i(s)}.$$

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

$$R_{\hat{\Sigma}_i}^T \leq |\mathcal{S}_i| \sqrt{T(p-1)4U^2(|\mathcal{A}| - 1)^{2/p} + 2U\epsilon_i^*}.$$

Corollary 8. *Given the polynomial link function f with $1 < p \leq 2$, let $x_i^t(s) \propto f(\tilde{Q}_i^{t-1}(s))$ be the policy that f-RCFR assigns to player i at iteration t in information state $s \in \mathcal{S}_i$ and denote the cumulative approximation error in s as*

$\epsilon_i(s) = \sum_{t=1}^{T} \left\| g(Q_i^{t-1}(s)) - g\left(\tilde{Q}_i^{t-1}(s)\right) \right\|_1$, *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,* $i \in \{1, 2\}$,

$$R_{\hat{\Sigma}_i}^T \leq \sum_{s \in \mathcal{S}_i} \left(T(2U)^p(|A(s)| - 1) + 2U\epsilon_i(s)\right)^{1/p}.$$

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

$$R_{\hat{\Sigma}_i}^T \leq |\mathcal{S}_i| \left(T(2U)^p(|\mathcal{A}| - 1) + 2U\epsilon_i^*\right)^{1/p}.$$

We close the chapter with a few remarks. First, the Corollaries 7 and 8, are very similar to Corollaries 4 and 5, except for the the function approximation error. One is measured with respect to $R_i^t(s)$ while the other is with respect to $Q_i^t(s)$. Second, using the results from Section 3.2.1, we can replace the error terms in the bounds with terms such as $\left\| R_i^t(s) - \tilde{R}_i^t(s) \right\|$ or $\left\| Q_i^t(s) - \tilde{Q}_i^t(s) \right\|$. Third, f-RCFR+ with a polynomial link function with $p = 2$, corresponds to Morrill's RCFR+ algorithm [42].

Chapter 5

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 [55] and imperfect information goofspiel [36] with linear function approximation. We then compare f-RCFR with f-RCFR+ using the same hyperparameters, with the only difference being the regret targets approximated by the function approximator, as explained in Section 4.4.

5.1 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

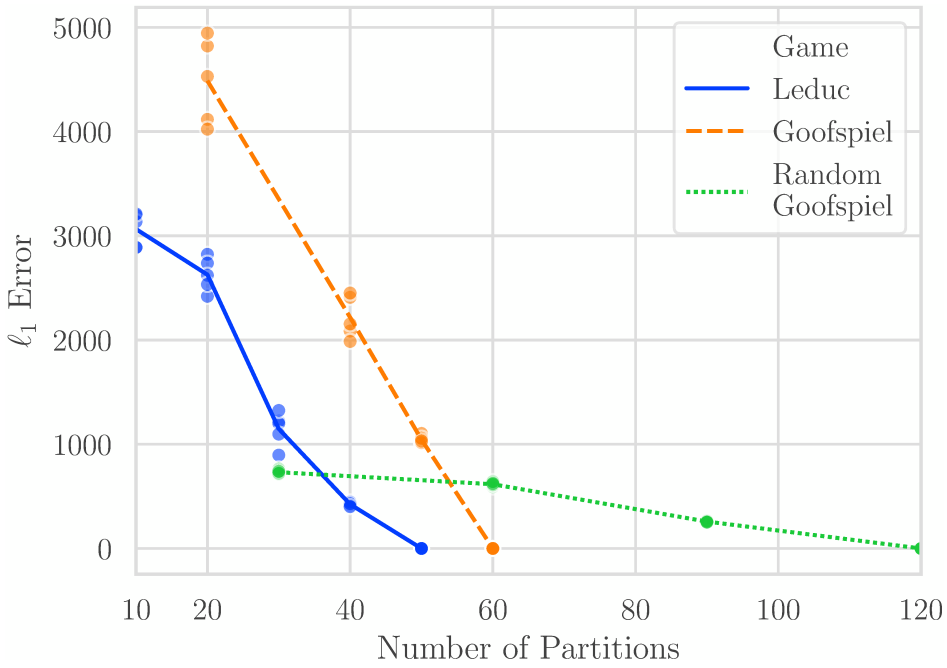

Figure 5.1: The cumulative counterfactual regret estimation error accumulated over time and information states for select f-RCFR instances in Leduc hold'em poker, goofspiel, and random goofspiel. For each game and setting of the number of partitions, we select the link function and the parameter with 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.

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

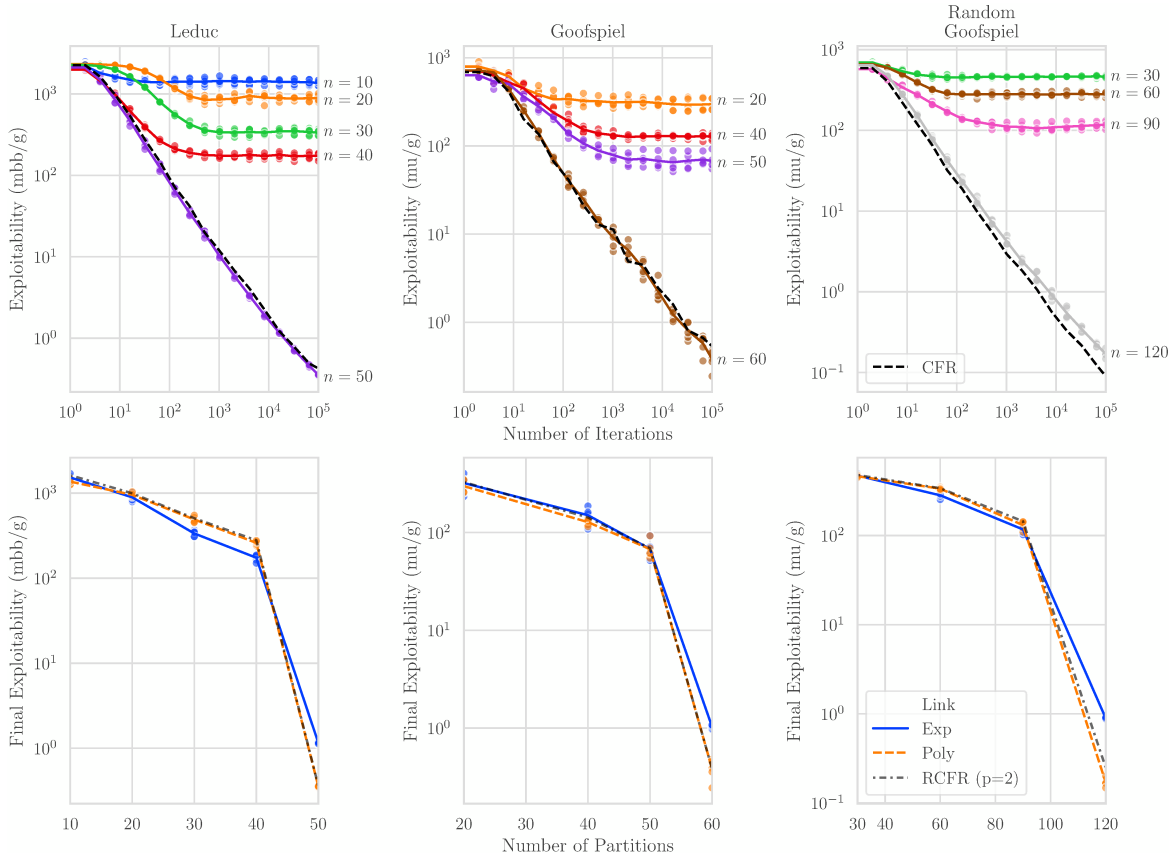

Figure 5.2: (top) The exploitability of the average strategy profile of tabular CFR and f-RCFR instances during the first 100K-iterations in Leduc hold'em (top left), goofspiel (top center), and random goofspiel (top right). For each setting of the number of partitions, we show the performance of the 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).

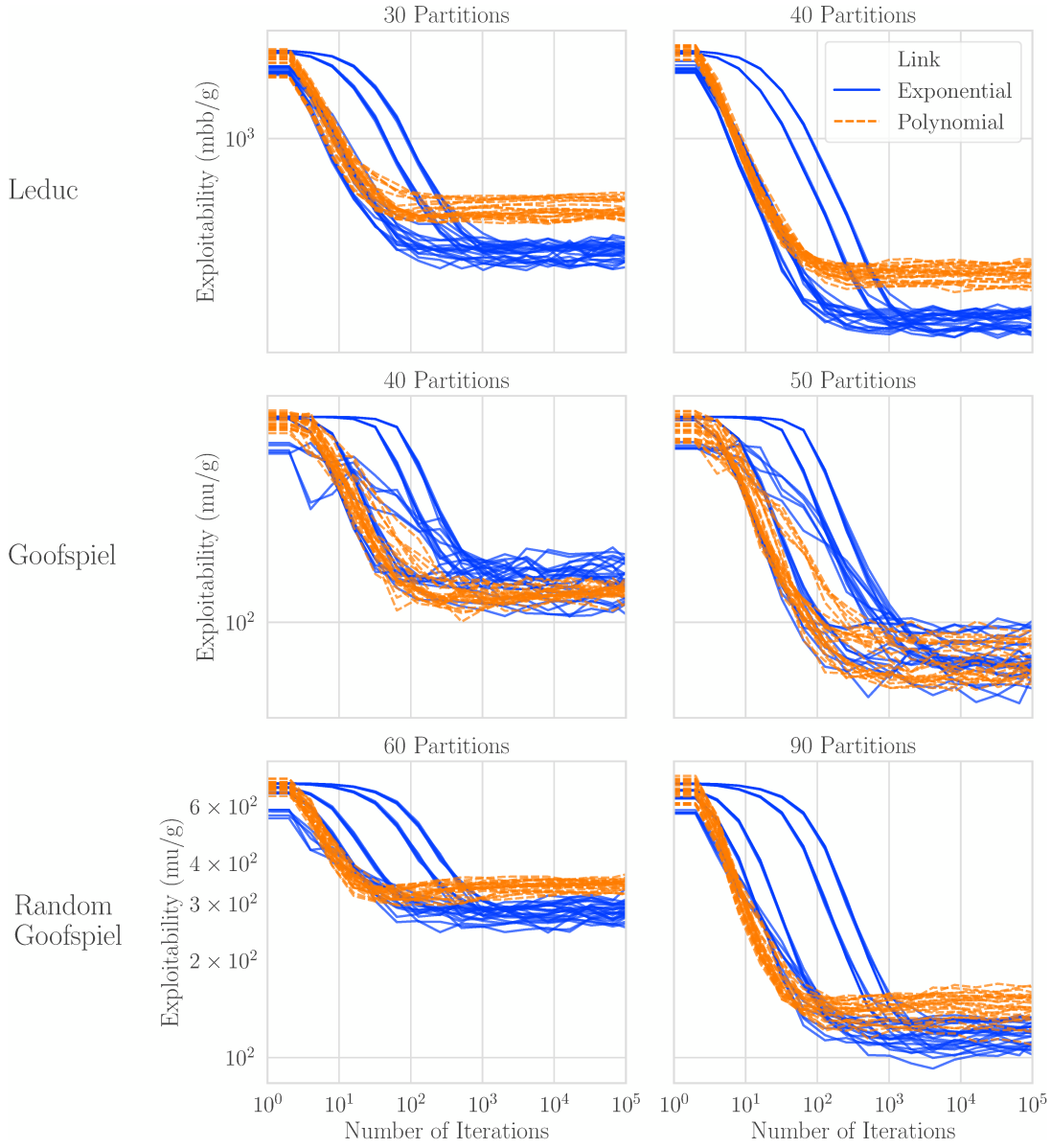

Figure 5.3: Exploitability of the average strategy profile for all configurations and runs with the exponential and polynomial link functions. The exponential link function achieves a lower exploitability than the polynomial link function when a moderate number of partitions (30 or 40) are used in Leduc 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).

on demand, similarly to Steinberger [57].

5.2 Games

In Leduc hold'em poker [55], 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 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 [36]. 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* [38] implementations of these games.

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

5.3 Parameters

From Theorems 1 and 11, 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 Sections 3.2 and 3.3, provide an exploitability bound for f-RCFR and f-RCFR+, for different link functions f, and when estimates of counterfactual regrets are used at each information state in place of true values (Corollaries 4, 5, 6, 9, and 10).

Most notably, the appearance of function approximator error within the regret bounds in Sections 3.2 and 3.3 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 5.1).

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 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.

5.4 Results and Analysis

Figure 5.2 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 5.1).[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 5.3, 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 5.3, center and Figure 5.2, 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 5.2 (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.

[1] We include statistical significance tests for Figures 5.2, 5.4, and 5.5 in the appendix (Section A.3).

5.4.1 f-RCFR+

Similar to our f-RCFR experiments we test the f-RCFR+ algorithm with the same link function and hyper-parameters on the games Leduc, goofspiel, and random goofspiel. To highlight the performance difference between f-RCFR+ and f-RCFR, in Figures 5.4 and 5.5, for each link function we show the exploitability of all the f-RCFR+ instances (dotted lines) with varying partition sizes, along with their f-RCFR counterparts (solid lines). For the polynomial link function (Figure 5.4), all instances of f-RCFR+ outperform their f-RCFR counterparts, except when $p = 1.1$. Interestingly, none of the f-RCFR+ instances plateau except for in the game of Leduc with large function approximation error (20 partitions). This suggests a more efficient use of the function approximator; indeed, in Figure 5.6 (right) the funtion approximation error accumulated by all the f-RCFR+ instances with a polynomial link function is much lower than all of the f-RCFR ones. This observation corroborates Morrill's experimental results, where using function approximation, in particular regression trees, to learn Q regret vectors with the polynomial link function with $p = 2$ leads to lower function approximation error and superior performance across different games [42]. In addition to the larger approximation error, a disadvantage for f-RCFR with the polynomial link function is the irrelevance of negative values in the regret vectors $R(s)$; a policy is invariant to negative values, therefore there is no need to accurately predict them. In a sense f-RCFR+ is also wasting capacity by learning targets of value 0, however, no capacity of the function approximator is spent on distinguishing between the bad actions for which the regret is 0.

Despite the much lower approximation error for f-RCFR+, it is likely not the major contributor to the observed performance gain. CFR with regret-matching+ at each state, *i.e.*, f-RCFR+ with $p = 2$ and zero function approximation error, is known to converge an order of magnitude faster than CFR in practice [10, 42, 59].[2] The performance gain of f-RCFR+ is therefore likely

[2]Tabular f-RCFR+ with $p = 2$ is CFR with regret-matching+ at each information set and is similar to CFR+ [59]. In the current literature CFR+ refers to a collection of modifications in addition to using regret-matching+, including linear weighting of iterates

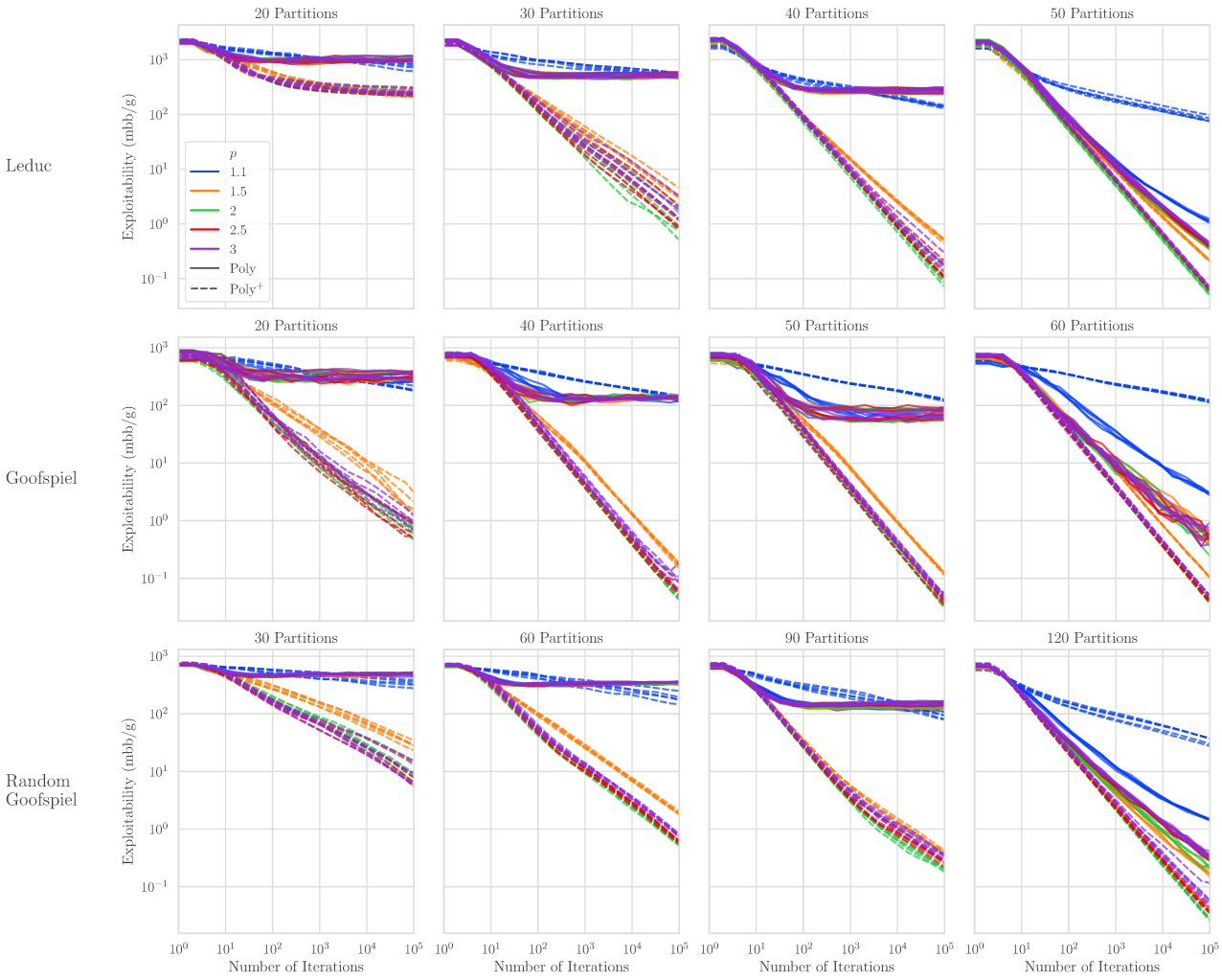

Figure 5.4: Exploitability of the average strategy profile for all instances of the polynomial link function and all runs. For each game, Leduc (top), goofspiel (center), and random goofspiel (bottom), f-RCFR+ outperforms f-RCFR with much lower exploitability.

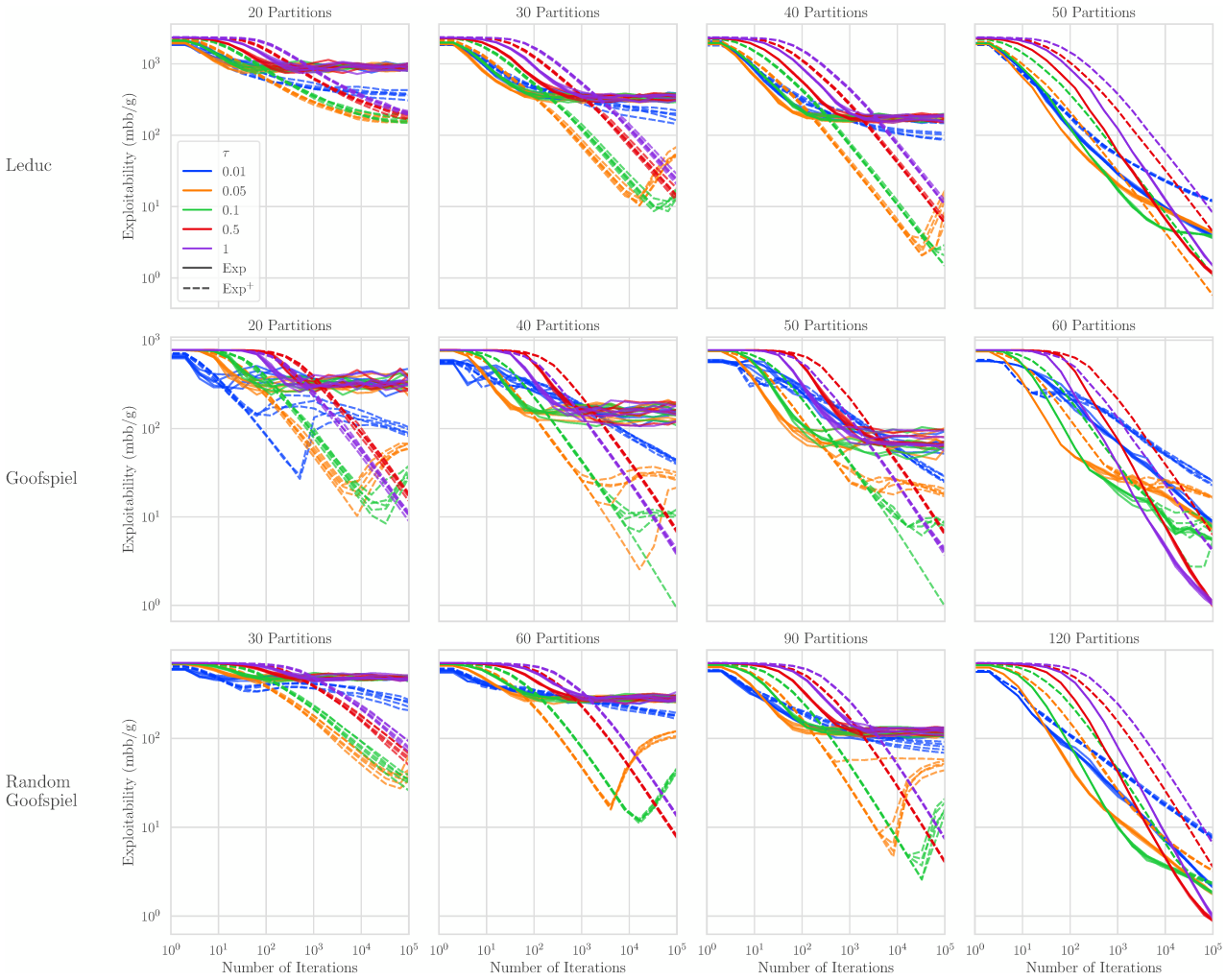

Figure 5.5: Exploitability of the average strategy profile for all instances of the exponential link function and all runs. For each game, Leduc (top), goofspiel (center), and random goofspiel (bottom), f-RCFR+ outperforms f-RCFR when there is large approximation error (columns 1-3).

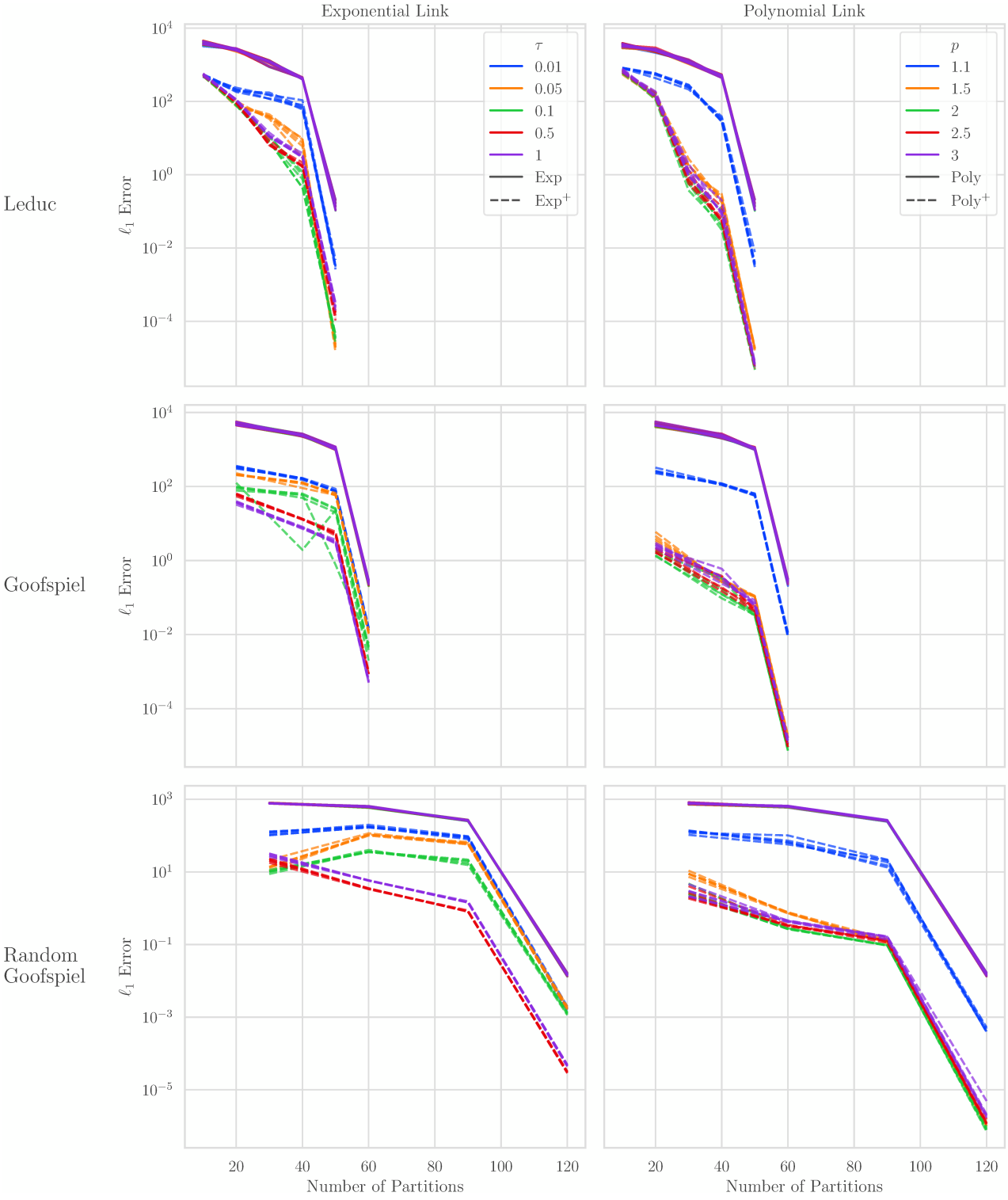

Figure 5.6: Cumulative function approximation error for f-RCFR+ (dotted lines) and f-RCFR (solid lines) with different different partitions and exponential link function (left) and polynomial link function (right). For each game, Leduc (top), goofspiel (center), random goofspiel (bottom), f-RCFR+ accumulates much lower approximation error than f-RCFR.

due to the acceleration properties of regret-matching+.[3]

Unlike the polynomial link function, there is no regret bound when using the exponential link function with Q-regrets. However, Figure 5.5 suggests some convergence properties might exist given the decrease in exploitability across all the partitions and games. In some games the exploitability is seen to increase after some iterations, this is expected and is observed even in f-RCFR with an exponential link function if the temperature τ is not properly tuned. Similar to the polynomial link function, we observe lower function approximation error with Q-regret targets and the exponential link function (Figure 5.6, left). However, unlike the polynomial link, the performance gain of f-RCFR+ with the exponential link depends on the function approximation error. With any degree of function approximation error f-RCFR+ outperforms f-RCFR (first three columns of Figure 5.5). When the function approximation error is negligible then both algorithms perform comparably (last column of Figure 5.5).

5.4.2 External Sampling

The main focus thus far has been scaling regret minimization in extensive form games by combining function approximation with CFR. For very large games, function approximation is not enough, computing the counterfactual value $v_i^x(s, a)$ requires a traversal of the whole subtree below state s. A more scalable approach is to approximate the counterfactual values via sampling. The Monte-Carlo counterfactual regret minimization algorithm (MCCFR) [36, 37], constructs unbiased estimates of the counterfactual values $v_i^x(s, a)$ and instantaneous regrets $\rho_i^x(s, a)$ by sampling a block of terminal histories $Q \subset \mathcal{Z}$. For a given sampled block Q, the sum in computing the counterfactual value $v_i^x(s, a)$ (Equation 4.2) is modified to be a sum over terminal histories that are in the sampled block. Ultimately, the algorithm only updates states that can reach a terminal history in the sampled block.

and alternating updates. f-RCFR+ is closest to Morrill's RCFR+ without bootstrapping.

[3]Understanding the acceleration of regret-matching+ is still an open problem, however, recently it has been shown that there exists a game where convergence is slower than $O(1/T)$ [16]; thus, it cannot be an optimistic method [48, 58].

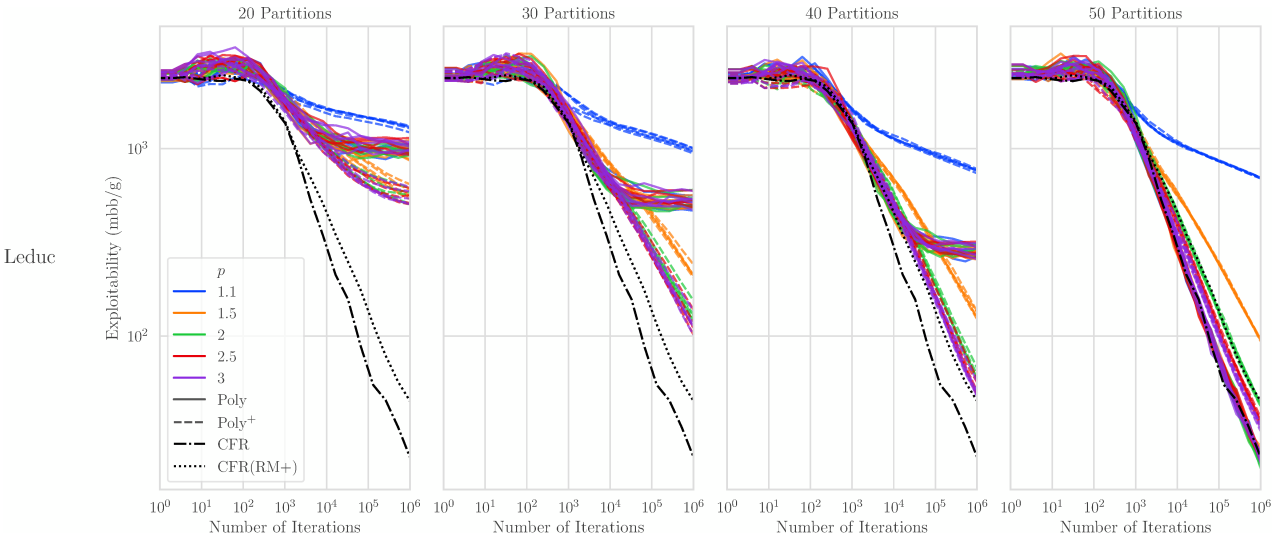

Figure 5.7: Exploitability of the average profile in the game of leduc for f-RCFR (solid lines) and f-RCFR+ (dotted lines) with external sampling and the the polynomial link function. Except for partition size 50 and when $p = 1.1$, f-RCFR+ achieves lower exploitability. For reference, exploitabiliy of tabular CFR and tabular CFR(RM+) are included.

MCCFR defines a family of algorithms, each instance within the family uses a particular method for sampling a block of histories at each iteration. Three popular algorithms are chance, outcome, and external sampling. Chance sampling entails sampling actions made by chance (*e.g.*, cards dealt by a dealer in poker) and taking the sampled block Q to be any terminal history reachable by the actions sampled. Outcome sampling samples one single outcome, or trajectory of the game from start to end by sampling an action for every player (including chance). External sampling samples a block for player i by sampling the actions *external* to that player, *i.e.*, the actions of the other players (including chance).

In Figures 5.7 and 5.8, we compare f-RCFR with f-RCFR+ in the game of Leduc with approximate counterfactual regrets constructed by MCCFR with external sampling.[4] For the polynomail link function (Figure 5.7), f-RCFR+ outperforms f-RCFR when there is function approximation error (20, 30, and

[4]For states that are not updated during one sampling iteration, a target value of 0 is assigned.

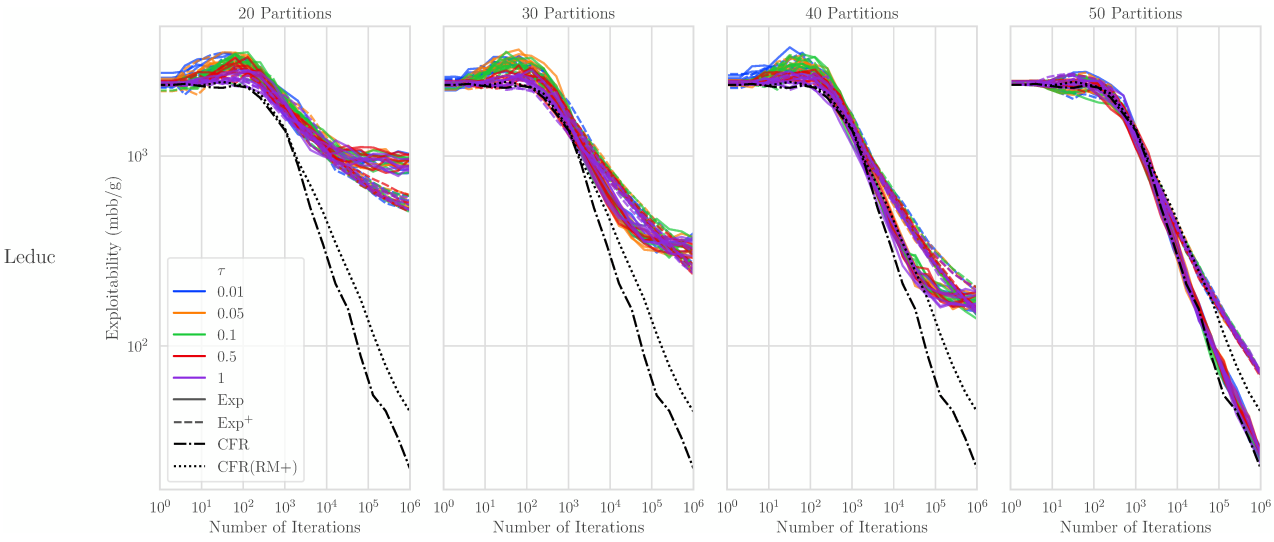

Figure 5.8: Exploitability of the average profile in the game of leduc for f-RCFR (solid lines) and f-RCFR+ (dotted lines) with external sampling and the exponential link function. Except for low function approximation error (50 partitions), f-RCFR+ achieves lower exploitability. For reference, exploitabiliy of tabular CFR and tabular CFR(RM+) are included.

40 partitions). Similar to the case without sampling, f-RCFR+ exhibits more efficient use of the function approximator capacity, the exploitability does not plateau except with coarse function approximation (20 partitions). As shown in Figure 5.9, the cumulative appoximation error is much lower for f-RCFR+ than f-RCFR. Interestingly, f-RCFR+ performs worse than f-RCFR with low function approximation error (Figure 5.7, 50 partitions). With 50 partitions in Leduc, the algorithms are close to tabular performance. Indeed, the tabular algorithm CFR (tabular f-RCFR with $p = 2$) outperforms CFR(RM+) (tabular f-RCFR+ with $p = 2$, see Figure 5.7). The loss in performance is consistent with the results of Burch [10], where it is shown that CFR+ (regret-matching+ and other modifications) fail to outperform CFR in the presence of sampling. Therefore it is surprising that f-RCFR+ presents tangible benefits when the algorithm is far from tabular. This suggests that the more efficient use of the function approximator is responsible for the performance gain of f-RCFR+.

Similar to the polynomial link function, f-RCFR+ with the exponential

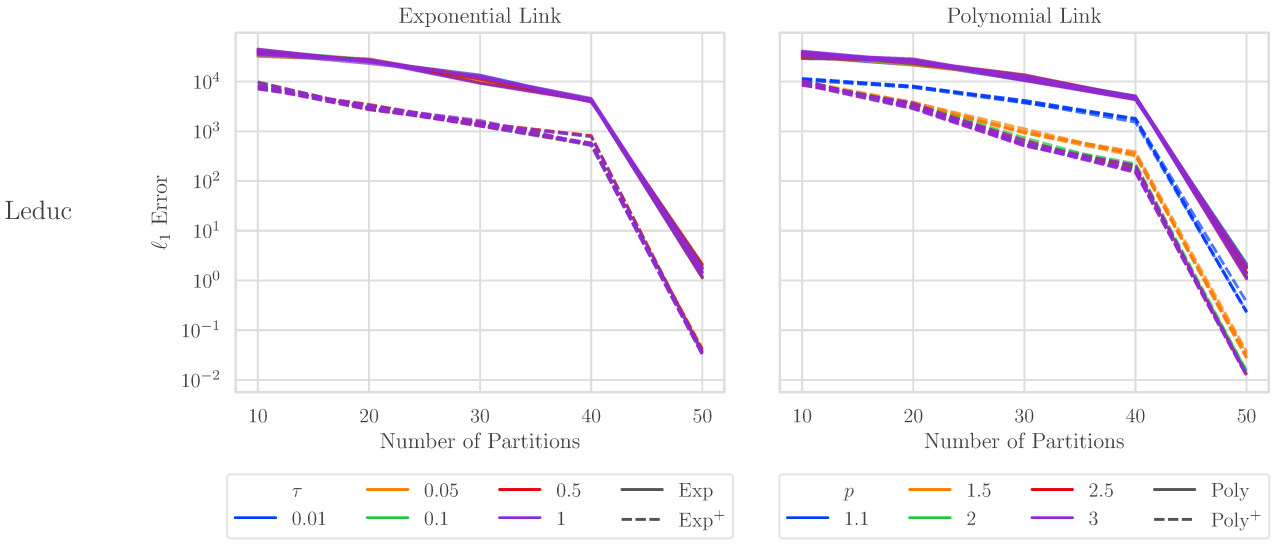

Figure 5.9: Cumulative function approximation error for f-RCFR (solid lines) and f-RCFR+ (dotted lines) with external sampling and the exponential link function (left) and the polynomial link function (right). For all instances, f-RCFR+ accumulates lower function approximation error.

link function accumulates lower function approximation error (Figure 5.9). However, the performance gain is not as prominent, both f-RCFR and f-RCFR+ attain similar exploitability except when there is coarse function approximation (Figure 5.8, 20 and 30 partitions).

Chapter 6

Conclusion

In this thesis, we generalize existing regret bounds with function approximation, allowing for different link functions, including the polynomial and exponential link functions, and regret metrics, including external and internal regret. Furthermore, we provide regret bounds for a new class of algorithms – approximate (Φ, f)-regret-matching+, a generalization regret-matching+. The generalization to different link functions and the new class of algorithms allows us to construct regret bounds for a general f-RCFR algorithm and f-RCFR+ algorithm, respectively. 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 and 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. In most cases, f-RCFR+ outperforms f-RCFR except with the polynomial link function and $p = 1.1$. In all cases the f-RCFR+ algorithm provides easier targets to learn than f-RCFR, permitting much lower function approximation error and faster convergence to an approximate Nash equilibrium. Within the family of f-RCFR algorithms, 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 [11] or extensive-form correlated equilibria [63] with function approximation.

NeuRD [46] 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, and if the their f-RCFR+ alternatives.

Finally, it would be interesting to test whether using Q-regrets improves performance beyond computing Nash equilibria in zero-sum games, such as in computing correlated equilibria N-player general-sum games, or single agent control problems.

References

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

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

[3] D. Blackwell. "An analog of the minimax theorem for vector payoffs." In: *Pacific Journal of Mathematics* 6 (1956), pp. 1–8.

[4] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. "Heads-up limit hold'em poker is solved." In: *Science* 347.6218 (2015), pp. 145–149.

[5] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

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

[7] Noam Brown and Tuomas Sandholm. "Superhuman AI for heads-up no-limit poker: Libratus beats top professionals." In: *Science* 359.6374 (2018), pp. 418–424.

[8] Noam Brown and Tuomas Sandholm. "Superhuman AI for multiplayer poker." In: *Science* 365.6456 (2019), pp. 885–890.

[9] Sébastien Bubeck et al. "Convex Optimization: Algorithms and Complexity." In: *Foundations and Trends® in Machine Learning* 8.3-4 (2015), pp. 231–357.

[10] Neil Burch. "Time and Space: Why Imperfect Information Games are Hard." PhD thesis. University of Alberta, 2017.

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

[12] Antonin Chambolle and Thomas Pock. "A first-order primal-dual algorithm for convex problems with applications to imaging." In: *Journal of mathematical imaging and vision* 40.1 (2011), pp. 120–145.

[13] Ching-An Cheng, Remi Tachet Combes, Byron Boots, and Geoff Gordon. "A reduction from reinforcement learning to no-regret online learning." In: *International Conference on Artificial Intelligence and Statistics*. 2020, pp. 3514–3524.

[14] Ryan D'Orazio, Dustin Morrill, James R Wright, and Michael Bowling. "Alternative Function Approximation Parameterizations for Solving Games: An Analysis of f-Regression Counterfactual Regret Minimization." In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. 2020, pp. 339–347.

[15] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. "Online convex optimization for sequential decision processes and extensive-form games." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 1917–1925.

[16] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. "Optimistic Regret Minimization for Extensive-Form Games via Dilated Distance-Generating Functions." In: *Advances in Neural Information Processing Systems*. 2019, pp. 5222–5232.

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

[18] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. "Stochastic regret minimization in extensive-form games." In: *arXiv preprint arXiv:2002.08493* (2020).

[19] Yoav Freund and Robert E Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139.

[20] Sam Ganzfried and Tuomas Sandholm. "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*. 2013.

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[22] Geoffrey J Gordon. *No-regret algorithms for structured prediction problems*. Tech. rep. CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 2005.

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

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

[25] Amy Greenwald, Zheng Li, and Warren Schudy. "More Efficient Internal-Regret-Minimizing Algorithms." In: *COLT*. 2008, pp. 239–250.

[26] James Hannan. "Approximation to Bayes risk in repeated play." In: *Contributions to the Theory of Games* 3 (1957), pp. 97–139.

[27] S. Hart and A. Mas-Colell. "A Simple Adaptive Procedure Leading to Correlated Equilibrium." In: *Econometrica* 68.5 (2000), pp. 1127–1150.

[28] Elad Hazan et al. "Introduction to online convex optimization." In: *Foundations and Trends® in Optimization* 2.3-4 (2016), pp. 157–325.

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

[30] Samid Hoda, Andrew Gilpin, Javier Pena, and Tuomas Sandholm. "Smoothing techniques for computing Nash equilibria of sequential games." In: *Mathematics of Operations Research* 35.2 (2010), pp. 494–512.

[31] Michael Johanson, Neil Burch, Richard Valenzano, and Michael Bowling. "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. 2013, pp. 271–278.

[32] Daphne Koller, Nimrod Megiddo, and Bernhard Von Stengel. "Efficient computation of equilibria for extensive two-person games." In: *Games and economic behavior* 14.2 (1996), pp. 247–259.

[33] Christian Kroer. *Economics, AI, and Optimization Lecture Note 4: Blackwell Approachability and Regret Matching*. URL: `http://www.columbia.edu/~ck2945/files/s20_8100/lecture_note_4_blackwell_rm_rmp.pdf`.

[34] Christian Kroer, Kevin Waugh, Fatma Kılınç-Karzan, and Tuomas Sandholm. "Faster algorithms for extensive-form game solving via improved smoothing functions." In: *Mathematical Programming* (2018), pp. 1–33.

[35] HW Kuhn. *Extensive Games and the Problem of Information," Contributions to the Theory of Games, II (Annals of Mathematics Studies).* 1953.

[36] Marc Lanctot. "Monte Carlo Sampling and Regret Minimization for Equilibrium Computation and Decision-Making in Large Extensive Form Games." PhD thesis. Edmonton, Alberta, Canada: Department of Computing Science, University of Alberta, June 2013.

[37] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. "Monte Carlo sampling for regret minimization in extensive games." In: *Advances in neural information processing systems*. 2009, pp. 1078–1086.

[38] Marc Lanctot et al. "OpenSpiel: A Framework for Reinforcement Learning in Games." In: *CoRR* abs/1908.09453 (2019). arXiv: 1908.09453 [cs.LG]. URL: http://arxiv.org/abs/1908.09453.

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

[40] Edward Lockhart, Marc Lanctot, Julien Pérolat, Jean-Baptiste Lespiau, Dustin Morrill, Finbarr Timbers, and Karl Tuyls. "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, July 2019, pp. 464–470. DOI: 10.24963/ijcai.2019/66. URL: https://doi.org/10.24963/ijcai.2019/66.

[41] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lis, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. "Deepstack: Expert-level artificial intelligence in heads-up no-limit poker." In: *Science* 356.6337 (2017), pp. 508–513.

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

[43] John F Nash et al. "Equilibrium points in n-person games." In: *Proceedings of the national academy of sciences* 36.1 (1950), pp. 48–49.

[44] Yurii Nesterov. *Lectures on convex optimization*. Vol. 137. Springer, 2018.

[45] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani, eds. *Algorithmic Game Theory*. Cambridge University Press, 2007.

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

[47] Francesco Orabona. "A modern introduction to online learning." In: *arXiv preprint arXiv:1912.13213* (2019).

[48] Sasha Rakhlin and Karthik Sridharan. "Optimization, learning, and games with predictable sequences." In: *Advances in Neural Information Processing Systems*. 2013, pp. 3066–3074.

[49] Dale Schuurmans and Martin A Zinkevich. "Deep learning games." In: *Advances in Neural Information Processing Systems*. 2016, pp. 1678–1686.

[50] Yoav Shoham, Rob Powers, and Trond Grenager. "If multi-agent learning is the answer, what is the question?" In: *Artificial intelligence* 171.7 (2007), pp. 365–377.

[51] 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. "Mastering the game of Go with deep neural networks and tree search." In: *Nature* 529.7587 (2016), pp. 484–489.

[52] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play." In: *Science* 362.6419 (2018), pp. 1140–1144.

[53] Maurice Sion et al. "On general minimax theorems." In: *Pacific Journal of mathematics* 8.1 (1958), pp. 171–176.

[54] Samuel Sokota, Ryan D'Orazio, Khurram Javed, Humza Haider, and Russell Greiner. "Simultaneous Prediction Intervals for Patient-Specific Survival Curves." In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 5975–5981. DOI: 10.24963/ijcai.2019/828. URL: https://doi.org/10.24963/ijcai.2019/828.

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

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

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

[58] Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. "Fast convergence of regularized learning in games." In: *Advances in Neural Information Processing Systems*. 2015, pp. 2989–2997.

[59] Oskari Tammelin. "Solving large imperfect information games using CFR+." In: *arXiv preprint arXiv:1407.5042* (2014).

[60] Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. "Solving heads-up limit Texas Hold'em." In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015.

[61] Oriol Vinyals et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning." In: *Nature* (2019). DOI: 10.1038/s41586-019-1724-z.

[62] Bernhard Von Stengel. "Efficient computation of behavior strategies." In: *Games and Economic Behavior* 14.2 (1996), pp. 220–246.

[63] Bernhard Von Stengel and Françoise Forges. "Extensive-form correlated equilibrium: Definition and computational complexity." In: *Mathematics of Operations Research* 33.4 (2008), pp. 1002–1022.

[64] Mengdi Wang and Yichen Chen. "An online primal-dual method for discounted Markov decision processes." In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE. 2016, pp. 4516–4521.

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

[66] Kevin Waugh, David Schnizlein, Michael Bowling, and Duane Szafron. "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. 2009, pp. 781–788.

[67] BL Welch. "The generalization of Student's problem when several different population variances are involved." In: *Biometrika* (1947).

[68] Martin Zinkevich. "Online convex programming and generalized infinitesimal gradient ascent." In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 2003, pp. 928–936.

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

Appendix A

Appendix

A.1 Existing Results

Below we recall results from Greenwald *et al.* [23] and include the detailed proofs omitted in the main body of the paper.

Lemma 1. *If x is a random vector that takes values in $\mathbb{R}^n$, then $(\mathbb{E}[\max_i x_i])^q \leq \mathbb{E}[\|x^+\|_p^q]$ for $p, q \geq 1$.*

See [Lemma 21][23].

Lemma 2. *Given a reward system $(A, \mathcal{R})$ and a finite set of action transformations $\Phi \subseteq \Phi_{ALL}$, then $\left\|\rho^{\Phi}(a, r)\right\|_p \leq 2U(\mu(\Phi))^{1/p}$ for any reward function $r \in \Pi$.*

The proof is indentical to [Lemma 22][23] except we have that regrets are bounded in $[-2U, 2U]$ instead of $[-1, 1]$. Also note that by assumption $\mathcal{R}$ is bounded.

Theorem 12 (Gordon 2005)**.** *Assume $\langle G, g, \gamma \rangle$ is a Gordon triple and $C : \mathcal{N} \to \mathbb{R}$. Let $X_0 \in \mathbb{R}^n$, let $x_1, x_2, ...$ be a sequence of random vectors over $\mathbb{R}^n$, and define $X_t = X_{t-1} + x_t$ for all times $t \geq 1$.*
If for all times $t \geq 1$,

$$\langle g(X_{t-1}), \mathbb{E}[x_t | X_{t-1}] \rangle + \mathbb{E}[\gamma(x_t) | X_{t-1}] \leq C(t) \quad a.s.$$

then, for all times $t \geq 0$,

$$\mathbb{E}[G(X_t)] \leq G(X_0) + \sum_{\tau=1}^{t} \mathbb{E}[C(\tau)].$$

It should be noted that the above theorem was originally proved by Gordon [22].

A.2 Proofs

An important observation of Theorem 2 is the following corollary:

Corollary 9. *For a reward system $(A, \mathcal{R})$, finite set of action transformations $\Phi \subseteq \Phi_{ALL}$, and two link functions f and f', if there exists a strictly positive function $\psi : \mathbb{R}^{|\Phi|} \to \mathbb{R}$ such that $f'(x) = \psi(x)f(x)$ then for any $\epsilon \in \mathbb{R}$, an approximate (Φ, f)-regret-matching algorithm satisfies*

$$\langle f'(R^{\Phi}_{t-1}(h)), \mathbb{E}_{a\sim L_t(h)}[\rho^{\Phi}(a,r)]\rangle \leq 2U \left\| f'(R^{\Phi}_{t-1}) - f'(\tilde{R}^{\Phi}_{t-1}) \right\|_1 .$$

Proof. The reasoning is similar to [Lemma 20][23]. The played fixed point is the same under both link functions, thus following the same steps to Theorem 1 provides the above bound. □

A.3 Statistical Significance Tests

Below we include statistical signifance tests for Figures 5.2, 5.4, and 5.5. The following tables use Welch's two-sided t-test [67], also known as the "unequal variances t-test." The test seeks to test whether samples from two populations have a different mean. It is assumed that both populations are sampled from a Normal distribution, though the *unknown* variances of both populations may differ. Given two sets of samples $\{X_{1i}\}_{i\leq N_1}$, and $\{X_{2i}\}_{i\leq N_2}$, with true means μ_1 and μ_2 respectively, the test statistic is given by

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}},$$

where $\bar{X}_i$ denotes the sample mean for population i and s_i^2 the unbiased sample variance, $s_i^2 = \sum_{j=1}^{N_i} \frac{(X_{ij}-\bar{X}_i)^2}{N_i-1}$. Under the null hypothesis ($\mu_1 = \mu_2$) the test statistic follows the Student's t-distribution with mean zero and approximately

$$\frac{\left(\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}\right)^2}{\frac{s_1^4}{N_1^3-N_1} + \frac{s_2^4}{N_2^3-N_2}} \tag{A.1}$$

degrees of freedom.

In the tables that follow, both the sampled t statistic according to equation A.1, as well as the p-value for the two-sided test are reported. A p-value less than 0.05 means the observed value of the t statistic lies within the critical region and we may reject the null-hypothesis at a 5% level of significance. For the case when the p-value is larger than 0.05 we highlight the cells in bold; in these cases we cannot reject the null hypothesis.

A.3.1 Figure 5.2

For the following three tables (Tables A.1, A.2, A.3), each row (population 1) is tested against each column (population 2) for significance in the difference in the mean. The data in Tables A.1, A.2, A.3, are from the top of Figure 5.2, where for each game and number partitions we consider the the average *final* exploitability over 5 independent runs.

The tables show there is sufficient evidence to reject the null hypothesis for all pairs of number of partitions in each game. As the number of partitions increases the average exploitability decreases.

Figure 5.2 (top Leduc)								
Partitions	20		30		40		50	
	t	p-value	t	p-value	t	p-value	t	p-value
10	7.562	8.16×10^{-5}	20.54	1.32×10^{-5}	24.20	1.19×10^{-5}	28.03	9.64×10^{-6}
20	—		13.50	5.90×10^{-5}	17.90	3.53×10^{-5}	22.59	2.27×10^{-5}
30	—		—		11.56	1.27×10^{-5}	28.03	9.64×10^{-6}
40	—		—		—		23.88	1.82×10^{-5}

Table A.1: Statistical tests for top of Figure 5.2 in Leduc.

Figure 5.2 (top goofspiel)						
Partitions	40		50		60	
	t	p-value	t	p-value	t	p-value
20	8.31	7.80×10^{-4}	10.91	1.33×10^{-4}	14.84	1.20×10^{-4}
40	—		7.63	1.38×10^{-4}	29.79	7.55×10^{-6}
50	—		—		10.16	5.38×10^{-4}

Table A.2: Statistical tests for top of Figure 5.2 in goofspiel.

Figure 5.2 (top random goofspiel)						
Partitions	60		90		120	
	t	p-value	t	p-value	t	p-value
30	17.27	1.93×10^{-7}	40.84	3.40×10^{-10}	69.81	2.52×10^{-7}
60	—		17.17	6.59×10^{-7}	35.00	3.98×10^{-6}
90	—		—		22.20	2.44×10^{-5}

Table A.3: Statistical tests for top of Figure 5.2 in random goofspiel.

The following three tables (Tables A.4, A.5, A.6), test for a significant difference in the mean for the data used in the bottom of Figure 5.2. For each game and each number of partitions, we compare the average final exploitability over five independent runs for the best polynomial f-RCFR (population 1) and best exponential f-RCFR (population 2) instances.

The performance of the best exponential link function and best polynomial link function differ with statistical significance for: 30, 40, and 50 paritions in Leduc; 60 paritions in goofspiel; 60, and 120 partitions in random goofspiel.

Figure 5.2 (bottom Leduc)		
Partitions	t	p-value
10	2.243	**0.0554**
20	-1.37	**0.217**
30	-6.933	2.58×10^{-4}
40	-8.88	2.23×10^{-5}
50	56.53	6.75×10^{-8}

Table A.4: Statistical tests for bottom of Figure 5.2 in Leduc.

Figure 5.2 (bottom goofspiel)		
Partitions	t	p-value
20	0.432	**0.540**
40	1.666	**0.158**
50	0.02555	**0.980**
60	13.86	6.42×10^{-6}

Table A.5: Statistical tests for bottom of Figure 5.2 in goofspiel.

Figure 5.2 (bottom random goofspiel)		
Partitions	t	p-value
30	0.6197	**0.555**
60	-5.454	9.10×10^{-4}
90	-1.658	**0.137**
120	55.28	2.035×10^{-11}

Table A.6: Statistical tests for bottom of Figure 5.2 in random goofspiel.

A.3.2 Figure 5.4

The following three tables (Tables A.7, A.8, A.9), test for statistical significance in Figure 5.4. The tables compare each f-RCFR instance with the *polynomial* link function against their respective f-RCFR+ counterpart with the same link function and link function parameters. For each game, number of partitions, and choice of the link function parameter p, the final average exploitability is compared between f-RCFR (population 1) and f-RCFR+ (population 2) over 5 independent runs.

Figure 5.4 (Leduc)										
Partitions	$p = 1.1$		$p = 1.5$		$p = 2$		$p = 2.5$		$p = 3$	
	t	p-value	t	p-value	t	p-value	t	p-value	t	p-value
20	5.033	2.94×10^{-3}	28.65	4.34×10^{-7}	23.07	1.30×10^{-6}	16.33	2.30×10^{-5}	18.04	7.41×10^{-6}
30	−0.662	**0.527**	25.93	1.30×10^{-5}	25.16	1.48×10^{-5}	23.98	1.79×10^{-5}	26.47	1.20×10^{-5}
40	17.15	4.81×10^{-6}	26.04	1.29×10^{-5}	23.0	2.12×10^{-5}	27.52	1.04×10^{-5}	23.96	1.80×10^{-5}
50	−19.15	4.38×10^{-7}	30.46	5.85×10^{-8}	31.72	3.04×10^{-6}	44.85	1.11×10^{-6}	47.15	7.79×10^{-7}

Table A.7: Statistical tests for Figure 5.4 in Leduc.

Figure 5.4 (goofspiel)										
Partitions	$p = 1.1$		$p = 1.5$		$p = 2$		$p = 2.5$		$p = 3$	
	t	p-value	t	p-value	t	p-value	t	p-value	t	p-value
20	5.07	3.65×10^{-3}	16.96	7.04×10^{-5}	15.83	9.32×10^{-5}	19.62	3.97×10^{-5}	12.85	2.11×10^{-4}
40	−2.34	**0.0529**	68.31	2.75×10^{-7}	35.50	3.76×10^{-6}	37.50	3.02×10^{-6}	43.25	1.71×10^{-6}
50	−8.073	5.31×10^{-4}	12.54	2.33×10^{-4}	11.22	3.60×10^{-4}	12.53	2.33×10^{-4}	14.91	1.18×10^{-4}
60	−51.10	8.60×10^{-7}	23.29	2.00×10^{-5}	8.426	1.08×10^{-3}	15.23	1.04×10^{-4}	6.842	2.39×10^{-3}

Table A.8: Statistical tests for Figure 5.4 in goofspiel.

Figure 5.4 (random goofspiel)										
Partitions	$p = 1.1$		$p = 1.5$		$p = 2$		$p = 2.5$		$p = 3$	
	t	p-value	t	p-value	t	p-value	t	p-value	t	p-value
30	6.33	1.50×10^{-3}	129.70	1.035×10^{-12}	76.69	2.69×10^{-8}	57.69	2.76×10^{-7}	63.63	1.25×10^{-7}
60	8.006	5.00×10^{-4}	62.74	3.86×10^{-7}	232.19	2.06×10^{-9}	169.80	7.21×10^{-9}	99.55	6.10×10^{-8}
90	4.879	1.46×10^{-3}	21.82	2.61×10^{-5}	28.55	8.95×10^{-6}	38.65	2.68×10^{-6}	29.04	8.37×10^{-6}
120	−14.98	1.16×10^{-4}	15.28	1.06×10^{-4}	11.47	4.48×10^{-5}	18.47	4.48×10^{-5}	12.45	4.62×10^{-6}

Table A.9: Statistical tests for Figure 5.4 in random goofspiel.

A.3.3 Figure 5.5

The following three tables (Tables A.10, A.11, A.12), test for statistical significance in Figure 5.4. The tables compare each f-RCFR instance with the *exponential* link function against their respective f-RCFR+ counterpart with the same link function and link function parameter. For each game, number of partitions, and choice of the link function parameter τ, the final average exploitability is compared between f-RCFR (population 1) and f-RCFR+ (population 2) over 5 independent runs.

Figure 5.5 (Leduc)										
Partitions	$\tau = 0.01$		$\tau = 0.05$		$\tau = 0.1$		$\tau = 0.5$		$\tau = 1$	
	t	p-value	t	p-value	t	p-value	t	p-value	t	p-value
20	17.24	2.36×10^{-7}	33.20	3.07×10^{-8}	18.60	2.06×10^{-5}	18.82	2.60×10^{-5}	17.38	5.15×10^{-5}
30	8.78	2.49×10^{-5}	26.66	2.34×10^{-6}	19.99	4.47×10^{-5}	26.77	1.02×10^{-5}	31.94	4.04×10^{-6}
40	5.12	1.43×10^{-3}	29.18	7.82×10^{-7}	22.41	2.30×10^{-5}	22.57	2.27×10^{-5}	18.62	4.81×10^{-5}
50	−84.79	3.64×10^{-12}	167.63	7.23×10^{-9}	46.60	1.27×10^{-6}	−242.9	1.72×10^{-9}	−1170	3.20×10^{-12}

Table A.10: Statistical tests for Figure 5.5 in Leduc.

Figure 5.5 (goofspiel)										
Partitions	$\tau = 0.1$		$\tau = 0.5$		$\tau = 1$		$\tau = 10$		$\tau = 5$	
	t	p-value	t	p-value	t	p-value	t	p-value	t	p-value
20	7.884	1.23×10^{-3}	8.83	8.70×10^{-4}	14.87	1.08×10^{-4}	10.95	3.93×10^{-4}	29.56	7.60×10^{-6}
40	14.88	9.64×10^{-5}	8.885	7.75×10^{-4}	10.90	3.12×10^{-4}	11.29	3.51×10^{-4}	13.03	2.00×10^{-4}
50	5.84	3.85×10^{-3}	6.944	2.15×10^{-3}	10.63	2.42×10^{-4}	11.40	3.37×10^{-4}	9.784	6.10×10^{-4}
60	−24.03	9.88×10^{-6}	−42.58	2.94×10^{-9}	−3.188	0.0312	−229.7	2.14×10^{-9}	−58.34	2.76×10^{-9}

Table A.11: Statistical tests for Figure 5.5 in goofspiel.

Figure 5.5 (random goofspiel)										
Partitions	$\tau = 0.01$		$\tau = 0.05$		$\tau = 0.1$		$\tau = 0.5$		$\tau = 1$	
	t	p-value	t	p-value	t	p-value	t	p-value	t	p-value
30	15.02	6.01×10^{-7}	47.87	3.05×10^{-10}	34.57	2.43×10^{-6}	61.51	1.46×10^{-10}	32.35	3.47×10^{-7}
60	13.30	2.27×10^{-6}	18.95	2.51×10^{-6}	29.07	6.50×10^{-6}	54.27	6.85×10^{-7}	22.65	2.24×10^{-5}
90	5.526	7.06×10^{-4}	13.07	1.03×10^{-5}	20.85	1.03×10^{-5}	26.86	1.14×10^{-5}	27.49	1.04×10^{-5}
120	-47.67	5.13×10^{-7}	-100.6	7.12×10^{-13}	-23.09	1.33×10^{-8}	-257.2	5.83×10^{-11}	-374.2	2.72×10^{-10}

Table A.12: Statistical tests for Figure 5.5 in random goofspiel.