Hindsight Rational Learning for Sequential Decision-Making: Foundations and Experimental Applications

by

Dustin Morrill

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

Doctor of Philosophy

Department of Computing Science
University of Alberta

Abstract

This thesis develops foundations for the development of dependable, scalable reinforcement learning algorithms with strong connections to game theory. I present a version of rationality for learning—one grounded in the learner's experience and connected with the rationality concepts of optimality and equilibrium—that demands resiliency to uncertainty, environmental changes, and adversarial pressures. This notion of *hindsight rationality* is based on *regret*, a well-known concept for evaluating a sequence of decisions with unilateral deviations. I show that in sequential decision-making tasks, there are many natural deviation sets with critical practical differences beyond those previously studied. I design and implement three extensions to the counterfactual regret minimization (CFR) algorithm, one that is observably sequentially hindsight rational for any given subset of deviations within a broad class; a second that generalizes regression CFR; and a third that applies to continuing Markov decision processes and robust optimization tasks.

The first part develops hindsight rationality and the partially observable history process (POHP) formalism for concisely describing multi-agent sequential decision-making from a single agent's perspective. The second part develops the foundations of defining, analyzing, and using deviations in finite-horizon POHPs to develop efficient hindsight rational algorithms, and the practical consequences of designing algorithms around different deviation sets. The third and final part describes experimental applications of these foundations that use function approximation and condensed domain representations to effectively play games and learn cautious behavior in safety challenges.

Preface

This thesis is original work by Dustin Morrill. It contains work from seven conference papers, five published at top conferences:

- the proceedings of the AAAI conference on artificial intelligence (Morrill, D'Orazio, Sarfati, Lanctot, Wright, A. R. Greenwald, et al. 2021),
- the international joint conference on artificial intelligence (Lockhart et al. 2019a),
- the international conference on autonomous agents and multiagent systems (D'Orazio, Morrill, et al. 2020; Hennes et al. 2020), and
- the international conference on machine learning (Morrill, D'Orazio, Lanctot, Wright, Bowling, and A. R. Greenwald 2021).

One (Morrill, A. R. Greenwald, et al. 2022) appears at the AAAI-22 reinforcement learning and games workshop and another (Mohammedalamen et al. 2021) is publicly available on `arXiv.org`. I am a first author on five of these works (D'Orazio, Morrill, et al. 2020; Hennes et al. 2020; Morrill, D'Orazio, Lanctot, Wright, Bowling, and A. R. Greenwald 2021; Morrill, D'Orazio, Sarfati, Lanctot, Wright, A. R. Greenwald, et al. 2021; Morrill, A. R. Greenwald, et al. 2022).

My specific contributions to each paper are outlined here.

- Morrill, D'Orazio, Lanctot, Wright, Bowling, and A. R. Greenwald (2021): I developed all the theoretical and experimental results in consultation with my advisors. My theoretical work built on early work that Ryan and I did together, but the final result is very different from and much more general than that initial work. Aside from small edits, all of the writing is mine.

- Morrill, D'Orazio, Sarfati, Lanctot, Wright, A. R. Greenwald, et al. (2021): I developed all of the theoretical results and game counterexamples, except the extended Shapley's game, in consultation with my advisors. Aside from small edits, all of the writing is mine.

- D'Orazio, Morrill, et al. (2020): I provided the vision for the paper and theory, developed the experiments, and wrote roughly half the text of the paper.

- Hennes et al. (2020): I independently developed the NeuRD algorithm at the same time as my coauthors before we consolidated our work into a single paper. I developed the online learning and extensive-form game analysis of NeuRD, and ran tabular NeuRD experiments. I developed the motivating example where Hedge/NeuRD succeeds where softmax policy gradient fails. I wrote sections of the paper corresponding to these contributions.

- Lockhart et al. (2019a): I proved and wrote Lemma 2, wrote about two variations of TabularED in the "Imperfect Information Games" section, wrote Appendix A, and edited the paper.

- Morrill, A. R. Greenwald, et al. (2022): I developed all of the paper in consultation with my advisors.

- Mohammedalamen et al. (2021): I developed all of the theoretical results in consultation with Michael Bowling. I provided the initial idea for the experiments and designed the final version of the experiments together with coauthors. I wrote the code for the driving gridworld environment, with some help from undergraduate Fatima Davelouis Gallardo. I completed preliminary experiments with approximate inference and k-of-N CFR that encouraged us to use neural network ensembles to form beliefs. I advised

Montaser and Alex on the development of the code and on running experiments. I wrote nearly all of the paper and appendix.

Morrill, D'Orazio, Sarfati, Lanctot, Wright, A. R. Greenwald, et al. (2021) and Morrill, D'Orazio, Lanctot, Wright, Bowling, and A. R. Greenwald (2021) have corrections regarding statements of deviation strength for the counterfactual and partial sequence deviations. MacQueen (2022) provides a counterexample with a beneficial external deviation in a counterfactual correlated equilibrium. Corrected versions of these papers are publicly available on `arXiv.org` (Morrill, D'Orazio, Lanctot, Wright, Bowling, and A. Greenwald 2021; Morrill, D'Orazio, Sarfati, Lanctot, Wright, A. Greenwald, et al. 2022).

Not every paper has its own chapter as some contributions in separate papers are grouped to create a more cohesive thesis and to limit redundancy. There are also some enhancements to contributions that were not previously published elsewhere. The hindsight rationality view of learning and agent design presented in Chapter 3 is from Morrill, D'Orazio, Sarfati, Lanctot, Wright, A. R. Greenwald, et al. (2021) while the motivating experiment (Section 3.3) is from Hennes et al. (2020). The remainder of my contributions to Morrill, D'Orazio, Sarfati, Lanctot, Wright, A. R. Greenwald, et al. (2021) are found in Sections 6.2 and 6.4 to 6.8, and those of Hennes et al. (2020) are found in Section 10.4. Equilibrium counterexample games involving behavioral deviations in Section 6.5 are entirely new contributions. Morrill, A. R. Greenwald, et al. (2022) is self-contained in Chapter 4, though this chapter also includes Theorem 6 from Morrill, D'Orazio, Lanctot, Wright, Bowling, and A. Greenwald (2021). The rest of Morrill, D'Orazio, Lanctot, Wright, Bowling, and A. R. Greenwald (2021) is spread throughout Part II in Sections 6.2 and 6.8 and Chapter 7. The regret bound presented for EFR in Section 7.4.4 is a new contribution based on an improved analysis in Section 7.2. D'Orazio, Morrill, et al. (2020) is presented in Chapter 10 minus Section 10.4. Mohammedalamen et al. (2021) is presented in Chapter 9 and my primary contribution to Lockhart et al. (2019a) is Lemma 8, which is presented in the same chapter.

What is play? Play is action done for its own sake. It's in a way the very paradigm of freedom because . . . action done for its own sake is what freedom really consists of. Play and freedom are ultimately the same thing.

– David Graeber (2019)

Acknowledgements

Thanks to my supervisors, Michael Bowling and Amy Greenwald, for their support, advice, and guidance. Mike has been consistent source of thought provoking ideas, fun discussions, and comradery for many years. Amy joined my committee at the same time as my candidacy exam but even though that was only two years before I would defend, she fit a lot of advising into that time. She was eager to understand my work and the background developed by Mike's previous students, and excited to help my research to progress.

Thanks to my research colleagues in Mike's group and the wider Reinforcement Learning and Artificial Intelligence (RLAI) and Alberta Machine Intelligence Institute (Amii) groups for interesting research discussions and friendship. In particular, thanks to everyone on the DeepStack team for an incredibly exciting start to my Ph.D. program.

Thanks to the DeepMind Alberta office, as well as the multi-agent and games groups in London and Paris for a fun and productive internship experience.

Thanks to Amii, the Natural Sciences and Engineering Research Council of Canada (NSERC), and Alberta Innovates for helping to fund this thesis. Thanks to Alberta Treasury Branch for additional funding and the opportunity to explore applied research with Mark Sebestyen. Thanks to Compute Canada for computing resources for experiments.

Thanks to my wife, Melanie, for sharing in my successes and helping me through my disappointments.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The world is filled with *agents* who are independent, perceptive decision-makers with their own incentives and abilities. To successfully navigate the world and fulfill their goals, an agent must recognize their limitations; that their perception of the world is imperfect and that parts of the world are out of their control. They must consider how to pursue their goals in spite of interference by other agents, either incidental or intentional. The complexity and dynamism of the world necessitates continual learning and adaptation, not towards any fixed behavioral plan, but to become better suited for current environmental conditions.

The field of reinforcement learning (RL) studies how to develop autonomous agents that refine their behavior over time in pursuit of a reward signal in complex decision-making settings. The field of game theory is the complementary study of multi-agent interactions and equilibrium behavior where each self-interested agent vies for their own payoff. In order to make progress in game theoretic analyses, it is common to add assumptions about the environment or the other agents. For example, it is typical to assume that each agent is rational, in the sense that they will optimize their own return given their beliefs about each of the other agents. This line of reasoning leads to the idea of *equilibrium*, which is an assignment of behavior to each agent where no agent has any incentive to deviate from their assignment. Another common assumption is that the environment contains exactly two agents and that their payoffs sum to zero, which drastically simplifies the space of equilibria and gives equilibria greater power as a prescriptive concept.

If our goal is to construct automated systems that win two-player, zero-sum games against any unknown opponent, then that system should naturally attempt to approximate a "maximin strategy", which is behavior that maximizes its minimum payoff under a worst-case opponent. However, if we want systems to play multi-player, general-sum games, then maximin strategies are unrealistically pessimistic because they optimize for the unlikely scenario where all of the other players ignore their own payoffs and collude specifically to minimize a single player's payoff. The Nash equilibrium concept is meant to be a more reasonable model of player behavior, however, strategies from Nash equilibria are risky to deploy in multi-player, general-sum games when the strategies of the other players are unknown. Nash equilibria are not interchangeable in these games, so players can play their parts of different Nash equilibria and not be in equilibrium together, and the payoff for a Nash equilibrium strategy could be the minimum payoff. These deficiencies leave us wanting for an objective that is better suited to the goal of designing algorithms for playing multi-player, general-sum games than Nash equilibrium or maximin strategies.

In addition, the reliance on assumptions and the inherently static nature of equilibrium ostensibly puts game theory in conflict with RL ideals. As its name suggests, RL is primarily focused on *learning*, which is inherently a dynamic process of change and adaptation. Ideally, an RL agent would learn everything they need to maximize their cumulative reward from contextual clues provided by the environment and the reward signal itself, without any prior knowledge. Assuming any given environment is filled with a particular number of agents and perhaps additionally assuming properties of their incentives does indeed seem to strain, if not entirely violate, the RL ethos. However, it is also true that much of the theoretical development of RL assumes that there is in fact a single agent in the environment, thereby violating this ethos as well.

This thesis develops foundations for the development of dependable, scalable RL algorithms with strong connections to game theory and a focus on the rationality of behavior during learning without relying on assumptions about other agents, or the absence of other agents, in the environment. Specifically, it attempts to answer the following questions:

> In sequential decision-making settings and with minimal assumptions, how can we define rationality in a learning context and how can computationally restricted learning agents utilize their own experience to strive toward this notion of rationality?

Of particular interest in this thesis are algorithms that perform self-evaluation based on the idea of regret to achieve resiliency in the face of environment non-stationarity and interference from other agents. These methods carefully update the agent's behavior to avoid both chasing fleeting rewards and sluggishly responding to opportunities for improvement. They also do not presuppose the existence of universally optimal behavior that will always be effective, regardless of how the environment or other agents change over time. Counterfactual regret minimization (CFR; Zinkevich, Johanson, et al. 2007) is an example of such an algorithm, and it has been extremely successful as a foundation for constructing solutions and expert players for human-scale games (Bowling et al. 2015; Brown and Sandholm 2018, 2019; Moravčík et al. 2017; Schmid, Moravčík, et al. 2021).

Part I introduces novel perspectives on foundations of RL in non-stationary and multi-agent environments. I present the idea of *hindsight rationality* as a notion of rationality for learning based on *regret*, a well-known concept for evaluating a sequence of decisions with unilateral deviations. I further examine the role of equilibria and their relation to hindsight rationality in the RL context, and establish a new formalism for modeling RL problems. Part II contains the theoretical and algorithmic developments that culminate in the extensive-form regret minimization (EFR) algorithm, a flexible generalization of CFR for a broad and natural set of deviations in sequential decision-making settings. Finally, Part III presents examples of how the ideas in previous chapters can be extended and scaled. Specifically, it shows how CFR can be adapted to a class of infinite-horizon problems, how this version of CFR can be used to automatically learn caution in the face of uncertainty, and how regression CFR (RCFR; Waugh, Morrill, et al. 2015) can be generalized to utilize alternative link functions and deviation sets.

This thesis presents the following specific contributions:

- hindsight rationality, an RL objective based on regret and intimately connected with game theory (Chapter 3, parts originally published in Morrill, D'Orazio, Sarfati, Lanctot, Wright, A. R. Greenwald, et al. 2021 and Hennes et al. 2020[1]);

- the partially observable history process (POHP), a formalism for describing complex, non-stationary, multi-agent environments from a single agent's perspective (Chapter 4, originally published in Morrill, A. R. Greenwald, et al. 2022);

- the behavioral deviations, a broad and natural class of deviations in POHPs, in addition to a thorough examination of the relationships between behavioral deviation subsets and equilibrium concepts (Chapter 6, parts originally published in Morrill, D'Orazio, Sarfati, Lanctot, Wright, A. R. Greenwald, et al. 2021 and Morrill, D'Orazio, Lanctot, Wright, Bowling, and A. R. Greenwald 2021);

- the EFR algorithm that generalizes CFR to any behavioral deviation subset, and time selection regret matching, a complementary regret minimizing algorithm (Chapter 7, parts originally published in Morrill, D'Orazio, Lanctot, Wright, Bowling, and A. R. Greenwald 2021 and Morrill, A. R. Greenwald, et al. 2022);

- the theory and procedure for practical versions of CFR and k-of-N CFR (K. Chen et al. 2012) that apply to continuing Markov decision processes (MDPs) with uncertain reward functions, as well as a procedure using k-of-N CFR that learns to behave cautiously in new situations (Chapter 9, contains work in Mohammedalamen et al. 2021 and published in Lockhart et al. 2019);

[1] I share first-authorship of Hennes et al. (2020).

- the f-RCFR generalization of RCFR and the neural replicator dynamics (NeuRD) instantiation thereof that represents a minimal conversion of the softmax policy gradient algorithm into one that strives for hindsight rationality (Chapter 10, originally published in D'Orazio, Morrill, et al. 2020 and Hennes et al. 2020[2]).

In summary, this thesis presents new perspectives on traditional concepts in artificial intelligence, a more complete understanding of equilibria in sequential decision-making environments, and new scalable learning algorithms with strong performance in multi-agent, sequential decision-making environments. Part I uses the small agent–complex environment idea that is central to RL as a lens to reveal new insights in traditional ideas. No-regret learning and the extensive-form game (EFG) formalism refracted through this lens become hindsight rationality and the POHP formalism, respectively. Part II enriches our understanding of deviations and equilibria in sequential decision-making settings, and reaps the benefits of this new understanding with algorithmic advances and empirical performance improvements. Part III shows that the ideas presented in the previous parts are extensible, and lays out a path toward further extensions and practical applications.

References

Bowling, M., N. Burch, M. Johanson, and O. Tammelin (2015). "Heads-up Limit Hold'em Poker is Solved". In: *Science* 347.6218, pp. 145–149.

Brown, N. and T. Sandholm (2018). "Superhuman AI for Heads-Up No-Limit Poker: Libratus Beats Top Professionals". In: *Science* 359.6374, pp. 418–424.

— (2019). "Superhuman AI for Multiplayer Poker". In: *Science* 365.6456, pp. 885–890.

Chen, K. and M. Bowling (2012). "Tractable Objectives for Robust Policy Optimization". In: *Advances in Neural Information Processing Systems*, pp. 2069–2077.

D'Orazio, R., D. Morrill, J. R. Wright, and M. Bowling (May 2020). "Alternative Function Approximation Parameterizations for Solving Games: An Analysis of f-Regression Counterfactual Regret Minimization". In: *19th International Conference on Autonomous Agents and Multi-Agent Systems.* International Foundation for Autonomous Agents and Multiagent Systems.

Hennes, D., D. Morrill, S. Omidshafiei, R. Munos, J. Perolat, M. Lanctot, A. Gruslys, J.-B. Lespiau, P. Parmas, E. Duéñez-Guzmán, et al. (2020). "Neural replicator dynamics: Multiagent learning via hedging policy gradients". In: *19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2020)*, pp. 492–501.

Lockhart, E., M. Lanctot, J. Pérolat, J.-B. Lespiau, D. Morrill, F. Timbers, and K. Tuyls (2019a). "Computing Approximate Equilibria in Sequential Adversarial Games by Exploitability Descent". In: *28th International Joint Conference on Artificial Intelligence (IJCAI 2019).*

[2] I share first-authorship on both of these papers.

Mohammedalamen, M., D. Morrill, A. Sieusahai, Y. Satsangi, and M. Bowling (2021). “Learning to Be Cautious”. In: *arXiv preprint arXiv:2110.15907*.

Moravčík, M., M. Schmid, N. Burch, V. Lisỳ, D. Morrill, N. Bard, T. Davis, K. Waugh, M. Johanson, and M. Bowling (2017). “DeepStack: Expert-Level Artificial Intelligence in Heads-Up No-Limit Poker”. In: *Science* 356.6337, pp. 508–513.

Morrill, D., R. D’Orazio, M. Lanctot, J. R. Wright, M. Bowling, and A. R. Greenwald (July 2021). “Efficient Deviation Types and Learning for Hindsight Rationality in Extensive-Form Games”. In: *38th International Conference on Machine Learning (ICML 2021)*. Vol. 139. virtual, pp. 7818–7828.

Morrill, D., R. D’Orazio, R. Sarfati, M. Lanctot, J. R. Wright, A. R. Greenwald, and M. Bowling (Feb. 2021). “Hindsight and Sequential Rationality of Correlated Play”. In: *35th AAAI Conference on Artificial Intelligence*. Vol. 35. 6. virtual, pp. 5584–5594.

Morrill, D., A. R. Greenwald, and M. Bowling (2022). “The Partially Observable History Process”. In: *AAAI-22 Workshop on Reinforcement Learning and Games*.

Schmid, M., M. Moravčík, et al. (2021). “Player of Games”. In: *arXiv preprint arXiv:2112.03178*.

Waugh, K., D. Morrill, J. A. Bagnell, and M. Bowling (2015). “Solving Games with Functional Regret Estimation”. In: *29th AAAI Conference on Artificial Intelligence (AAAI-15)*. Vol. 29. 1, pp. 2138–2144.

Zinkevich, M., M. Johanson, M. Bowling, and C. Piccione (Dec. 2007b). “Regret Minimization in Games with Incomplete Information”. In: *Advances in Neural Information Processing Systems (NeurIPS 2007)*. Vancouver, British Columbia, pp. 1729–1736.

Chapter 2

Background

2.1 Introduction

This thesis draws on problem settings and algorithms from three distinct fields: RL, game theory, and online learning. This chapter introduces the basic aspects of game theory and online learning, as well as elementary RL and online learning algorithms that are relevant to discussions in later chapters.

Rather than developing a traditional RL formalism here, such as the Markov decision process (MDP), this thesis introduces a new formalism, the partially observable history process (POHP; pronounced "pop"). The POHP model generalizes the MDP model, as well as various models that account for the presence of multiple agents and imperfect information. The development of the POHP formalism is deferred to Chapter 4.

The background in this chapter is all that is required for Part I, but additional background is introduced at the start of each subsequent Part.

2.2 Games

A *game* is an N player interaction where each player simultaneously chooses a strategy and immediately receives a payoff from a utility function (Neumann et al. 1947). There may also be an extra "chance player", denoted c, who "decides" random events like die rolls. The payoff for each player is determined by the strategies of each player, and we assume that the payoffs are bounded. A game described in this way is called a *normal-form game* (*NFG*).

Each player, $i \in \mathcal{P} \cup \{c\}$, is assigned a set of *pure strategies*, $\mathcal{X}_i$, where $\mathcal{P} = \{j\}_{j=1}^{N}$ is the set of real players. For simplicity, we assume that each set of pure strategies is finite, though we will allow chance's strategy set to be continuous when we later discuss robustness measures. A joint selection of pure strategies from each real player, $x_{\mathcal{P}} = (x_i \in \mathcal{X}_i)_{i \in \mathcal{P}}$, is called a *pure strategy profile.*[1] A pure strategy profile and a chance strategy together determines the game's outcome.

Each player i receives the payoff $\upsilon_i(x_{\mathcal{P}}, x_c) \in [-U, U]$. While the real players choose their strategies (presumably to achieve a large payoff), chance's strategy is sampled according to the *mixed strategy* $\pi_c \in \Delta(\mathcal{X}_c)$, where $\Delta(\mathcal{X}_c)$ is the probability simplex over $\mathcal{X}_c$, and this strategy is given as part of the game definition.[2] We overload $\upsilon_i : x_{\mathcal{P}} \mapsto \mathbb{E}[\upsilon_i(x_{\mathcal{P}}, X_c)]$ as the expected payoff for player i over pure chance strategies $X_c \sim \pi_c$ and $\upsilon_i(\pi_{\mathcal{P}}) = \mathbb{E}_{X_{\mathcal{P}}=(X_j \sim \pi_j)_{j \in \mathcal{P}}}[\upsilon_i(X_{\mathcal{P}})]$ over *mixed strategy profile* $\pi_{\mathcal{P}} = (\pi_i \in \Delta(\mathcal{X}_i))_{i \in \mathcal{P}}$. We drop the subscript on the utility function when it is called as a function of player i's strategy, π_i, parameterized by the strategies for the other players, $\sigma = \pi_{-i} \doteq (\pi_j)_{j \in \mathcal{P} \setminus \{i\}}$, *i.e.*, $\upsilon(\pi_i; \sigma) = \upsilon_i(\pi_{\mathcal{P}})$.

An *optimal strategy* or *best response* for player i is a strategy that achieves the *best response value* $\max_{x_i \in \mathcal{X}_i} \upsilon(x_i; \sigma)$ given mixed strategies for the other players, σ. A *Nash equilibrium* is a mixed strategy profile where all players are playing best responses simultaneously. Said another way, a Nash equilibrium requires that no player can benefit from a unilateral deviation to a different strategy. There are games for which no pure-strategy Nash equilibria exist, *e.g.*, rock-paper-scissors, where the game's three pure strategies have an intransitive dominance relationship. Randomization is sufficient to guarantee the existence of at least one Nash equilibrium expressed as a mixed strategy profile in finite games (Nash 1951).

2.2.1 Nash Equilibria and Maximin Strategies

We can characterize a strategy profile's distance to a Nash equilibrium with the concept of approximate Nash equilibrium. A mixed strategy profile, $\pi_{\mathcal{P}} = (\pi_i)_{i \in \mathcal{P}}$, is an ε-(approximate) Nash equilibrium if no player can gain more than ε in expectation by unilaterally deviating to another strategy, *i.e.*, for each player i, $\max_{\pi^* \in \Delta(\mathcal{X}_i)} \upsilon(\pi^*; \sigma) \leq \upsilon_i(\pi_{\mathcal{P}}) + \varepsilon$, where $\sigma = \pi_{-i}$. If $\varepsilon = 0$, $\pi_{\mathcal{P}}$ is an exact Nash equilibrium.

A Nash equilibrium models jointly rational play that is factored in that players act entirely independently from one another: each chooses a single mixed strategy and sample from it independently. A more pessimistic assumption is that all but one of the players collude to minimize the payoff of the remaining player. A *maximin strategy* for player i, π, maximizes their minimum expected payoff against colluding opponents, *i.e.*,

$$\min_{\sigma^* \in \times_{j \in \mathcal{P} \setminus \{i\}} \Pi_j} \upsilon(\pi; \sigma^*) = \upsilon_i^{\text{MXMN}} \doteq \max_{\pi^* \in \Delta(\Pi_i)} \min_{\sigma^* \in \times_{j \in \mathcal{P} \setminus \{i\}} \Pi_j} \upsilon(\pi^*; \sigma^*).$$

[1] Since the influence of chance will often be marginalized away in this work, we do not include chance's strategy in the definition of a strategy profile, as done in some other works.

[2] A pure strategy can also be represented as a point-mass mixed strategy.

Maximin strategies are also said to be *minimax optimal* since they minimize the maximum loss under negated payoffs, and a strategy π with bounded minimax optimality gap $\upsilon_i^{\text{MXMN}} - \min_{\sigma^* \in \times_{j \in \mathcal{P} \setminus \{i\}} \Pi_j} \upsilon(\pi; \sigma^*) \leq \varepsilon$ is ε-minimax.

Proposition 1. *In a two-player, zero-sum game, an ε-Nash equilibrium, $\pi_{\mathcal{P}}$, is a pair of 2ε-maximin strategies.*

Proof. Let $\pi = \pi_1$ and $\sigma = \pi_2$. We begin by proving that player one's strategy is 2ε-maximin. Since $\pi_{\mathcal{P}}$ is a Nash equilibrium, $\upsilon_2(\pi_{\mathcal{P}}) \geq \max_{\sigma^* \in \Delta(\mathcal{X}_2)} \upsilon(\sigma^*; \pi) - \varepsilon$. Since the game is zero-sum, this inequality can be negated to yield

$$\begin{aligned} -\upsilon_2(\pi_{\mathcal{P}}) &= \upsilon_1(\pi_{\mathcal{P}}) && (2.1) \\ &\leq - \max_{\sigma^* \in \Delta(\mathcal{X}_2)} \upsilon(\sigma^*; \pi) + \varepsilon && (2.2) \\ &\leq \min_{\sigma^* \in \Delta(\mathcal{X}_2)} \upsilon(\pi; \sigma^*) + \varepsilon. && (2.3) \end{aligned}$$

Again, since $\pi_{\mathcal{P}}$ is a Nash equilibrium,

$$\begin{aligned} \max_{\pi^* \in \Delta(\mathcal{X}_1)} \upsilon(\pi^*; \sigma) - \varepsilon \leq \upsilon(\pi; \sigma) &\leq \min_{\sigma^* \in \Delta(\mathcal{X}_2)} \upsilon(\pi; \sigma^*) + \varepsilon && (2.4) \\ \min_{\sigma^* \in \Delta(\mathcal{X}_2)} \upsilon(\pi; \sigma^*) &\geq \max_{\pi^* \in \Delta(\mathcal{X}_1)} \upsilon(\pi^*; \sigma) - 2\varepsilon && (2.5) \\ &\geq \upsilon_1^{\text{MXMN}} - 2\varepsilon, && (2.6) \end{aligned}$$

which proves player one's strategy is within 2ε of minimax optimality and is therefore 2ε-maximin.

Repeating the same argument for player two with the roles of player one and two exchanged proves that player two's strategy is also 2ε-maximin, which proves the initial claim. □

The minimax optimality gap is often called *exploitability* in two-player, zero-sum games since it quantifies the extent to which weaknesses in a given strategy can be exploited by the opponent. We can see this by negating and rearranging Eq. (2.6):

$$\begin{aligned} -\min_{\sigma^* \in \Delta(\mathcal{X}_2)} \upsilon(\pi; \sigma^*) &\leq 2\varepsilon - \upsilon_1^{\text{MXMN}} \\ \max_{\sigma^* \in \Delta(\mathcal{X}_2)} \upsilon(\sigma^*; \pi) &\leq 2\varepsilon + \upsilon_2^{\text{MXMN}} \\ \max_{\sigma^* \in \Delta(\mathcal{X}_2)} \upsilon(\sigma^*; \pi) - \upsilon_2^{\text{MXMN}} &\leq 2\varepsilon. \end{aligned}$$

The exploitability of a strategy profile is the average exploitability of its component strategies. In a two-player, zero-sum game, the sum of exploitabilities simplifies to the sum of best response values as

$$\begin{aligned} &\max_{\sigma^* \in \Delta(\mathcal{X}_2)} \upsilon(\sigma^*; \pi) - \upsilon_2^{\text{MXMN}} + \max_{\pi^* \in \Delta(\mathcal{X}_1)} \upsilon(\pi^*; \sigma) - \upsilon_1^{\text{MXMN}} \\ &= \max_{\sigma^* \in \Delta(\mathcal{X}_2)} \upsilon(\sigma^*; \pi) + \max_{\pi^* \in \Delta(\mathcal{X}_1)} \upsilon(\pi^*; \sigma) + \underbrace{\upsilon_2^{\text{MXMN}} - \upsilon_2^{\text{MXMN}}}_{0}. \end{aligned}$$

The exploitability of an ε-Nash in two-player, zero-sum games is therefore at most ε, since

$$\frac{1}{2}\left(\max_{\pi^* \in \Delta(\mathcal{X}_1)} \upsilon(\pi^*; \sigma) + \max_{\sigma^* \in \Delta(\mathcal{X}_2)} \upsilon(\sigma^*; \pi)\right) \tag{2.7}$$

$$= \frac{1}{2}\left(\max_{\pi^* \in \Delta(\mathcal{X}_1)} \upsilon(\pi^*; \sigma) + \max_{\sigma^* \in \Delta(\mathcal{X}_2)} \upsilon(\sigma^*; \pi) - \overbrace{(\upsilon_1(\pi_{\mathcal{P}}) + \upsilon_2(\pi_{\mathcal{P}}))}^{0}\right) \tag{2.8}$$

$$\leq \frac{1}{2}(\varepsilon + \varepsilon) = \varepsilon. \tag{2.9}$$

The function

$$\pi_{\mathcal{P}} \mapsto \sum_{i \in \mathcal{P}} \max_{\pi^*_i \in \Delta(\mathcal{X}_i)} \upsilon(\pi^*_i; \pi_{-i}) - \upsilon_i(\pi_{\mathcal{P}}) \tag{2.10}$$

generalizes Eq. (2.8) beyond two-player, zero-sum games into a measure of how close a given strategy profile is to Nash equilibrium, and we call this measure *Nash convergence* (*NashConv*). A strategy profile with a NashConv of ε could be improved by ε, which implies this profile is an ε-Nash equilibrium.

2.2.2 Correlated Equilibria

The *correlated equilibrium* concept is a generalization of the Nash equilibrium concept to correlated play.[3] Play in a correlated equilibrium is correlated rather than factored because a correlated equilibrium is a distribution over pure strategy profiles. We can imagine that strategies are jointly sampled from the correlated equilibrium and given to the players to play by a neutral mediator. This scenario is helpful in thinking about how a given correlated equilibrium could be evaluated or executed, but it is worth highlighting that an explicit mediator is not required to implement the behavior of a correlated equilibrium or to explicitly construct one. As the players are given their strategies from the mediator, pure strategy profiles sampled in this manner are called *recommendations* and the mediator's distribution is then a *recommendation distribution.* The equilibrium condition for recommendation distributions is that there is no unilateral deviation from the recommendations that would benefit any player in expectation.

It is common for "deviation" to refer to a switch from one strategy to another, as we saw in the definition of Nash equilibrium. However, in a correlated equilibrium, a player could

[3] In the work this chapter is based on (Morrill, D'Orazio, Lanctot, Wright, Bowling, and A. R. Greenwald 2021; Morrill, D'Orazio, Sarfati, Lanctot, Wright, A. R. Greenwald, et al. 2021), the correlated equilibrium concept was called "mediated equilibrium" to distinguish it from Aumann (1974)'s specific equilibrium type, which is the original example of a correlated equilibrium and has the same name. However, the term "mediated equilibrium" perhaps suggests that an explicit mediator is required to implement such an equilibrium when in fact play can be correlated without a non-player mediator or without the players even knowing anything about each other.

be recommended to play many different strategies, each one in the support of the recommendation distribution. Therefore, the concept of a deviation here is generalized to pure strategy transformation functions. Formally, the benefit to player i of deviating from recommendations $(X_i, X_{-i}) \sim \mu$ sampled from $\mu \in \Delta(\bigtimes_{j \in \mathcal{P}} \mathcal{X}_j)$ according to deviation function $\phi : \mathcal{X}_i \to \mathcal{X}_i$ is $\mathbb{E}[\upsilon(\phi(X_i); X_{-i}) - \upsilon(X_i; X_{-i})]$. An ε-correlated equilibrium with respect to a deviation set profile $(\Phi_i)_{i \in \mathcal{P}}$, where each deviation set is a subset of all possible strategy transformations, is a recommendation distribution where the most any player can gain by unilaterally deviating is ε, *i.e.*, $\max_{i \in \mathcal{P}, \phi \in \Phi_i} \mathbb{E}[\upsilon(\phi(X_i); X_{-i}) - \upsilon(X_i; X_{-i})] \leq \varepsilon$.

Deviation sets constrain the extent to which individual strategy modifications can condition on input strategies, thereby providing a mechanism for varying the strength and character of equilibrium rationality constraints. The set of constant or *external deviations*, $\Phi^{\text{EX}}_{\mathcal{X}_i} = \{\phi^{\to x} : \mathcal{X}_i \to x\}_{x \in \mathcal{X}_i}$, captures the set of deviations to a fixed strategy, which is used in the definition of best response, *i.e.*, $\max_{\phi^{\to x'} \in \Phi^{\text{EX}}_{\mathcal{X}_i}} \upsilon\big(\phi^{\to x'}(x); x_{-i}\big) = \max_{x' \in \mathcal{X}_i} \upsilon(x'; x_{-i})$ for all strategy profiles (x, x_{-i}). Consequently, the definitions of Nash equilibrium and minimax optimality are also based on competitions between a player's actual behavior and the set of external deviations. The set of all pure strategy transformations, $\Phi^{\text{SW}}_{\mathcal{X}_i} = \{\phi : \mathcal{X}_i \to \mathcal{X}_i\}$, is called the set of *swap deviations* and a correlated equilibrium with respect to swap deviations is called an *Aumann-correlated equilibrium* (Aumann 1974). The *internal deviations* (Foster et al. 1999), $\Phi^{\text{IN}}_{\mathcal{X}_i} = \big\{\phi^{x \to x'} \mid \phi^{x \to x'}(\bar{x}) = x' \text{ if } \bar{x} = x \text{ else } \bar{x}\big\}_{x, x' \in \mathcal{X}_i}$, is a special set of deviations that is much smaller than the set of swap deviations ($|\Phi^{\text{IN}}_{\mathcal{X}_i}| = |\mathcal{X}_i|^2$) but it nevertheless subsumes the set of swap deviations strategically. That is, if a player has no beneficial internal deviations, then they do not have a beneficial swap deviation either, and a correlated equilibrium with respect to internal deviations is an Aumann-correlated equilibrium (A. Greenwald, Jafari, et al. 2003). In contrast, the external deviations do not have this property; a player may have a beneficial internal or swap deviation even if they do not have a beneficial external deviation. A correlated equilibrium with respect to external deviations is called a *coarse-correlated equilibrium* (Moulin et al. 1978) and the set of such equilibria is larger than the set of Aumann-correlated equilibria.

2.3 Online Decision Processes

In an *online decision process* (*ODP*), an agent repeatedly chooses from a compact decision set. In this thesis, we always assume that the decision set is a finite set of pure strategies, $\mathcal{X}$. On each round t, the agent chooses a mixed strategy $\pi^t \in \Delta(\mathcal{X})$ and samples a pure strategy $X^t \sim \pi^t$ to play. The agent's sampled strategy is evaluated by a round-dependent, bounded utility function, $\upsilon^t : \mathcal{X} \to [-U, U]$. The agent receives υ^t at the end of the round to learn from but the utility function on the next round can be arbitrarily different, within the payoff bounds. This is the *full monitoring* version of the ODP model and it is the default ODP setting assumed in this work. The *bandit* setting is an ODP model only provides the agent with the payoff for the pure strategy they sampled rather than the entire utility function, and it will also be mentioned occasionally.

In an ODP, the agent's goal is to accumulate a larger payoff over time than any deviation from a pre-defined subset of swap deviations, $\Phi \subseteq \Phi^{\text{sw}}_{\mathcal{X}}$. In this thesis, we typically consider the agent's performance in expectation, which motivates overloading $\upsilon^t(\pi) = \mathbb{E}_{X \sim \pi}[\upsilon^t(X)]$ as the expected payoff under mixed strategy π and $\phi(\pi) \in \Delta(\mathcal{X})$ as the mixed strategy generated by applying deviation ϕ to π. $\phi(\pi)$ is formally defined as the pushforward measure defined by accumulating the probability of each pre-transformation decision that could be sampled from π, *i.e.*, the probability of decision X' under the deviation distribution $\phi(\pi)$ is $[\phi\pi](X') = \sum_{X \in \phi^{-1}(X')} \pi(X)$, where $\phi^{-1} : X' \mapsto \{X \mid \phi(X) = X'\}$ is the pre-image of deviation ϕ.

The expected benefit to the agent of deviation function ϕ on round t is the *instantaneous (expected) regret*, $\rho(\phi, \pi^t; \upsilon^t) = \upsilon^t(\phi(\pi^t)) - \upsilon^t(\pi^t)$. When the sequence of strategies and utility functions is clear, we write $\rho^{1:T}(\phi) = \sum_{t=1}^{T} \rho(\phi, \pi^t; \upsilon^t)$ as shorthand for the cumulative (expected) regret of ϕ. If the positive part of the maximum average (expected) regret over time vanishes, *i.e.*, $\max_{\phi \in \Phi} \lim_{T \to \infty} \frac{1}{T} \left[\rho^{1:T}(\phi)\right]_+ = 0$, where $[\cdot]_+ = \max\{\cdot, 0\}$ is the *ramp function* function, then the agent (equivalently, the sequence of mixed strategies $(\pi^t)_{t=1}^{\infty}$) is *no-regret* with respect to Φ. We would equivalently say that the agent is *no-Φ-regret.*

2.3.1 Approximating Correlated Equilibria

We can use the ODP formalism to study learning in a repeated game and make a connection with correlated equilibrium by considering N ODPs, each derived from a particular player's perspective. Set each player i's ODP decision set to their set of pure strategies, *i.e.*, $\mathcal{X} = \mathcal{X}_i$. On round t of a repeated game, each player i selects a mixed strategy, π^t_i, and samples a pure strategy, $X^t_i \sim \pi^t_i$. For a given player i, the tuple $\sigma^t = \left(\pi^t_j\right)_{j \in \mathcal{P} \setminus \{i\}}$ aggregates the mixed strategies of the other players and a sample $D^t \sim \sigma^t$ is a tuple of independent samples from each mixed strategy, *i.e.*, $D^t = (X^t_j \sim \pi^t_j)_{j \in \mathcal{P} \setminus \{i\}}$. Player i's ODP utility function is then naturally $\upsilon^t(\cdot) = \upsilon(\cdot; D^t)$. To show the influence of the other players' strategies on regret, we parameterize the ρ function by their strategies, *i.e.*, $\rho(\phi, \cdot; D^t) = \rho(\phi, \cdot; \upsilon(\cdot; D^t))$.

After T rounds, the players have generated T mixed strategy profiles that form their *empirical distribution of play*,

$$\mu^T : (x_i)_{i=1}^N \mapsto \frac{1}{T} \sum_{t=1}^{T} \prod_{i=1}^{N} \pi^t_i(x_i).$$

Proposition 2. *If each player in a repeated game has a maximum cumulative regret no more than $f(T) > 0$ after T rounds with respect to a set of deviations Φ in their derived ODP, then the players' empirical distribution of play, μ^T, is an ${}^{f(T)}/_{T}$-correlated equilibrium in that game. Furthermore, if each player is no-Φ-regret, then ${}^{f(T)}/_{T} \to 0$ in the limit as $T \to \infty$.*

Proof. Consider player i's ODP, denoting $\pi^t = \pi^t_i$, $X^t \sim \pi^t$, $\sigma^t = \left(\pi^t_j\right)_{j \in \mathcal{P} \setminus \{i\}}$, and $D^t \sim \sigma^t$ for each round t. Evaluating the benefit of a deviation $\phi \in \Phi$ under μ^T, we can see that

$$\mathbb{E}_{(X,D)\sim\mu^T}[\upsilon(\phi(X); D) - \upsilon(X; D)] \tag{2.11}$$

$$= \mathbb{E}_{t\sim\mathrm{Unif}(\{t'\}^T_{t'=1})}\mathbb{E}\big[\upsilon(\phi(X^t); D^t) - \upsilon(X^t; D^t)\big] \tag{2.12}$$

$$= \mathbb{E}_{t\sim\mathrm{Unif}(\{t'\}^T_{t'=1})}\big[\rho(\phi, \pi^t; \sigma^t)\big] \tag{2.13}$$

$$= \frac{1}{T}\sum_{t=1}^{T} \rho(\phi, \pi^t; \sigma^t), \tag{2.14}$$

where $\mathrm{Unif}(\{t'\}^T_{t'=1})$ is the uniform distribution over the set $\{t'\}^T_{t'=1}$. By assumption, the maximum cumulative regret for player i after T rounds is $f(T)$, *i.e.*, $\max_{\phi\in\Phi} \sum_{t=1}^{T} \rho(\phi, \pi^t; \sigma^t) \leq f(T)$, so according to Eq. (2.14), the maximum benefit of any deviation from μ^T is

$$\max_{\phi\in\Phi} \mathbb{E}_{(X,D)\sim\mu^T}[\upsilon(\phi(X); D) - \upsilon(X; D)] \leq \frac{f(T)}{T}.$$

Since this holds for each player i, μ^T is an ${}^{f(T)}/_{T}$-correlated equilibrium, as claimed. Furthermore, if each player is no-Φ-regret, then $\lim_{T\to\infty} {}^{f(T)}/_{T} = 0$. □

2.3.2 Approximating Nash Equilibria

In a two-player, zero-sum game, no-regret learning can be used to construct ε-Nash equilibria.

Proposition 3. *Folk theorem. If both players in a repeated two-player, zero-sum game have a maximum cumulative external regret (regret with respect to the external deviations) of $f(T)$ after T rounds in their derived ODPs, then their average strategy profile, $\left(\bar{\pi}^T_i \in \Delta(\mathcal{X}_i)\right)^2_{i=1}$, where $\bar{\pi}^T_i(x_i) = \frac{1}{T}\sum_{t=1}^{T} \pi^t_i(x_i)$ for all players i and pure strategies $x_i \in \mathcal{X}_i$, is a ${}^{2f(T)}/_{T}$-Nash equilibrium in that game. Furthermore, if both players are no-external-regret, then $\left(\bar{\pi}^T_i\right)^2_{i=1}$ approaches an exact Nash equilibrium in the limit as $T \to \infty$.*

Proof. Let player one's strategy on round t be $\pi^t = \pi^t_1$ and let player two's be $\sigma^t = \pi^t_2$. Player two's regret bound implies that player two performs nearly as well as each deviation $\phi_2 \in \Phi^{\mathrm{EX}}_{\mathcal{X}_2}$ in hindsight, that is,

$$\sum_{t=1}^{T} \upsilon(\sigma^t; \pi^t) \geq \sum_{t=1}^{T} \upsilon\big(\phi_2(\sigma^t); \pi^t\big) - f(T). \tag{2.15}$$

Player one also achieves this with respect to each deviation $\phi_1 \in \Phi^{\text{EX}}_{\mathcal{X}_1}$. Negating Eq. (2.15) and utilizing the zero-sum property of υ, we see that

$$\sum_{t=1}^{T} \upsilon(\pi^t; \sigma^t) \leq f(T) - \sum_{t=1}^{T} \upsilon\big(\phi_2(\sigma^t); \pi^t\big). \tag{2.16}$$

Starting from player one's version of Eq. (2.15) and replacing $\sum_{t=1}^{T} \upsilon(\pi^t; \sigma^t)$ with Eq. (2.16) yields

$$\sum_{t=1}^{T} \upsilon\big(\phi_1(\pi^t); \sigma^t\big) - f(T) \leq \sum_{t=1}^{T} \upsilon(\pi^t; \sigma^t) \leq f(T) - \sum_{t=1}^{T} \upsilon\big(\phi_2(\sigma^t); \pi^t\big)$$

$$\sum_{t=1}^{T} \upsilon\big(\phi_1(\pi^t); \sigma^t\big) + \upsilon\big(\phi_2(\sigma^t); \pi^t\big) \leq 2f(T). \tag{2.17}$$

For each player i where $\pi^t = \pi^t_i$ and $\sigma^t = \pi^t_{-i}$, we can marginalize the other player's strategy since ϕ_1 and ϕ_2 are external.

$$\begin{aligned}
\max_{\phi_i \in \Phi^{\text{EX}}_{\mathcal{X}_i}} \frac{1}{T} \sum_{t=1}^{T} \upsilon\big(\phi_i(\pi^t); \sigma^t\big) &= \max_{x^* \in \mathcal{X}_i} \frac{1}{T} \sum_{t=1}^{T} \sum_{d \in \mathcal{X}_{-i}} \sigma^t(d) \upsilon(x^*; d) && (2.18) \\
&= \max_{x^* \in \mathcal{X}_i} \sum_{d \in \mathcal{X}_{-i}} \upsilon(x^*; d) \frac{1}{T} \sum_{t=1}^{T} \sigma^t(d). && (2.19) \\
&= \max_{x^* \in \mathcal{X}_i} \upsilon\left(x^*; \frac{1}{T} \sum_{t=1}^{T} \sigma^t(d) \right), && (2.20)
\end{aligned}$$

where the first and last step just apply our overloading of υ to return expected payoff when parameterized by a mixed strategy.

Maximizing Eq. (2.17) over deviations, dividing by T, applying Eq. (2.20), and using the definitions of $\bar{\pi}^T_1$ and $\bar{\pi}^T_2$ yields

$$\underbrace{\max_{x^*_1 \in \mathcal{X}_1} \upsilon(x^*_1; \bar{\pi}^T_2) + \max_{x^*_2 \in \mathcal{X}_2} \upsilon(x^*_2; \bar{\pi}^T_1)}_{\text{The NashConv of } (\bar{\pi}^T_i)^2_{i=1}.} \leq \frac{2f(T)}{T}. \tag{2.21}$$

Eq. (2.21) shows that $(\bar{\pi}^T_i)^2_{i=1}$ has a NashConv no larger than ${}^{2f(T)}/_{T}$, which implies that this strategy profile is a ${}^{2f(T)}/_{T}$-Nash equilibrium, as claimed. Furthermore, if all players are no-regret, then $\lim_{T \to \infty} {}^{f(T)}/_{T} = 0$. Therefore, in the limit as $T \to \infty$, $(\bar{\pi}^T_i)^2_{i=1}$ approaches an exact Nash equilibrium. □

2.3.3 Approximating an Optimal Strategy

Regret minimization can be used to approximate an optimal strategy in a static environment where the utility function on each round is the same single utility function, υ.

Proposition 4. *In a static environment, the average mixed strategy, $\bar{\pi}^T = \frac{1}{T}\sum_{t=1}^T \pi^t \in \Delta(\mathcal{X})$, is an ε-optimal strategy, where $\varepsilon = \max_{\phi^{\to x} \in \Phi_{\mathcal{X}}^{\text{EX}}} \frac{1}{T}\left[\sum_{t=1}^T \rho(\phi^{\to x}, \pi^t; \upsilon)\right]_+$. Thus, if the agent is no-external-regret, then $\bar{\pi}^T$ approaches an optimal decision in the limit as $T \to \infty$.*

Proof. In a static environment, the average external regret with respect to $\phi^{\to X} : \cdot \mapsto X \in \mathcal{X}$ after T rounds simplifies to

$$\frac{1}{T}\sum_{t=1}^{T} \upsilon(x) - \upsilon(\pi^t) = \upsilon(x) - \frac{1}{T}\sum_{t=1}^{T} \upsilon(\pi^t) = \upsilon(x) - \upsilon(\bar{\pi}^T), \tag{2.22}$$

where the second equality follows from Jensen's inequality, which is satisfied with equality because the utility function is linear. Thus,

$$\max_{x \in \mathcal{X}} \upsilon(x) - \upsilon(\bar{\pi}^T) \leq \max_{\phi^{\to x} \in \Phi_{\mathcal{X}}^{\text{EX}}} \frac{1}{T}\left[\sum_{t=1}^{T} \rho(\phi^{\to x}, \pi^t; \upsilon)\right]_+, \tag{2.23}$$

as claimed. Furthermore, if the agent is no-external-regret, then the positive part of their maximum average regret goes to zero as $T \to \infty$, so the optimality gap of $\bar{\pi}^T$ also approaches zero. □

One notable consequence of Proposition 4 is that if an ODP represents player i's perspective of a game where the strategies of all the other players are fixed, then a no-external-regret algorithm can be used to approximate a best response to those fixed strategies.

2.3.4 Time Selection

An *online time selection decision process* (*OTSDP*; Lehrer 2003; Blum et al. 2007) generalizes ODPs to weighted utility functions. A *time selection function* w maps each round t to a weight $w^t \in [0, 1]$ so that on round t, the utility function is weighted by w^t. The agent is given a finite set of time selection functions and their goal is to minimize regret with respect to all deviation–time-selection function pairs simultaneously. Formally, a finite set of time selection functions, $\mathcal{W} = \{t \mapsto w_j^t \in [0, 1]\}_{j=1}^m$, $1 \leq m < \infty$, is given, and the cumulative regret with respect to deviation $\phi \in \Phi \subseteq \Phi_{\mathcal{X}}^{\text{SW}}$ and time selection function $w \in \mathcal{W}(\phi)$ after T rounds is $\sum_{t=1}^T w^t \rho(\phi, \pi^t; \upsilon^t)$. A no-$(\Phi, \mathcal{W})$-regret algorithm in this setting ensures that the positive part of the average regret, maximizing over deviations and time selection functions, vanishes, *i.e.*,

$$\max_{\substack{\phi \in \Phi, \\ w \in \mathcal{W}}} \lim_{T \to \infty} \left[\sum_{t=1}^{T} w^t \rho(\phi, \pi^t; \upsilon^t)\right]_+ = 0$$

.

2.4 Regret Matching

Regret matching is a learning algorithm framework based on updating a vector of cumulative regrets, $\rho^{1:t-1} = [\rho^{1:t-1}(\phi) \doteq \sum_{k=1}^{t-1} \rho(\phi, \pi^k; \upsilon^k)]_{\phi \in \Phi}$, one element for each deviation in a given subset of swap deviations $\Phi \subseteq \Phi^{\text{sw}}_{\mathcal{X}}$. Regret matching selects the strategy on each round that is a fixed point of the convex combination of deviations across Φ according to deviation preferences generated by passing regrets through a non-negative *link function*, $f : \mathbb{R}^{|\Phi|} \to \mathbb{R}^{|\Phi|}_{+}$. The preferences evaluate each deviation's performance across all past rounds so we can think about the combined deviation, $\bar{\phi}^t$, as the best average deviation in hindsight given these preferences. We represent each deviation ϕ as a $|\mathcal{X}| \times |\mathcal{X}|$ matrix and mixed strategy π as a $|\mathcal{X}|$-length probability vector where the transformation $\phi(\pi)$ is the matrix–vector product $\phi\pi \in \Delta^{|\mathcal{X}|}$. On round t, preferences are generated as $y^t = f(\rho^{1:t-1})$ and the best average deviation in hindsight is

$$\bar{\phi}^t = \begin{cases} \dfrac{1}{\langle \mathbf{1},\, y^t \rangle} \sum_{\phi \in \Phi} y^t_{\phi} \phi & \text{if } \langle \mathbf{1},\, y^t \rangle > 0 \\ I & \text{o.w.}, \end{cases} \tag{2.24}$$

where I is the identity matrix. The strategy π^t is then selected as a fixed point of the linear transformation $\bar{\phi}^t$, *i.e.*, $\bar{\phi}^t \pi^t = \pi^t$.[4]

In general, we can compute a fixed point of $\bar{\phi}^t$ with a linear system solver or approximate it with power iteration (A. Greenwald, Z. Li, and Schudy 2008). A special case where a fixed point can be found very quickly is if each column of $\bar{\phi}^t$ is identical, *e.g.*, when Φ contains only external deviations. In that case, each column of $\bar{\phi}^t$ is a fixed point, so we can simply set π^t to be its first column, $\bar{\phi}^t_{:,1}$. If Φ is the full set of external deviations, then this special case simplifies to

$$\pi^t = \begin{cases} \dfrac{1}{\langle \mathbf{1},\, y^t \rangle} y^t & \text{if } \langle \mathbf{1},\, y^t \rangle > 0 \\ \pi & \text{o.w.}, \end{cases} \tag{2.25}$$

where π is an arbitrary mixed strategy like uniform random.

Regret bounds for regret matching algorithms are based on Blackwell approachability (Blackwell 1956). The first step is choosing the link function to be $f = \alpha g$ for some $\alpha > 0$, where g is part of a Gordon triple (Gordon 2005), (G, g, γ), where a Gordon triple consists of a potential function, $G : \mathbb{R}^n \to \mathbb{R}$, a scaled link function $g : \mathbb{R}^n \to \mathbb{R}^n_{+}$, and a size function, $\gamma : \mathbb{R}^n \to \mathbb{R}_{+}$, that satisfy the generalized smoothness condition $G(x + x') \leq G(x) + x' \cdot g(x) + \gamma(x')$ for any $x, x' \in \mathbb{R}^n$. By applying the potential function to the cumulative regret, we can unroll the recursive bound to get a simple bound on the cumulative regret itself.

[4] Since every strategy is a fixed point of the identity matrix, setting the average deviation to the identity matrix when the learner has no preferences implies that the learner can choose any strategy on those rounds.

2.4.1 Example Instantiations

Softmax regret matching (that is, regret matching with the exponential or softmax link function) is the *Hedge* (Freund et al. 1997) algorithm or the *exponential weighted average forecaster* (Cesa-Bianchi et al. 2006). Representing the utility function as a vector, Hedge on the external deviations chooses the mixed strategy

$$\pi^t = \frac{\exp\left(\frac{1}{\tau^t}\sum_{k=1}^{t-1} \upsilon^k\right)}{\left\langle \mathbf{1}, \exp\left(\frac{1}{\tau^t}\sum_{k=1}^{t-1} \upsilon^k\right)\right\rangle}$$

on each round t, where $\tau^t > 0$ is a *temperature* parameter. The regret matching analysis of this algorithm (D'Orazio and R. Huang 2021; Freund et al. 1997; A. Greenwald, Z. Li, and Marks 2006a) leads to optimal regret bounds (with respect to the number of rounds and deviations) when temperature parameters are chosen appropriately (*e.g.*, increasing with $\sqrt{t}$).

Theorem 1 (Theorem 15 of A. Greenwald, Z. Li, and Marks (2006a) for expected cumulative regret and payoff magnitude U)**.** *Softmax regret matching (Hedge) with constant temperature* $\tau > 0$ *ensures that expected cumulative regret after* T *rounds for any deviation* $\phi \in \Phi \subseteq \Phi^{\text{SW}}_{\mathcal{X}}$ *is upper bounded as*

$$\rho^{1:T}(\phi) \leq U\tau \ln|\mathcal{X}| + \frac{U}{2\tau}T.$$

See A. Greenwald, Z. Li, and Marks (ibid.) for a proof.

The original regret matching algorithm, described by Hart et al. (2000) and implied by Blackwell (1956), is defined with the ramp link function, $[\cdot]_+ = \max\{\cdot, 0\}$. The usual Gordon triple for this link function is $\gamma : x \mapsto \frac{1}{2}\|x\|_2^2$, $G : x \mapsto \gamma([x]_+)$, and $g(\cdot) = [\cdot]_+$, which leads to the following regret bound.

Theorem 2 (Theorem 11 of A. Greenwald, Z. Li, and Marks (2006a) for $p = 2$, expected cumulative regret, and payoff magnitude U)**.** *Ramp regret matching ensures that expected cumulative regret after* T *rounds for any deviation* $\phi \in \Phi \subseteq \Phi^{\text{SW}}_{\mathcal{X}}$ *is upper bounded as* $\rho^{1:T}(\phi) \leq U\sqrt{\alpha(\Phi)T}$*, where* $\alpha(\Phi) = \max_{x\in\mathcal{X}} \sum_{\phi\in\Phi} \mathbb{1}\{\phi(x) \neq x\}$ *is the maximal activation of* Φ*.*

See A. Greenwald, Z. Li, and Marks (ibid.) for a proof.

Ramp regret matching has a suboptimal dependence on the number of deviations, but it is often exceptionally effective in practice (see, *e.g.*, Burch (2017) and Waugh and Bagnell (2015)) without parameter tuning.

2.4.2 Extensions

Three notable extensions are regret matching$^+$, optimistic regret matching, and approximate regret matching.

Instead of the cumulative regrets, regret matching$^+$ updates a vector of pseudo regrets, $q^{1:t} = [q^{1:t-1} + \rho^t]_+ \geq \rho^{1:t}$, where $\rho^t = [\rho(\phi, \pi^t; \upsilon^t)]_{\phi \in \Phi}$ is the next vector of instantaneous regrets (Tammelin 2014; Tammelin et al. 2015). If we assume a *positive invariant* potential function where $G([x + x']_+) \leq G(x + x')$, then the same regret bounds follow from the same arguments used in the analysis of ordinary regret matching D'Orazio (2020). Note that this condition is satisfied with equality for the quadratic ramp potential $G : x \mapsto \frac{1}{2}\|[x]_+\|_2^2$.

Optimistic regret matching augments its link inputs by adding a prediction of the instantaneous regret on the next round, *i.e.*, $m^t \approx \rho^t$. If the predictions are accurate then the algorithm's cumulative regret will be very small. This is a direct application of optimistic Lagrangian Hedging (D'Orazio and R. Huang 2021) to Φ-regret. The general approach of adding predictions to improve the performance of regret minimizers originates with Rakhlin et al. (2013) and Syrgkanis et al. (2015) and has also been adapted for external regret matching by Farina, Kroer, and Sandholm (2021). D'Orazio and R. Huang (2021)'s analysis requires that G and g satisfy $G(x') \geq G(x) + \langle g(x), x' - x \rangle$, which is achieved, for example, if G is convex and g is a subgradient of G. Hart et al. (2000)'s regret matching satisfies this condition because the ramp function is the gradient of the quadratic ramp potential, and this potential function is convex (A. Greenwald, Z. Li, and Marks 2006a).

Approximate regret matching is regret matching with approximate cumulative regrets, $\widetilde{\rho}^{1:t-1} \approx \rho^{1:t-1}$ (Waugh, Morrill, et al. 2015) or pseudo regrets, $\widetilde{q}^{1:t-1} \approx q^{1:t-1}$ (D'Orazio 2020; Morrill 2016). The regret of approximate regret matching depends on its approximation accuracy and motivates the use of function approximation when it is impractical to store and update the regret for each deviation individually. The regret bound for this algorithm is presented in the background for Part III as Theorem 17. Only approximate external regret matching with the ramp link function was studied prior to the work in Chapter 10, where we generalize this approach and analysis. Section 7.4.3 also provides a unified analysis for regret matching in the time selection setting with regret approximations and predictions.

2.5 Policy Gradient

The current predominant approach to construct reward-seeking agents in the field of RL is to design algorithms that approach an optimal strategy (though in this field, a strategy is typically called a *policy*). In order for this motivation to be coherent, the environment must be sufficiently static for an optimal policy to exist, which is potentially limiting. However, RL algorithms are typically designed to be incrementally updated, so this limitation is usually conceptual rather than procedural.

Policy gradient (R. S. Sutton et al. 2000; Williams 1992) is a foundational RL method and follows this pattern closely. This algorithm uses a parameterized differentiable function to generate the agent's policy, and updates the policy parameters in the direction of steepest payoff increase, *i.e.*, the gradient of the utility function.

Policy architectures are generally composed of at least two layers, the first generating parameters that encapsulates the preference the agent has for each action and the second that constructs a distribution over actions from the output of the first layer.[5] If the set of actions is finite, the first layer typically outputs a preference for each action and the conventional distribution layer is a softmax function that simply exponentiates and normalizes each preference.

As long as the utility function is static and the update step size decreases appropriately, *e.g.*, at a $\mathcal{O}({}^1\!/t)$ rate, this method converges towards a policy where there are no unilateral parameter changes that could improve the policy (a "local maximum"; R. S. Sutton et al. 2000). If the policy architecture is sufficiently expressive, this algorithm is even guaranteed to converge toward an optimal policy, although the convergence rate with a softmax distribution layer can be unreasonbly slow (Mei et al. 2020). Section 3.3 shows that softmax policy gradient also behaves poorly to environmental non-stationarity.

The underlying idea of *gradient ascent* is sound even in adversarial environments where the utility function is constantly changing (Zinkevich 2003). However, the agent's payoff must be a concave function of their parameters for policy gradient to inherit this soundness. The softmax distribution layer in particular ensures that agent's payoff is not a concave function of their parameters, regardless of the rest of the policy's architecture.

References

Aumann, R. J. (1974). "Subjectivity and Correlation in Randomized Strategies". In: *Journal of Mathematical Economics* 1.1, pp. 67–96.

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

Blum, A. and Y. Mansour (2007). "From External to Internal Regret". In: *Journal of Machine Learning Research* 8.6, pp. 1307–1324.

Burch, N. (2017). "Time and space: Why imperfect information games are hard". PhD thesis. University of Alberta.

Cesa-Bianchi, N. and G. Lugosi (2006). *Prediction, Learning, and Games.* Cambridge University Press.

D'Orazio, R. (2020). "Regret Minimization with Function Approximation in Extensive-Form Games". Master's thesis. University of Alberta.

D'Orazio, R. and R. Huang (2021). "Optimistic and Adaptive Lagrangian Hedging". In: *AAAI Reinforcement Learning in Games Workshop.*

[5] We will define an action more precisely as a component of pure strategies in Chapter 4 but for now it is fine to imagine an action as a pure strategy.

Farina, G., C. Kroer, and T. Sandholm (2021). "Faster Game Solving via Predictive Blackwell Approachability: Connecting Regret Matching and Mirror Descent". In: *35th AAAI Conference on Artificial Intelligence*. Vol. 35. 6, pp. 5363–5371.

Foster, D. P. and R. Vohra (1999). "Regret in the On-Line Decision Problem". In: *Games and Economic Behavior* 29.1-2, pp. 7–35.

Freund, Y. and R. E. Schapire (1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1, pp. 119–139.

Gordon, G. J. (2005). *No-Regret Algorithms for Structured Prediction Problems*. Tech. rep. CMU-CALD-05-112. Carnegie Mellon University.

Greenwald, A., A. Jafari, and C. Marks (Aug. 2003). "A general class of no-regret learning algorithms and game-theoretic equilibria". In: *2003 Computational Learning Theory Conference*, pp. 1–11.

Greenwald, A., Z. Li, and C. Marks (Jan. 2006a). "Bounds for Regret-Matching Algorithms". In: *International Symposium on Artificial Intelligence and Mathematics (ISAIM 2006)*. Fort Lauderdale, Florida, USA.

Greenwald, A., Z. Li, and W. Schudy (2008). "More Efficient Internal-Regret-Minimizing Algorithms." In: *COLT*, pp. 239–250.

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

Lehrer, E. (2003). "A Wide Range No-Regret Theorem". In: *Games and Economic Behavior* 42.1, pp. 101–115.

Mei, J., C. Xiao, B. Dai, L. Li, C. Szepesvári, and D. Schuurmans (2020). "Escaping the Gravitational Pull of Softmax." In: *NeurIPS*.

Morrill, D. (2016). "Using Regret Estimation to Solve Games Compactly". Master's thesis. University of Alberta.

Morrill, D., R. D'Orazio, M. Lanctot, J. R. Wright, M. Bowling, and A. R. Greenwald (July 2021). "Efficient Deviation Types and Learning for Hindsight Rationality in Extensive-Form Games". In: *38th International Conference on Machine Learning (ICML 2021)*. Vol. 139. virtual, pp. 7818–7828.

Morrill, D., R. D'Orazio, R. Sarfati, M. Lanctot, J. R. Wright, A. R. Greenwald, and M. Bowling (Feb. 2021). "Hindsight and Sequential Rationality of Correlated Play". In: *35th AAAI Conference on Artificial Intelligence*. Vol. 35. 6. virtual, pp. 5584–5594.

Moulin, H. and J.-P. Vial (1978). "Strategically zero-sum games: the class of games whose completely mixed equilibria cannot be improved upon". In: *International Journal of Game Theory* 7.3-4, pp. 201–221.

Nash, J. (1951). "Non-Cooperative Games". In: *The Annals of Mathematics* 54.2, pp. 286–295.

Neumann, J. von and O. Morgenstern (1947). *The Theory of Games and Economic Behavior*. 2nd. Princeton University Press.

Rakhlin, S. and K. Sridharan (2013). "Optimization, learning, and games with predictable sequences". In: *Advances in Neural Information Processing Systems*, pp. 3066–3074.

Sutton, R. S., D. McAllester, S. Singh, and Y. Mansour (2000). "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Advances in Neural Information Processing Systems 12*. MIT Press, pp. 1057–1063.

Syrgkanis, V., A. Agarwal, H. Luo, and R. E. Schapire (2015). “Fast convergence of regularized learning in games”. In: *Advances in Neural Information Processing Systems*, pp. 2989–2997.

Tammelin, O. (2014). “Solving Large Imperfect Information Games Using CFR+”. In: *arXiv preprint arXiv:1407.5042*.

Tammelin, O., N. Burch, M. Johanson, and M. Bowling (2015). “Solving Heads-up Limit Texas Hold’em”. In: *24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*.

Waugh, K. and J. A. Bagnell (2015). “A Unified View of Large-Scale Zero-Sum Equilibrium Computation”. In: *Workshops at the 29th AAAI Conference on Artificial Intelligence*.

Waugh, K., D. Morrill, J. A. Bagnell, and M. Bowling (2015). “Solving Games with Functional Regret Estimation”. In: *29th AAAI Conference on Artificial Intelligence (AAAI-15)*. Vol. 29. 1, pp. 2138–2144.

Williams, R. J. (May 1992). “Simple statistical gradient-following algorithms for connectionist reinforcement learning”. In: *Machine Learning* 8.3, pp. 229–256. ISSN: 1573-0565. DOI: 10.1007/BF00992696. URL: https://doi.org/10.1007/BF00992696.

Zinkevich, M. (2003). “Online Convex Programming and Generalized Infinitesimal Gradient Ascent”. In: *20th International Conference on Machine Learning (ICML 2003)*.

Part I

Modeling Foundations

Chapter 3

Hindsight Rationality

> Philosophy is perfectly right in saying that life must be understood backwards. But then one forgets the other clause—that it must be lived forwards.
>
> Søren Kierkegaard[1]

3.1 Introduction

What does it mean for an AI system to be "intelligent" or to "behave intelligently"? One intuitive description is that we want AI systems to make good decisions in service of our goals or at least those we instill within them. At a technical level, we can equate intelligence with rationality, and rationality with optimal behavior, which in our context translates to payoff maximization. In stationary RL environments, it is natural that an intelligent system would achieve the maximum expected payoff by following an optimal policy.

Environments containing multiple dynamic agents, however, are non-stationary from a given agent A's perspective. Whenever A's behavior changes, the observations that the other agents make change as well. While A can play a best response to any set of strategies employed by the other agents, A may not know what strategies the others will play or the other agents may change their strategies over time. In general, there is no universal optimal policy that A can use to maximize their payoff in multi-agent environments. How then should we design intelligent agents for multi-agent settings?

If there are only two agents in the environment and their incentives conflict exactly (a two-player, zero-sum game), then one reasonable choice is to have A play a maximin strategy. Agent A could do better by tailoring their strategy to the other agent's particular strategy,

[1] Recounted in a collection of Kierkegaard's journals and papers (1967).

but a maximin strategy sacrifices this potential advantage in exchange for robustness. In addition, since it is in the best interest of each agent to exploit the other's weaknesses, such pessimism is warranted.

However, pessimism is less useful in general-sum games or those with more than two players because minimizing A's payoff may not be in the best interest of each of the other players and it may not even be possible for them to coordinate perfectly. An alternative ostensibly reasonable proposal is for A to play a strategy from a Nash equilibrium. Here, rational play is that where no player can improve by unilaterally deviating to a different strategy. A nice feature of this definition is that in two-player, zero-sum games, every Nash equilibrium strategy is a maximin strategy. However, a critical problem is Nash equilibrium selection; different Nash equilibria may assign different payoffs to each player and Nash equilibrium strategies are not interchangeable outside of two-player, zero-sum games. Thus, if A plays a strategy from a Nash equilibrium, it is possible to unilaterally improve by changing to a different strategy even if all the other agents play parts of a different Nash equilibrium!

A key limitation shared by all of these behavior proposals (optimal policy, maximin strategy, and Nash equilibrium strategy) is that they suggest that the agent should play a single strategy and forgo the agent's potential to tailor their behavior to changing conditions. The only role that learning can play under these proposals is to produce a static artifact representing a strategy with the requisite properties. Perhaps the problem of designing agents to behave intelligently in complex domains filled with dynamic agents likewise generally requires a dynamic solution concept. I propose hindsight rationality as just such a solution concept.

3.2 The Rationality of Regret Minimization

We have already seen an objective that evaluates a sequence of decisions or an entire learning algorithm: regret. Every time a learning agent A updates their behavior, the new behavior can be characterized by a strategy. After after T updates, A has generated T strategies, $(\pi^t)_{t=1}^{T}$. Regret compares A's performance across these T updates with alternative behavior generated by a deviation function, ϕ, based on the actual utility functions that A observed while learning. Thus, a regret evaluation takes place in hindsight and requires no assumptions about the environment's future dynamics. If A chooses strategies so that their regret is zero, for all deviations in a set Φ, A is optimal with respect to Φ and their own experience; they are optimal in hindsight. In this way, regret expresses a parameterized form of rationality for learning algorithms grounded in the learning agent's actual experience, and we call this *hindsight rationality.*

Of course, exact optimality in a learning context where A is unable to predict the future is generally impossible. It is thus natural to allow for some mistakes where the total value of those mistakes becomes negligible over time. This intuition corresponds to the no-regret objective where regret is guaranteed to grow sublinearly. The interpretation of regret minimization as rationality in hindsight leads us to say that A is *hindsight rational* with respect to a given deviation set Φ if they are no-regret with respect to Φ.

Static solution concepts are useful in the design of algorithms in scenarios where they confer a practical guarantee about the payoff the algorithm will achieve when deployed, *e.g.*, in stationary environments or in two-player, zero-sum games. For example, an optimal policy achieves the maximum payoff in its target environment, and we know this by virtue of its optimality before the policy is actually deployed, which motivates the computation and subsequent deployment of optimal policies. Hindsight rationality does not directly tell us how successful an algorithm will be when deployed and instead makes a claim about how well that algorithm will adapt to its experience. Rationality in hindsight can, of course, only be evaluated after agent A has already made their choices, but since A's future eventually becomes their past, hindsight rationality ensures that A is prepared to adapt to any eventuality.

Hindsight rationality actually subsumes optimal policy rationality. In a static environment with reward function υ, an agent that is hindsight rational for the external deviations learns to achieve essentially the same expected reward as an optimal policy, π^*, *i.e.*,

$$\frac{1}{T}\sum_{t=1}^{T} \upsilon(\pi^t) \leq \frac{1}{T}\sum_{t=1}^{T} \upsilon(\pi^*) - \mathrm{o}(1) = \upsilon(\pi^*) - \mathrm{o}(1),$$

where the agent chooses strategies $(\pi^t)_{t=1}^{T}$. Additionally, hindsight rationality is connected to static notions of rationality via Propositions 2 to 4. In contrast to the prescriptive nature of the regret minimization objective, these connections represent descriptions of hindsight rational behavior in special scenarios. Of particular importance is Proposition 2, which shows that hindsight rationality subsumes the rationality of correlated equilibria in general, and that a society of such agents will learn to optimally correlate with each other (with respect to unilateral deviations).

3.3 Hindsight Rationality Versus Incremental Optimization

Does it really matter if an agent is hindsight rational in terms of the payoff they accumulate over time, or is it enough that the agent incrementally improves? Softmax policy gradient (SPG; see Section 2.5) is an elementary example of the latter type of incremental optimization algorithm that does not in general boast a no-regret property. Hedge (see Section 2.4.1) is a very similar algorithm procedurally to SPG (applied to an ODP), but Hedge is no-regret. These two algorithms are ideal avatars to experimentally compare the incremental optimization paradigm with that of hindsight rationality.

3.3.1 The Similarities and Differences of Hedge and SPG

To begin, let us examine the similarities and differences between full monitoring ODP implementations of SPG and Hedge on the external deviations, both instantiated with constant parameters.

On round t, Hedge, with temperature $\tau > 0$, chooses the strategy $\pi^t \propto \exp(\theta^t)$ where Hedge's preference vector $\theta^t = \sum_{k=1}^{t-1} \frac{1}{\tau} \upsilon^k \in \mathbb{R}^{|\mathcal{A}|}$ and υ^k is the utility function on round k treated as a vector. Since the exponential function is shift invariant, we could also set Hedge's preference vector to $\theta^t = \sum_{k=1}^{t-1} \frac{1}{\tau} \rho^k$ without changing its strategies.

On round t, SPG, with step size $\frac{1}{\tau}$ and a single parameter for each pure strategy, chooses its mixed strategy in the same way as Hedge except that it uses the preference vector $\theta^t = \sum_{k=1}^{t-1} \frac{1}{\tau} \nabla \upsilon^k(\pi^k)$. We can see more precisely the difference between the SPG and Hedge preferences if we evaluate the payoff gradients. $\nabla \upsilon^k(\pi^k) = \left[\frac{\partial \upsilon^k(\pi^k)}{\partial \theta^k_x}\right]_{x \in \mathcal{X}}$ where the partial derivatives are, by Section 2.8 of R. Sutton et al. (2018),

$$\begin{aligned}
\frac{\partial \upsilon^k(\pi^k)}{\partial \theta^k_x} &= \sum_{x' \in \mathcal{X}} \upsilon^k(x') \frac{\partial \pi^k(x')}{\partial \theta^k_x} && (3.1)\\
&= \sum_{x' \in \mathcal{X}} \upsilon^k(x') \pi^k(x') \big(\mathbb{1}\{x = x'\} - \pi^k(x)\big) && (3.2)\\
&= \pi^k(x) \upsilon^k(x) \big(1 - \pi^k(x)\big) - \pi^k(x) \sum_{x' \neq x} \pi^k(x') \upsilon^k(x') && (3.3)\\
&= \pi^k(x) \left(\upsilon^k(x) - \pi^k(x) \upsilon^k(x) - \sum_{x' \neq x} \pi^k(x') \upsilon^k(x') \right) && (3.4)\\
&= \pi^k(x) \left(\upsilon^k(x) - \sum_{x' \in \mathcal{X}} \pi^k(x') \upsilon^k(x') \right) && (3.5)\\
&= \pi^k(x) \rho(\phi^{\to x}, \pi^k; \upsilon^k). && (3.6)
\end{aligned}$$

Therefore, the SPG preferences simplify to $\theta^t = \sum_{k=1}^{t-1} \frac{1}{\tau} \pi^k \odot \rho^k$, where π^k is treated as a vector and $\odot$ is the elementwise product operation.

Now we can see that the only difference between the SPG and Hedge preferences is that SPG accumulates the elementwise product $\pi^k \odot \rho^k$ instead of ρ^k or υ^k alone. The problem that this can cause for SPG is that elements of π^k can be close to zero, which prevents the SPG preferences from changing quickly if a historically bad strategy becomes good after a change in the environment. In contrast, it is easy to see that Hedge requires exactly as strong a utility signal to "unlearn" a behavior as it does to learn that behavior initially since the Hedge accumulates utility vectors with the same weight on each round, *e.g.*, Hedge's preference on round t for strategy x is equal to that on round $t+2$ if $\upsilon^t(x) = -\upsilon^{t+1}(x)$.

3.3.2 A Non-Stationary Matching Pennies Experiment

Consider matching pennies, a symmetric, two-player zero-sum game with two pure strategies, HEADS and TAILS. The EVEN player receives a payoff of 1 if both players choose the same pure strategy and -1 otherwise. The ODD player receives the negative of the EVEN player's payoff, but the ODD player's payoff incentive does not matter for this experiment because their strategy is going to be pre-determined. Across a pre-established number of rounds T, the ODD player is going to choose HEADS for the first 40% of the rounds and TAILS for the remaining 60%. The ODD player only changes their strategy once and the 40%–60% split is tuned to cause difficulties for SPG. The EVEN player's perspective forms an ODP where the utility function is $\upsilon^t(x) = 1$ if $x =$ HEADS and -1 otherwise for $t \leq 0.4T$ and its negation $\upsilon^t = -\upsilon^1$ afterward ($t > 0.4T$).

This experiment tests SPG and Hedge in the EVEN player's ODP and illustrates their performance as T increases. Their step size and temperature parameters are set to constants and tuned to a given T for simplicity and ease of comparison. Since this particular environment has two long stationary phases, Hedge performs better in this particular environment as the temperature is driven to zero, but the Hedge agent should be hindsight rational. Therefore, we use the temperature $\tau = \sqrt{\frac{T}{\ln|\mathcal{A}|}}$, which is the best choice in a worst-case environment according to the Hedge analysis of A. Greenwald, Z. Li, and Marks (2006a). To ensure a charitable evaluation for SPG, its step size is empirically tuned to this particular environment for each given horizon length. See Section 3.A for information about the step size tuning.

3.3.3 Results

Figure 3.1 (left) shows the average payoff of SPG, Hedge, and always TAILS as the horizon length, T, increases, evaluated with $T \in \{10^i\}_{i=1}^6$. SPG and Hedge perform similarly, but remember that SPG's performance here is with step sizes that are precisely tuned to each T. Hedge, in contrast, uses a temperature for each T that is only best in a worst-case environment. In fact, Hedge's average payoff approaches $0.2-\frac{1}{T}$ as its temperature approaches zero, so Hedge's performance in this environment could be substantially improved for small T with tuning.

SPG performs reasonably well in this environment because it only has two pure strategies to choose from, HEADS or TAILS. Even if the probability of playing TAILS is small after 40% of the rounds, the SPG strategy quickly decreases the probability of playing TAILS because the preferences for HEADS decreases quickly. The SPG update to the HEADS preference is

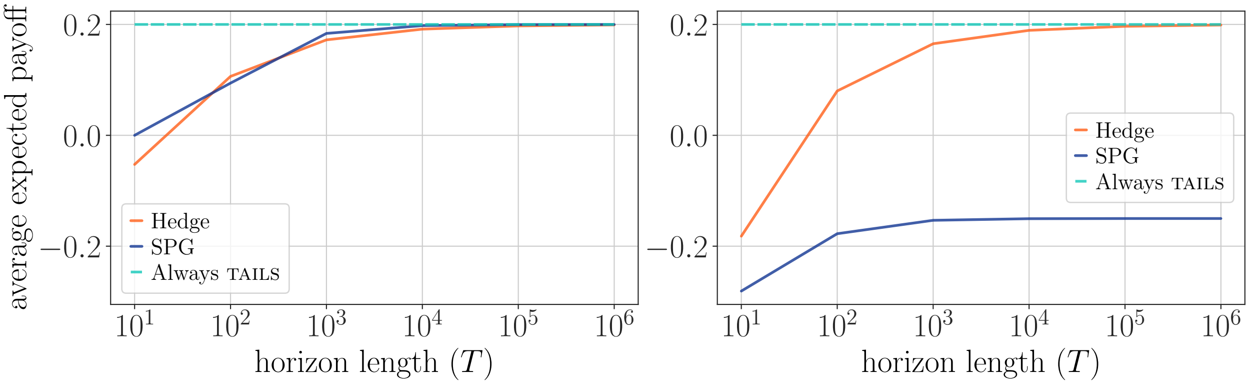


Figure 3.1: The average expected payoff of SPG, Hedge, and always TAILS in repeated matching pennies against fixed ODD player behavior (HEADS for the first 40% of the rounds and TAILS for the remaining 60%) across various horizon lengths. (left) Without FORFEIT. (right) With FORFEIT.

large because the probability SPG plays HEADS is large and all probability mass taken away from HEADS must go to TAILS.

Figure 3.1 (right) shows the average payoff of SPG, Hedge, and always TAILS again, but this time, the pure strategy set includes an a third strategy: FORFEIT. This strategy simply forfeits the round to the ODD player, giving the EVEN player -1. In this case, SPG's performance is drastically reduced while Hedge is nearly unaffected. With three actions, the probability mass moved from HEADS after 40% of the rounds is split between TAILS and FORFEIT, which ensures that FORFEIT is played much more often than is beneficial. SPG's average payoff approaches $-0.15 < +0.2$ as T increases so its regret grows linearly in this environment.

3.4 Conclusion

An intelligent system must respond well in complex domains inhabited by dynamic agents each with their own incentives, capabilities, and perspectives. Perhaps unsurprisingly, a static solution like an optimal policy or an equilibrium, is generally unsatisfactory when no assumptions can be made about the behavior of the other agents. Learning online and analyzing agent behavior in hindsight is a dynamic alternative. The hindsight rationality perspective suggests that agents ought to reduce their regret for deviations in hindsight and in doing so, their behavior approaches rationally in hindsight. Rationality in hindsight measures an agent's performance according to its actual behavior and experience rather than the hypothetical future scenarios of equilibrium rationality. However, hindsight rationality does not discard the concept of equilibrium entirely. Instead, an equilibrium merely describes the behavior of long running interactions between hindsight rational agents. The hindsight rationality perspective thus returns equilibria to a descriptive role, as originally introduced within the field of game theory, rather than the prescriptive role it often holds in the field of artificial intelligence. As our motivating example shows, the difference in lifetime performance between a hindsight rational algorithm and an optimization algorithm can be stark because of the hindsight rational algorithm's greater resiliency to environmental changes.

References

Greenwald, A., Z. Li, and C. Marks (Jan. 2006a). "Bounds for Regret-Matching Algorithms". In: *International Symposium on Artificial Intelligence and Mathematics (ISAIM 2006)*. Fort Lauderdale, Florida, USA.

Kierkegaard, S. (1967). "Søren Kierkegaard's Journals and Papers. Volume 1, A-E". In: 01. Ed. by H. V. Hong, E. H. Hong, and G. Malantschuk.

Sutton, R. and A. Barto (2018). *Reinforcement Learning: An Introduction*. 2nd. MIT Press.

3.A SPG Step-Size Tuning in Non-Stationary Matching Pennies Experiment

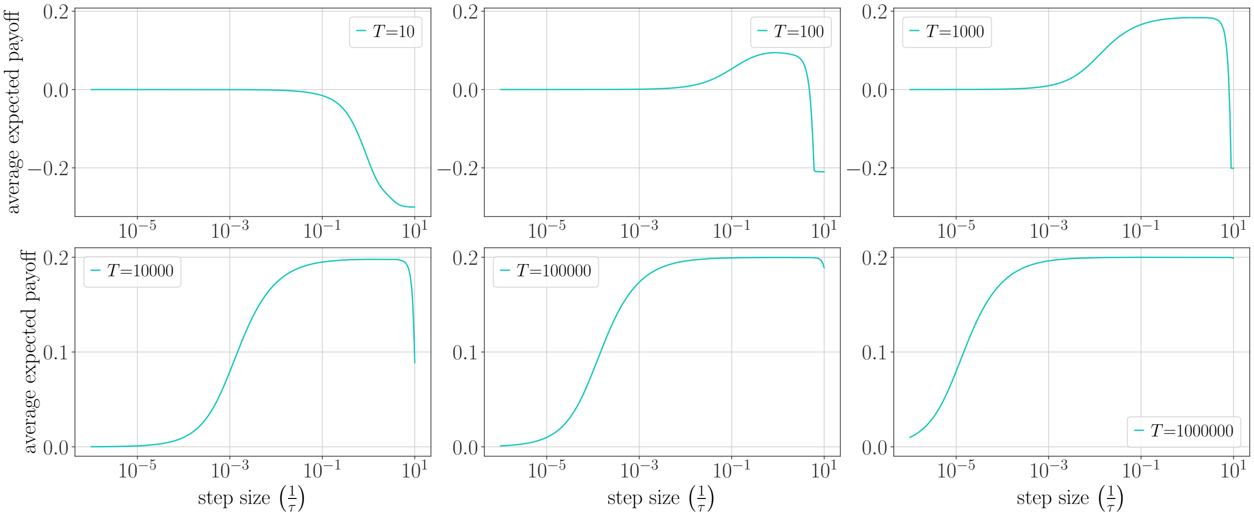


Figure 3.A.2: The average expected payoff of SPG in the non-stationary matching pennies environment without FORFEIT for a given horizon length T across step sizes evaluated in linear and logarithmic grid searches.

To find good step sizes for SPG in the non-stationary matching pennies environment with each horizon length T, two uniform grid searches were completed for each $T \in \{10^i\}_{i=1}^{6}$. One search was done in the linear scale and the other in the logarithmic scale and both grid searches evaluated $10,000$ step sizes for each T.

The smallest step size tested was 10^{-6} and the largest was 10. For the linear search, this results in an increment of $\Delta = (10 - 10^{-6})/(10,000 - 1) \approx 0.01$ and step sizes $\{10^{-6} + i\Delta\}_{i=0}^{9999}$. For the logarithmic search, this results in a factor of $m = \sqrt[9999]{\frac{10}{10^{-6}}} \approx 1.0018$ and step sizes $\{10^{-6} m^i\}_{i=0}^{9999}$. After removing duplicates between the two grid searches, $19,986$ unique step sizes were evaluated.

Figure 3.A.2 shows the performance of SPG across step sizes without FORFEIT and Fig. 3.A.3 is the same with FORFEIT. Table 3.1 lists the best step sizes found for each T along with the average payoff and cumulative regret achieved. The full experiment took 1 hour and 45 minutes using two cores from a 3.40GHz Intel® Core™ i5-3570K CPU with 8 GB of memory.

Table 3.1: The Hedge and best SPG performance achieved for each horizon length in the non-stationary matching pennies environment with and without FORFEIT (to two significant digits).

		T	$\frac{1}{\tau}$	$\frac{1}{T}\sum_{t=1}^{T} \upsilon^t(\pi^t)$	$\max_x \rho^{1:T}(x)$
without FORFEIT	Hedge	10^1	0.26	-0.053	2.5
		10^2	0.083	$+0.11$	9.4
		10^3	0.026	$+0.17$	28
		10^4	0.0083	$+0.19$	85
		10^5	0.0026	$+0.2$	$2.6 \cdot 10^2$
		10^6	0.00083	$+0.2$	$8.3 \cdot 10^2$
	SPG	10^1	$1 \cdot 10^{-6}$	$-1.5 \cdot 10^{-7}$	2
		10^2	0.88	$+0.094$	11
		10^3	1.6	$+0.18$	16
		10^4	1.9	$+0.2$	21
		10^5	2.3	$+0.2$	27
		10^6	1.4	$+0.2$	32
with FORFEIT	Hedge	10^1	0.33	-0.18	3.8
		10^2	0.1	$+0.08$	12
		10^3	0.033	$+0.17$	35
		10^4	0.01	$+0.19$	$1.1 \cdot 10^2$
		10^5	0.0033	$+0.2$	$3.3 \cdot 10^2$
		10^6	0.001	$+0.2$	$1 \cdot 10^3$
	SPG	10^1	0.38	-0.28	4.8
		10^2	0.1	-0.18	38
		10^3	0.014	-0.15	$3.5 \cdot 10^2$
		10^4	0.0015	-0.15	$3.5 \cdot 10^3$
		10^5	0.00015	-0.15	$3.5 \cdot 10^4$
		10^6	$1.5 \cdot 10^{-5}$	-0.15	$3.5 \cdot 10^5$

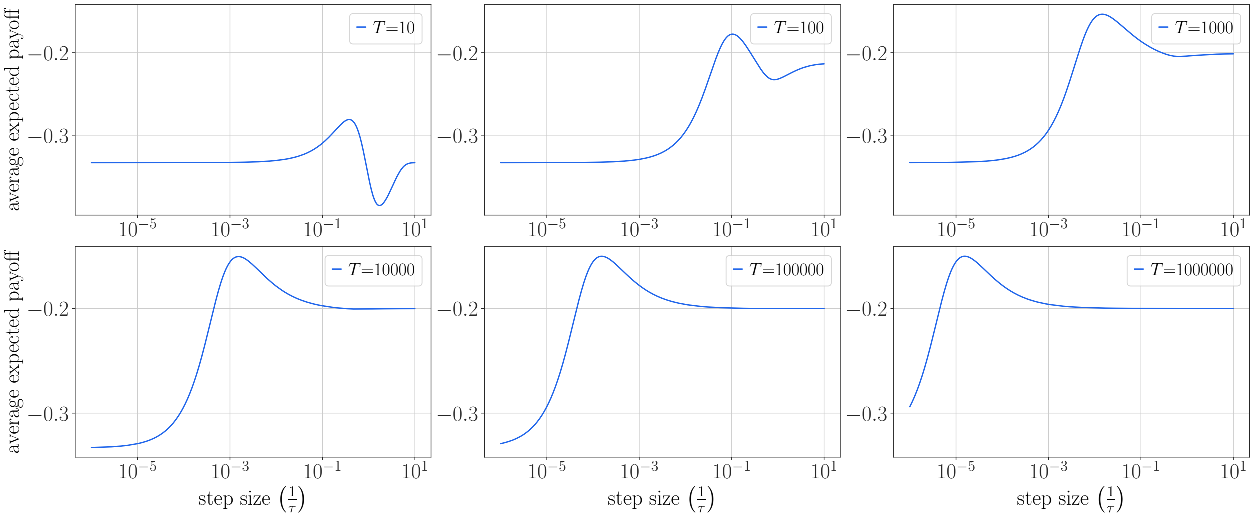


Figure 3.A.3: The average expected payoff of SPG in the non-stationary matching pennies environment with FORFEIT for a given horizon length T across step sizes evaluated in linear and logarithmic grid searches.

Chapter 4

The Partially Observable History Process

4.1 Introduction

We develop the partially observable history process (POHP) that embodies the philosophical aspects of RL. That is, the formalism codifies principle tenets of RL; for example, that the agent is responsible for managing their own representation of an environment that is, by default, massively more complicated than themselves, and that the agent is capable of evaluating their own progress towards goals established by their designer or themself. The POHP formalism is built on a small number of elementary mechanisms to be easy to understand and to use without sacrificing generality. A POHP accurately models reward-driven sequential decision-making with multiple-agents, information asymmetry, and stochasticity, specifically from a single agent's perspective. In contrast to other general models with similar capabilities, a POHP model abstracts away any other agents into a fictional aggregate entity (a "daimon"), which facilitates streamlined analyses of single-agent RL algorithms. The POHP formalism is precisely tuned for the analysis and development of algorithms that are agnostic to the number of other agents in the environment.

The individual components of the POHP formalism are taken from two sequential decision-making frameworks, the extensive-form game (EFG; Kuhn 1953) and the partially observable Markov decision process (POMDP; Smallwood et al. 1973), along with a repeated game framework, the online decision process (see, *e.g.*, A. Greenwald, Z. Li, and Marks (2006a)). The result is a sequential decision-making formalism that is both conceptually simpler and more general than either of its two sequential decision-making progenitors. Other general formalisms with nearly the same expressiveness such as the partially observable Markov game (POMG; Hansen et al. 2004), the turn-taking POMG (TT-POMG; A. Greenwald, J. Li, et al. 2017), and the factored observation stochastic game (FOSG; Kovařík et al. 2019) bring with them unnecessary complications for agent-centric RL. The sequential decision-making setting presented by Farina, Kroer, and Sandholm (2019) shares spiritual similarities but it represents a less radical departure from the EFG model. And while the POHP model deviates substantially from the EFG model, Srinivasan et al. (2018)'s presentation of the EFG model using RL and Markov decision process (MDP) terminology provided substantial inspiration for the POHP model's development.

The POHP model is not meant to replace any of these established formalisms, but rather to fill a particular niche. Consider using a POHP over an MDP or POMDP when there may be more than one in the agent in the environment or when the environment is non-stationary. Consider a POHP over an EFG if the environment to model is naturally a continuing process, if rewards are naturally provided to agents incrementally as they choose actions rather than all at once upon termination, or if explicitly reasoning about more than one agent as an individual is unnecessary. Consider a POHP over a POMG, TT-POMG of FOSG when the Markovian state assumption is unnecessary or unrealistic, or if explicitly reasoning about more than one agent as an individual is unnecessary.

A convenient emergent feature of the POHP model is how it lends itself to a recursive analysis. A POHP can often be decomposed into smaller "sub-POHPs" that are themselves well-defined POHPs. This allows us to conveniently describe sequential rationality (Kreps et al. 1982) for POHPs and define a new variation thereof called *observable sequential rationality*, which incorporates the correlated behavior between multiple agents that naturally arises in hindsight evaluation.

4.2 Partially Observable History Process

We begin from the premise that an *agent* observes and influences an *environment*. We are principally concerned with the design of the agent and how well they navigate the environment. The environment may change without the agent's input and we attribute these changes to a *daimon*. Inspired by depictions in Greek mythology, our daimon is an inexplicable force that partially determines the evolution of the environment and shapes the agent's learning. The concept of a daimon is flexible enough that it can represent an adversary, a teammate, a teacher, chance, or any combination thereof.

4.2.1 The Environment and Daimon

The environment dynamics follow a simple continuing history model. *History* in this model refers to a simple ledger that permanently records *actions*. Given history h from the set of possible histories, $\mathcal{H}$, and action a from the set of legal actions, $\mathcal{A}(h)$, the next history is always $ha \in \mathcal{H}$. The daimon and the agent take turns choosing actions until the process terminates, which divides the set of histories into the *active histories*, $\mathcal{H}_{\mathcal{A}}$, where it is the

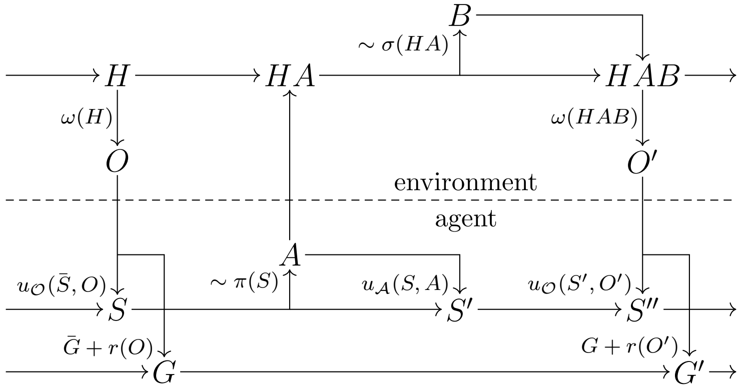


Figure 4.1: The evolution of a POHP environment and agent, steered by a daimon through its strategy, σ.

agent's turn to act, and the *passive histories*, $\mathcal{H}_{\mathcal{O}}$, where it is the daimon's turn to act (and where the agent waits for an observation). Histories are partially ordered action strings so we use $h \sqsubset h'$ to denote that h is a predecessor of h', $|h|$ to denote the length of h, and use subscripts to reference substrings, *e.g.*, h_i is the i^{th} action in h (counting from 1) and $h_{\leq n}$ are the first n actions of h. If the process begins from the empty history, $\varnothing$, we assume without loss of generality that the daimon acts first so that the length of the history determines if the agent is acting or waiting, *i.e.*, $\mathcal{H}_{\mathcal{A}} = \{h \mid |h| \bmod 2 = 1\}$ and $\mathcal{H}_{\mathcal{O}} = \{h \mid |h| \bmod 2 = 0\}$. Although, a POHP need not begin at the empty history; instead, one can be sampled from a probability distribution over histories $\xi : \mathcal{H} \to [0, 1]$.

The agent receives limited information about the daimon's actions through *observations* from a set $\mathcal{O}$, generated by an *observation function*, $\omega : \mathcal{H} \to \mathcal{O}$, while the daimon may observe the agent's actions directly. The agent knows exactly the action they choose when they choose it, so we set the observation of each passive history to the special symbol $\textsc{wait} \in \mathcal{O}$, which indicates that the agent needs to wait for the daimon to act before the agent can receive a proper observation and act.

The history process continues or terminates probabilistically according to $\gamma : \mathcal{H} \to [0, 1]$. $\gamma(h)$ is the probability that the process continues beyond history h and $1 - \gamma(h)$ is the complementary probability that the process terminates at h. Without loss of generality, we assume that the process only terminates after daimon actions so that the agent always receives at least one observation after each agent action.

The daimon behaves according to a *behavioral strategy* (also called a *policy*), σ, that assigns a probability distribution over legal actions to each passive history. In passive history h, the daimon chooses action $B \sim \sigma(h)$ and the history advances to hB, at which point the process continues only if $\Gamma = 1$ where $\Gamma \sim \gamma(hB)$. We do not ascribe any a priori motivation or perceptual limitations to the daimon, though these constraints could be added as extra assumptions for a specific application.

4.2.2 An Abstract POHP Agent

The agent in a POHP is decoupled from the history process by the observation and action interfaces, and as such, could be designed in various ways. The agent architecture we assume in this thesis has three conceptual modules: a state of mind, a behavior plan, and goals. The agent pursues their goals by choosing different actions depending on their state of mind. Formally, the agent is defined by a tuple, $(s_\varnothing, u_\mathcal{A}, u_\mathcal{O}, \pi, r)$, where the components are described as follows.

State of mind. A snapshot of the agent's state of mind is given a concrete form in their *agent state*, which is initialized to $s_\varnothing$ at the beginning of the POHP.[1] The agent's state evolves according to update functions $u_\mathcal{A}$ and $u_\mathcal{O}$, which describe how actions and observations are "remembered" (encoded in the next agent state), respectively.[2] At the start of the POHP, the agent receives an observation, either of WAIT if the agent must wait for the daimon to act or a proper observation o related to the daimon's previous action. Eventually, the agent receives a proper observation o, at which point the agent updates their agent state to $s' = u_\mathcal{O}(s, o)$. The agent then chooses an action a and updates their agent state to $u_\mathcal{A}(s', a)$ to begin the observation–action cycle again. Ultimately, each history $h_\varnothing h$, composed of an initial history prefix $h_\varnothing$ and a postfix h, yields an agent state, so we recursively define a unified update function,

$$u : h_\varnothing h \mapsto \begin{cases} s_\varnothing & \text{if } h = \varnothing \\ u_\mathcal{A}\big(u\big(h_\varnothing h_{<|h|}\big), h_\varnothing h_{|h|}\big) & \text{if } h_\varnothing h_{<|h|} \in \mathcal{H}_\mathcal{A} \\ u_\mathcal{O}\big(u\big(h_\varnothing h_{<|h|}\big), \omega(h_\varnothing h)\big) & \text{o.w.} \end{cases}$$

Each agent state corresponds to an *information set*, $I(s) = \{h \mid u(h) = s\}$, which is the set of histories the environment could be in, given the agent's state is s. We denote the set of agent states that could ever be generated as $\mathcal{S}$, and we partition them into the passive agent states where the agent awaits an observation, $\mathcal{S}_\mathcal{O}$, and the active agent states where

[1] In the work that this chapter is based on (Morrill, A. R. Greenwald, et al. 2022), the agent state concept was originally called "information state" to make connections with previous work (*e.g.*, Srinivasan et al. 2018 and D'Orazio, Morrill, et al. 2020) and the concept of information sets from extensive-form games. However, calling this concept "information state" makes discussing states and information sets as separate concepts difficult, and alienates those who are unfamiliar with the extensive-form game notion of information sets. The term "agent state" that this thesis uses also has the benefit that it is already used within the RL community to reference the same concept (*e.g.*, Dong et al. 2021).

[2] Splitting agent state updates into action and observation specific updates has a couple benefits: it allows us to refer to the agent's state of mind immediately after choosing an action, which we make use of in our reduction to Markov decision processes and in intermediate proof steps, and it allows the agent to forget the action they just chose without waiting for an observation, which could be useful in some applications involving asynchronous processing.

the agent acts, $\mathcal{S}_{\mathcal{A}}$. We overload $\mathcal{S}_{\mathcal{A}}(s, a) = \{u_{\mathcal{O}}(u_{\mathcal{A}}(s, a), \omega(hab))\}_{h \in I(s), b \in \mathcal{A}(ha)}$ as the set of child active agent states following s and action a, allowing the daimon to choose any action b in passive history ha.

Behavior. The agent acts by sampling actions from a behavioral strategy (also called a policy), $\pi \in \Pi$, where probability distributions over legal actions are assigned to agent states. At active history $h \in I(s)$ associated with agent state s, the agent chooses an action by sampling from their *immediate strategy* at s, $\pi(s) \in \Delta(\mathcal{A}(h))$, where $\Delta(\mathcal{A}(h))$ is the probability simplex over $\mathcal{A}(h)$. We assume that the agent can always determine the legal actions from their agent state so we overload $\mathcal{A}(s) = \mathcal{A}(h)$ for all $s \in \mathcal{S}$ and $h \in I(s)$.

Goals. A bounded reward function, $r : \mathcal{O} \to [-U, U]$, provides quantitative feedback to the agent about their progress toward their goals. The *return* (cumulative reward) that the agent acquires from active history $h \in \mathcal{H}_{\mathcal{A}}$ is $G_h(\pi; \sigma) = \sum_{i=1}^{\infty} Y_i r\big(\omega(H_i)\big)$, where the initial history in the trajectory is $H_1 = h$, the agent's action on each step is $A_i \sim \pi\big(u(H_i)\big)$, the daimon's action on each step is $B_i \sim \sigma(H_i A_i)$, the history is updated as the concatenation $H_{i+1} = H_i A_i B_i$, and the continuation indicator is the product $Y_{i+1} = Y_i \Gamma_i \in \{0, 1\}$ with $Y_1 = 1$ and $\Gamma_i \sim \gamma(H_i)$.

Generally, the agent's goal is to maximize their return. The fact that the daimon's strategy is unknown and their actions are only partially observed prevents us from immediately formulating this goal as an optimization problem. Neither can an equilibrium concept be proposed as a solution concept without presupposing incentives and a level of rationality for the daimon. Hindsight rationality, in contrast, is well suited as a solution concept for POHPs as it focuses on self-improvement grounded in experience and requires no assumptions about the daimon.

Repetition is a key requirement of hindsight rationality, and while no history may ever repeat *within* a POHP, this is not a problem in a *repeated POHP*. Before each round t begins, the agent chooses strategy π^t and the daimon chooses strategy σ^t. The POHP plays out according to these strategies, after which the agent receives reward information. The agent can then compare the returns they achieved with π^t with those they could have achieved with alternative behavior.[3] A repeated POHP is a well defined online decision process as long as the POHP terminates almost surely so that the agent is unlikely to be stuck in a single round forever. In a repeated POHP, learning occurs across rounds rather than across actions within a single POHP evaluation.[4]

[3] The agent may estimate the returns for alternative behavior using importance corrections if this information is not provided explicitly at the end of each round, similarly to how reward functions are estimated in adversarial bandit contexts (see, *e.g.*, Lattimore et al. (2020)).

[4] Though it ought to be possible to construct a hindsight rationality objective within a single POHP evaluation since the agent need only encounter the same or similar agent states repeatedly.

4.2.3 Reach Probabilities

Consider random history H generated according to agent strategy π, daimon strategy σ, and continuation function γ. The probability that history h is a prefix of H follows from the chain rule of probability, $\mathbb{P}_{\pi,\sigma}[h \sqsubseteq H] = \prod_{i=1}^{|h|} \mathbb{P}_{\pi,\sigma}[h_i \,|\, h_{<i}]$ where

$$\mathbb{P}_{\pi,\sigma}[h_i \,|\, h_{<i}] = \begin{cases} \pi(h_i \,|\, u(h_{<i})) & \text{if } h_{<i} \in \mathcal{H}_{\mathcal{A}} \\ \sigma(h_i \,|\, h_{<i})\gamma(h_{<i-1}) & \text{o.w.} \end{cases}$$

We denote $\mathbb{P}_{\pi,\sigma}[h] = \mathbb{P}_{\pi,\sigma}[h \sqsubseteq H]$ and refer to this quantity as h's *reach probability.* We can decompose $\mathbb{P}_{\pi,\sigma}[h] = \mathbb{P}_{\pi}[h]\mathbb{P}_{\sigma}[h]$ by grouping alternating terms

$$\mathbb{P}_{\pi}[h] = \prod_{i=1,\, h_{<i}\in\mathcal{H}_{\mathcal{A}}}^{|h|} \pi(h_i \,|\, u(h_{<i}))$$
$$\mathbb{P}_{\sigma}[h] = \prod_{i=1,\, h_{<i}\in\mathcal{H}_{\mathcal{O}}}^{|h|} \sigma(h_i \,|\, h_{<i})\gamma(h_{<i-1}).$$

$\mathbb{P}_{\pi}[h]$ represents the probability that the agent plays to reach h and $\mathbb{P}_{\sigma}[h]$ represents the joint probability that the daimon plays to reach h and that the history process continues long enough to reach h.

The conditional probability

$$\mathbb{P}_{\pi,\sigma}[h' \,|\, h] = \frac{\mathbb{P}_{\pi,\sigma}[h \sqsubseteq H, h' \sqsubseteq H]}{\mathbb{P}_{\pi,\sigma}[h]}$$

is the probability that history $h' \sqsubseteq H$ given $h \sqsubseteq H$. If h' and h are unrelated in that $h' \not\sqsubseteq h \not\sqsubseteq h'$, then it is not possible for H to realize both, so the joint probability $\mathbb{P}_{\pi,\sigma}[h, h'] = 0$, and consequently $\mathbb{P}_{\pi,\sigma}[h' \,|\, h] = 0$. If $h' \sqsubseteq h$ then H always realizes h' when h is realized, then $\mathbb{P}_{\pi,\sigma}[h, h'] = \mathbb{P}_{\pi,\sigma}[h]$ and $\mathbb{P}_{\pi,\sigma}[h' \,|\, h] = 1$. The last case is $h \sqsubseteq h'$, where $\mathbb{P}_{\pi,\sigma}[h', h] = \mathbb{P}_{\pi,\sigma}[h']$ so that

$$\mathbb{P}_{\pi,\sigma}[h' \,|\, h] = \frac{\mathbb{P}_{\pi,\sigma}[h']}{\mathbb{P}_{\pi,\sigma}[h]}.$$

4.3 Representing Traditional Models

The POHP model generalizes many traditional models. Here we describe reductions to game and Markov models.

4.3.1 Games

A *game* is an N player interaction where each player simultaneously chooses a strategy and immediately receives a payoff from a bounded utility function (Neumann et al. 1947). There

Algorithm 1 The procedure for playing an N player game in POHP-form. The input to this algorithm is either a pre-constructed POHP-form game or an EFG from which the POHP-form.

```
Input: turn function $p : \mathcal{H} \to \{c\} \cup \{i\}_{i=1}^{N}$,
    legal actions function $\mathcal{A}$,
    terminal histories $\mathcal{Z} \subseteq \mathcal{H}$
    or continuation function $\gamma : \mathcal{H} \to \Delta\{0, 1\}$,
    information partitions $\{\mathcal{I}_i\}_{i \in \{c\} \cup \{j\}_{j=1}^{N}}$
    or observation functions $\{\omega_i : \mathcal{H} \to \mathcal{S}_i\}_{i \in \{c\} \cup \{j\}_{j=1}^{N}}$,
    and utility functions $\{\upsilon_i : \mathcal{Z} \to [-U, U]\}_{i=1}^{N}$.
for $i \in \{c\} \cup \{j\}_{j=1}^{N}$ do
    $\omega_i(h) \leftarrow I$ for $h \in I \in \mathcal{I}_i$ if $\omega_i$ undefined
$\gamma \leftarrow h \mapsto \mathbb{1}\{h \notin \mathcal{Z}\}$ if $\gamma$ undefined
$H \leftarrow \varnothing$
$\Gamma \leftarrow 1$
while $\Gamma$ do
    send $\omega_i(H)$ to player $p(H)$
    receive $A \in \mathcal{A}(H)$ from player $p(H)$
    $H \leftarrow HA$
    sample $\Gamma \sim \gamma(H)$
for $i = 1, 2, \ldots, N$ do
    send $\omega_i(H) = (H, \upsilon_i(H))$ to player $i$
```

may also be an extra "chance player", denoted c, who "decides" chance events like die rolls with strategy π_c. A game described in this way is called a *normal-form game* (*NFG*).

For any given player, i, we can represent i's view of the game with a POHP, $\mathcal{G}_i$, where the agent represents i and the daimon represents the other $N-1$ players and chance in aggregate. We can also represent chance's view of the game with a POHP where the chance agent's strategy is fixed to π_c. The histories, action sets, and continuation function across all $N+1$ of these POHPs are shared but the first turn indicator and observation functions are specific to each player. The reward functions for each player must also reflect the game's payoffs. After each player chooses an agent strategy for their POHP, all the POHPs are evaluated together, sharing the same history, and each player receives a return in their POHP that equals their payoff in the game.

Together, the set of POHPs, $\{\mathcal{G}_i\}_{i \in \{c\} \cup \{j\}_{j=1}^{N}}$, represents what we could call a *POHP-form game*. In each $\mathcal{G}_i$, the daimon's strategy, σ_i, must reflect those of the other players. If the game is defined with a *turn function* $p : \mathcal{H} \to \{c\} \cup \{j\}_{j=1}^{N}$ that determines which player acts after a given history h, we can set σ_i to conform to the agent strategies from the other POHPs as $\sigma_i(h) = \pi_{p(h)}(u_{p(h)}(h))$.

A turn-based game described with histories is called an *extensive-form game* (*EFG*; Kuhn

1953). Any NFG can be converted into extensive form by serializing each decision. In the process, of course, players who act later are not allowed to observe previous actions, and this is traditionally specified through information partitions. Each player, i, is assigned an *information partition*, denoted $\mathcal{I}_i$, constructed by partitioning all of player i's active histories into information sets. Typically, EFGs also define a set of *terminal histories*, $\mathcal{Z} \subseteq \mathcal{H}$, which is constructed so that every history eventually terminates. As in a NFG, players receive payoffs upon termination.

Since the POHP and EFG share the same history-based progression, representing an EFG in POHP-form simply requires that information partitions, terminal histories, and utility functions are faithfully reconstructed in the POHP. To reconstruct information partitions, we must construct the POHP for each player i so that each history $h \in I' \in \mathcal{I}_i$ yields an active agent state $s = u(h)$ where the POHP information set $I(s) = I'$ matches the EFG information set. We can complete this reconstruction by showing player i their EFG information set through their observations, *e.g.*, $\omega_i(h) = I'$, and by setting player i's observation update function to $u_{\mathcal{O}} : s, o \mapsto o$. We can add terminal histories to a POHP by setting the continuation function to $\gamma : h \mapsto \mathbb{1}\{h \notin \mathcal{Z}\}$. To respect the EFG's utility function for each player i, $\upsilon_i : \mathcal{Z} \to [-U, U]$, we set player i's rewards in the POHP to zero except those on a terminal history, z, at which point $r_i(\omega_i(z)) = \upsilon_i(z)$.

See Algorithm 1 for a programmatic description of how a game, given in either POHP or extensive form, can be played out in POHP form.

4.3.2 Markov Models

The POHP model has extremely simple dynamics, the next action is merely appended to the current history, but this leads to a constantly expanding, extremely complex history set. To manage this complexity, agents can construct their own abstractions through agent state, which operates in the opposite way; the agent state dynamics may be mechanically complicated but they can produce compact agent state sets. Instead of proposing a complex but mechanically simple environment, the popular class of Markov models build complicated dynamics into the environment to make them more compact and to allow agents to take advantage of this structure.

In a Markov model, the environment has a state S that evolves according to the actions of agents and a transition distribution. In a general *partially observable Markov game* (*POMG*; Hansen et al. 2004), all of the agents in a Markovian environment choose an action, and the combination of these actions, $(A_i)_{i=1}^N$, determines the distribution over next environment states given the current environment state, $\mathbb{P}[\cdot \mid S, (A_i)_{i=1}^N]$.[5] The next state is then sampled

[5] A Markov game is also often called a "stochastic game", but a core feature of this model is Markovian

according to $\mathbb{P}[\cdot \mid S, (A_i)_{i=1}^N]$ and the cycle repeats. The environment is *Markovian* because the next state distribution is conditionally independent of all previous states and actions.

To reproduce a POMG with a POHP-form game, we must construct a Markovian environment state and each player's observations must depend only on the environment state rather than the underlying action history. Agent states in a POHP may not satisfy the Markov property even if the daimon's strategy is fixed because the daimon's strategy may depend on the underlying action history. However, we can easily reproduce a Markovian environment state with the passive agent states of the chance player in a POHP-form game.

At each of chance's passive histories h, each non-chance player chooses an action in turn, which advances the history to $h' = ha_1 \dots a_N$ where chance updates their agent state to $s_{h'} = u_c(h') \in \mathcal{S}_{c,\mathcal{A}}$. Since each player acts in turns rather than simultaneously, we are technically constructing a *turn-taking POMG* (*TT-POMG*; A. Greenwald, J. Li, et al. 2017), though a TT-POMG is functionally equivalent to a simultaneous action POMG. Chance then chooses which of their passive agent states is next by sampling A_c from $\pi_c(s_{h'})$, resulting in a transition to $s_{h'A_c} = u_c(h'A_c) \in \mathcal{S}_{c,\mathcal{O}}$. A Markovian transition between s_h and $s_{h'A_c}$ can be enforced by constructing chance's observation function so that $\omega_c(ha_1 \dots a_N) = \omega_c(\bar{h}a_1 \dots a_N)$ for all joint player actions $a_1 \dots a_N$ and histories $\bar{h} \in I(s_h)$, which, *e.g.*, is satisfied by the simple observation function $\omega_c(ha_1 \dots a_N) = a_1 \dots a_N$. Enforcing this constraint for each pair of histories $h, \bar{h}$ ensures that if $u_c(\bar{h}) = s_h$, then, given joint player actions $a_1 \dots a_N$,

$$\begin{aligned} u_c(\bar{h}a_1 \dots a_N A_c) &= u_{c,\mathcal{A}}(u_{c,\mathcal{O}}(u_c(\bar{h}), \omega_c(\bar{h}a_1 \dots a_N)), A_c) \\ &= u_{c,\mathcal{A}}(u_{c,\mathcal{O}}(s_h, \omega_c(ha_1 \dots a_N)), A_c) \\ &= s_{h'A_c} \end{aligned}$$

with transition probability $\mathbb{P}[\cdot \mid s_h, a_1 \dots a_N] = \pi_c(A_c \mid s_{h'})$.

To complete the reduction, we must construct each player's observation function so that it only depends on chance's passive agent state. For each player i and active history $h \in \mathcal{H}_{\mathcal{A},i}$, set $\omega_i(h) = \omega'_i(u_c(h))$, where ω'_i is player i's POMG observation function. The POMG model is also typically presented as a continuing process with discounting, which we can replicate by setting the continuation probability $\gamma(h)$ to the POMG discount factor for each of chance's passive histories h and $\gamma(h') = 1$ for all other histories h'.

Providing full observability to player i in a POHP-form POMG is simply a matter of revealing chance's passive agent state to player i, *i.e.*, set $\omega_i(h) = u_c(h)$ for each active history $h \in \mathcal{H}_{\mathcal{A},i}$. If all players are granted full observability, then a POMG becomes, naturally, a *Markov game* (L. S. Shapley 1953). Furthermore, a single-player Markov game or POMG reduces to a *Markov decision process* (*MDP*) or *partially observable MDP* (*POMDP*;

transitions, not stochasticity. This leads us to prefer the term "Markov game".

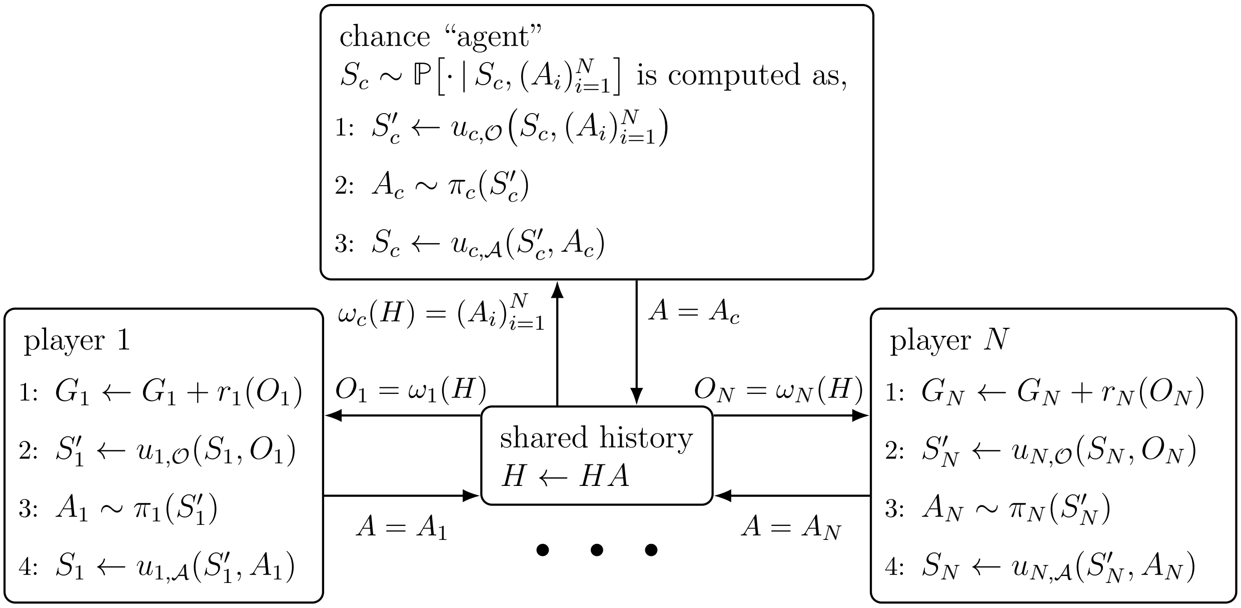

chance "agent"
$S_c \sim \mathbb{P}\big[\cdot \mid S_c, (A_i)_{i=1}^N\big]$ is computed as,
1: $S'_c \leftarrow u_{c,\mathcal{O}}\big(S_c, (A_i)_{i=1}^N\big)$
2: $A_c \sim \pi_c(S'_c)$
3: $S_c \leftarrow u_{c,\mathcal{A}}(S'_c, A_c)$
$\omega_c(H) = (A_i)_{i=1}^N$
$A = A_c$
player 1
1: $G_1 \leftarrow G_1 + r_1(O_1)$
2: $S'_1 \leftarrow u_{1,\mathcal{O}}(S_1, O_1)$
3: $A_1 \sim \pi_1(S'_1)$
4: $S_1 \leftarrow u_{1,\mathcal{A}}(S'_1, A_1)$
$O_1 = \omega_1(H)$
shared history
$H \leftarrow HA$
$A = A_1$
$O_N = \omega_N(H)$
$A = A_N$
player N
1: $G_N \leftarrow G_N + r_N(O_N)$
2: $S'_N \leftarrow u_{N,\mathcal{O}}(S_N, O_N)$
3: $A_N \sim \pi_N(S'_N)$
4: $S_N \leftarrow u_{N,\mathcal{A}}(S'_N, A_N)$

Figure 4.2: An N-player, POHP-form game where the agents for players 2 through $N-1$ are not shown. (**POMG**) If O_i for each player i only depends on S_c (*Markovian*), then the POHP-form game reproduces a POMG where S_c is the POMG state. (**Markov game**) If $N > 1$ and $O_i = S_c$ for each player i, then it reproduces a Markov game where S_c is the Markov game state. (**POMDP**) If $N = 1$ and the POHP is Markovian, then it reproduces a POMDP where S_c is the POMDP state. (**MDP**) If $N = 1$ and $O_1 = S_c$, then it reproduces an MDP where S_c is the MDP state. (**EFG**) If $\gamma(H) = \mathbb{1}\{H \notin \mathcal{Z}\}$ and $r_i(\omega_i(H)) = \upsilon_i(H)$ if $H \in \mathcal{Z}$ and 0 otherwise, then the POHP-form game reproduces an EFG with terminal histories $\mathcal{Z}$ and payoff functions $(\upsilon_i)_{i=1}^N$.

Smallwood et al. 1973), respectively, and this is true when the model is represented in either canonical or POHP-form.

Figure 4.2 visualizes a POHP-form game and summarizes all of the reductions to traditional models.

4.4 The Sub-POHP

For the rest of this chapter, we consider finite-horizon POHPs with timed update functions. A POHP has a *finite horizon* if every history eventually terminates deterministically. We enforce this condition by selecting a subset of histories, $\mathcal{Z} \subseteq \mathcal{H}$ where $\gamma(z) = 0$ for all $z \in \mathcal{Z}$. The agent's updates are *timed* as long as the agent's action update function records the number of actions the agent has taken. A finite horizon and timed updates ensure that the number of histories in each information set is finite and the same agent state is never encountered twice before termination. Thus, the agent states are partially ordered and we can write $s \prec s'$ to denote that agent state s is a predecessor of s'.

We now describe how sub-POHPs can be constructed in finite-horizon POHPs with timed updates and show how observable sequential rationality is naturally defined in terms of sub-POHPs. This exploration will provide the base for the development and analysis of the extensive-form regret minimization (EFR) algorithm for finite-horizon POHPs with perfect-recall updates in Chapter 7.

4.4.1 Beliefs and Realization Weights

Given that the agent's state is s, how likely is it that the agent is in a particular history $h \in I(s)$? Traditionally, this is called the agent's *belief* (about which history they are in) at s. According to Bayes' rule, $\mathbb{P}_{\pi,\sigma}[h \,|\, s] = \mathbb{P}_{\pi,\sigma}[s \,|\, h]\mathbb{P}_{\pi,\sigma}[h]/\mathbb{P}_{\pi,\sigma}[s]$. Since $h \in I(s)$, $\mathbb{P}_{\pi,\sigma}[s \,|\, h] = 1$. The agent's belief at s is then $\xi^{\pi,\sigma}_s : h \mapsto \mathbb{P}_{\pi,\sigma}[h]/\mathbb{P}_{\pi,\sigma}[s]$.

To evaluate $\mathbb{P}_{\pi,\sigma}[s]$, consider that the agent's state is s only if the random history H lands in $I(s)$, so we can describe the event of realizing s as the union of history realization events. Since we assume the agent's updates are timed, there is at most one prefix of H in $I(s)$, which means that each $h \sqsubseteq H$ event for $h \in I(s)$ is disjoint. The probability of their union is thus the sum

$$\mathbb{P}_{\pi,\sigma}[s] = \mathbb{P}_{\pi,\sigma}\left[\bigcup_{h \in I(s)} h \sqsubseteq H\right] = \sum_{h \in I(s)} \mathbb{P}_{\pi,\sigma}[h].$$

An assignment of beliefs to each agent state is called a *system of beliefs*. A problem that arises in defining a complete system of beliefs from a given π–σ pair is that some agent states may be unrealizable ($\mathbb{P}_{\pi,\sigma}[s] = 0$). Motivated by a desire to describe how rational players would play games or to deploy strong static artifacts, various rationality assumptions have been studied that complete belief systems in different ways and lead to different equilibrium concepts (see, *e.g.*, Breitmoser et al. (2010), Dekel et al. (2015), and Kreps et al. (1982)). However, from a hindsight rationality perspective, only realizable agent states could have been observed by the agent, and only behavior in realizable states could have impacted the agent's return. Thus, beliefs at unreachable agent states are naturally left undefined.

As a consequence, agent state realization probabilities hold special significance in hindsight analysis, as they determine whether or not a state is observable. More generally, they provide a measure of importance to each agent state. Let J be the (possibly infinite) random step a trajectory of active histories $\{H_i\}_{i=1}^{\infty}$ where s is realized at history H_J, *i.e.*, $u(H_J) = s$. The return from H_1 can be split as

$$G_{H_1}(\pi;\sigma) = \sum_{i=1}^{\infty} \underbrace{\mathbb{1}\{i < J\} Y_i r\big(\omega(H_i)\big)}_{\text{Return before } s.} + \underbrace{\mathbb{1}\{i \geq J\} Y_i r\big(\omega(H_i)\big)}_{\text{Return after } s.}.$$

Setting $H_1 \sim \xi$ to be an initial history and taking the expectation,

$$\mathbb{E}[G_{H_1}(\pi;\sigma)] = \mathbb{E}\left[\sum_{i=1}^{\infty} \mathbb{1}\{i < J\} Y_i r\big(\omega(H_i)\big)\right] + \mathbb{P}_{\pi,\sigma}[s]\mathbb{E}_{H_J \sim \xi_s^{\pi,\sigma}}[G_{H_J}(\pi;\sigma)].$$

We define the *realization-weighted expected return* from s,

$$v_s(\pi;\sigma) = \mathbb{P}_{\pi,\sigma}[s]\mathbb{E}_{H \sim \xi_s^{\pi,\sigma}}[G_H(\pi;\sigma)], \tag{4.1}$$

where $v_s(\pi;\sigma)$ is naturally zero if $\xi_s^{\pi,\sigma}$ is undefined, to summarize s's contribution to the agent's expected return.

4.4.2 Observable Sequential Rationality

Here we capitalize on the generality of our POHP definition. An agent belief can be used as a distribution over initial histories to define a POHP, which in this context we call a *sub-POHP*. Thus, every realizable agent state s admits a sub-POHP with the initial history distribution $\xi(h) = \xi_s^{\pi,\sigma}(h)$ if $h \in I(s)$ and zero otherwise.

Sequential rationality can then be defined as optimal behavior within every sub-POHP with respect to an assignment of beliefs to unrealizable agent states. This definition is equivalent to sequential rationality in a single-player EFG (Kreps et al. 1982). Our new extension, *observable sequential rationality* (*OSR*), merely drops the requirement that play must be rational at unrealizable (and therefore unobservable) agent states. OSR is a weaker condition than any previous form of sequential rationality, including that of weak sequential equilibrium (Hillas 1987; Myerson 1997), because OSR can be achieved while choosing dominated actions at unobservable agent states. Nonetheless, OSR is indistinguishable from sequential rationality under Bayesian beliefs according to all outcomes that could be observed by the agent given a daimon strategy, which makes it a natural refinement of hindsight rationality.

We can generalize the idea of OSR to samples from a joint distribution of agent strategy–daimon strategy pairs (traditionally called a *recommendation distribution*) and deviations, which we use to construct a general definition of OSR in a POHP. The key value determining OSR is in fact Eq. (4.1), the realization-weighted expected return. The OSR condition can thus be written in terms of a generalized full regret.

Definition 1. *Define the* full regret *from agent state s as the difference in realization-weighted expected return under ϕ compared with $\phi_{\prec s}$,* i.e., $\rho_s(\phi,\pi;\sigma) = v_s(\phi(\pi);\sigma) - v_s(\phi_{\prec s}(\pi);\sigma)$*, where $\phi_{\prec s}$ is the deviation that applies ϕ only before s,* i.e., $[\phi_{\prec s}x](\bar{s}) = [\phi x](\bar{s})$ *if $\bar{s} \prec s$ and $x(\bar{s})$ otherwise.*

Definition 2. *A recommendation distribution, $\mu \in \Delta(\mathcal{X} \times \mathcal{D})$, where $\mathcal{X}$ and $\mathcal{D}$ are the sets of pure strategies for the agent and daimon, respectively, is OSR for the agent with respect to* *a set of deviations, $\Phi \subseteq \Phi^{\text{sw}}_{\mathcal{X}}$, if the maximum benefit for every deviation, $\phi \in \Phi$, according to the realization-weighted expected return from every agent state, $s \in \mathcal{S}$, is non-positive,*

$$\mathbb{E}_{(x,d)\sim\mu}[\rho_s(\phi, x; d)] \leq 0.$$

The hindsight analogue to Definition 2 follows.

Definition 3. *An agent is observably sequentially (OS) hindsight rational if they are a no-full-regret learner in every realizable agent state within a given POHP with respect to $\Phi \subseteq \Phi^{\text{sw}}_{\mathcal{X}}$. That is, the agent generates for any $T > 0$ a sequence of strategies, $(\pi^t)_{t=1}^T$, where $\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \rho_s(\phi, \pi^t; \sigma^t) \leq 0$ at each s for each $\phi \in \Phi$ under any sequence of daimon strategies $(\sigma^t)_{t=1}^T$. The positive part of the agent's maximum average full regret across active agent states is their* OSR gap*.*

The discussion that Myerson (1997) has at the start of Section 4.4 is illustrative of the difference between the equilibrium and hindsight rationality perspectives on sequential rationality. Myerson (ibid.) explains that it is insufficient to consider how sequentially-rational play is only at agent states that are observed in equilibrium play because equilibrium play may only be motivated under particular behavior at unobserved agent states. One way to resolve this issue is to assign arbitrary beliefs to each unobservable agent state and ensure sequential rationality at each agent state, *i.e.*, weak sequential rationality. Observable sequential equilibrium shows that, at least in a learning context, a more natural approach is to consider the play at *observable* rather than the *observed* agent states, the difference being that observable states could be observed under a deviation by the agent, without assigning beliefs at unobservable states.

On each round, the daimon fixes their strategy, thereby constraining which agent states are observable by the agent. After this round, the agent can look back on their behavior and potential deviations at each observable agent state to improve upon their strategy. If the daimon knows that the agent's strategy is poor in particular sub-POHPs, then the daimon can play to those sub-POHPs and exploit the agent's weakness. The agent states in these sub-POHPs become observable on the next round and the agent learns to improve their play there. If the daimon never leads the agent to an agent state, then there is no reward-based motivation for the agent to even consider how they would play in that state, which would presumably require time and an allocation of computational resources.

4.4.3 An Attempt at Local Learning

Consider a local learning problem in a repeated finite-horizon POHP with timed updates based on the realization-weighted expected return at each active agent state s. Given a set of deviations, $\Phi \subseteq \Phi^{\text{SW}}_{\mathcal{X}}$, we can construct a set of truncated deviations, $\Phi_{\preceq s} = \{\phi_{\preceq s}\}_{\phi \in \Phi}$, where each deviation in $\Phi_{\preceq s}$ applies a deviation from Φ until after an action has been taken in s, at which point the rest of the strategy is left unmodified. Each truncated deviation represents a way that the agent could play to and in s so a natural local learning problem is for the agent to choose their actions at s so that there is no beneficial truncated deviation.

To apply deviations to the agent's behavioral strategies, notice that sampling an action for each agent state under timed updates yields a pure strategy. Thus, a behavioral strategy defines a probability distribution over the set of pure strategies, $\mathcal{X}$. We overload $\pi : \mathcal{X} \to \Delta(\mathcal{X})$ to return the probability of a given pure strategy under behavioral strategy $\pi \in \Pi$. From this perspective, π may be called a *mixed strategy.* The transformation of π by deviation ϕ is the pushforward measure $\phi(\pi)$ defined pointwise by $[\phi\pi](x') = \sum_{x \in \phi^{-1}(x')} \pi(x)$ for all $x' \in \mathcal{X}$, where $\phi^{-1} : x' \mapsto \{x \mid \phi(x) = x'\}$ is the pre-image of ϕ.

The *immediate regret* at agent state s for not employing truncated deviation $\phi_{\preceq s}$ is a difference in realization-weighted expected return under $\xi_s^{\phi_{\prec s}(\pi),\sigma}$:

$$\begin{aligned}\rho_s(\phi_{\preceq s}, \pi; \sigma) &= v_s(\phi_{\preceq s}(\pi); \sigma) - v_s(\phi_{\prec s}(\pi); \sigma)\\ &= \mathbb{P}_{\phi_{\prec s}(\pi),\sigma}[s]\mathbb{E}[G_H(\phi_{\preceq s}(\pi); \sigma) - G_H(\pi; \sigma)].\end{aligned}$$

Intuitively, it is the advantage that $\phi_{\preceq s}(\pi)$ has over π in s assuming that the agent plays to s according to $\phi_{\prec s}(\pi)$.[6] Sadly, it can be impossible to prevent agent state s's immediate regret with respect to $\Phi_{\preceq s}$ from growing linearly in a repeated POHP.

Theorem 3. *An agent with timed updates cannot generally prevent immediate regret from growing linearly in a finite-horizon repeated POHP.*

Proof. Consider a two action, two agent state POHP where agent state s transitions to s' where the reward is $+1$ if the agent chooses the same action in both s and s', and -1 otherwise. The two *external* (constant) deviations, $\phi^{\to 1}$ and $\phi^{\to 2}$, that choose the same actions in both agent states always achieve a value of $+1$. At s', the agent has to choose between achieving value with respect to the play of $\phi^{\to 1}$ or $\phi^{\to 2}$ in s. If the agent chooses action #1, then $v_s(\phi^{\to 1}_{\prec s}(\pi); \sigma) = +1$ but $v_s(\phi^{\to 2}_{\prec s}(\pi); \sigma) = -1$, and *vice-versa* if they choose action #2. Therefore, the agent minimizes their maximum regret by always playing uniform random and suffering an expected regret of $+1$ on every round. □

Since timed updates are insufficient to guarantee the existence of no-immediate-regret algorithms, we will use a stronger property: perfect recall.

[6] The term "advantage" here is chosen deliberately as immediate regret is analogous to advantage in MDPs (Baird 1994), with respect to a given, rather than optimal, policy (see, *e.g.*, Kakade (2003)).

4.5 General Immediate Regret Minimization with Perfect Recall

Perfect recall requires that every bit of information from every action and observation is encoded in the agent state, *e.g.*, update functions that concatenate the previous agent state with the given action or observation. As the terminology suggests, agents with perfect recall "remember" each of their actions and observations. This ensures that each agent state s' is either the initial agent state or has a single parent agent state s, *i.e.*, $u(h_{<|h|}) = s$ for each history $h \in I(s')$. As a result, perfect recall requires that there is a unique sequence of agent states leading up to each agent state $s \in \mathcal{S}$, *i.e.*, $\big(u(\bar{h})\big)_{\bar{h} \sqsubseteq h} = \Big(u(\bar{h}')\Big)_{\bar{h}' \sqsubseteq h'}$ for each pair of histories $h, h' \in I(s)$.

Denote the sequence of active history prefixes of history h as $\eta_{\mathcal{H}_{\mathcal{A}}}(h) = (h_{<2i})_{i=1}^{\lceil |h|/2 \rceil}$ and the sequence of agent actions as $\eta_{\mathcal{A}}(h) = (h_{2i})_{i=1}^{\lceil |h|/2 \rceil}$. As long as the agent always remembers their own actions, which a prerequisite for perfect recall, there is a unique sequence of agent actions that leads to s and we can overload $\eta_{\mathcal{A}}(s) = \eta_{\mathcal{A}}(h)$ where $h \in I(s)$ is arbitrary. Consequently, a perfect recall agent can only play to reach each history in s equally, *i.e.*, $\mathbb{P}_\pi[h] = \mathbb{P}_\pi[h']$ for all $h, h' \in I(s)$. If we define $\mathbb{P}_\pi[s] = \mathbb{P}_\pi\Big[\bigcup_{h \in I(s)} h\Big]$, then perfect recall implies that $\mathbb{P}_\pi[s] = \mathbb{P}_\pi[h']$ for any history $h' \in I(s)$.

Therefore, under perfect recall, the probability of realizing s simplifies to

$$\mathbb{P}_{\pi,\sigma}[s] = \sum_{h \in I(s)} \mathbb{P}_\pi[h]\mathbb{P}_\sigma[h] = \mathbb{P}_\pi[s] \sum_{h \in I(s)} \mathbb{P}_\sigma[h].$$

The belief about any history $h \in I(s)$ then simplifies to

$$\xi_s^{\pi,\sigma}(h) = \frac{\mathbb{P}_\pi[h]\mathbb{P}_\sigma[h]}{\mathbb{P}_\pi[s]\sum_{h \in I(s)} \mathbb{P}_\sigma[h]} = \frac{\mathbb{P}_\sigma[h]}{\sum_{h \in I(s)} \mathbb{P}_\sigma[h]}.$$

The realization-weighted expected return simplifies to

$$v_s(\pi; \sigma) = \mathbb{P}_\pi[s] \underbrace{\sum_{h \in I(s)} \mathbb{P}_\sigma[h]\mathbb{E}[G_h(\pi; \sigma)]}_{v_s^{\text{CF}}(\pi;\sigma)}. \tag{4.2}$$

We can recognize the sum denoted $v_s^{\text{CF}}(\pi; \sigma)$ as the counterfactual value of s, which does not depend on π's play at s's predecessors. General immediate regret becomes weighted immediate counterfactual regret,

$$\rho_s(\phi_{\preceq s}, \pi; \sigma) = \mathbb{P}_{\phi_{\prec s}(\pi)}[s](v_s^{\text{CF}}(\phi_s(\pi); \sigma) - v_s^{\text{CF}}(\pi; \sigma)).$$

Since the counterfactual value function does not depend on the agent's play at s's predecessors, perfect recall avoids the problem that leads to Theorem 3 under updates that are only timed, where no algorithm can ensure sublinear growth of cumulative immediate regret.

Perfect recall allows immediate regret minimization via a reduction to online time selection decision processes (OTSDPs; Section 2.3.4).

Theorem 4. *If the agent has perfect-recall in a repeated, finite-horizon POHP and there is no deviation $\phi \in \Phi \subseteq \Phi^{\mathrm{SW}}_{\mathcal{X}}$ for which the deviation reach-probability of active agent state s, $\mathbb{P}_{\phi_{\prec s}(\pi)}[s]$, depends on the agent's immediate strategy at s, $\pi(s)$, then the problem of minimizing immediate regret with respect to Φ at s reduces to that of minimizing regret with respect to action transformations $\Phi_s = \{\phi_s\}_{\phi \in \Phi}$ in an OTSDP.*

Proof. In a repeated POHP where the agent and daimon choose $\pi^t \in \Pi$ and $\sigma^t \in \Sigma$ on each round t the (round dependent) counterfactual value function $t \mapsto v^{\mathrm{CF}}_s(\cdot; \sigma^t)$ fills the role of the OTSDP reward function, the set of (round dependent) reach probability functions $\left\{w_{s,\phi} : t \mapsto \mathbb{P}_{\phi_{\prec s}(\pi^t)}[s]\right\}_{\phi \in \Phi}$ fills the role of the set of time selection functions, and the set of action transformations that could be made at s, Φ_s, fills the role of OTSDP deviations. An OTSDP requires giving the agent all weights on time t, *i.e.*, $\{w^t_{s,\phi}\}_{\phi \in \Phi}$, before they must choose a strategy. Since we assume that $\mathbb{P}_{\phi_{\prec s}(\pi^t)}[s]$ does not depend on $\pi^t(s)$, each weight $w^t_{s,\phi}$ can be computed at the start of each OTSDP round, which completes the reduction. □

With perfect recall, we can also give a bound on full regret as a function of immediate regret. This result is achieved by decomposing full regret into the sum of immediate regrets across active agent states and generalizes Zinkevich, Johanson, et al. (2007b)'s original counterfactual regret decomposition (Lemma 5) so that it applies to any deviation.

Lemma 1. *In a finite-horizon POHP, the realization-weighted expected return of active agent state s under perfect recall recursively decomposes as*

$$v_s(\pi; \sigma) = \mathbb{P}_\pi[s] r_s(\pi; \sigma) + \sum_{s' \in \bigcup_{a \in \mathcal{A}(s)} \mathcal{S}_{\mathcal{A}}(s,a)} v_{s'}(\pi; \sigma).$$

Proof. Multiplying the decomposed counterfactual value (see Lemma 3) by the reach weight,

$$\begin{aligned} v_s(\pi; \sigma) &= \mathbb{P}_\pi[s] r_s(\pi; \sigma) + \sum_{a \in \mathcal{A}(s)} \mathbb{P}_\pi[s] \pi(a \mid s) v^{\mathrm{CF}}_{u_{\mathcal{A}}(s,a)}(\pi; \sigma) \\ &= \mathbb{P}_\pi[s] r_s(\pi; \sigma) + \sum_{a \in \mathcal{A}(s)} v_{u_{\mathcal{A}}(s,a)}(\pi; \sigma). \qquad (4.3) \end{aligned}$$

Furthermore,

$$\begin{aligned} v_{u_{\mathcal{A}}(s,a)}(\pi; \sigma) =& \sum_{s' \in \mathcal{S}_{\mathcal{A}}(s,a)} \sum_{h \in u_{\mathcal{A}}(s,a)} \mathbb{1}\{u(h) = s'\} \mathbb{P}_{\pi,\sigma}[h] \mathbb{E}[G_h(\pi; \sigma)] \\ =& \sum_{s' \in \mathcal{S}_{\mathcal{A}}(s,a)} \underbrace{\sum_{h' \in I(s')} \mathbb{P}_{\pi,\sigma}[h'] \mathbb{E}[G_{h'}(\pi; \sigma)]}_{v_{s'}(\pi;\sigma)}. \qquad (4.4) \end{aligned}$$

Substituting Eq. (4.4) into Eq. (4.3) completes the proof. □

Lemma 2. *In a finite-horizon POHP, the full regret with respect to $\phi \in \Phi^{\text{SW}}_{\mathcal{X}}$ under perfect recall at active agent state s recursively decomposes as*

$$\rho_s(\phi, \pi; \sigma) = \rho_s(\phi_{\preceq s}, \pi; \sigma) + \sum_{s' \in \bigcup_{a \in \mathcal{A}(s)} \mathcal{S}_{\mathcal{A}}(s,a)} \rho_{s'}(\phi, \pi; \sigma).$$

Proof.

$$\begin{aligned}
\rho_s(\phi, \pi; \sigma) &= v_s(\phi(\pi); \sigma) \overbrace{- v_s(\phi_{\preceq s}(\pi); \sigma) + \mathrlap{\underbrace{\phantom{v_s(\phi_{\preceq s}(\pi); \sigma) - v_s(\phi_{\prec s}(\pi); \sigma)}}_{\rho_s(\phi_{\preceq s}, \pi; \sigma)}} v_s(\phi_{\preceq s}(\pi); \sigma)}^{0} - v_s(\phi_{\prec s}(\pi); \sigma) \\
&= \rho_s(\phi_{\preceq s}, \pi; \sigma) \\
&\quad + \underbrace{\mathbb{P}_{\phi(\pi)}[s] r_s(\pi; \sigma) - \mathbb{P}_{\phi_{\preceq s}(\pi)}[s] r_s(\pi; \sigma)}_{0} \\
&\quad + \sum_{a \in \mathcal{A}(s)} \underbrace{v_{u_{\mathcal{A}}(s,a)}(\phi(\pi); \sigma) - v_{u_{\mathcal{A}}(s,a)}(\phi_{\preceq s}(\pi); \sigma)}_{\rho_{u_{\mathcal{A}}(s,a)}(\phi, \pi; \sigma)}.
\end{aligned}$$

Applying Eq. (4.4) to sum over active agent states,

$$= \rho_s(\phi_{\preceq s}, \pi; \sigma) + \sum_{s' \in \bigcup_{a \in \mathcal{A}(s)} \mathcal{S}_{\mathcal{A}}(s,a)} \underbrace{v_{s'}(\phi(\pi); \sigma) - v_{s'}(\phi_{\prec s'}(\pi); \sigma)}_{\rho_{s'}(\phi, \pi; \sigma)}. \qquad \square$$

Theorem 5. *If an agent with perfect recall chooses their strategy in a repeated finite-horizon sub-POHP rooted at agent state s on each round t so that after T rounds, their immediate regret with respect to $\Phi \subseteq \Phi^{\text{SW}}_{\mathcal{X}}$ is upper bounded by $f(T) \geq 0$, $f(T) \in \mathrm{o}(T)$, then the agent's full regret at s is sublinear, upper bounded according to $\rho^{1:T}_s(\phi) = \sum_{t=1}^{T} \rho_s(\phi, \pi^t; \sigma^t) \leq |\mathcal{S}_{s,\mathcal{A}}| f(T)$, where $\mathcal{S}_{s,\mathcal{A}} = \{s' \in \mathcal{S}_{\mathcal{A}} \mid s \preceq s'\}$ is the number of active agent states in s's sub-POHP. Following this procedure every agent state thus guarantees OS hindsight rationality with respect to Φ.*

Proof. Working from each terminal agent state where the full and immediate regret are equal toward s, we recursively bound the cumulative full regret at every agent state according to Lemma 2. Every active agent state adds at most $f(T)$ to the cumulative full regret at s and there are $|\mathcal{S}_{s,\mathcal{A}}|$ active agent states in s's sub-POHP so the cumulative full regret at s is no more than $|\mathcal{S}_{s,\mathcal{A}}| f(T)$. □

The fact that the procedure described by Theorem 5 is OS hindsight rational is critical as OSR can elevate the strength of a special class of deviation type.

Definition 4. *The set of* single-target deviations *generated from an arbitrary set of deviations $\Phi \subseteq \Phi^{\text{SW}}_{\mathcal{X}}$ is*

$$\Phi_{\preceq \odot} = \{\phi' \mid \forall x,\ [\phi' x](\bar{s}) = [\phi x](\bar{s}) \text{ if } \bar{s} \preceq s \text{ and } x(\bar{s}) \text{ o.w.}\}_{\phi \in \Phi,\, s \in \mathcal{S}_{\mathcal{A}}}.$$

$\Phi_{\preceq\odot}$ is the set of deviations constructed from Φ that only deviate along a single sequence of agent states up to a "target" agent state and behave identically to the input strategy at all other agent states.

The set of single-target deviations is special because it captures all of the ways that the input strategy could be modified along any sequence of agent states without including the combinations of these modifications across multiple sequences. The set of single-target deviations can therefore be much smaller than its generating set while preserving the generating set's capacity to express different behavior modifications.

The next result formalizes the intuition that the set of single-target deviations preserves the essential expressive capacity of its generating set by proving that there is no beneficial deviation in an arbitrary set of deviations Φ if OSR is achieved with respect to $\Phi_{\preceq\odot}$.

Theorem 6. *If a perfect-recall agent's full regret at each active agent state s with respect to each single-target deviation $\phi \in \Phi_{\preceq\odot}$, is $d_s(\phi)f(T) \geq 0$, where $d_s(\phi)$ is the number of non-identity action transformations ϕ applies from s to the end of the finite-horizon POHP, then that agent's OSR gap with respect to $\Phi_{\preceq\odot}$ and $\Phi \subseteq \Phi_{\mathcal{X}}^{\mathrm{sw}}$ is no more than $|\mathcal{S}_{\mathcal{A}}|f(T)$.*

Proof. First, establish a simple fact about single-target deviations. At each active agent state s, the full regret with respect to each single-target deviation $\phi_{\preceq s}$ that transforms each action up to and at s, and then immediately re-correlates is no more than $f(T)$ since $d_s(\phi_{\preceq s}) = 1$. Formally, denote this value as

$$\rho_s^{1:T}(\phi_{\preceq s}) = \sum_{t=1}^{T} \rho_s(\phi_{\preceq s}, \pi^t; \sigma^t) \leq f(T). \tag{4.5}$$

Next, consider the terminal or height 1 active agent states for player i, *i.e.*, those without successors. The maximum full regret with respect to $\Phi_{\preceq\odot}$ is the positive part of the maximum full regret with respect to Φ at each terminal active agent state s; either the single-target deviation can change the action at s or it can re-correlate. Therefore, the theorem is proved if all active agent states are terminal as $d < |\mathcal{S}_{\mathcal{A}}|$. This serves as the base case of a proof by induction.

For the induction step, assume that the maximum full regret of a deviation in Φ and a single-target deviation in $\Phi_{\preceq\odot}$ at active agent state s is upper bounded at each immediate successor $s' \in \mathcal{S}_{\mathcal{A}}(s, a)$ by $(d-1)f(T)$, where d is the height of s ($d-1$ agent actions leads to a terminal active agent state). The full regret of deviation $\phi \in \Phi$ decomposes as

$$\rho_s^{1:T}(\phi) = \rho_s^{1:T}(\phi_{\preceq s}) + \sum_{s' \in \bigcup_{a \in \mathcal{A}(s)} \mathcal{S}_{\mathcal{A}}(s,a)} \rho_{s'}^{1:T}(\phi). \tag{4.6}$$

according to Lemma 2. We can bound the full regret at each s' by the induction assumption,

$$\rho_s^{1:T}(\phi) \leq \rho_s^{1:T}(\phi_{\preceq s}) + |\mathcal{S}_{\mathcal{A}}(s, a)|(d-1)f(T). \tag{4.7}$$

We can then bound the maximum immediate regret at s by Eq. (4.5),

$$\begin{aligned} \rho_s^{1:T}(\phi) &\leq f(T) + |\mathcal{S}_{\mathcal{A}}(s, a)|(d-1)f(T) && (4.8) \\ &\leq |\mathcal{S}_{\mathcal{A}}|f(T), && (4.9) \end{aligned}$$

where the last inequality follows from the fact that $(d-1)|\mathcal{S}_{\mathcal{A}}(s, a)| \leq |\mathcal{S}_{\mathcal{A}}| - 1$ under perfect recall. Equation (4.9) completes the proof. □

One important contribution of Part II is an investigation into the use of single-target deviations with OSR in the design of deviation types and algorithms.

4.6 Conclusion

This chapter introduced the POHP formalism for modeling complex, multi-agent RL environments from a single agent's perspective. The POHP formalism provides a mechanically simple alternative to a more complicated general formalism like the POMG, EFG or POMDP, and in fact generalizes all of these models. In contrast to the MDP formalism, the POHP formalism achieves this mechanical simplicity without sacrificing the ability to accurately model multi-agent interactions and partial observability.

A POHP model is also recursive in that sub-POHPs can naturally be constructed from the agent's states. Using this property, we showed how OSR can be formulated using the sub-POHP concept. We also showed that updates which are merely timed prevent efficient agent-state-local learning as such updates are insufficient to guarantee that the agent can minimize immediate regret. However, with perfect-recall updates, we showed that immediate regret minimization is possible and that it leads to full regret minimization via a generalized regret decomposition. Furthermore, we showed how OSR elevates the strength of single-target deviations, which is a key result that Part II will build on.

The generality and simplicity of the POHP formalism suggests that it may be useful in modeling continual learning problems where environments are expansive, unpredictable, and dynamic. Good performance in such environments demands that the agent continually learns, adapts, and re-evaluates their assumptions. Perhaps hindsight rationality could serve as the learning objective for such problems if it could be formulated for a single agent lifetime rather than over a repeated POHP.

The POHP formalism allows agents to determine their own representation of the environment, which opens the way to direct discussions and comparisons of agent state representations. One particular direction that is made natural by the POHP model's action–observation interface is predictive state representations (PSRs; S. Singh et al.; S. P. Singh et al. 2012; 2003). While PSRs were developed to model Markovian dynamical systems with at most one controller, the POHP model could facilitate an extension to multi-agent settings.

References

Baird, L. C. (1994). "Reinforcement learning in continuous time: Advantage updating". In: *1994 IEEE International Conference on Neural Networks (ICNN'94)*. Vol. 4. IEEE, pp. 2448–2453.

Breitmoser, Y., J. H. Tan, and D. J. Zizzo (2010). "On the beliefs off the path: Equilibrium refinement due to quantal response and level-k". In: *Nottingham University Business School Research Paper* 2010-07.

D'Orazio, R., D. Morrill, J. R. Wright, and M. Bowling (May 2020). "Alternative Function Approximation Parameterizations for Solving Games: An Analysis of f-Regression Counterfactual Regret Minimization". In: *19th International Conference on Autonomous Agents and Multi-Agent Systems*. International Foundation for Autonomous Agents and Multiagent Systems.

Dekel, E. and M. M. Siniscalchi (2015). "Epistemic Game Theory". In: *Handbook of Game Theory With Economic Applications* 4, pp. 619–702.

Dong, S., B. Van Roy, and Z. Zhou (2021). "Simple Agent, Complex Environment: Efficient Reinforcement Learning with Agent States". In: *CoRR* abs/2102.05261.

Farina, G., C. Kroer, and T. Sandholm (2019). "Online convex optimization for sequential decision processes and extensive-form games". In: *33rd AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 1917–1925.

Greenwald, A., J. Li, and E. Sodomka (2017). "Solving for Best Responses and Equilibria in Extensive-Form Games with Reinforcement Learning Methods". In: *Rohit Parikh on Logic, Language and Society*. Ed. by C. Başkent, L. S. Moss, and R. Ramanujam. Cham: Springer International Publishing, pp. 185–226. ISBN: 978-3-319-47843-2. DOI: 10.1007/978-3-319-47843-2_11.

Greenwald, A., Z. Li, and C. Marks (Jan. 2006a). "Bounds for Regret-Matching Algorithms". In: *International Symposium on Artificial Intelligence and Mathematics (ISAIM 2006)*. Fort Lauderdale, Florida, USA.

Hansen, E. A., D. S. Bernstein, and S. Zilberstein (2004). "Dynamic programming for partially observable stochastic games". In: *20th AAAI Conference on Artificial Intelligence (AAAI-04)*. Vol. 4, pp. 709–715.

Hillas, J. (1987). *Sequential equilibria and stable sets of beliefs*. Institute for Mathematical Studies in the Social Sciences, Stanford University.

Kakade, S. M. (2003). "On the sample complexity of reinforcement learning". PhD thesis. UCL (University College London).

Kovařík, V., M. Schmid, N. Burch, M. Bowling, and V. Lisỳ (2019). "Rethinking formal models of partially observable multiagent decision making". In: *arXiv preprint arXiv:1906.11110*.

Kreps, D. M. and R. Wilson (1982). "Sequential equilibria". In: *Econometrica* 50.4, pp. 863–894.

Kuhn, H. W. (1953). “Extensive Games and the Problem of Information”. In: *Contributions to the Theory of Games* 2. Ed. by H. W. Kuhn and A. W. Tucker, pp. 193–216.

Lattimore, T. and C. Szepesvári (2020). *Bandit algorithms.* Cambridge University Press.

Morrill, D., A. R. Greenwald, and M. Bowling (2022). “The Partially Observable History Process”. In: *AAAI-22 Workshop on Reinforcement Learning and Games.*

Myerson, R. B. (1997). *Game Theory: Analysis of Conflict.* Harvard university press.

Neumann, J. von and O. Morgenstern (1947). *The Theory of Games and Economic Behavior.* 2nd. Princeton University Press.

Shapley, L. S. (1953). “Stochastic games”. In: *national academy of sciences* 39.10, pp. 1095–1100.

Singh, S., M. James, and M. Rudary (2012). “Predictive state representations: A new theory for modeling dynamical systems”. In: *arXiv preprint arXiv:1207.4167.*

Singh, S. P., M. L. Littman, N. K. Jong, D. Pardoe, and P. Stone (2003). “Learning predictive state representations”. In: *20th International Conference on Machine Learning (ICML 2003)*, pp. 712–719.

Smallwood, R. D. and E. J. Sondik (1973). “The optimal control of partially observable Markov processes over a finite horizon”. In: *Operations research* 21.5, pp. 1071–1088.

Srinivasan, S., M. Lanctot, V. Zambaldi, J. Pérolat, K. Tuyls, R. Munos, and M. Bowling (2018). “Actor-Critic Policy Optimization in Partially Observable Multiagent Environments”. In: *Advances in Neural Information Processing Systems.*

Zinkevich, M., M. Johanson, M. Bowling, and C. Piccione (Dec. 2007b). “Regret Minimization in Games with Incomplete Information”. In: *Advances in Neural Information Processing Systems (NeurIPS 2007).* Vancouver, British Columbia, pp. 1729–1736.

Part II

Algorithm and Deviation Foundations

Chapter 5

Background

5.1 Introduction

The external, internal, and swap deviations are natural in normal-form games, but what deviation types and equilibrium concepts are natural in perfect-recall extensive-form games and POHPs? What learning algorithms already exist for perfect-recall POHPs? This chapter describes background relevant to these questions before answering them in the following chapter. To build up to new deviation types and a new algorithm for achieving hindsight rationality in perfect-recall POHPs, this chapter presents the necessary background on previously studied deviation types and algorithms.

5.2 The Deviation Player and Extensive-Form Correlated Equilibrium

Extensive-form correlated equilibrium (*EFCE*) is defined by Definition 2.2 of von Stengel et al. (2008) as a correlated equilibrium with respect to deviations that are constructed according to the play of a *deviation player*. At the beginning of the game, the mediator samples a pure strategy profile (strategy recommendations), $\{x_i\}_{i=1}^{N}$, and the game plays out according to this profile until it is player i's turn to act. Player i's decision at their active agent state s is determined by the deviation player who observes the action recommendation for player i, namely $x_i(s)$, which is the action recommended to player i by the mediator at s, and then chooses an action by either following this recommendation or deviating to a different action. After choosing an action and waiting for the other players to act according to their recommended strategies, the deviation player arrives at player i's next active agent state. Knowing the actions that were previously recommended to player i, the deviation player again chooses to follow the next recommendation or to deviate from it. This process continues until the game ends. If the deviation player cannot achieve a better value than that of player i, for each player i in the game, then the recommendation distribution is an EFCE.

Let $n_{\mathcal{A}} = \max_{h \in \mathcal{H}_{\mathcal{A}}} |\mathcal{A}(h)|$ denote the maximum number of possible agent actions and $d_{u(h)} = \lfloor |h|/2 \rfloor$ be the depth, *i.e.*, the number of agent actions, of the agent state at active history $h \in \mathcal{H}_{\mathcal{A}}$ under perfect recall.[1] The number of different states that the deviation player's memory could be in upon reaching agent state s at depth d_s is $(n_{\mathcal{A}})^{d_s}$, corresponding to the number of action combinations across s's predecessors. This exponential growth is computationally problematic.

One way to avoid this exponential growth is to assume that recommended strategies are in *reduced form*. A *reduced strategy* does not assign actions to agent states that could not be reached according to actions assigned to previous agent states. If a reduced recommended strategy x plays action a in agent state s and the deviation plays action a' there instead, every agent state $s' \succ s$ that the deviation player encounters thereafter would never have been encountered if the recommended strategy had been followed. Therefore, $x(s')$ is undefined for all $s' \succ s$ and the deviation player does not even have the opportunity to observe any more actions from the recommended strategy; the recommended strategy never set them to begin with. A deviation player can reach a given agent state s by deviating at any predecessor $\bar{s} \preceq s$ or not deviating at all, which means that the number of possible memory states associated with s grows linearly with depth.

This reduced strategy assumption effectively forces the deviation player to behave according to an *informed causal deviation* (Dudík et al. 2009; Gordon et al. 2008) defined by a *trigger action* and *trigger agent-state* pair, along with a strategy to play after triggering. Defining EFCE as a correlated equilibrium with respect to informed causal deviations allows them to be computed efficiently, which has led to this becoming the conventional definition of EFCE.

An *agent-form correlated equilibrium* (*AFCE*) is a different type of equilibrium introduced by Forges (1986) based on Selten (1974)'s *agent normal-form* of a game where we imagine that a different agent determines the action for each player at each agent state. Then there is a different deviation player for each agent state, not just each actual player, and they can only deviate from the recommendation at their own agent state.

von Stengel et al. (2008) shows that under the reduced strategy assumption, EFCE and AFCE are equivalent; a reduced strategy EFCE is a reduced strategy AFCE and *vice-versa*. In addition, a full strategy EFCE is a reduced strategy EFCE as well as an AFCE in both full and reduced strategies. One important contribution of this chapter is an analysis of the relative strength of reduced strategy EFCE and full strategy AFCE, two deviations which had not previously been compared.

[1] The depth is the length of the history divided by 2 to exclude the daimon actions from the count.

5.3 Counterfactual Regret Minimization

Counterfactual regret minimization performs external regret minimization locally at each active agent state where the reward function is *counterfactual value* (Zinkevich, Johanson, et al. 2007b). Given a strategy profile, (π, σ), and assuming perfect recall, the counterfactual value for taking action a in agent state s is the agent's expected return assuming they play to reach s and play a before playing π thereafter, *i.e.*,

$$v_s^{\text{CF}}(\phi_s^{\to a}(\pi); \sigma) = \sum_{h \in I(s)} \mathbb{P}_\sigma[h] \mathbb{E}[G_h(\phi_s^{\to a}(\pi); \sigma)], \tag{5.1}$$

where we overload the action transformation $\phi_s^{\to a}$ to a strategy transformation that changes the strategy only at s to play a. The learner's performance is then measured at each agent state in isolation according to *immediate counterfactual regret*, which is the extra counterfactual value achieved by choosing a given action instead of following π at s, *i.e.*, $\rho_s^{\text{CF}}(\phi_s^{\to a}, \pi; \sigma) = v_s^{\text{CF}}(\phi_s^{\to a}(\pi); \sigma) - v_s^{\text{CF}}(\pi; \sigma)$.

CFR is the application of a no-external-regret algorithm to ensure that cumulative immediate counterfactual regret, $\rho_s^{1:T,\text{IMM},\text{CF}}(\phi_s^{\to a}) \doteq \sum_{t=1}^{T} \rho_s^{\text{CF}}(\phi_s^{\to a}, \pi^t; \sigma^t)$, with respect to each action transformation $\phi_s^{\to a}$ at each active agent state s, grows sublinearly with T. The following result from Zinkevich, Johanson, et al. (ibid.) shows that minimizing immediate counterfactual regret everywhere actually minimizes external regret.

Theorem 7. *Cumualtive external regret cannot be larger than a sum of the positive part of cumulative immediate counterfactual regrets across active agent states. Let the cumulative regret with respect to external deviation $\phi^{\to x}$ to pure strategy x be $\rho^{1:T}(\phi^{\to x}) = \sum_{t=1}^{T} \rho(\phi^{\to x}, \pi^t; \sigma^t)$, then the maximum cumulative external regret is upper bounded as*

$$\max_{\phi^{\to x} \in \Phi_{\mathcal{X}}^{\text{EX}}} \rho^{1:T}(\phi^{\to x}) \leq \sum_{s \in \mathcal{S}_{\mathcal{A}}} \left[\max_{\phi_s^{\to a} \in \Phi_{\mathcal{A}(s)}^{\text{EX}}} \rho_s^{1:T,\text{IMM},\text{CF}}(\phi_s^{\to a}) \right]_+ .$$

Consequently, a no-regret algorithm such as ramp regret matching can be deployed at each active agent state so that together they minimize external regret.

Theorem 8. *If the agent selects actions according to ramp regret matching trained to choose from the external action transformations according to the counterfactual value function, then the agent's maximum cumulative immediate counterfactual regret is upper bounded as*

$$\max_{\phi_s^{\to a} \in \Phi_{\mathcal{A}(s)}^{\text{EX}}} \rho_s^{1:T,\text{IMM},\text{CF}}(\phi_s^{\to a}) \leq U d_* \sqrt{n_{\mathcal{A}} T},$$

where $d_* = \max_{s \in \mathcal{S}_{\mathcal{A}}} d_s$ is the maximal depth of any agent state. Therefore, by Theorem 7,

$$\max_{\phi^{\to x} \in \Phi_{\mathcal{X}}^{\text{EX}}} \rho^{1:T}(\phi^{\to x}) \leq U d_* |\mathcal{S}_{\mathcal{A}}| \sqrt{n_{\mathcal{A}} T}.$$

Theorem 7 (and by extension Theorem 8) follows directly from an elementary decomposition relationship consisting of the following three lemmas.

Lemma 3. *In a finite-horizon POHP, the counterfactual value of active agent state s under perfect recall recursively decomposes as*

$$v_s^{\text{CF}}(\pi;\sigma) = r_s(\pi;\sigma) + \mathbb{E}\big[v_{u_{\mathcal{A}}(s,A)}^{\text{CF}}(\pi;\sigma)\big]$$

where $r_s(\pi;\sigma) = \sum_{h\in I(s)} \mathbb{P}_\sigma[h]\mathbb{E}[r(\omega(hAB))]$, and expectations are taken over actions $A \sim \pi(s)$ and $B \sim \sigma(hA)$.

Proof.

$$\begin{aligned}
v_s^{\text{CF}}(\pi;\sigma) &= \sum_{h\in I(s)} \mathbb{P}_\sigma[h]\mathbb{E}[r(\omega(hAB)) + \Gamma G_{hAB}(\pi;\sigma)] \\
&= r_s(\pi;\sigma) + \mathbb{E}[\sum_{h\in I(s), b\in\mathcal{A}(hA)} \mathbb{P}_\sigma[hAb]G_{hAb}(\pi;\sigma)] \\
&= r_s(\pi;\sigma) + \mathbb{E}\big[v_{u_{\mathcal{A}}(s,A)}^{\text{CF}}(\pi;\sigma)\big],
\end{aligned}$$

where $\Gamma \sim \gamma(h)$. □

Lemma 4. *Define the* full counterfactual regret *with respect to pure strategy x as*

$$\rho_s(\phi^{\to x},\pi;\sigma) = v_s^{\text{CF}}(\phi^{\to x}(\pi);\sigma) - v_s^{\text{CF}}(\pi;\sigma).$$

If x uses action a in s, then its full counterfactual regret decomposes as

$$\rho_s^{\text{CF}}(\phi^{\to x},\pi;\sigma) = \rho_s^{\text{CF}}(\phi_s^{\to a},\pi;\sigma) + \sum_{s'\in\mathcal{S}_{\mathcal{A}}(s,a)} \rho_{s'}^{\text{CF}}(\phi^{\to x},\pi;\sigma).$$

Proof.

$$\rho_s^{\text{CF}}(\phi^{\to x},\pi;\sigma) \tag{5.2}$$

$$= v_s^{\text{CF}}(\phi^{\to x}(\pi);\sigma) - v_s^{\text{CF}}(\pi;\sigma). \tag{5.3}$$

Decompose into immediate and future value:

$$= r_s(\phi_s^{\to a}(\pi);\sigma) - v_s^{\text{CF}}(\pi;\sigma) + \sum_{s'\in\mathcal{S}_{\mathcal{A}}(s,a)} v_{s'}^{\text{CF}}(x;\sigma). \tag{5.4}$$

Add and subtract $\sum_{s'\in\mathcal{S}_{\mathcal{A}}(s,a)} v_{s'}^{\text{CF}}(\pi;\sigma)$:

$$\begin{aligned}
= \quad & \underbrace{r_s(\phi_s^{\to a}(\pi);\sigma) + \sum_{s'\in\mathcal{S}_{\mathcal{A}}(s,a)} v_{s'}^{\text{CF}}(\pi;\sigma)}_{\text{Value from } a \text{ assuming } \pi \text{ is played thereafter, } v_s^{\text{CF}}(\phi_s^{\to a}(\pi);\sigma).} \qquad - v_s^{\text{CF}}(\pi;\sigma) \\
+ \quad & \underbrace{\sum_{s'\in\mathcal{S}_{\mathcal{A}}(s,a)} v_{s'}^{\text{CF}}(x;\sigma) - v_{s'}^{\text{CF}}(\pi;\sigma).}_{\text{Suboptimality after } a,\ \sum_{s'\in\mathcal{S}_{\mathcal{A}}(s,a)} \rho_{s'}^{\text{CF}}(\phi^{\to x},\pi;\sigma).}
\end{aligned} \tag{5.5}$$

$$= \rho_s^{\text{CF}}(\phi_s^{\to a},\pi;\sigma) + \sum_{s'\in\mathcal{S}_{\mathcal{A}}(s,a)} \rho_{s'}^{\text{CF}}(\phi^{\to x},\pi;\sigma). \tag{5.6}$$

□

Lemma 5. *(Zinkevich, Johanson, et al. 2007a, Equation 13, Lemma 5) Denote the cumulative full counterfactual regret at s of deviation ϕ as $\rho_s^{1:T,\text{FULL},\text{CF}}(\phi) = \sum_{t=1}^{T} \rho_s^{\text{CF}}(\phi, \pi^t; \sigma^t)$. The maximum cumulative full counterfactual regret at s over the external deviations is upper bounded as*

$$\begin{aligned}
&\max_{\phi^{\to x} \in \Phi^{\text{EX}}_{\mathcal{X}}} \rho_s^{1:T,\text{FULL},\text{CF}}(\phi^{\to x}) \\
&\leq \max_{\phi_s^{\to a} \in \Phi^{\text{EX}}_{\mathcal{A}(s)}} \rho_s^{1:T,\text{IMM},\text{CF}}(\phi_s^{\to a}) + \max_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}_{\mathcal{A}}(s,a)} \max_{\phi^{\to x} \in \Phi^{\text{EX}}_{\mathcal{X}}} \rho_{s'}^{1:T,\text{FULL},\text{CF}}(\phi^{\to x}).
\end{aligned}$$

Proof. Sum across rounds on both sides of Lemma 4. Take the maximum over $\Phi^{\text{EX}}_{\mathcal{X}}$ on both sides and split the maximum on the right-hand-side into three separate maximizations over $\Phi^{\text{EX}}_{\mathcal{A}(s)}$, $\mathcal{A}(s)$, and $\Phi^{\text{EX}}_{\mathcal{X}}$. □

5.4 Internal CFR

Celli et al. (2020) adapts CFR so that its empirical distribution of play in self-play converges to a (reduced strategy) EFCE almost surely. They accomplish this by swapping out immediate counterfactual regret for *laminar subtree regret*, a form of immediate regret similar to cumulative immediate counterfactual regret except that instantaneous counterfactual regret terms are conditionally dropped from the sum across rounds. A "trigger condition" is defined by a predecessor active agent state $s^!$ and an action $a^!$, and is satisfied only if the agent plays to $s^!$ and plays $a^!$ once there. Terms from the cumulative immediate counterfactual regret sum are dropped unless the trigger condition is met.

Assume that on each round t of a repeated finite-horizon POHP, a perfect recall agent chooses a mixed strategy π^t and samples a pure strategy $X^t \sim \pi^t$. The laminar subtree regret for not choosing action a at agent state s triggering on $(s^!, a^!)$ accumulated after T rounds is[2]

$$\sum_{t=1}^{T} \mathbb{1}\{s^! \preceq s,\, X^t(s^!) = a^!\} \rho_s^{\text{CF}}(\phi_s^{\to a}, X^t; \sigma^t).$$

Laminar subtree regret can be controlled throughout the POHP by a set of no-regret learners at each agent state. Conceptually, there is one for each trigger, $(s^!, a^!)$. A learner only activates if the trigger condition is met, otherwise they are "asleep" and do not produce

[2] Celli et al. (2020) works from the EFG formalism and purifies the strategies of all non-chance players in their definition of laminar subtree regret but take the expectation over chance events. Since a POHP presents the view of a game from a single player's perspective and abstracts away the other players and chance into the daimon, our definition of laminar subtree regret allows but does not require that the daimon use a partially purified strategy that conforms with Celli et al. (ibid.)'s definition.

an immediate strategy or update their internal state. Given a sampling of actions leading up to a given agent state s, only one learner is awake, so the next action can be sampled from this learner's immediate strategy. All but one of these learners at s are no-external-regret learners. The single no-internal-regret learner is shared across all triggers where $s^! = s$. Celli et al. (2020) call this algorithm *internal CFR* (*ICFR*) and they prove that

Theorem 9. *When all players play according to ICFR, their empirical distribution of play converges almost surely to a (reduced strategy) EFCE.*

Since actions are sampled in each agent state, ICFR can be recognized as an extension to *pure CFR* (Gibson 2014). Pure CFR performs CFR except that counterfactual regrets are computed with respect to a purified agent strategy where an action was sampled at each agent state according to the local no-external-regret learner's immediate strategy.

5.5 Policy Gradient Policy Iteration

If we apply gradient ascent to the expected return from a POHP's root agent states, we arrive at a policy gradient algorithm for POHPs. The policy gradient strategy is a behavioral strategy π determined by parameters learned with gradient ascent on the agent's expected return function. In a finite-horizon POHP with timed updates, the capability of a single action probability at a single agent state s to change the root expected return depends on how often π plays to s and the strategy's behavior after s. Adding up all the contributions to the change in the root expected return across each agent state and action yields the partial derivative for a given gradient ascent parameter.

Restating a result from Srinivasan et al. (2018)'s Appendix E, if PGPI is *tabular*, *i.e.*, there is a single parameter, $\theta_{s,a}$, for each action a in each active agent state s, then each active agent state has its own local gradient. The partial derivative of the expected return with respect to $\theta_{s,a}$ is then the realization-weighted expected return assuming that action a is chosen in s multiplied by the partial derivative of the strategy with respect to $\theta_{s,a}$, *i.e.*,

$$\begin{aligned}
\frac{\partial \mathbb{E}_{H \sim \xi}[G_H(\pi;\sigma)]}{\partial \theta_{s,a}} &= \frac{\partial \mathbb{P}_{\pi,\sigma}[s] \mathbb{E}_{H \sim \xi_s^{\pi,\sigma}}[G_H(\pi;\sigma)]}{\partial \theta_{s,a}} && (5.7) \\
&= \mathbb{P}_{\pi,\sigma}[s] \mathbb{E}_{H \sim \xi_s^{\pi,\sigma}}\left[\frac{\partial G_H(\pi;\sigma)}{\partial \pi(a \,|\, s)}\right] \frac{\partial \pi(a \,|\, s)}{\partial \theta_{s,a}} && (5.8) \\
&= \mathbb{P}_{\pi,\sigma}[s] \mathbb{E}_{H \sim \xi_s^{\pi,\sigma}}[G_{Ha}(\pi;\sigma)] \frac{\partial \pi(a \,|\, s)}{\partial \theta_{s,a}}. && (5.9)
\end{aligned}$$

Eq. (5.8) shows that we can implement tabular PGPI with a set of policy gradient instances, each localized to a particular agent state. The policy gradient instance for active agent state s trains on the realization-weighted expected return (as a function of the action taken in s) from s as its utility function and it produces the immediate strategy at s. As noted by Srinivasan et al. (2018), this procedure resembles CFR except for two differences: the local utility functions in PGPI are

$$a \mapsto \mathbb{P}_{\pi,\sigma}[s]\mathbb{E}_{H \sim \xi_s^{\pi,\sigma}}[G_{Ha}(\pi;\sigma)] = \sum_{h \in I(s)} \mathbb{P}_{\pi,\sigma}[h] G_{ha}(\pi;\sigma) \tag{5.10}$$

rather than

$$a \mapsto \sum_{h \in I(s)} \mathbb{P}_{\sigma}[h]\mathbb{E}[G_{ha}(\pi;\sigma)] \tag{5.11}$$

in CFR, and PGPI uses policy gradient as its local learner rather than a no-regret learner.[3] The only difference between Eq. (5.10) and Eq. (5.11) is that the former (PGPI) uses the joint reach probability $\mathbb{P}_{\pi,\sigma}[h]$ that includes the agent's reach probability contribution, while the latter (CFR), uses $\mathbb{P}_{\sigma}[h]$, which only includes the daimon's reach probability contribution.

References

Celli, A., A. Marchesi, G. Farina, and N. Gatti (2020). "No-regret learning dynamics for extensive-form correlated equilibrium". In: *Advances in Neural Information Processing Systems* 33.

Dudík, M. and G. J. Gordon (2009). "A Sampling-Based Approach to Computing Equilibria in Succinct Extensive-Form Games". In: *25th Conference on Uncertainty in Artificial Intelligence (UAI-2009).*

Forges, F. (1986). "Correlated equilibria in repeated games with lack of information on one side: a model with verifiable types". In: *International Journal of Game Theory* 15.2, pp. 65–82.

Gibson, R. (2014). "Regret Minimization in Games and the Development of Champion Multiplayer Computer Poker-Playing Agents". PhD thesis. University of Alberta.

Gordon, G. J., A. Greenwald, and C. Marks (2008). "No-Regret Learning in Convex Games". In: *25th international conference on Machine learning*, pp. 360–367.

Selten, R. (1974). "Reexamination of the Perfectness Concept for Equilibrium Points in Extensive Games". In: *Economics.*

Srinivasan, S., M. Lanctot, V. Zambaldi, J. Pérolat, K. Tuyls, R. Munos, and M. Bowling (2018). "Actor-Critic Policy Optimization in Partially Observable Multiagent Environments". In: *Advances in Neural Information Processing Systems.*

von Stengel, B. and F. Forges (2008). "Extensive-form correlated equilibrium: Definition and computational complexity". In: *Mathematics of Operations Research* 33.4, pp. 1002–1022.

Zinkevich, M., M. Johanson, M. Bowling, and C. Piccione (Sept. 2007a). *Regret Minimization in Games with Incomplete Information.* Tech. rep. TR07-14. University of Alberta.

[3] As discussed in Section 2.5, policy gradient may not be no-regret depending on the method used to convert its gradient ascent parameters into its strategy.

Zinkevich, M., M. Johanson, M. Bowling, and C. Piccione (Dec. 2007b). "Regret Minimization in Games with Incomplete Information". In: *Advances in Neural Information Processing Systems (NeurIPS 2007)*. Vancouver, British Columbia, pp. 1729–1736.

Chapter 6

Behavioral Deviations

6.1 Introduction

In this chapter, the behavioral deviations, a broad and natural class of deviation functions for finite-horizon POHPs with perfect recall, are developed. They are inspired by von Stengel et al. (2008)'s deviation player that forms the basis of the EFCE concept. von Stengel et al. (ibid.), as well as subsequent works (Celli et al. 2020; Dudík et al. 2009; Farina, Bianchi, et al. 2020; Farina, Ling, et al. 2019; Gordon et al. 2008; W. Huang 2011), add assumptions to restrict deviations that a deviation player could express in order to construct efficient algorithms. The behavioral deviations provide a unified mechanism for selecting subsets to replicate all previously studied deviation player restrictions and for creating new restrictions. The new partial sequence deviation restriction introduced in this chapter yields more powerful deviation types, when combined with observable sequential rationality (OSR), than those previously studied without substantially increasing complexity.

This chapter also explores some of the tradeoffs between different behavioral deviation subsets. Different subsets have different strategic strengths and weaknesses, as well as different computational requirements. Elementary strategic differences between deviation types are illustrated with game examples.

6.2 Definition

It does not appear to be feasible, in general, to compete with von Stengel et al. (2008)'s deviation player at full strength. Thus, it is conventional to weaken the deviation player by forcing strategy recommendations to be in reduced form, thereby limiting the amount of information the deviation player can use to construct beneficial deviations. Instead of restricting the information in the strategy recommendations, what if we intentionally hide information from the deviation player?

Assuming full pure strategy recommendations (*e.g.*, in normal or behavioral form), we now provide the deviation player with a subset of three options at each agent state, s: (i) commit to following the (pure) action recommendation at s, $x(s)$, before seeing $x(s)$; (ii) choose a new action without ever seeing $x(s)$; or (iii) observe $x(s)$ and then choose a possibly different action. Option (iii) is traditionally the only option provided to the deviation player and it does in fact dominate the other two. However, in our construction, Option (iii) need not be available to the deviation player in every, or possibly any, agent state.

The deviation player's behavior at agent state s can be modeled as an action transformation $\phi_s : \mathcal{A}(s) \to \mathcal{A}(s)$. We can then implement controls on the deviation player's options by allowing or prohibiting particular action transformations. The deviation player is allowed Option (i) if they are allowed to choose the *identity transformation*, $\phi^1 : a \mapsto a$. They are allowed Option (ii) at s if they are allowed to choose an external transformations from $\Phi^{\text{EX}}_{\mathcal{A}(s)}$. Finally, they are allowed Option (iii) if they are allowed to choose an internal transformation from $\Phi^{\text{IN}}_{\mathcal{A}(s)}$.[1]

The action transformation that the deviation player chooses does not merely represent their behavior, it also reflects the information that the deviation player observes during play. Each internal transformation (including the identity transformation) must "observe" the recommended action in order to return the appropriate transformed action, therefore, if the deviation player uses one of these transformations, they also observe the recommended action. Since each external transformation is a constant function, executing such a transformation does not require observing the recommended action, and therefore, if the deviation player uses an external transformation, they do not observe the recommended action.

Let $\mathcal{A}_* = \bigcup_{s \in \mathcal{S}_\mathcal{A}} \mathcal{A}(s)$ denote the union of the agent's action sets. We can describe the deviation player's *memory state*, $\lambda \in \mathcal{G} \subseteq (\{*\} \cup \mathcal{A}_*)^{d_*}$, as a string that begins empty and gains a character after each of the agent's actions. The recommendation, $x(s)$, at agent state s is hidden from or revealed to the deviation player depending on which action transformation the deviation player employs at s. The choice of this action transformation can naturally depend on the deviation player's memory state at s. The deviation player observes the recommendation with an internal transformation (including the identity transformation), so $x(s)$ would be appended to λ in that case. The deviation player does not observe the recommendation with an external transformation, so "$*$" would be appended to λ in that case.

Limiting the action transformations available to the deviation player thus restricts the set of memory states that they can realize. Given a memory state λ, there is only one realizable

[1] While an internal transformation can only swap one action particular with another, there is no loss in generality because every multi-action swap can be a represented as the combination of single swaps (Dudík et al. 2009; A. Greenwald, Jafari, et al. 2003). Thus, any strategy sequence that can be improved upon by a swap transformation can also be improved upon by at least one internal transformation.

child memory state at the next agent state if the deviation player is allowed either the identity transformation ($\lambda x(s)$) or any set of external transformations ($\lambda *$). If both of these options are allowed, then naturally there are two possible child memory states, $\lambda x(s)$ and $\lambda *$. The internal transformation from action a to a' leads to memory state $\lambda a'$ at each agent state following both action a and a', as well $\lambda \bar{a}$ at each agent state following each other action $\bar{a}$. If the deviation player is allowed the union of the external and internal transformations, then there are $|\mathcal{A}(s)| + 1$ child memory states at each child agent state following each action a, one for each action that could be transformed into a plus "$*$".

A complete strategy for the deviation player can then be represented as a complete assignment of action transformations to each active agent state and realizable memory state. We call such an assignment a *behavioral deviation* in analogy with behavioral strategies, and denote the full set of behavioral deviations as $\Phi^{\text{SW}}_{\mathcal{S}_\mathcal{A}}$ since they are a natural analog of the swap transformations for POHPs.[2] Any strategy that von Stengel et al. (2008)'s deviation player could employ can be implemented by a behavioral deviation. The behavioral deviations therefore allow us to disambiguate between full and reduced strategy EFCE by referring to full strategy EFCE as *behavioral correlated equilibrium* (*BCE*) and referring to reduced strategy EFCE simply as EFCE. The latter terminology is already the norm in AI literature (Celli et al. 2020; Dudík et al. 2009; Farina, Bianchi, et al. 2020; Farina, Ling, et al. 2019; Gordon et al. 2008).

As we see next, previously studied deviation types are actually captured by behavioral deviation subsets.

6.3 Reductions to Previously Studied Deviation Types

As discussed in Section 5.2, von Stengel et al. (2008) introduce two restrictions to their deviation player. Here we show how both of the resulting deviation sets can be defined as behavioral deviation subsets.

Causal deviations. An informed causal deviation is traditionally defined by a trigger agent state $s^!$, a trigger action $a^!$, and a strategy π'. The following behavioral deviation reproduces any such deviation: assign (i) the internal transformation $\phi^{a^! \to a^\odot}$ to the sole memory state at $s^!$, (ii) external transformations to all successors $s' \succ s^!$ where $a^!$ is in the deviation player's memory to reproduce π', and (iii) identity transformations to every memory

[2] In Morrill, D'Orazio, Lanctot, Wright, Bowling, and A. R. Greenwald (2021), behavioral deviations were previously labeled as $\Phi^{\text{IN}}_{\mathcal{S}_\mathcal{A}}$ to make a connection with the internal transformations. However, since we want to be able to assign external transformations to agent states, not just internal transformations, it is better to denote them as $\Phi^{\text{SW}}_{\mathcal{S}_\mathcal{A}}$. Regardless, it is still sufficient to achieve hindsight rationality with respect to the full set of behavioral deviations by achieving hindsight rationality with respect to the set of behavioral deviations that only use internal transformations. This latter set ought to be denoted $\Phi^{\text{IN}}_{\mathcal{S}_\mathcal{A}}$.

state in every other agent state. The analogous *blind causal deviation* (Farina, Bianchi, et al. 2020) always triggers in $s^!$, which is reproduced with the same behavioral deviation except that the external transformation $\phi^{\to \pi'(s^!)}$ is assigned to $s^!$.

Action deviations. An agent-form equilibrium deviation can only change a single action at a single agent state, so we naturally call it an *action deviation*. Precisely, an action deviation modifies the immediate strategy at $s^!$, $\pi_i(s^!)$, only, either conditioning on $\pi_i(s^!)$ (an informed action deviation) or not (a blind action deviation). Any informed or blind action deviation is reproduced by assigning either an internal or external transformation to the sole memory state at $s^!$, respectively, and identity transformations elsewhere.

6.4 Counterfactual Deviations

A causal deviation modifies strategies from a particular agent state to the end of the game. What if we instead modify strategies at agent states leading up to a particular agent state? That is, we define a target agent state, $s^\odot$, and assign external transformations at the sole memory state of each predecessor $s \prec s^\odot$ that play to reach $s^\odot$. At the sole "**...*" memory state at $s^\odot$ we can assign an arbitrary action transformation and we assign identity transformations everywhere else. I call this a *counterfactual deviation* due to its connection with CFR, which will be described in Section 6.6.2, and because its distinguishing feature is that it plays to reach a given agent state, even if that agent state would never be reached by the strategy under transformation. Just like causal and action deviations, we can distinguish between a blind counterfactual deviation that assigns an external transformation to $s^\odot$ and an informed counterfactual deviation that assigns an internal transformation there instead. Section 6.6 makes a connection between these new counterfactual deviations and the CFR algorithm.

Definition 5. *Let $a_s^{\to s'} \in \mathcal{A}(s)$ be the unique action that must be played in active agent state s to reach agent state $s' \succ s$ under perfect recall. A blind counterfactual deviation, ϕ, is determined by a pair, $(s^\odot, a^\odot)$, where $s^\odot \in \mathcal{S}_\mathcal{A}$ is a target agent state that the deviation plays to reach deterministically from the start of the POHP and $a^\odot$ is the action taken at $s^\odot$. The deviation leaves the input strategy, $x \in \mathcal{X}$, unmodified at every other agent state. Formally,*

$$\phi = \left\{ \phi_s \in \Phi^{\text{SW}}_{\mathcal{A}(s)} \mid \phi_s = \begin{cases} \phi_s^{\to a_s^{\to s^\odot}} & \text{if } s \prec s^\odot \\ \phi_s^{\to a^\odot} & \text{if } s = s^\odot \\ \phi^{\mathbf{1}} & \text{o.w.} \end{cases} \right\}_{s \in \mathcal{S}_\mathcal{A}}.$$

Definition 6. *An informed counterfactual deviation, ϕ, is defined by a triple, $(s^\odot, a^!, a^\odot)$, where $s^\odot \in \mathcal{S}_\mathcal{A}$ is a target agent state that the deviation plays to reach deterministically from* *the start of the POHP. If $a^!$ is the input strategy's action at $s^\odot$, then $a^\odot$ is played at $s^\odot$, otherwise the input strategy is followed at $s^\odot$. The deviation leaves the input strategy, $x \in \mathcal{X}$, unmodified at every other agent state. Formally,*

$$\phi = \left\{ \phi_s \in \Phi^{\text{SW}}_{\mathcal{A}(s)} \mid \phi_s = \begin{cases} \phi_s^{\to a_s^{\to s^\odot}} & \textit{if } s \prec s^\odot \\ \phi_s^{a^! \to a^\odot} & \textit{if } s = s^\odot \\ \phi^{\mathbf{1}} & o.w. \end{cases} \right\}_{s \in \mathcal{S}_{\mathcal{A}}} .$$

Every counterfactual deviation where the target agent state is at the start of the POHP is an action deviation, just as every causal deviation that triggers at a terminal agent state is an action deviation. Unlike the set of action deviations, however, the set of counterfactual deviations is more like the external deviations since terminal agent states can be chosen as targets. In fact, the set of counterfactual deviations is the set of single-target deviations (Definition 4) generated from the external deviations.

Naturally, an equilibrium with respect to counterfactual deviations is a *counterfactual correlated equilibrium* (*CFCE*) or a *counterfactual coarse-correlated equilibrium* (*CFCCE*) depending on whether the equilibrium is with respect to informed or blind counterfactual deviations, respectively. Formally:

Definition 7. *A recommendation distribution is a counterfactual coarse-correlated equilibria (CFCCE) if there are no beneficial blind counterfactual deviations.*

Definition 8. *A recommendation distribution is a counterfactual correlated equilibria (CFCE) if there are no beneficial informed counterfactual deviations.*

In a depth-two POHP with a single root agent state (where depth here is measured in sequential agent actions), the set of behavioral deviations is actually the union of the counterfactual and causal deviations. The set of action deviations is the set of behavioral deviations minus the external deviations and the counterfactual deviations that target a terminal agent state. These facts are important for understanding the examples that follow.

6.5 Practical Relationships Between Elementary Deviation Types

I now illustrate some practical differences between the six elementary deviation types discussed so far: swap, external, behavioral, causal, action, and counterfactual. These differences are important because algorithms tied to more limited deviation types may not achieve as much reward as ones using stronger deviation types, and the types of mistakes an algorithm makes can depend on the structure of its associated deviation type. Moreover, even if an algorithm is not designed with a deviation type in mind, it may implicitly use one. For example, "on-policy" RL algorithms like PGPI and Monte Carlo approximations thereof (*i.e.*, standard formulations of policy gradient) are implicitly tied to action deviations, as we later discuss. Thus, these results have substantial generality and widespread impact.

I use a two-player game as the basis for a series of examples. In this game, player one acts twice and player two acts once, but player two's recommendations are the same for each example. The only difference between each example is the player one recommendations. Each recommendation distribution is uniform random over a pair of strategy profiles.

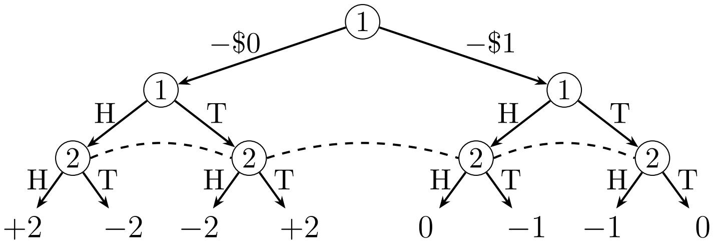


Figure 6.1: An extended matching pennies game with payoffs defined for player one. Dashed lines indicate that each of player two's histories are in the same information set.

The game is matching pennies with an additional gambling action for player one before the matching pennies game starts. Player one privately chooses whether or not to pay $1. If player one pays (chooses the −$1 action), then losing matching pennies has no additional cost and winning refunds their $1. If player one does not pay (chooses the −$0 action), then they play matching pennies for $2. The game is visualized in Fig. 6.1. The recommendation distribution of each example is uniform random over two pure strategy profiles: player one is assigned strategies specific to that example and player two is assigned heads (H) in the first strategy profile and tails (T) in the second.

Example #1: BCE that is not a CE. If the recommendations for player one are $\{-\$0; \mathrm{T} \,|\, -\$0; \mathrm{H} \,|\, -\$1\}$ and $\{-\$0; \mathrm{T} \,|\, -\$0; \mathrm{T} \,|\, -\$1\}$, then player one achieves an expected value of zero. The only way to improve on this value is to switch T to H given −$0 only in the first recommendation. However, the only difference between the two recommendations for player one is the action given −$1 and behavioral deviations cannot correlate the behavior after −$0 with the action recommendation after −$1. Thus, there is no behavioral deviation that achieves more than zero and the recommendation distribution is a BCE. There is, however, a swap deviation that does correlate the behavior after −$0 with the action recommendation after −$1 and achieves +2. Therefore, the recommendation distribution is not a CE. These recommendations and notable deviations are visualized in Fig. 6.2. Nearly all deviations are visualized in Fig. 6.A.11 in this chapter's appendix for completeness, only some uninteresting swap deviations are omitted for brevity. To the best of my knowledge, this is

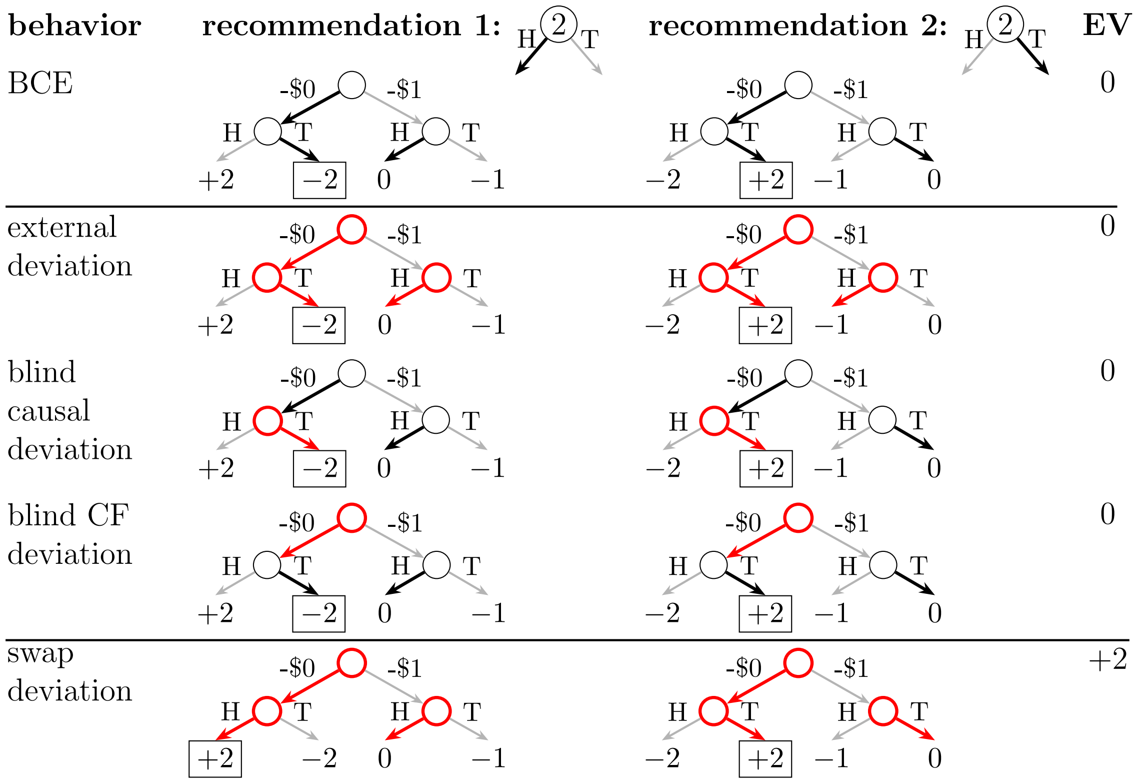


Figure 6.2: A gambling matching pennies example that is a BCE but not a CE. Recommendations are shown in the first row, the next three rows are three of the best behavioral deviations. The first is an external deviation, the second is a blind causal deviation, and the third is a blind counterfactual deviation. These blind causal and blind counterfactual deviations are both blind action deviations. The bottom row shows a beneficial swap deviation that is not a behavioral deviation.

the first example to show directly how the swap deviations are stronger than the behavioral deviations.

Example #2: AFCE that is not a CCE. If the recommendations for player one are $\{-\$1; \mathrm{T} \,|\, -\$0; \mathrm{H} \,|\, -\$1\}$ and $\{-\$1; \mathrm{H} \,|\, -\$0, \mathrm{H} \,|\, -\$1\}$, then player one achieves an expected value of -0.5. Achieving a greater value requires transforming two actions in separate agent states (the root agent state and the one following $-\$0$). Since an action deviation can only transform the action for a single agent state, there is no beneficial action deviation and this recommendation distribution is an AFCE. There is, however, an external deviation that always plays $\{-\$0; \mathrm{H} \,|\, -\$0; \mathrm{H} \,|\, -\$1\}$ and achieves a better value of 0. This recommendation distribution is therefore not a CCE. These recommendations and notable deviations are visualized in Fig. 6.3 and more deviations are visualized in Fig. 6.B.12. von Stengel et al. (2008) provided a similar example in their In-or-Out game so the example here is merely provided for completeness.

Example #3: CFCE that is not an EFCCE or an AFCCE. If the recommenda-

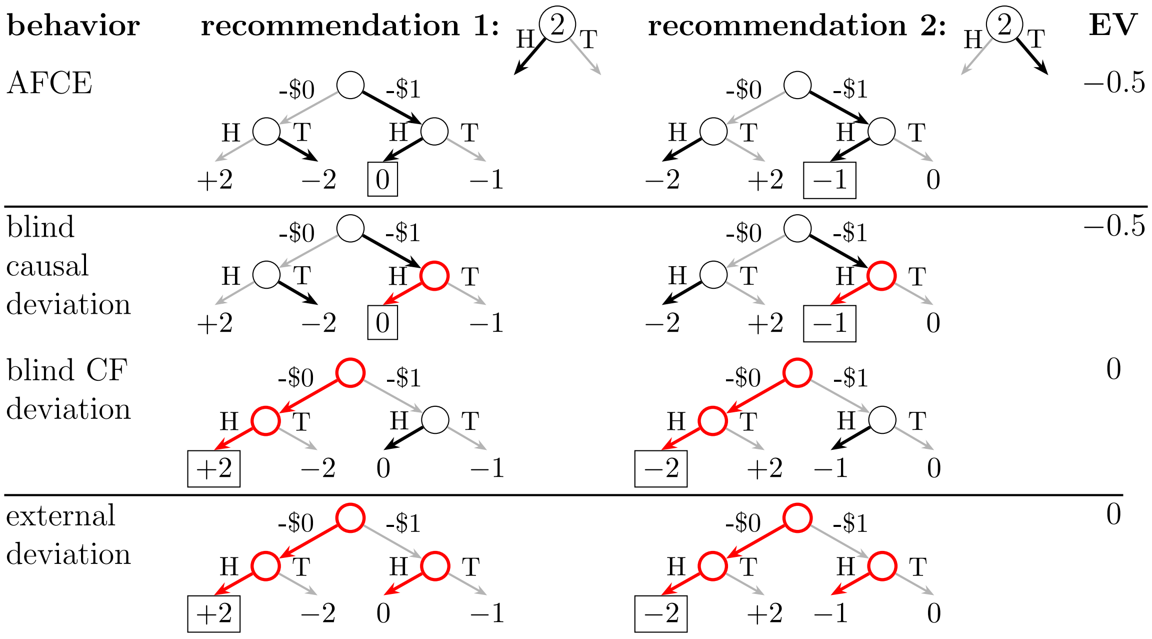


Figure 6.3: A gambling matching pennies example that is an AFCE but not a CCE.

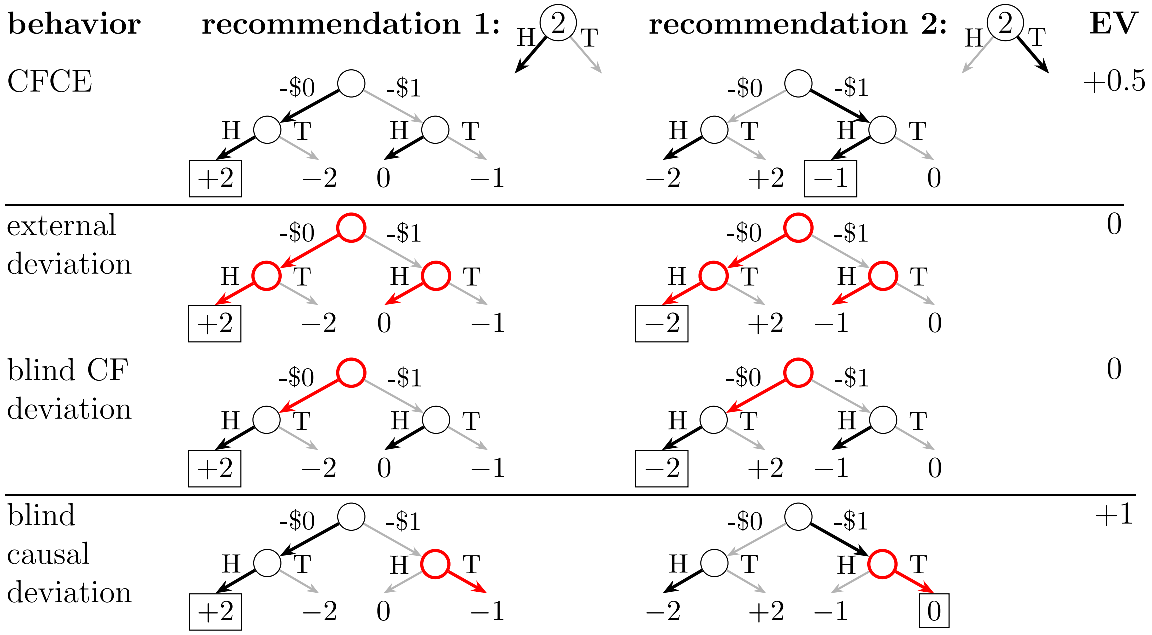


Figure 6.4: A gambling matching pennies example that is a CFCE but not an EFCCE or an AFCCE.

tions for player one are $\{-\$0; \mathrm{H} \,|\, -\$0; \mathrm{H} \,|\, -\$1\}$ and $\{-\$1; \mathrm{H} \,|\, -\$0; \mathrm{H} \,|\, -\$1\}$, then player one achieves an expected value of +0.5. Achieving a greater value requires choosing T after −\$1 without deviating from the recommendation at the root. A counterfactual deviation can only deviate from the recommendation after −\$1 if it deviates to a fixed action at the root, making the recommendation distribution a CFCE. A blind causal or blind action deviation, however, can follow the recommendations at the root and trigger after −\$1 to always choose T there and achieve a value of +1. This recommendation distribution is therefore not an EFCCE or an AFCCE. These recommendations and notable deviations are visualized in Fig. 6.4 and more deviations are visualized in Fig. 6.C.13. To the best of my knowledge, this example is the first to show directly how a blind causal or blind action deviation can outperform all external deviations (except for a different but analogous example I presented in work leading up to this thesis).

Example #4: EFCE that is not an AFCCE or a CFCCE. If the recommenda-

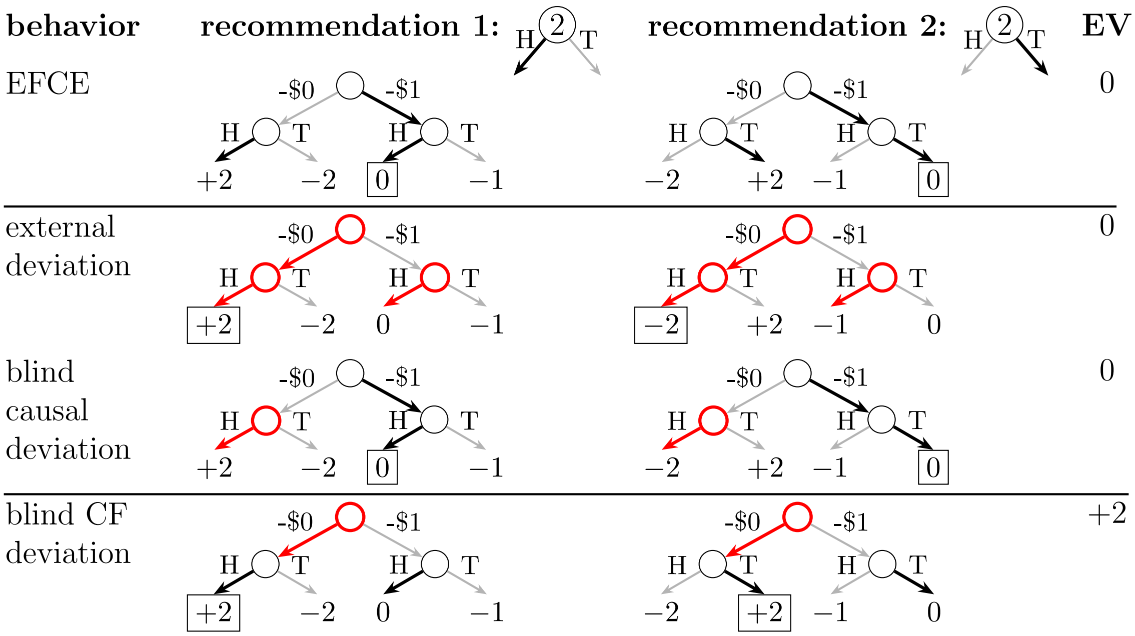


Figure 6.5: A gambling matching pennies example that is an EFCE but not an AFCCE or a CFCCE.

tions for player one are $\{-\$1; \mathrm{H} \,|\, -\$0; \mathrm{H} \,|\, -\$1\}$ and $\{-\$1; \mathrm{T} \,|\, -\$0; \mathrm{T} \,|\, -\$1\}$, then player one achieves an expected value of zero. Achieving a greater value requires always choosing −\$0 and following the recommendation after −\$0. A causal deviation must either follow the recommendation at the root agent state and trigger afterward or trigger in the root agent state, making the recommendation distribution an EFCE. A blind action deviation, however, can choose −\$0 and follow the subsequent action recommendation to achieve a value of +2. This recommendation distribution is therefore not an AFCCE. These recommendations and notable deviations are visualized in Fig. 6.5 and more deviations are visualized in Fig. 6.D.14.

This example appears to contradict von Stengel et al. (2008)'s statement that "in general extensive-form games, any EFCE is an AFCE, by giving arbitrary recommendations at unreachable agent states that in an EFCE are left unspecified." The confusion stems from the fact that here von Stengel et al. (ibid.) uses a definition of EFCE based on behavioral

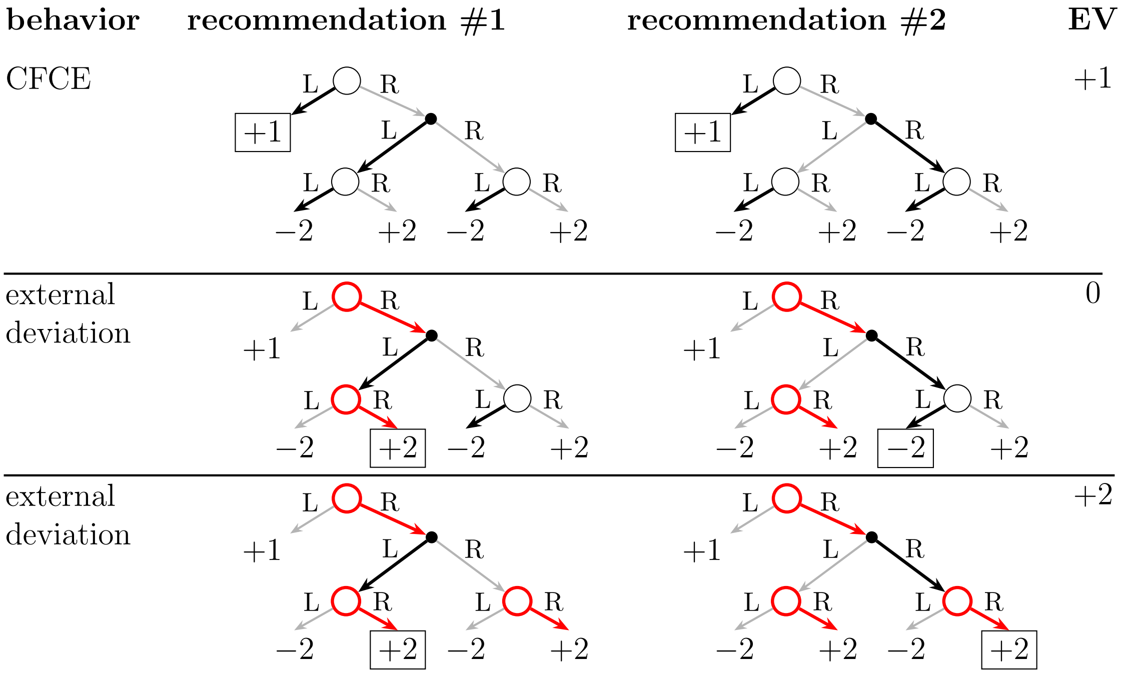


Figure 6.6: A beneficial external deviation for player one in a CFCE in MacQueen (2022)'s counterexample. Black lines denote recommendations, red lines denote deviations from unobserved recommendations, and grey lines mark actions that are not recommended or used by the deviation.

deviations rather than the informed causal deviations, and as mentioned before, the informed causal deviation definition of EFCE has become standard in the AI literature (see, for example, Celli et al. (2020), Dudík et al. (2009), Farina, Bianchi, et al. (2020), Farina, Ling, et al. (2019), and Gordon et al. (2008)). Now that the BCE concept has been defined, we could state this more clearly and concisely as: in general extensive-form games, any BCE is an AFCE. Conveniently, this statement follows immediately from the definition of the set of informed action deviations as a restricted set of behavioral deviations.

In addition to resolving confusion around the definition of EFCE, this example is an important contribution because it shows that blind action and blind counterfactual deviations can actually outperform informed causal deviations. Each of these three deviation types have distinct strengths and weaknesses, which further means that there are no strict domination relationships between them or their equilibrium concepts.

Example #5: CFCE that is not a CCE. This example uses a different game from the previous examples and comes from MacQueen (2022). Player one first chooses between actions Left (L) and Right (R). If they choose L, the game ends and they receive +1. If they choose R, player two publicly chooses between L and R, and player one ends up in one of two active agent states with identical payoffs. If player one's second action is L, they receive −2, if they choose R, they receive +2.

↓ implies →	CE	CCE	EF	AF	BCE	CF	OS-CCE	OS-CF
CE	=	✓	✓	✓	✓	✓	**#6**	**#6**
CCE	#1	=	#3	**#3**	**#3**	**#4**	**#4**	**#6**
EF	#1	✓	=	**#4**	**#4**	**#4**	**#4**	**#6**
AF	#1	#2	#2	=	**#2**	**#2**	**#2**	**#6**
BCE	**#1**	✓	✓	✓	=	✓	**#6**	**#6**
CF	**#1**	**#5**	**#3**	**#3**	**#3**	=	**#6**	**#6**
OS-CCE	**#1**	✓	**#3**	**#3**	**#4**	✓	=	**#4**
OS-CF	**#1**	✓	**#3**	**#3**	**#3**	✓	✓	=

Table 6.1: Equilibrium class relationships. The relationships between the coarse-correlated and correlated versions of each EFG equilibrium concept are the same, *e.g.*, the table is identical if "EF" is replaced with "EFCE" or "EFCCE". Cyan cells highlight where the row concept implies the column concept (*e.g.*, EF $\subseteq$ CCE) either by equality (=) or by definition (✓). Otherwise, the cell is colored red and references one of the counterexamples illustrated by Figs. 6.2 to 6.5, *e.g.*, EF $\nsubseteq$ AF according to example #4 illustrated by Fig. 6.5. Bold entries were previously unknown or unclear *e.g.*, EF $\not\Rightarrow$ AF and CF $\not\Rightarrow$ EF.Dividers separate previously defined equilibrium concepts from those formalized in this thesis.

The CFCE recommendations are for player one to always choose L and for player two to switch between L and R: $[\{(\text{L},\ \text{L}\,|\,\text{R L},\ \text{L}\,|\,\text{R R}), (\text{L})\}, \{(\text{L},\ \text{L}\,|\,\text{R L},\ \text{L}\,|\,\text{R R}), (\text{R})\}]$. Player one achieves a value of +1 averaged across these two strategy profiles since they always end the game before the other player can act. We know that these recommendations are a CFCE because the counterfactual deviations either leave the recommendations unmodified or they play the R action in the root active agent state and then play R in exactly *one* of the two successor active agent states. Making any such modification ensures that player one achieves +2 under one recommendation and −2 under the other, resulting in an average value of 0. According to Definitions 5 and 6, counterfactual deviations can only modify strategies along a single path, so there is no counterfactual deviation that plays R in *both* of the successor active agent states.

However, the external deviation that always plays R achieves a value of +2 averaged across these two strategy profiles. Thus, we have a beneficial external deviation in a CFCE. The recommendations and the beneficial external deviation are visualized in Fig. 6.6

Summary. The relationships revealed by these examples are summarized in a table of relationships between equilibrium concepts (Table 6.1). The bottom row and the rightmost column references observable sequential (OS) counterfactual equilibria, which will later be connected with CFR. An example separating OS-CCE and OS-CFCCE from CE will also be presented then.

6.6 A Refined Analysis of CFR

The CFR algorithm was originally designed to solve two-player, zero-sum games by minimizing external regret, which it accomplishes through local no-external-regret learning at each agent state. This locality resembles the locality inherent to action deviations, so does CFR also happen to minimize action deviation regret? For that matter, could CFR happen to minimize causal deviation regret as well?

6.6.1 Failure on Causal and Action Deviations

We present an extension of Shapley's game (L. Shapley 1964) where CFR fails to behave according to an EFCCE or an AFCCE and therefore does not necessarily eliminate incentives for causal or action deviations.

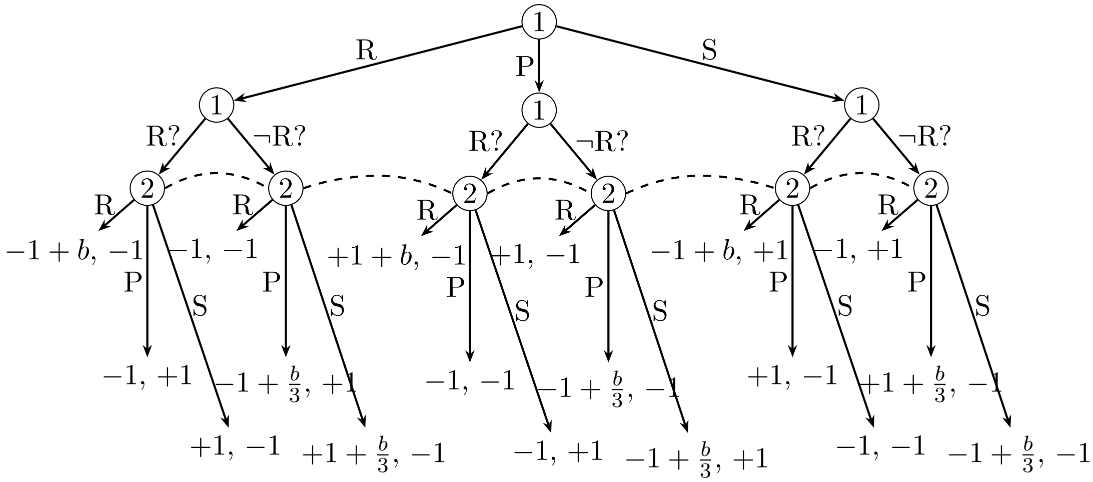


Figure 6.7: An extended Shapley's game where the first player privately predicts whether or not their opponent will play Rock, denoted R? and ¬R? respectively. Rock, Paper, and Scissors are denoted by R, P, and S, respectively. Dashed lines indicate that each of player two's histories are in the same information set.

In Shapley's game, both players simultaneously choose between Rock, Paper, and Scissors. Rock beats Scissors, Scissors beats Paper, and Paper beats Rock, but both players lose if they choose the same item. A winning player gets +1 while losing players get −1. Our extension is that player one privately predicts whether or not player two will choose Rock after choosing their action. If they accurately predict a Rock play, they receive a bonus, b, in addition to their usual reward, and if they accurately predict that they will not play Rock,

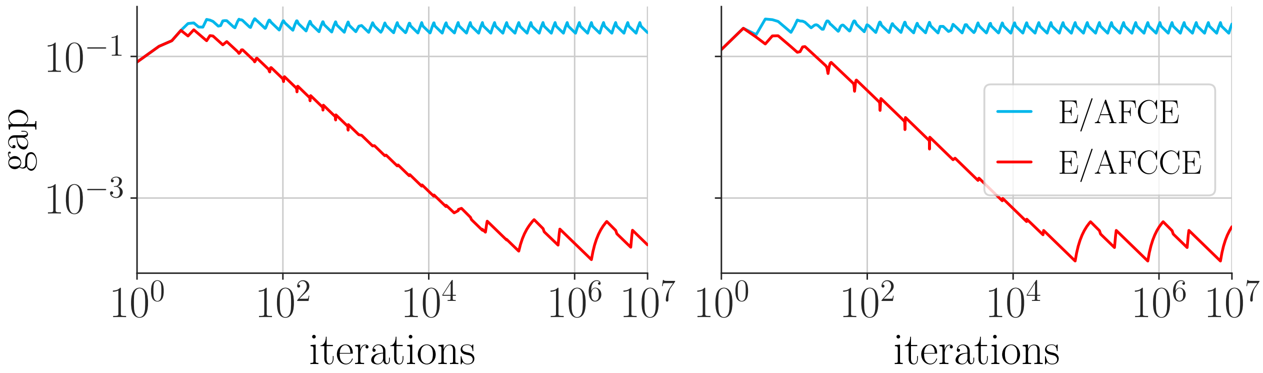


Figure 6.8: The gap between CFR's self-play empirical distribution and an extensive-form or agent-form (C)CE (E/AF(C)CE) in the extended Shapley's game with $b = 0.003$. (Left) simultaneous-update CFR. (Right) alternating-update (Burch, Moravčík, et al. 2019) CFR.

they receive a smaller bonus, $b/3$. There is no cost for inaccurate predictions, and the second player's decisions and payoffs are unchanged. The game can be found in OpenSpiel (Lanctot, Lockhart, et al. 2019) under the name `extended_shapleys.efg`. The game's extensive-form is drawn in Fig. 6.7.

Figure 6.8 shows the gap between the expected payoff achieved by CFR's self-play empirical distribution (summed across players) and an optimal causal or action deviation across iterations. In this experiment, the causal and action deviations that maximize the expected payoff across CFR's self-play empirical distribution achieve the same payoff. The E/AFCE lines correspond to the gap between CFR's payoff and the payoff of the best informed deviation (the E/AFCE gap). The E/AFCCE lines are the same except that CFR's performance is compared to the best blind deviation (the E/AFCCE gap). In all figures, the gap does not continue to decrease over time as we would expect if CFR were to minimize causal or action regret.

6.6.2 CFR and Counterfactual Deviations

Here we show that conventional CFR with no-external-regret learners at each agent state is observably sequentially (OS) hindsight rational for blind counterfactual deviations and that CFR with no-internal-regret learners is OS hindsight rational for informed counterfactual deviations. CFR's behavior in self-play therefore conforms to OS-CFCCE or OS-CFCE.

The full counterfactual regret decomposition, Lemma 4, provides a one-step recursive connection between full counterfactual regret and immediate counterfactual regret. When we unroll this recursion completely from the start of the game, we arrive at Theorem 8. But what if instead we unroll this recursion a fixed number of steps? This *intermediate counterfactual regret* is exactly the benefit of a counterfactual deviation from a given agent state and intermediate counterfactual regret can be exactly decomposed into a sum of immediate counterfactual regrets.

Definition 9. *Under perfect recall, a counterfactual deviation, ϕ, encounters a sequence of n active agent states, $(s_j)_{j=0}^{n-1}$ where $s_j \prec s_{j+1}$, on the path to target agent state s_n within the POHP rooted at active agent state s_0. The expected return of this deviation by virtue of the fact that it is a counterfactual deviation is the* intermediate counterfactual value

$$\underbrace{v_{s_0}^{\text{CF}}(\phi(\pi); \sigma)}_{\textit{Expected return from } s_0.} = \underbrace{v_{s_n}^{\text{CF}}(\phi_{s_n}(\pi); \sigma) + \sum_{j=0}^{n-1} r_{s_j}(\phi_{s_j}^{\to a_{s_j}^{\to s_n}}(\pi); \sigma).}_{\textit{Intermediate counterfactual value.}}$$

The full counterfactual regret of counterfactual deviation ϕ is the intermediate counterfactual regret $\underbrace{\rho_{s_0}^{\text{CF}}(\phi(\pi), \pi; \sigma)}_{\textit{Full counterfactual regret of } \phi.} = \underbrace{v_{s_0}^{\text{CF}}(\phi(\pi); \sigma) - v_{s_0}^{\text{CF}}(\pi; \sigma).}_{\textit{Intermediate counterfactual regret.}}$

Lemma 6. *n-step intermediate counterfactual regret decomposes exactly into immediate counterfactual regret as $\rho_{s_0}^{\text{CF}}(\phi(\pi), \pi; \sigma) = \sum_{i=0}^{n} \rho_{s_i}^{\text{CF}}(\phi_{s_i}(\pi), \pi; \sigma)$, where ϕ is a counterfactual deviation that plays to target agent state s_n from root agent state s_0 through the sequence of intermediate active agent states $(s_j)_{j=1}^{n-1}$.*

Proof. If $n = 0$, $\rho_{s_0}^{\text{CF}}(\phi(\pi), \pi; \sigma) = \rho_{s_0}^{\text{CF}}(\phi_{s_0}(\pi), \pi; \sigma)$ and the statement is trivially true. If $n > 0$, the action transformation at s_0 is external so the proof of Lemma 4 shows that $\rho_{s_0}^{\text{CF}}(\phi(\pi), \pi; \sigma) = \rho_{s_0}^{\text{CF}}(\phi_{s_0}(\pi), \pi; \sigma) + \sum_{s' \in \mathcal{S}_{\mathcal{A}}(s_0, a_{s_0}^{\to s^{\odot}})} \rho_{s'}^{\text{CF}}(\phi(\pi), \pi; \sigma)$. Since ϕ applies the identity transformation at all s' except for s_1, we can simplify this to $\rho_{s_0}^{\text{CF}}(\phi(\pi), \pi; \sigma) = \rho_{s_0}^{\text{CF}}(\phi_{s_0}(\pi), \pi; \sigma) + \rho_{s_1}^{\text{CF}}(\phi(\pi), \pi; \sigma)$. Applying this logic recursively yields the sum $\rho_{s_0}^{\text{CF}}(\phi(\pi), \pi; \sigma) = \sum_{i=0}^{n} \rho_{s_i}^{\text{CF}}(\phi_{s_i}(\pi), \pi; \sigma)$, which completes the proof. □

Since CFR minimizes immediate counterfactual regret at each agent state, Lemma 6 shows that it also minimizes intermediate counterfactual regret. Therefore, by Lemma 6 and Definitions 3 and 7 to 9:

Theorem 10. *CFR is OS hindsight rational with respect to blind counterfactual deviations in a finite-horizon POHP with perfect recall updates. If the cumulative external immediate counterfactual regret at each agent state s is upper bounded by $f(T) \geq 0$, $f(T) \in \mathrm{o}(T)$ after T rounds, then CFR's blind counterfactual regret from s (*i.e.*, full regret with respect to any blind counterfactual deviation ϕ) is upper bounded according to $\sum_{t=1}^{T} \rho_{s}^{\text{CF}}(\phi, \pi^t; \sigma^t) \leq |\mathcal{S}_{s,\mathcal{A}}| f(T)$, where $\mathcal{S}_{s,\mathcal{A}} = \{s' \in \mathcal{S}_{\mathcal{A}} \mid s \preceq s'\}$ is the active agent states in the sub-POHP rooted at s. CFR's empirical play approaches exact rationality at the same rate as its average blind counterfactual regret vanishes.*

Theorem 11. *CFR instantiated with no-internal-regret learners is OS hindsight rational with respect to informed counterfactual deviations in a finite-horizon POHP with perfect recall updates. If the cumulative internal immediate counterfactual regret at each agent state s is upper bounded by $f(T) \geq 0$, $f(T) \in \mathrm{o}(T)$ after T rounds, then CFR's full regret from s with respect to any informed counterfactual deviation ϕ is upper bounded according to $\sum_{t=1}^{T} \rho_s^{\mathrm{CF}}(\phi, \pi^t; \sigma^t) \leq |\mathcal{S}_{s,\mathcal{A}}| f(T)$. This algorithm's empirical play approaches exact rationality at the same rate as its average informed counterfactual regret vanishes.*

Theorem 12. *CFR in self-play converges to an OS-CFCCE at the same rate as its average blind counterfactual regret vanishes. CFR with no-internal-regret learners converges to an OS-CFCE at the same rate as its average informed counterfactual regret vanishes. See Theorems 10 and 11 for both of these rates.*

Theorems 10 to 12 provide the most thorough characterization of CFR's behavior, both for a single player and in self-play, to date. See the OS-CF row and column in Table 6.1 for a summary of how observable sequential counterfactual equilibria relate to the other equilibrium concepts.

Example #6: CE that is not an OS-CCE. To show how observable sequential rationality differs from ordinary rationality in practice, we now examine an example where a CE is not an OS-CCE. If the recommendations for player one are $\{-\$0; \mathrm{H} \,|\, -\$0; \mathrm{T} \,|\, -\$1\}$ and $\{-\$0; \mathrm{T} \,|\, -\$0; \mathrm{H} \,|\, -\$1\}$, then player one achieves an expected value of $+2$ at the root but an expected value of -1 after $-\$1$. There is no swap deviation that can improve the root value so this recommendation distribution is a CE. However, deviating to always choose either H or T after $-\$1$ increases the payoff after $-\$1$ by $+0.5$. This improvement could be achieved by an external deviation, and therefore, this is not an observable sequential CCE. Of course, always playing $-\$1$ and always choosing either H or T thereafter is also a blind counterfactual deviation that improves on the expected return following $-\$1$, so this recommendation distribution is also not an OS-CFCCE. The recommendations for this example are visualized in Fig. 6.9 and deviations are visualized in Fig. 6.E.15.

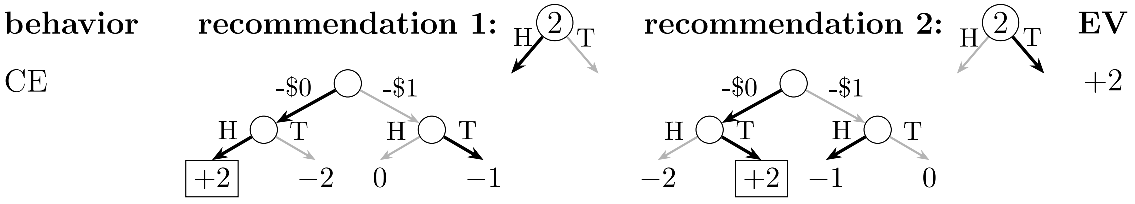


Figure 6.9: A gambling matching pennies example that is a CE but not an OS-CCE.

6.7 CFR for Action Deviations

We showed that CFR does not behave according to an EFCCE or AFCCE, but could we modify CFR to do so? Modifying CFR to address causal deviations requires recently developed non-trivial modifications (Celli et al. 2020), but we can easily modify CFR for action deviations by weighting counterfactual regrets by reach probabilities.

Since by definition an action deviation applies an action transformation at a single trigger agent state, the benefit of such a deviation is simply the average reach-probability-weighted immediate regret, no decomposition required. Therefore, employing no-regret learners on the reach-probability-weighted counterfactual value functions in each agent state ensures that the average reach-probability-weighted immediate regret vanishes.

Theorem 13. *The CFR-like algorithm that trains a no-regret learner at each agent state on the reach-probability-weighted counterfactual value functions, $\hat{v}^t_s(\cdot) : a \mapsto \mathbb{P}_{\pi^t}[s] v^{\mathrm{CF}}_s(\phi^{\to a}_s(\pi^t); \sigma^t) = \mathbb{P}_{\pi^t,\sigma^t}[s] \mathbb{E}_{H \sim \xi^{\pi^t,\sigma^t}_s}[G_{Ha}(\pi^t; \sigma^t)]$, is no-regret/hindsight rational with respect to action deviations. If the learners minimize external regret, then the algorithm minimizes blind action deviation regret, and if the learners minimize internal regret, then the algorithm minimizes informed action deviation regret. At all times, the algorithm's action deviation regret cannot be more than the maximum regret suffered by any single learner.*

Celli et al. (ibid.)'s algorithm is also no-regret with respect to informed action deviations because it uses no-internal-regret learners trained on reach-probability-weighted immediate counterfactual values. Theorem 13 gives an algorithm that is weaker in that it does not minimize causal, counterfactual, or external regret, but it does not require action sampling, nor does it require multiple learners at every agent state.

Notice that the local utility functions for CFR for action deviations is exactly PGPI's local utility functions defined by Eq. (5.10). Thus, Theorem 13 establishes a new connection between CFR and reinforcement learning algorithms via action deviations and PGPI. The primary difference between CFR for action deviations and PGPI is that CFR requires its local learners to be no regret while PGPI's may not be (see Section 5.5 for a discussion).

6.8 Partial Sequence Deviations

Representing the causal, action, and counterfactual deviation types as behavioral deviations allows us to identify complexity differences between these deviation types by counting the number of realizable memory states they admit. Across all action or counterfactual deviations, there is always exactly one memory state at each agent state to which a non-identity transformation is assigned. Thus, a hindsight rational algorithm need only ensure its strategy cannot be improved by applying a single action transformation at each agent state. Under the causal deviations, in contrast, the number of realizable memory states at agent state s is at least the number of s's predecessors since there is at least one causal deviation that triggers at each of them and plays to s. Causal deviations are therefore more costly to compete with and gives them strategic power, though notably not enough to subsume either the action or counterfactual deviations, as we saw in Section 6.5. Are there sets of behavioral deviations that subsume the causal, action, and counterfactual deviations without being much more costly than the causal deviations?

Looking at Fig. 6.10, we can see that the causal, action, and counterfactual deviations are composed of contiguous blocks of the same type of action transformation. For causal and action, the first block is made of identity transformations while for counterfactual it is made of external transformations. For causal, the second block is made of external transformations, separated from the first block by a single internal transformation in the informed case. For counterfactual and action, the second block is made of identity transformations, separated from the first by an external or internal transformation in the blind and informed cases, respectively.

Building on this observation, we can understand these deviations as having distinct phases. The *correlation phase* is an initial sequence of identity transformations, where "correlation" references the fact that the identity transformation preserves any correlation that player i's behavior has with those of the other players. There are causal and action deviations with a correlation phase, but no counterfactual deviation exhibits such behavior. All of these deviation types permit a *de-correlation phase* that modifies the input strategy with external transformations, breaking correlation. Finally, the *re-correlation phase* is where identity transformations follow a de-correlation phase, but it is only present in action and counterfactual deviations. The informed variant of each deviation type separates these phases with a single internal transformation, which both modifies the strategy and preserves correlation. The action deviation type is the only one that permits all three phases, but the de-correlation phase is limited to a single action transformation.

Why not permit all three phases at arbitrary lengths to subsume the causal, action, and counterfactual deviations? We now introduce four types of *partial sequence deviations* based on exactly this idea, where each phase spans a "partial sequence" through the game.

The *blind partial sequence* (*BPS*) deviation has all three phases and lacks any internal transformations. The set of BPS deviations is also the set of single-target deviations generated from the blind causal deviations. Just as there are exponentially fewer counterfactual deviations than external deviations, there are exponentially fewer BPS deviations than blind causal deviations.[3] There are $d_* n_{\mathcal{A}} |\mathcal{S}_{\mathcal{A}}|$ BPS deviations compared with $\mathcal{O}(n_{\mathcal{A}}^{|\mathcal{S}_{\mathcal{A}}|} |\mathcal{S}_{\mathcal{A}}|)$ blind

[3] The complexity referred to here is that of directly competing with a deviation set in general, which

causal deviations. Combined with OSR via Theorem 6, the BPS deviations capture the same strategic power with an exponential reduction in complexity. Even better, the set of BPS deviations includes the sets of blind action and blind counterfactual deviations. The empirical distribution of hindsight rational play for BPS deviations thus converges toward what we call a *BPS correlated equilibrium.* An OS BPS correlated equilibrium (OS-BPSCE) is in the intersection of the OS versions of three equilibrium sets: extensive-form coarse-correlated equilibrium (EFCCE) (Farina, Bianchi, et al. 2020), agent-form coarse-correlated equilibrium (AFCCE) (Morrill, D'Orazio, Sarfati, Lanctot, Wright, A. R. Greenwald, et al. 2021), and counterfactual coarse-correlated equilibrium (CFCCE) (ibid.).

In general, re-correlation is strategically useful and adding it to a deviation type (transforming it into a single-target deviation type) *decreases* its complexity! While this observation may be new in its generality, Zinkevich, Johanson, et al. (2007b) implicitly uses this property of deviations in EFGs and specifically the fact that the set of blind counterfactual deviations is the set of single-target deviations generated from the set of external deviations.

There are three versions of informed partial sequence deviations due to the asymmetry between informed causal and informed counterfactual deviations. A *causal partial sequence* (*CSPS*) deviation uses an internal transformation at the end of the correlation phase while a *counterfactual partial sequence* (*CFPS*) deviation uses an internal transformation at the start of the re-correlation phase. A *twice informed partial sequence* (*TIPS*) deviation uses internal transformations at both positions, making it the strongest of our partial sequence deviation types.

The set of CSPS deviations is the set of single-target deviations generated from the set of informed causal deviations, and therefore subsumes the informed causal deviations, when used with OSR, while being exponentially smaller. TIPS achieves our initial goal as it subsumes the informed causal, informed action, and informed counterfactual deviations at the cost of an $n_{\mathcal{A}}$ factor compared to CSPS or CFPS, when used with OSR. Each type of informed partial sequence deviation corresponds to a new equilibrium concept and a new OS equilibrium concept in the intersection of previously studied equilibrium concepts.

Table 7.1 gives a formal definition of each deviation type derived from behavioral deviations, Fig. 6.10 gives a visualization of each type along with their relationships, and Table 6.3 summarizes the complexity of each type.

is determined by the number of such deviations. Algorithms may be less complex than the deviation sets they minimize regret for, either by utilizing the structure of the environment and the deviation set (*e.g.*, by representing pure strategies in sequence-form), or by directly competing with a stronger but less complex set of deviations. For example, CFR's complexity is linear in the number of agent states and is hindsight rational for the external deviations even though the number of external deviations grows exponentially with the number of agent states since CFR directly competes with the less numerous blind counterfactual deviations.

Table 6.2: Formal definition of the strategy generated by a deviation of each type given pure strategy $x \in \mathcal{X}$ at each active agent state $s \in \mathcal{S}_{\mathcal{A}}$.

single-target behavioral	$\forall s',\ \exists a^{!}_{s'}, a'_{s'},$ $\begin{cases} a'_s & \text{if } \forall \bar{s}{\preceq}s, x(\bar{s}){=}a^{!}_{\bar{s}} \\ x(s) & \text{o.w.} \end{cases}$	in. causal	$\exists s^{!}, a^{!}, x',$ $\begin{cases} x'(s) & \text{if } s{\succeq}s^{!}, x(s^{!}){=}a^{!} \\ x(s) & \text{o.w.} \end{cases}$
TIPS	$\exists s^{!}, a^{!}, s^{\odot}, a^{\odot}, a^{\odot!},$ $\begin{cases} a^{\odot} & \text{if } s{=}s^{\odot}, x(s^{\odot}){=}a^{\odot!}, \\ & x(s^{!}){=}a^{!} \\ a_s^{\to s^{\odot}} & \text{if } s{\succeq}s^{!}, x(s^{!}){=}a^{!} \\ x(s) & \text{o.w.} \end{cases}$	in. action	$\exists s^{!}, a^{\odot}, a^{!},$ $\begin{cases} a^{\odot} & \text{if } s{=}s^{!}, x(s^{!}){=}a^{!} \\ x(s) & \text{o.w.} \end{cases}$
CSPS	$\exists s^{!}, a^{!}, s^{\odot}, a^{\odot},$ $\begin{cases} a^{\odot} & \text{if } s{=}s^{\odot}, x(s^{!}){=}a^{!} \\ a_s^{\to s^{\odot}} & \text{if } s{\succeq}s^{!}, x(s^{!}){=}a^{!} \\ x(s) & \text{o.w.} \end{cases}$	in. CF	$\exists s^{\odot}, a^{\odot}, a^{\odot!},$ $\begin{cases} a^{\odot} & \text{if } s{=}s^{\odot}, x(s^{\odot}){=}a^{\odot!} \\ a_s^{\to s^{\odot}} & \text{if } s{\preceq}s^{\odot} \\ x(s) & \text{o.w.} \end{cases}$
CFPS	$\exists s^{!}, s^{\odot}, a^{\odot}, a^{\odot!},$ $\begin{cases} a^{\odot} & \text{if } s{=}s^{\odot}, x(s^{\odot}){=}a^{\odot!} \\ a_s^{\to s^{\odot}} & \text{if } s{\succeq}s^{!} \\ x(s) & \text{o.w.} \end{cases}$	blind causal	$\exists s^{!}, x',$ $\begin{cases} x'(s) & \text{if } s{\succeq}s^{!} \\ x(s) & \text{o.w.} \end{cases}$
BPS	$\exists s^{!}, s^{\odot}, a^{\odot},$ $\begin{cases} a^{\odot} & \text{if } s{=}s^{\odot} \\ a_s^{\to s^{\odot}} & \text{if } s{\succeq}s^{!} \\ x(s) & \text{o.w.} \end{cases}$	blind action	$\exists s^{!}, a^{\odot},$ $\begin{cases} a^{\odot} & \text{if } s{=}s^{!} \\ x(s) & \text{o.w.} \end{cases}$
		blind CF	$\exists s^{\odot}, a^{\odot},$ $\begin{cases} a^{\odot} & \text{if } s{=}s^{\odot} \\ a_s^{\to s^{\odot}} & \text{if } s{\preceq}s^{\odot} \\ x(s) & \text{o.w.} \end{cases}$

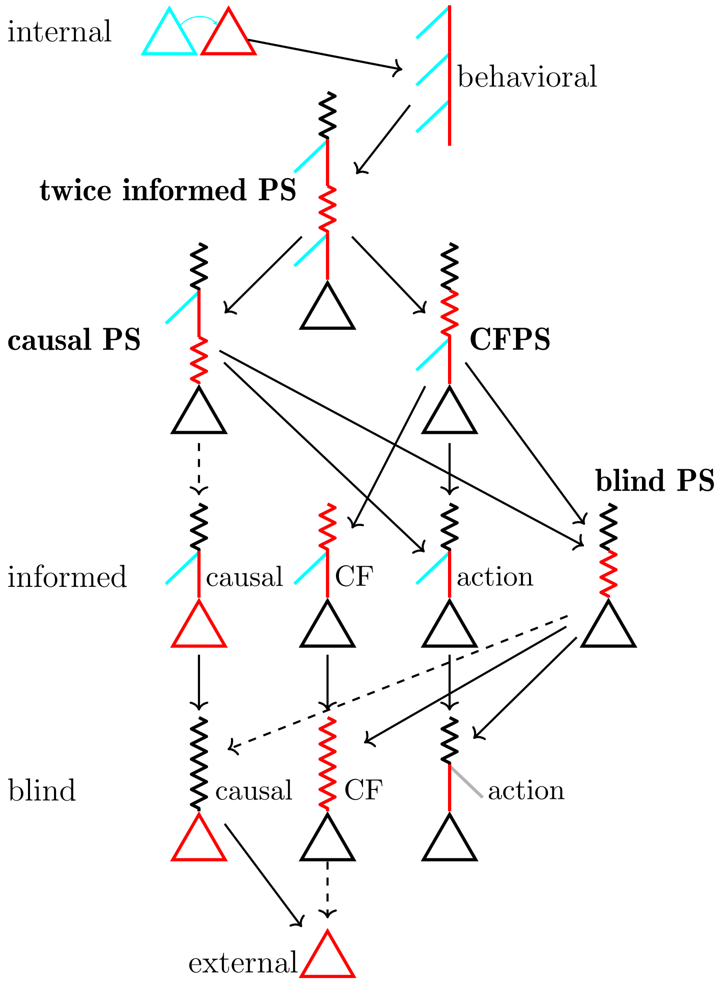


Figure 6.10: A summary of the deviation landscape in finite-horizon POHPs under perfect recall. Each pictogram is an abstract prototypical deviation representing a named set of deviations (a deviation type). Games play out from top to bottom. Straight lines represent action transformations, zigzags are transformation sequences, and triangles are transformations of entire decision trees. Identity transformations are colored black; internal transformations have a cyan component representing the trigger action or strategy and a red component representing the deviation action or strategy; and external transformations only have a red component. Arrows denote ordering from a stronger to a weaker deviation type (and therefore a subset to superset equilibrium relationship), the dashed arrow denotes that this relationship holds only under observable sequential rationality.

Table 6.3: A rough accounting of (i) realizable memory states, (ii) action transformations, and (iii) the total number of deviations showing dominant terms. Columns (i) and (ii) are with respect to a single agent state.

type	(i)	(ii)	(iii)
internal	N/A	N/A	$n_{\mathcal{A}}^{2\lvert\mathcal{S}_{\mathcal{A}}\rvert}$
single-target behavioral	$n_{\mathcal{A}}^{d_*}$	$n_{\mathcal{A}}^2$	$n_{\mathcal{A}}^{d_*+2}\lvert\mathcal{S}_{\mathcal{A}}\rvert$
TIPS	$d_* n_{\mathcal{A}}$	$n_{\mathcal{A}}^2$	$d_* n_{\mathcal{A}}^3\lvert\mathcal{S}_{\mathcal{A}}\rvert$
CSPS	$d_* n_{\mathcal{A}}$	$n_{\mathcal{A}}{}^\dagger$	$d_* n_{\mathcal{A}}^2\lvert\mathcal{S}_{\mathcal{A}}\rvert$
CFPS	d_*	$n_{\mathcal{A}}^2$	$d_* n_{\mathcal{A}}^2\lvert\mathcal{S}_{\mathcal{A}}\rvert$
BPS	d_*	$n_{\mathcal{A}}$	$d_* n_{\mathcal{A}}\lvert\mathcal{S}_{\mathcal{A}}\rvert$
informed causal	d_*	N/A	$n_{\mathcal{A}}^{\lvert\mathcal{S}_{\mathcal{A}}\rvert+1}\lvert\mathcal{S}_{\mathcal{A}}\rvert$
informed action	1	$n_{\mathcal{A}}^2$	$n_{\mathcal{A}}^2\lvert\mathcal{S}_{\mathcal{A}}\rvert$
informed CF	1	$n_{\mathcal{A}}^2$	$n_{\mathcal{A}}^2\lvert\mathcal{S}_{\mathcal{A}}\rvert$
blind causal	d_*	N/A	$n_{\mathcal{A}}^{\lvert\mathcal{S}_{\mathcal{A}}\rvert}\lvert\mathcal{S}_{\mathcal{A}}\rvert$
blind action	1	$n_{\mathcal{A}}$	$n_{\mathcal{A}}\lvert\mathcal{S}_{\mathcal{A}}\rvert$
blind CF	1	$n_{\mathcal{A}}$	$n_{\mathcal{A}}\lvert\mathcal{S}_{\mathcal{A}}\rvert$
external	N/A	N/A	$n_{\mathcal{A}}^{\lvert\mathcal{S}_{\mathcal{A}}\rvert}$

† One memory state at each agent state is associated with the set of internal transformations which contains $\mathcal{O}(n_{\mathcal{A}}^2)$ transformations, but this is dominated by the number of external transformations associated with every other memory state in non-root agent states.

6.9 Conclusion

The correlated equilibrium and deviation landscapes of POHPs are particularly rich because there is substantial space for deviation types that have intermediate power and computational requirements between external and internal deviations. To develop these deviation and equilibrium types, we re-examined causal and action deviations along with their corresponding equilibrium concepts, extensive-form and agent-form equilibria. We showed how action deviations can outperform causal deviations, which dispels a common misunderstanding that an EFCE is always an AFCE.

We showed that CFR's empirical play does not converge toward an EFCCE or AFCCE in self-play and thus CFR is not hindsight rational for causal or action deviations. Instead, we defined blind and informed counterfactual deviations to more precisely characterize CFR's behavior. CFR is OS hindsight rational with respect to blind counterfactual deviations and CFR with internal learners has the same property with respect to informed counterfactual deviations. In self-play, these algorithms converge toward OS-CFCCEs or OS-CFCEs, respectively.

Table 6.1 summarizes all of the equilibrium relationships investigated in this chapter.

In the next chapter, CFR is modified to handle any subset of the behavioral deviations and the practical benefits of this new algorithm are illustrated.

References

Burch, N., M. Moravčík, and M. Schmid (2019). "Revisiting CFR$^+$ and Alternating Updates". In: *Journal of Artificial Intelligence Research* 64, pp. 429–443.

Celli, A., A. Marchesi, G. Farina, and N. Gatti (2020). "No-regret learning dynamics for extensive-form correlated equilibrium". In: *Advances in Neural Information Processing Systems* 33.

Dudík, M. and G. J. Gordon (2009). "A Sampling-Based Approach to Computing Equilibria in Succinct Extensive-Form Games". In: *25th Conference on Uncertainty in Artificial Intelligence (UAI-2009).*

Farina, G., T. Bianchi, and T. Sandholm (Feb. 2020). "Coarse Correlation in Extensive-Form Games". In: *34th AAAI Conference on Artificial Intelligence*. New York, New York, USA.

Farina, G., C. K. Ling, F. Fang, and T. Sandholm (2019). "Correlation in Extensive-Form Games: Saddle-Point Formulation and Benchmarks". In: *Advances in Neural Information Processing Systems (NeurIPS).*

Gordon, G. J., A. Greenwald, and C. Marks (2008). "No-Regret Learning in Convex Games". In: *25th international conference on Machine learning*, pp. 360–367.

Greenwald, A., A. Jafari, and C. Marks (Aug. 2003). "A general class of no-regret learning algorithms and game-theoretic equilibria". In: *2003 Computational Learning Theory Conference*, pp. 1–11.

Huang, W. (2011). “Equilibrium Computation for Extensive Games”. PhD thesis. London School of Economics and Political Science.

Lanctot, M., E. Lockhart, et al. (2019). “OpenSpiel: A Framework for Reinforcement Learning in Games”. In: *CoRR* abs/1908.09453. arXiv: 1908.09453 `[cs.LG]`.

MacQueen, R. (May 2022). “Personal communication”.

Morrill, D., R. D’Orazio, M. Lanctot, J. R. Wright, M. Bowling, and A. R. Greenwald (July 2021). “Efficient Deviation Types and Learning for Hindsight Rationality in Extensive-Form Games”. In: *38th International Conference on Machine Learning (ICML 2021).* Vol. 139. virtual, pp. 7818–7828.

Morrill, D., R. D’Orazio, R. Sarfati, M. Lanctot, J. R. Wright, A. R. Greenwald, and M. Bowling (Feb. 2021). “Hindsight and Sequential Rationality of Correlated Play”. In: *35th AAAI Conference on Artificial Intelligence*. Vol. 35. 6. virtual, pp. 5584–5594.

Shapley, L. (1964). “Some Topics in Two-Person Games”. In: *Advances in Game Theory.* Princeton University Press.

von Stengel, B. and F. Forges (2008). “Extensive-form correlated equilibrium: Definition and computational complexity”. In: *Mathematics of Operations Research* 33.4, pp. 1002–1022.

Zinkevich, M., M. Johanson, M. Bowling, and C. Piccione (Dec. 2007b). “Regret Minimization in Games with Incomplete Information”. In: *Advances in Neural Information Processing Systems (NeurIPS 2007).* Vancouver, British Columbia, pp. 1729–1736.

6.A Example #1: BCE That Is Not a CE

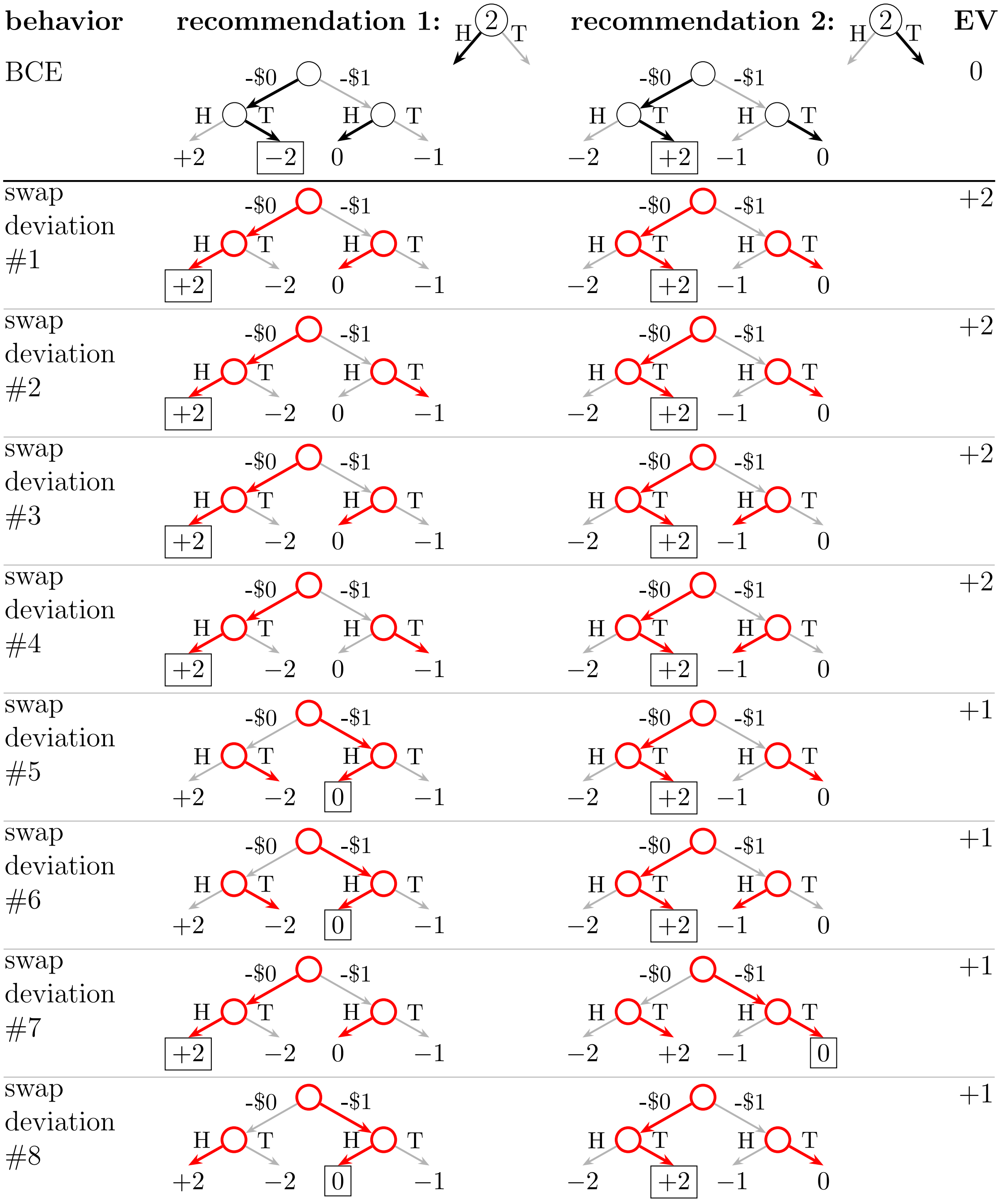


Figure 6.A.11: A gambling matching pennies example where a BCE is not a CE. Only eight of the highest value swap deviations are shown for brevity.

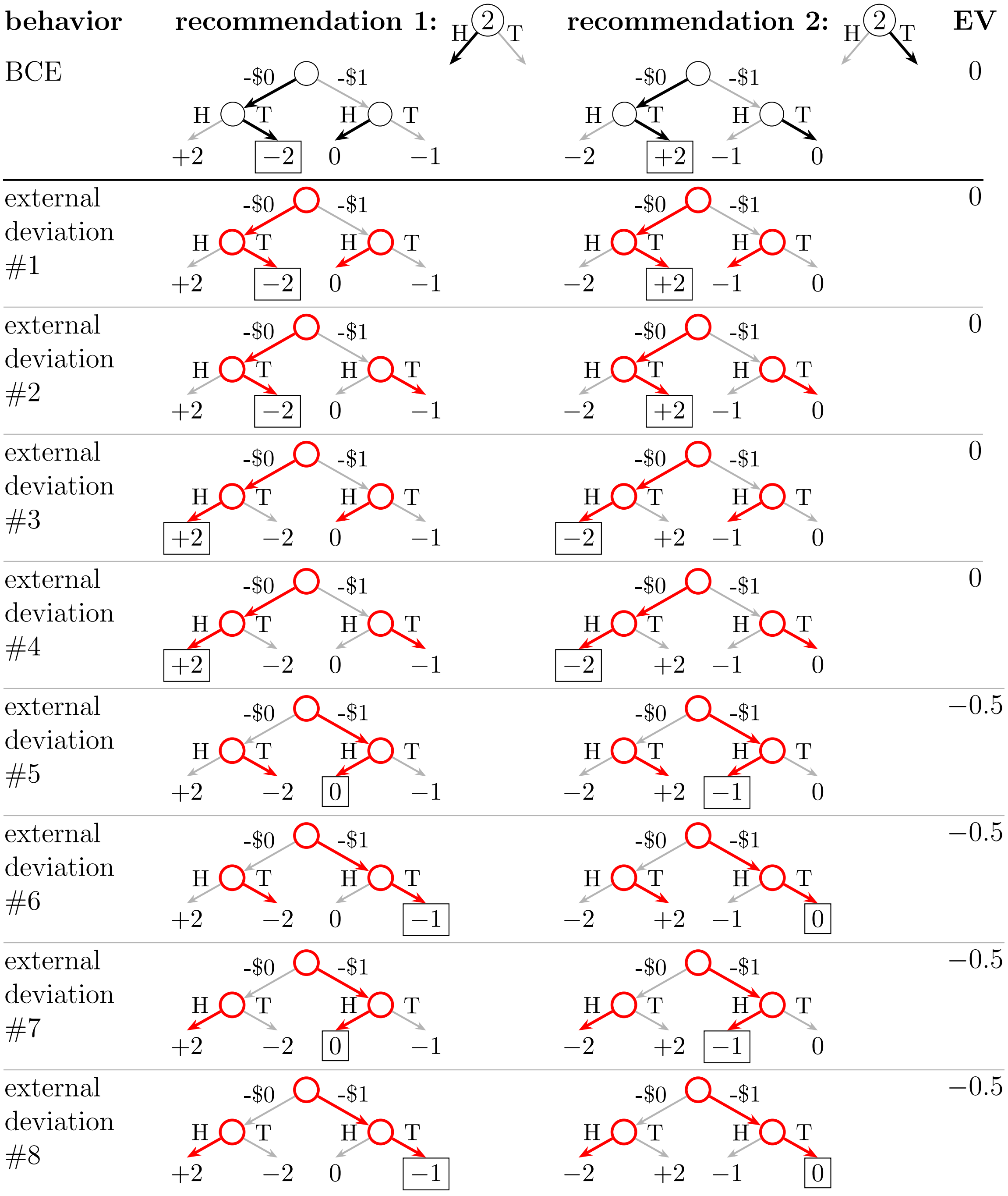


Figure 6.A.11 (Cont.): All external deviations in the gambling matching pennies example of a BCE that is not a CE.

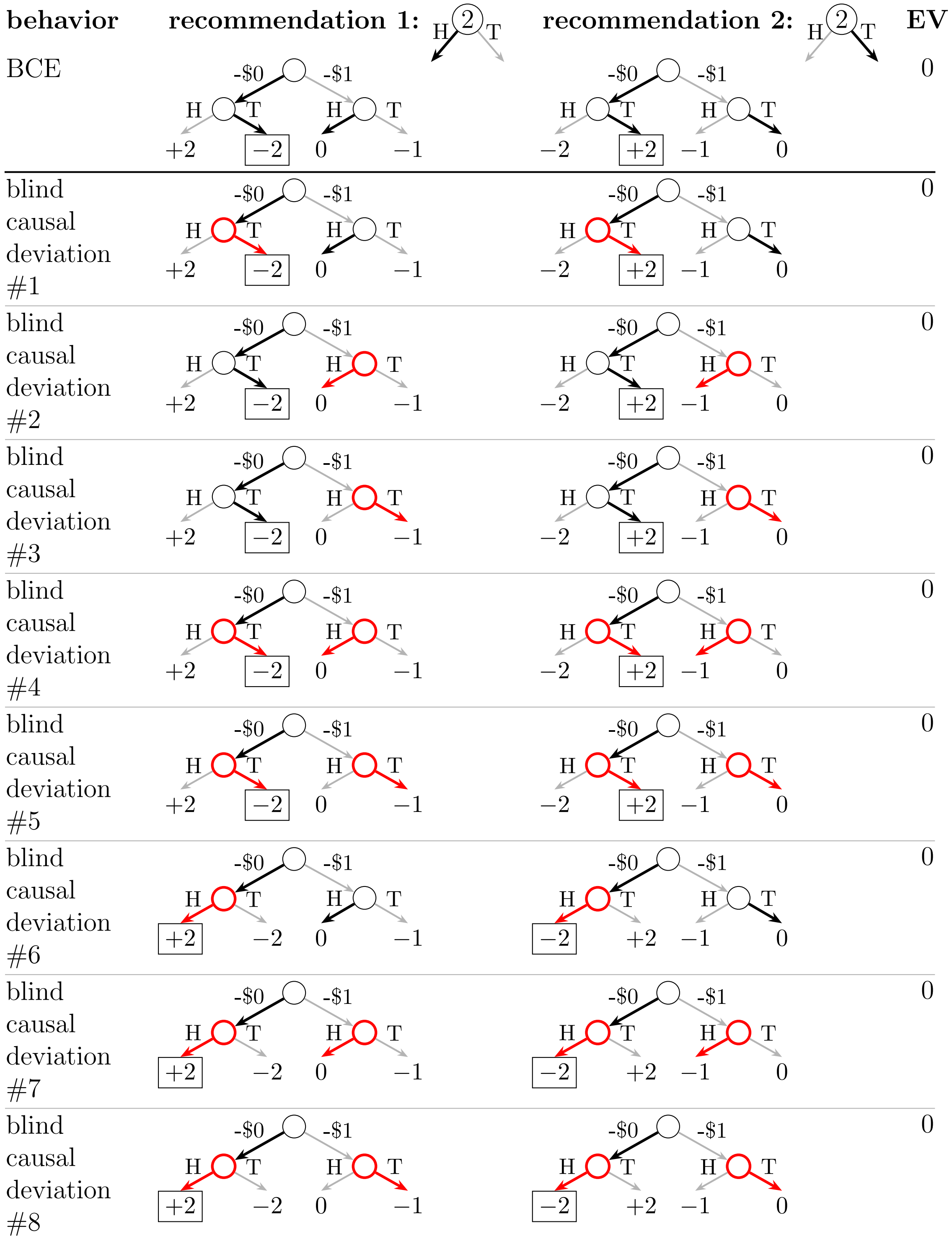

behavior
recommendation 1:
recommendation 2:
EV
2
H
T
BCE
blind causal deviation #1
blind causal deviation #2
blind causal deviation #3
blind causal deviation #4
blind causal deviation #5
blind causal deviation #6
blind causal deviation #7
blind causal deviation #8
-$0
-$1
+2
−2
0
−1
−2
+2
−1
0
0

Figure 6.A.11 (Cont.): All non-external blind causal deviations in the gambling matching pennies example of a BCE that is not a CE.

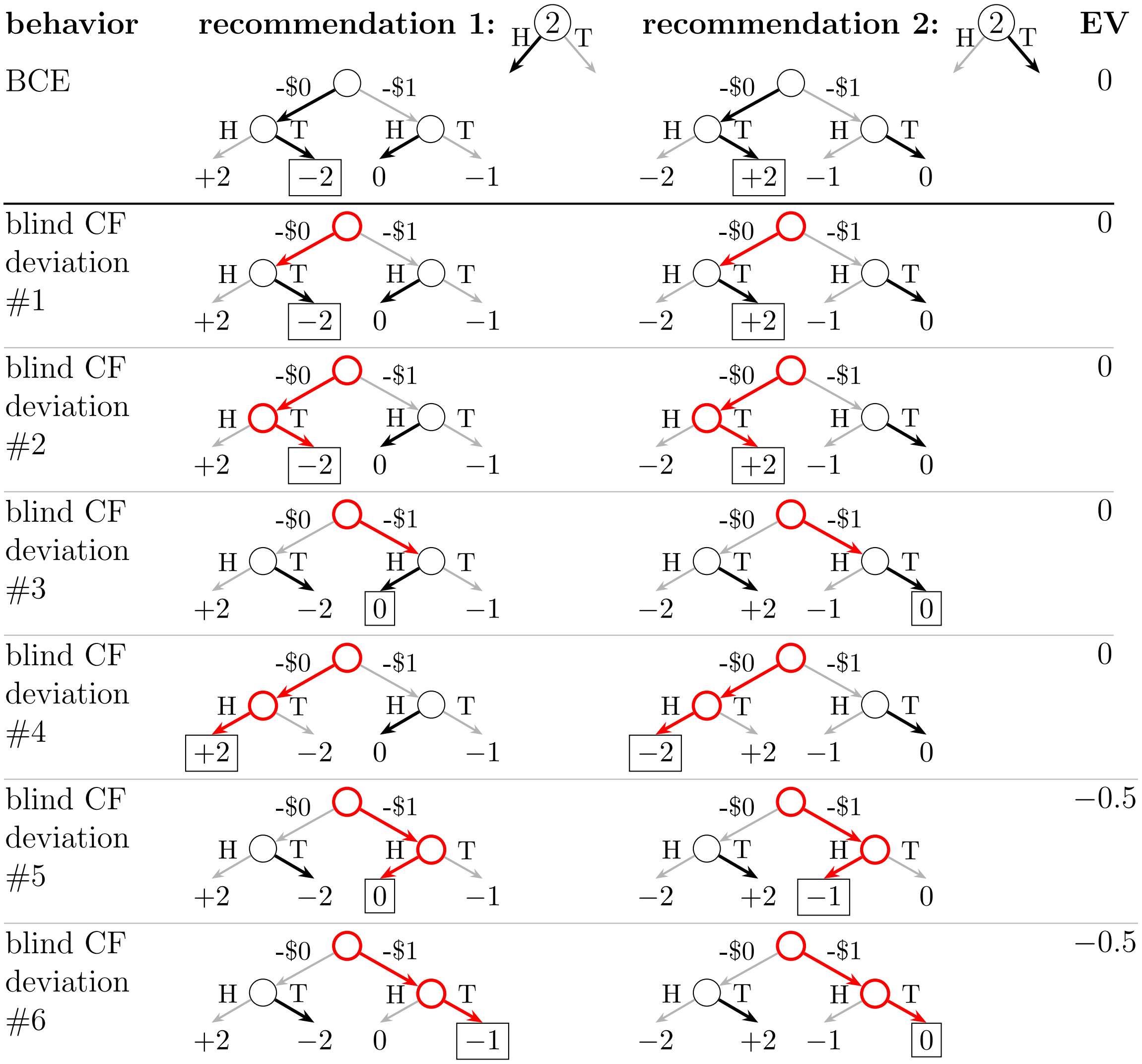


Figure 6.A.11 (Cont.): All blind counterfactual deviations in the gambling matching pennies example of a BCE that is not a CE.

6.B Example #2: AFCE That Is Not a CCE

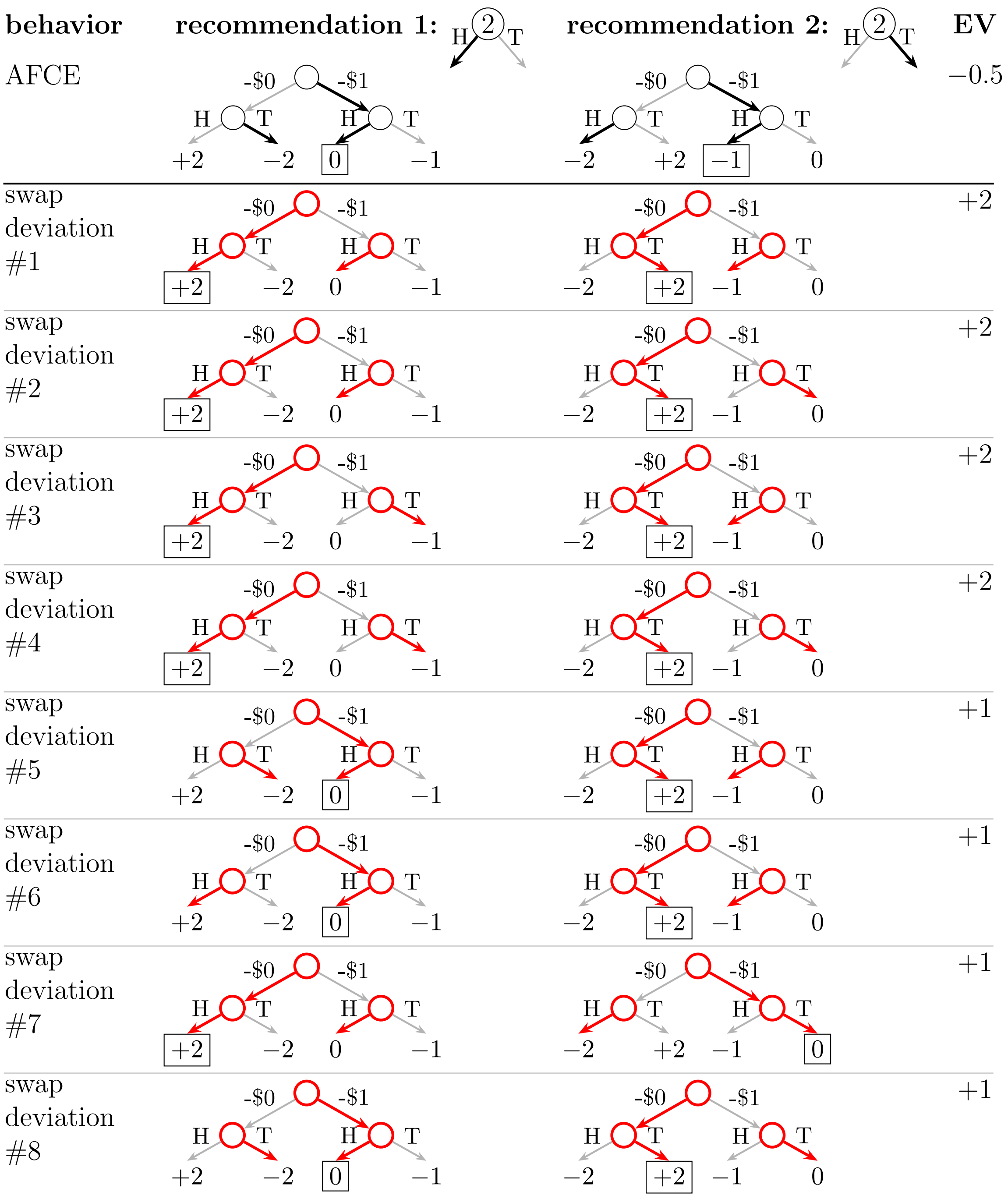


Figure 6.B.12: A gambling matching pennies example where an AFCE is not a CCE. Only eight of the highest value swap deviations are shown for brevity.

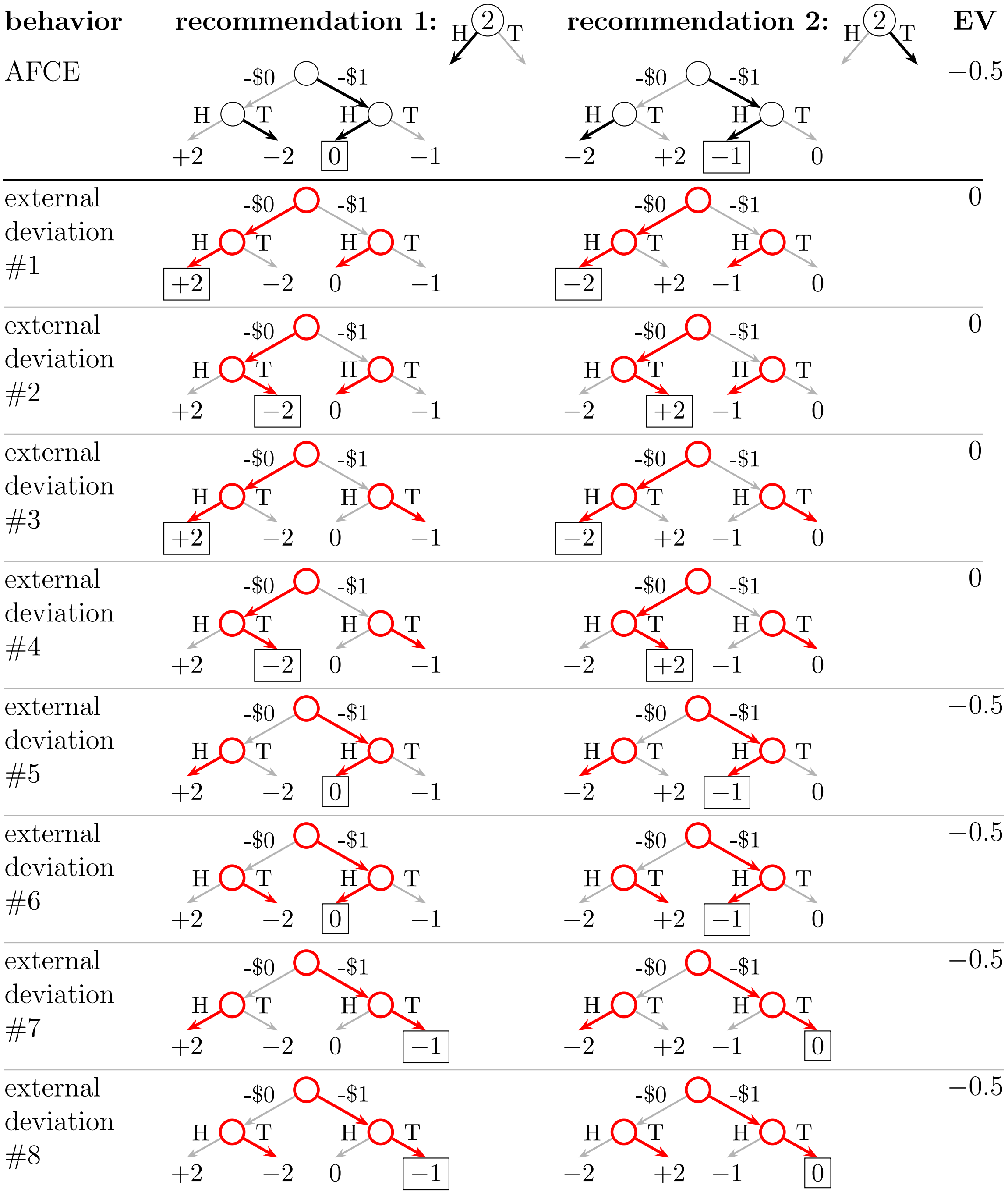


Figure 6.B.12 (Cont.): All external deviations in the gambling matching pennies example of an AFCE that is not a CCE.

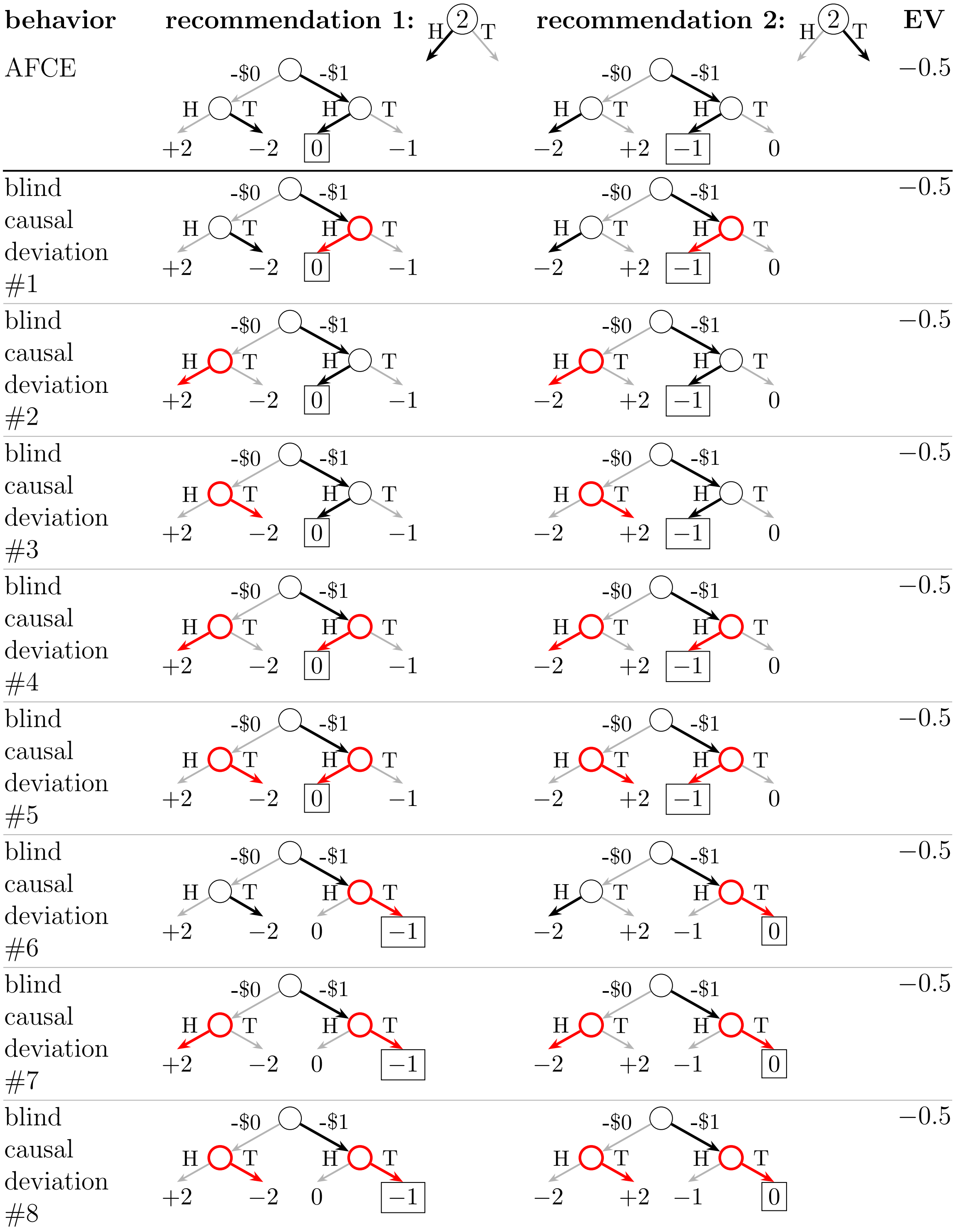

behavior
recommendation 1:
H 2 T
recommendation 2:
H 2 T
EV
AFCE
-$0
-$1
H T
H T
+2
−2
0
−1
−2
+2
−1
0
−0.5
blind causal deviation #1
−0.5
blind causal deviation #2
−0.5
blind causal deviation #3
−0.5
blind causal deviation #4
−0.5
blind causal deviation #5
−0.5
blind causal deviation #6
−0.5
blind causal deviation #7
−0.5
blind causal deviation #8
−0.5

Figure 6.B.12 (Cont.): All non-external blind causal deviations in the gambling matching pennies example of an AFCE that is not a CCE.

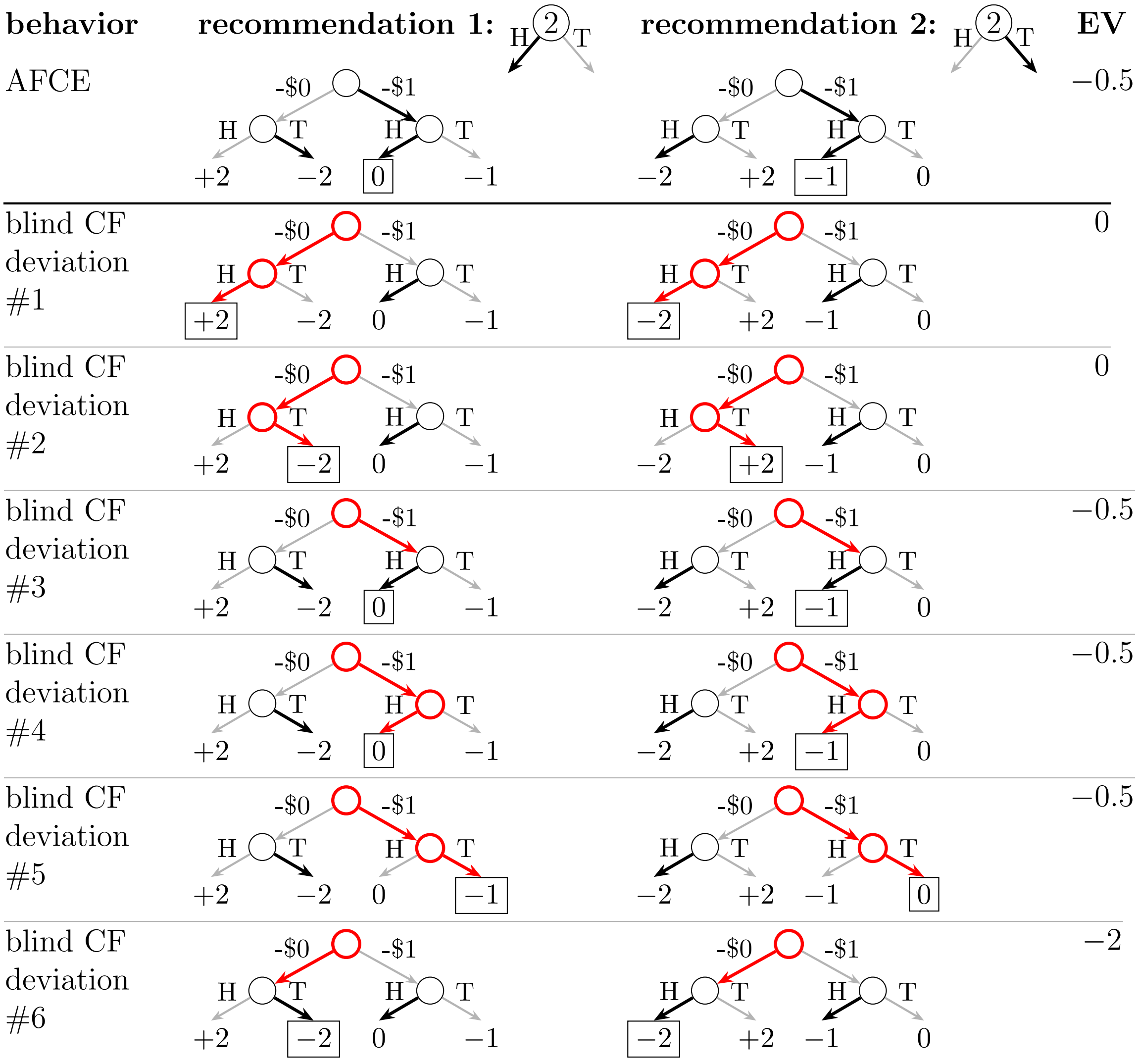


Figure 6.B.12 (Cont.): All blind counterfactual deviations in the gambling matching pennies example of an AFCE that is not a CCE.

6.C Example #3: AFCE and CFCE That Is Not an EFCCE

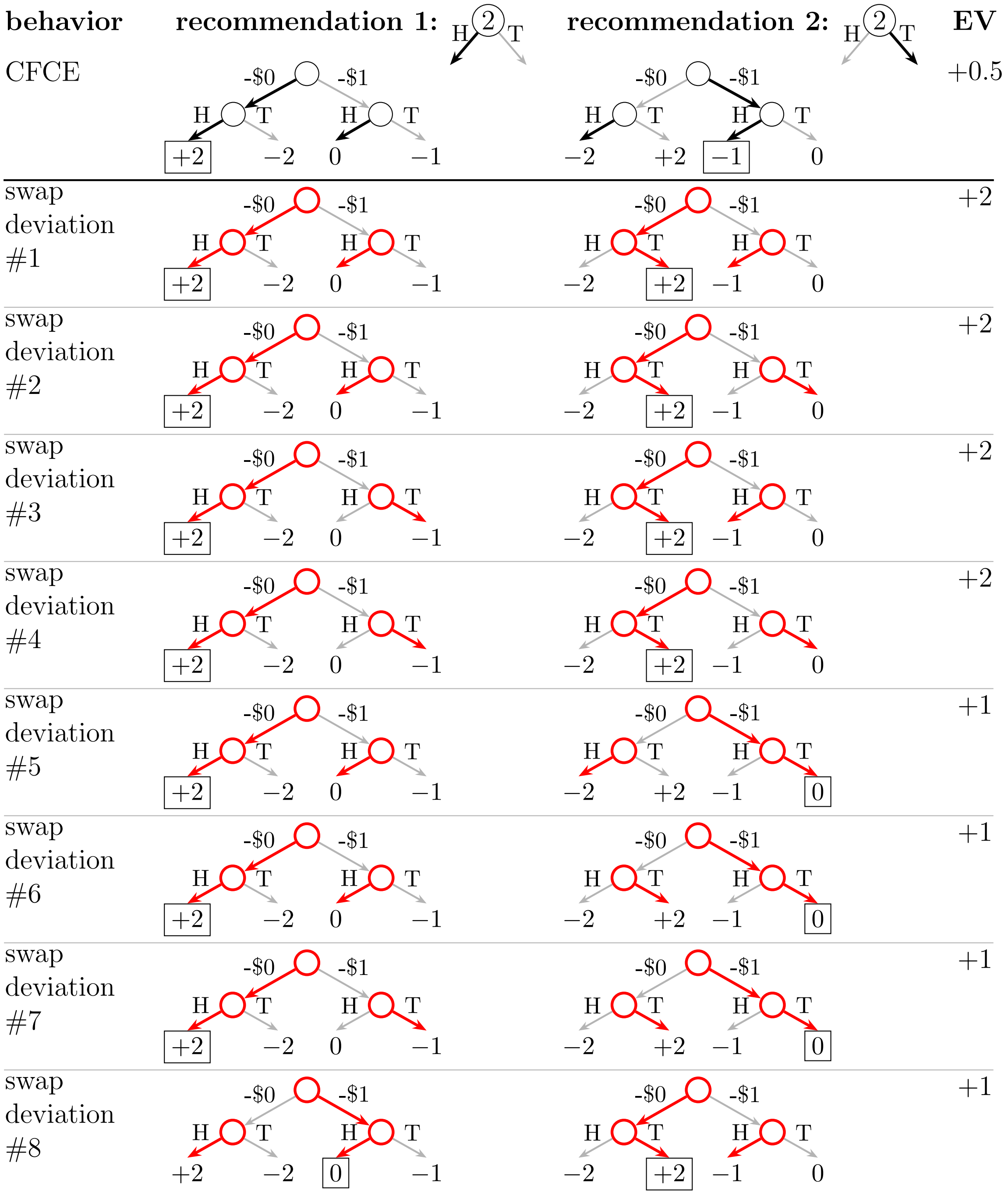


Figure 6.C.13: A gambling matching pennies example where a CFCE is not an EFCCE. Only eight of the highest value swap deviations are shown for brevity.

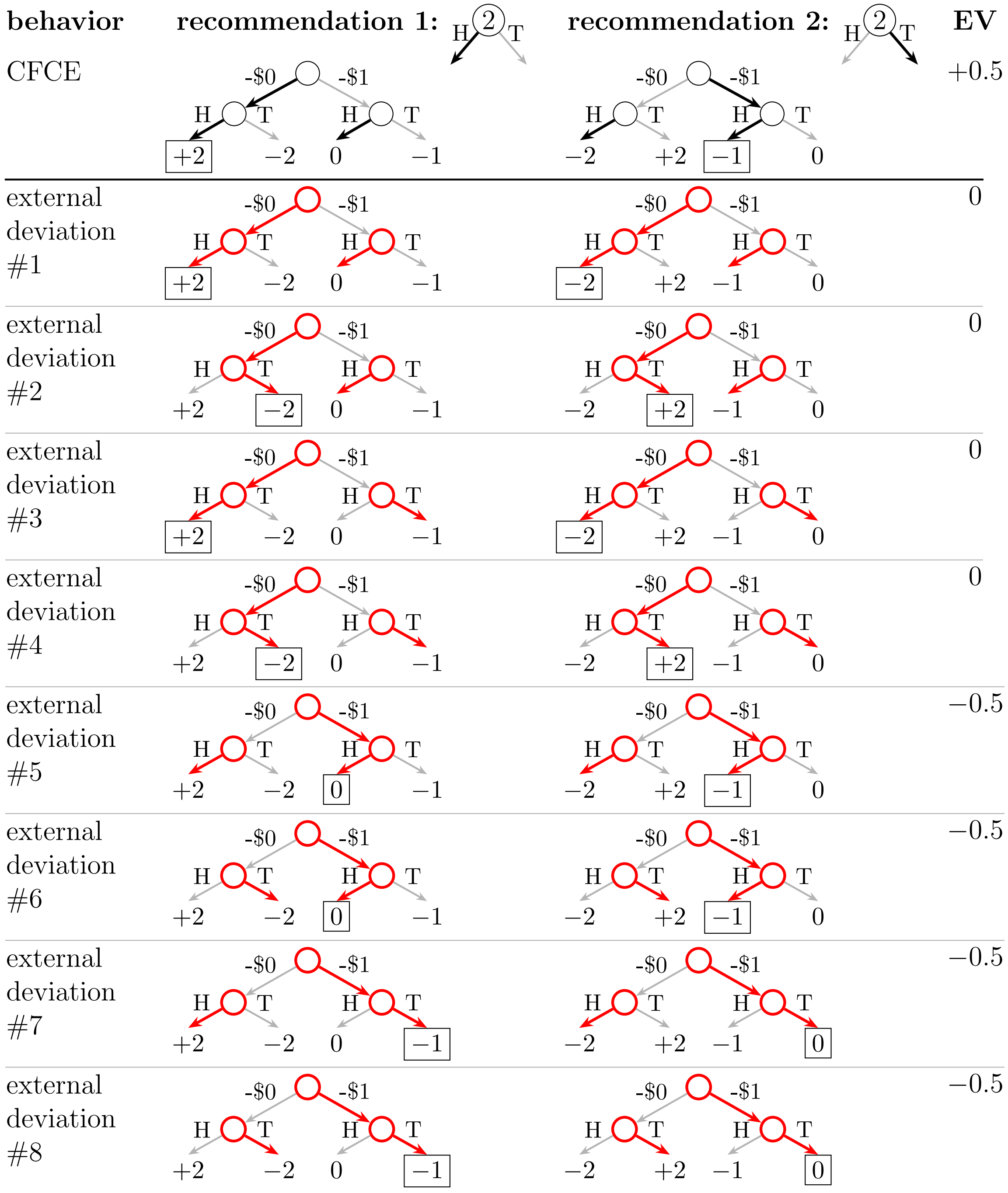


Figure 6.C.13 (Cont.): All external deviations in the gambling matching pennies example of a CFCE that is not an EFCCE.

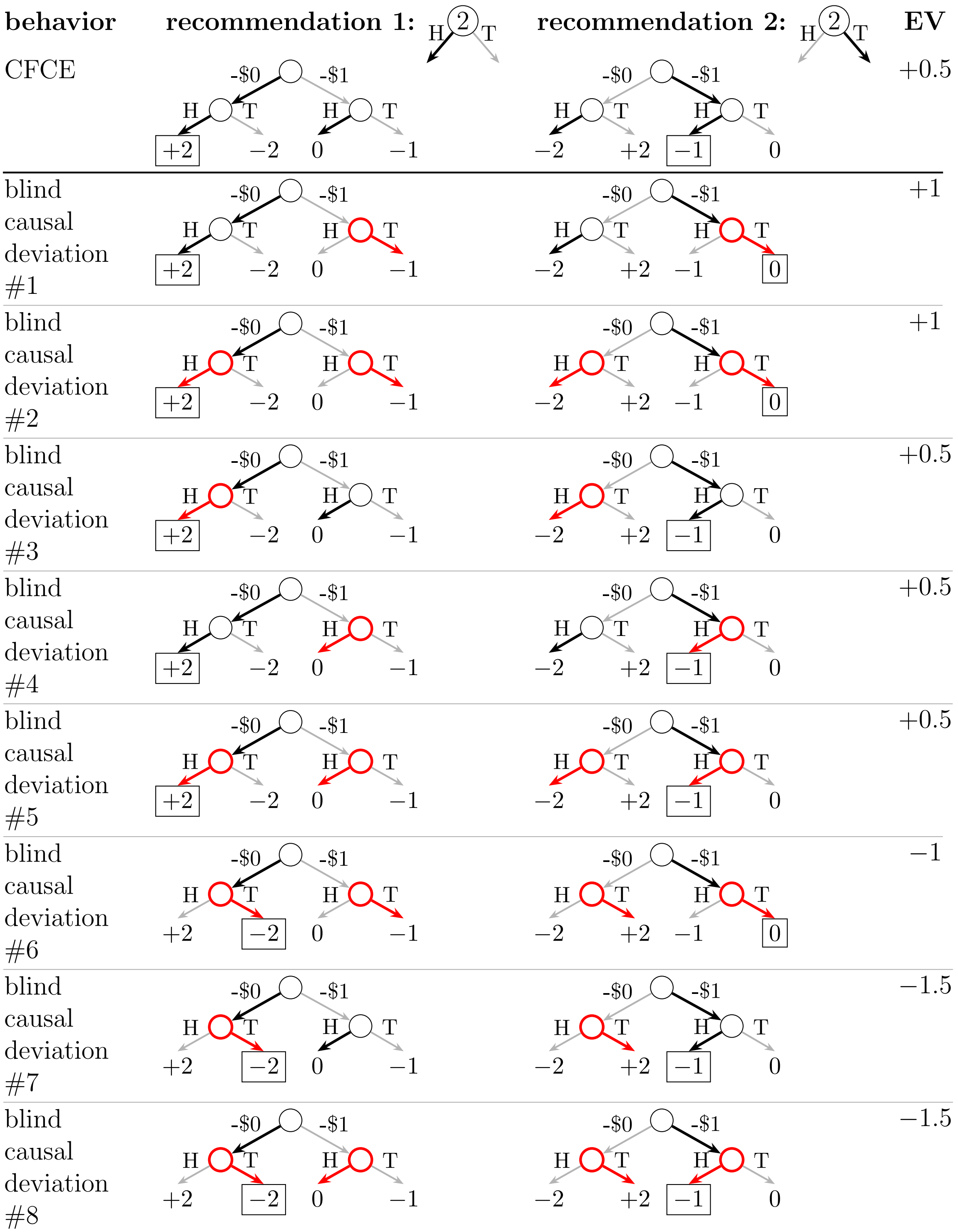

behavior
recommendation 1:
H
2
T
recommendation 2:
H
2
T
EV
CFCE
-$0
-$1
H
T
H
T
+2
−2
0
−1
−2
+2
−1
0
+0.5
blind causal deviation #1
+1
blind causal deviation #2
+1
blind causal deviation #3
+0.5
blind causal deviation #4
+0.5
blind causal deviation #5
+0.5
blind causal deviation #6
−1
blind causal deviation #7
−1.5
blind causal deviation #8
−1.5

Figure 6.C.13 (Cont.): All non-external blind causal deviations in the gambling matching pennies example of a CFCE that is not an EFCCE.

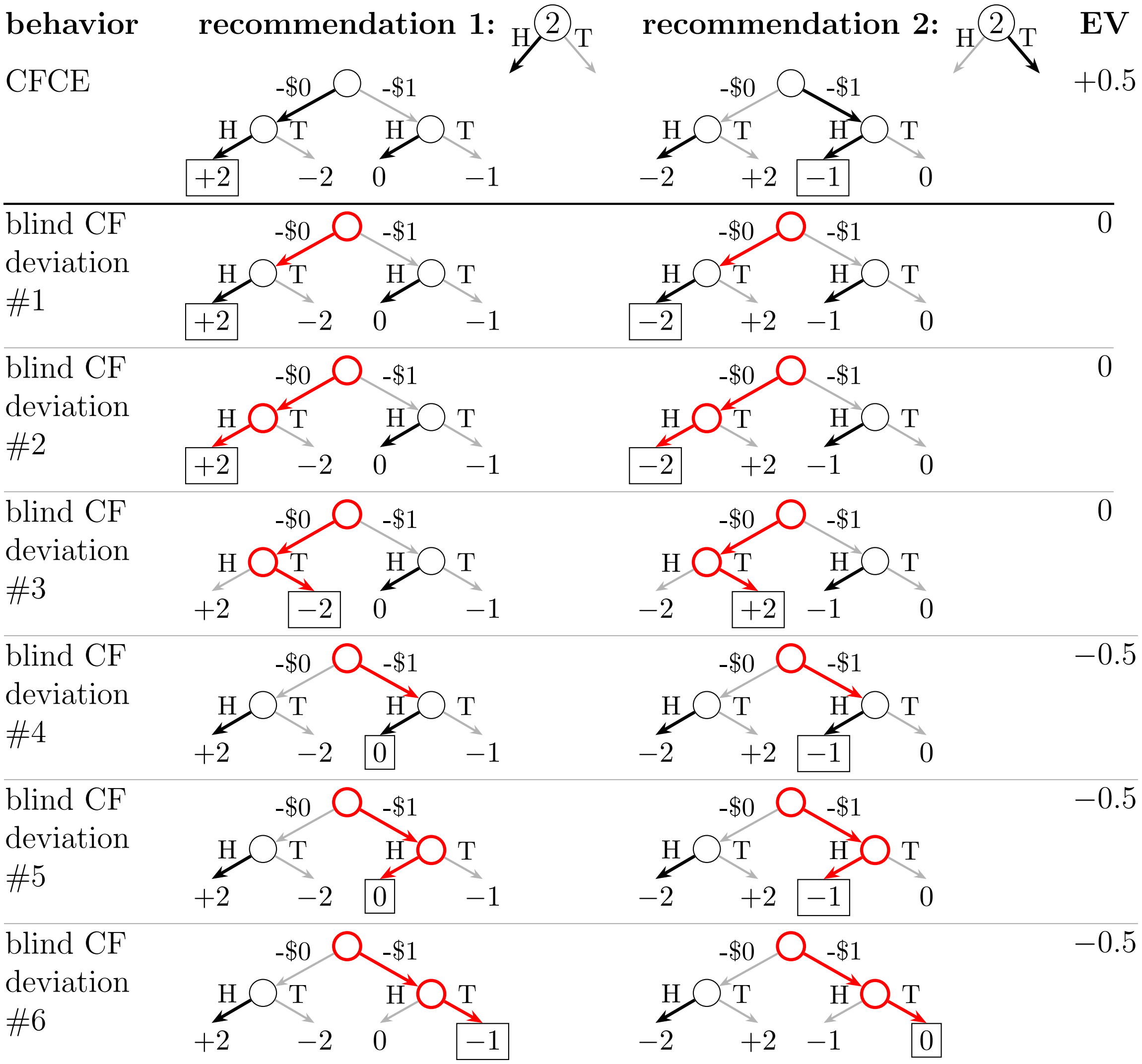


Figure 6.C.13 (Cont.): All blind counterfactual deviations in the gambling matching pennies example of a CFCE that is not an EFCCE.

6.D Example #4: EFCE That Is Not an AFCCE or CFCCE

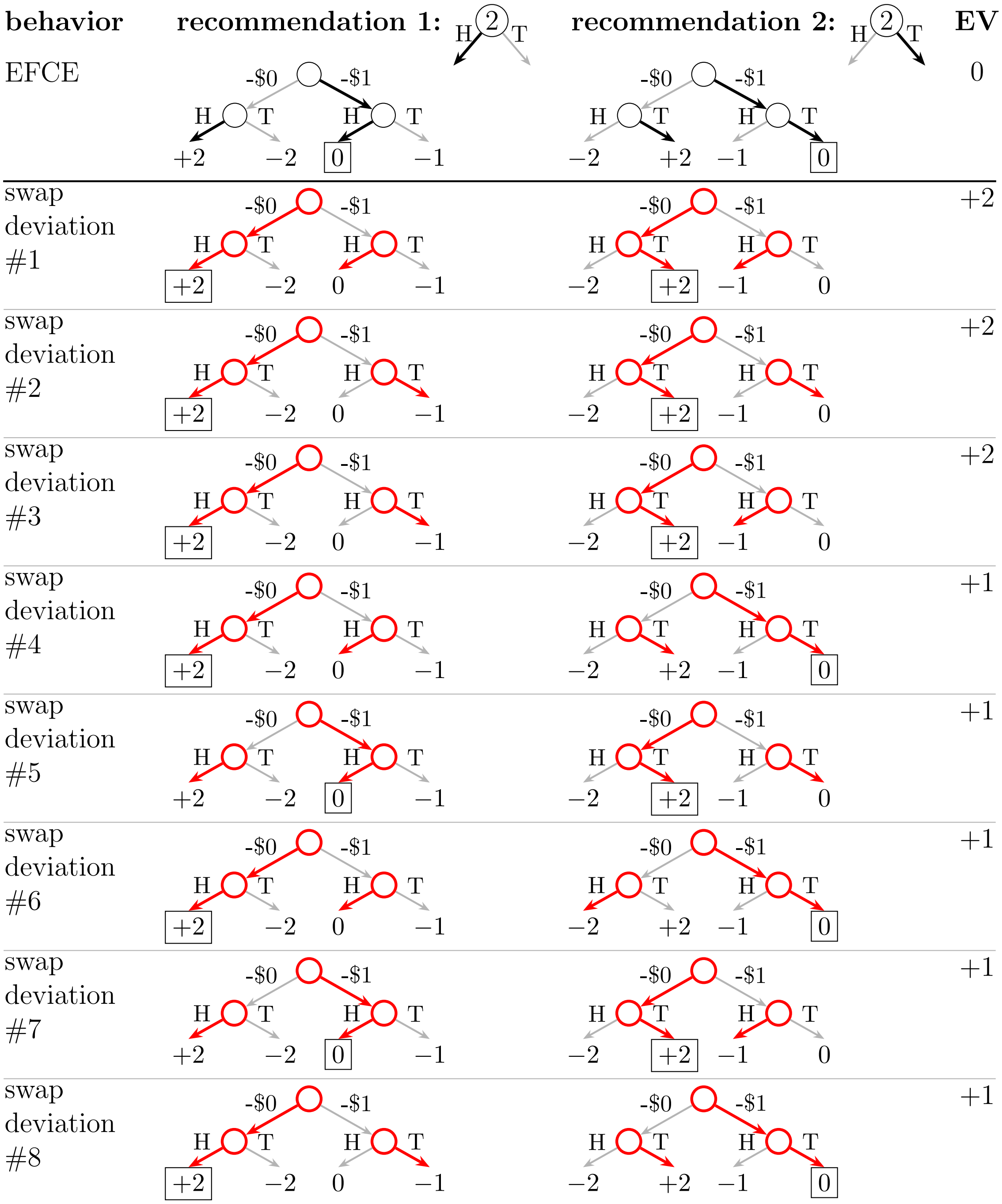


Figure 6.D.14: A gambling matching pennies example where an EFCE is not a CFCCE. Only eight of the highest value swap deviations are shown for brevity.

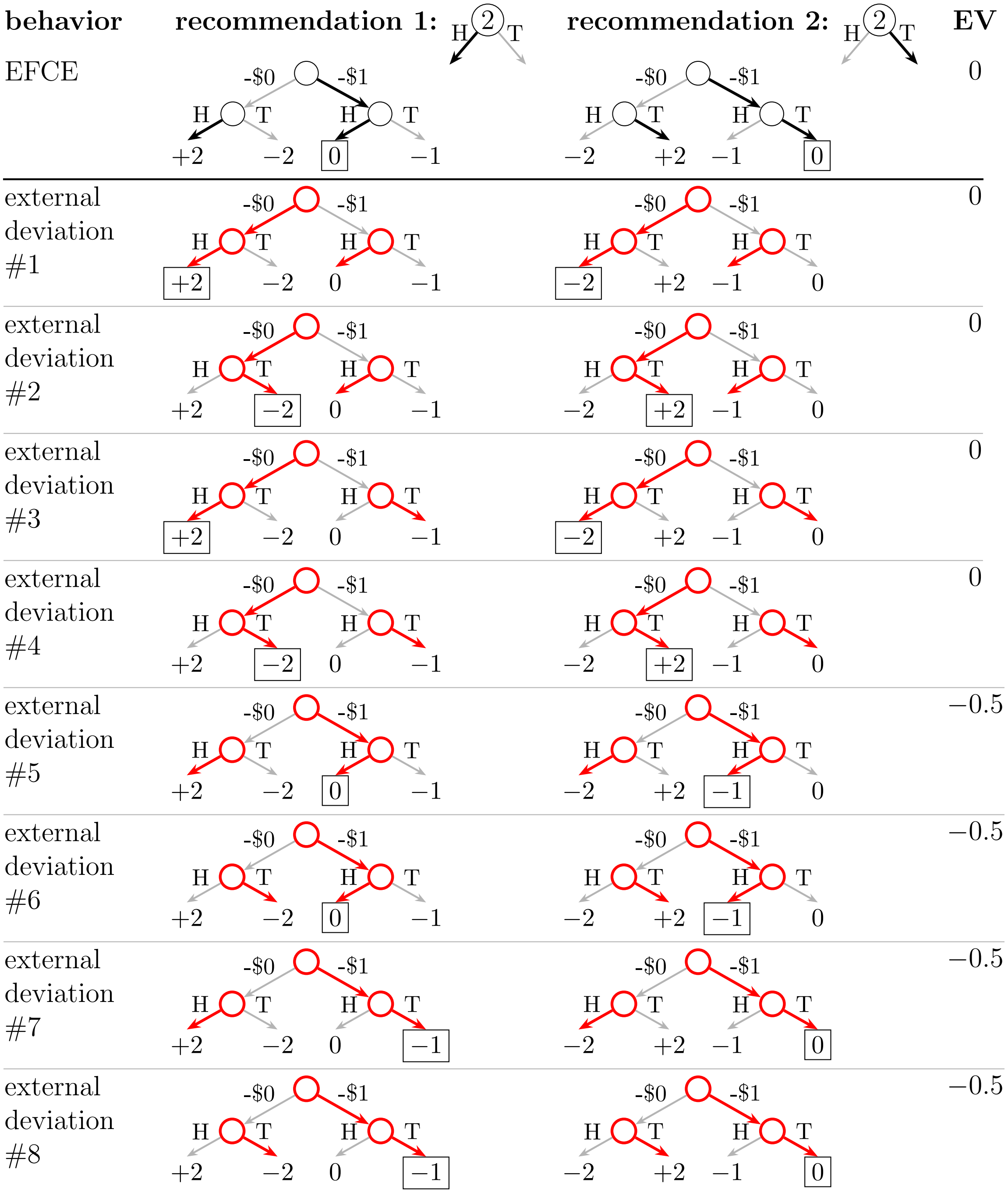


Figure 6.D.14 (Cont.): All external deviations in the gambling matching pennies example of an EFCE that is not a CFCCE.

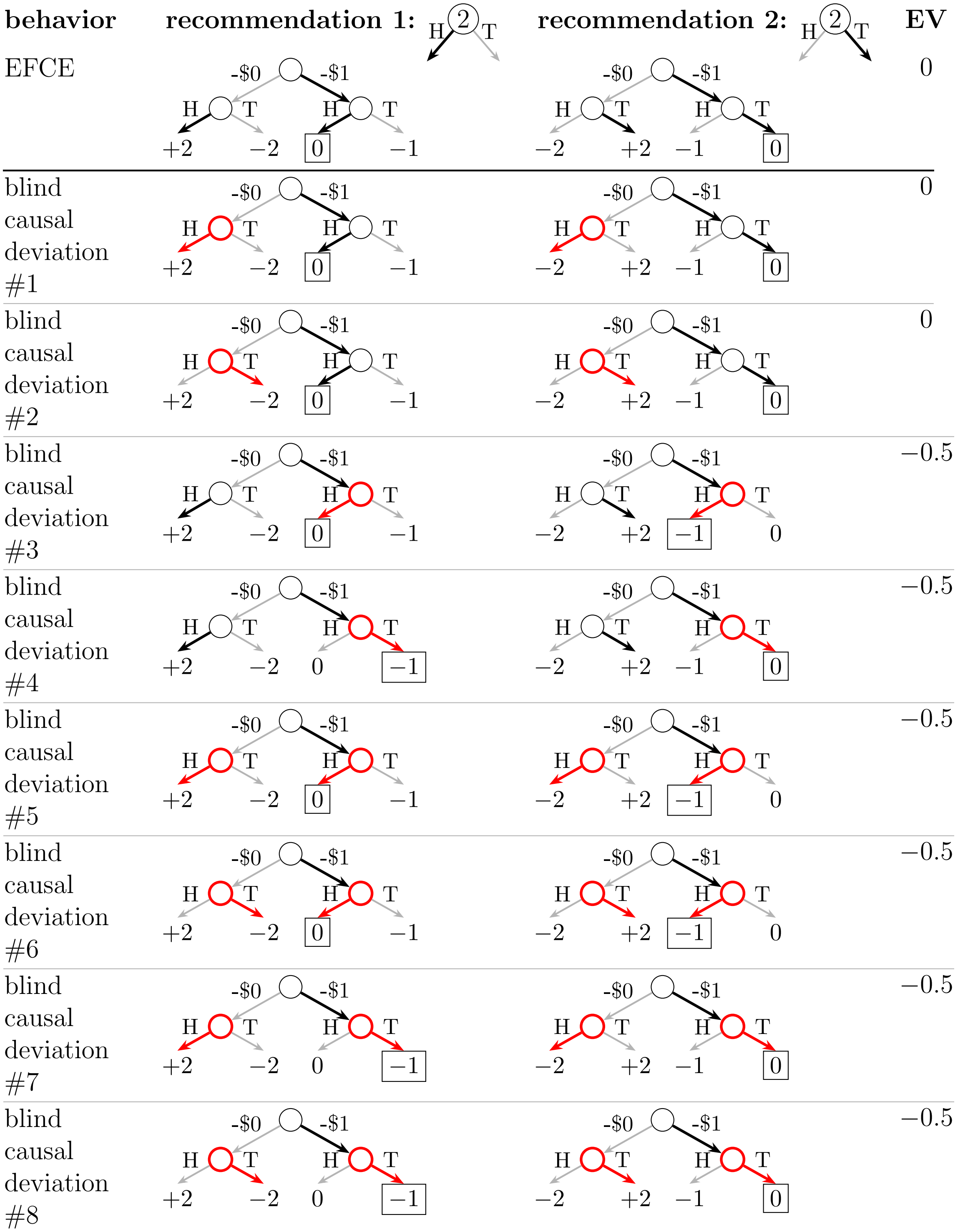

behavior
recommendation 1:
recommendation 2:
EV
EFCE
0
blind causal deviation #1
0
blind causal deviation #2
0
blind causal deviation #3
−0.5
blind causal deviation #4
−0.5
blind causal deviation #5
−0.5
blind causal deviation #6
−0.5
blind causal deviation #7
−0.5
blind causal deviation #8
−0.5
2
H
T
-$0
-$1
+2
−2
0
−1
−2
+2
−1
0

Figure 6.D.14 (Cont.): All non-external blind causal deviations in the gambling matching pennies example of an EFCE that is not a CFCCE.

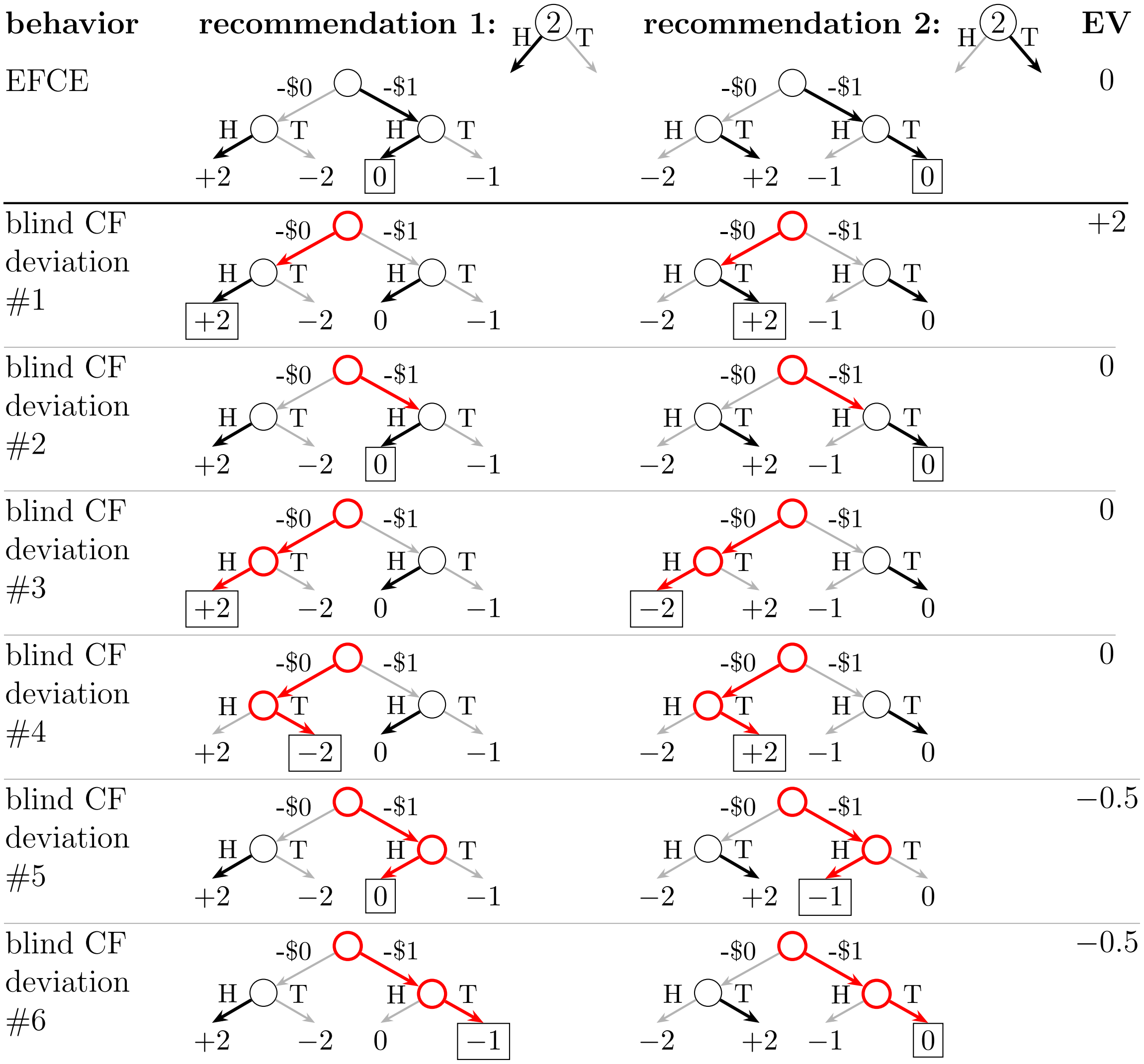


Figure 6.D.14 (Cont.): All blind counterfactual deviations in the gambling matching pennies example of an EFCE that is not a CFCCE.

6.E Example #6: CE That Is Not an Observable Sequential CCE

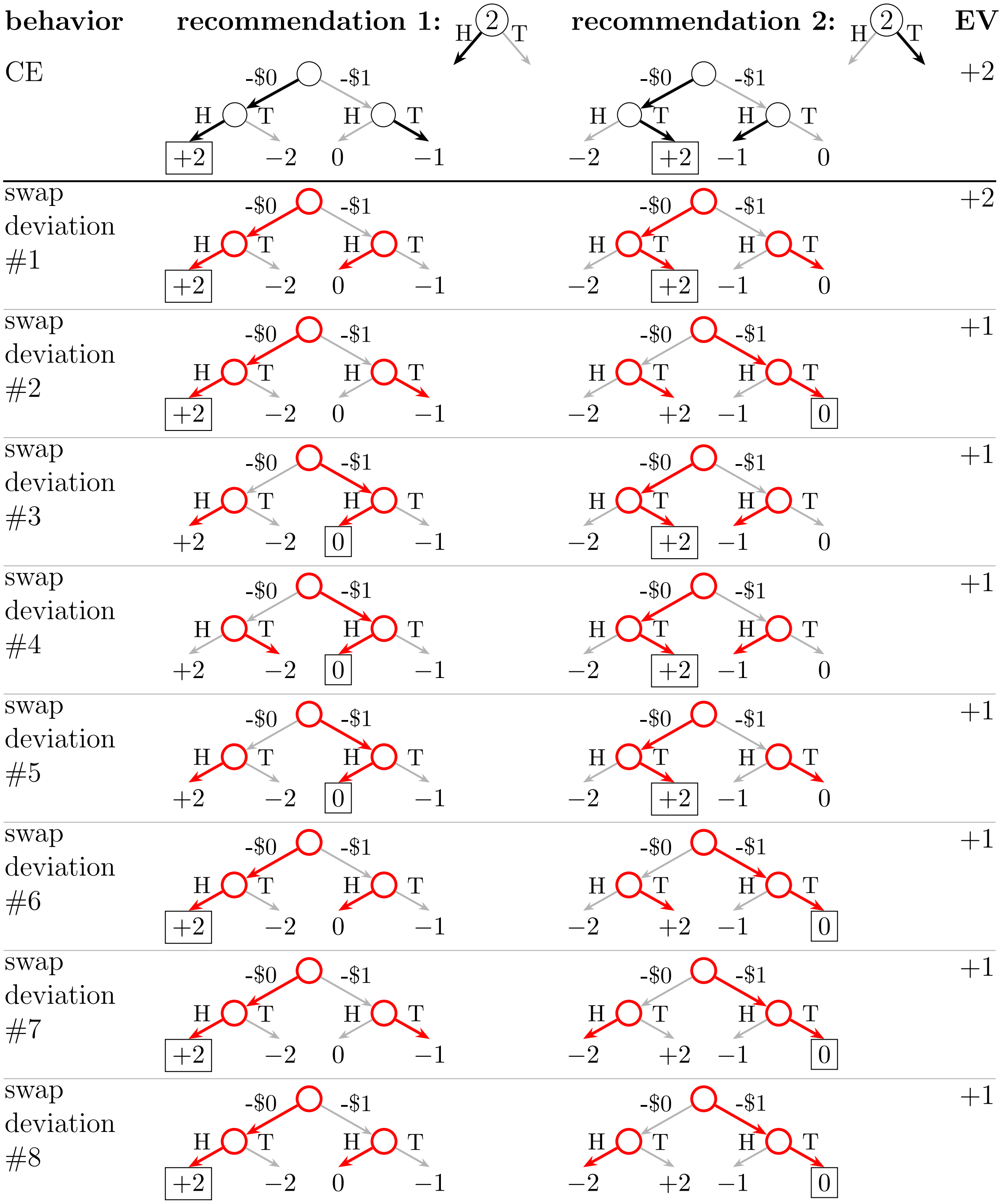


Figure 6.E.15: A gambling matching pennies example where a CE is not an observable sequential CCE. Only eight of the highest value swap deviations are shown for brevity.

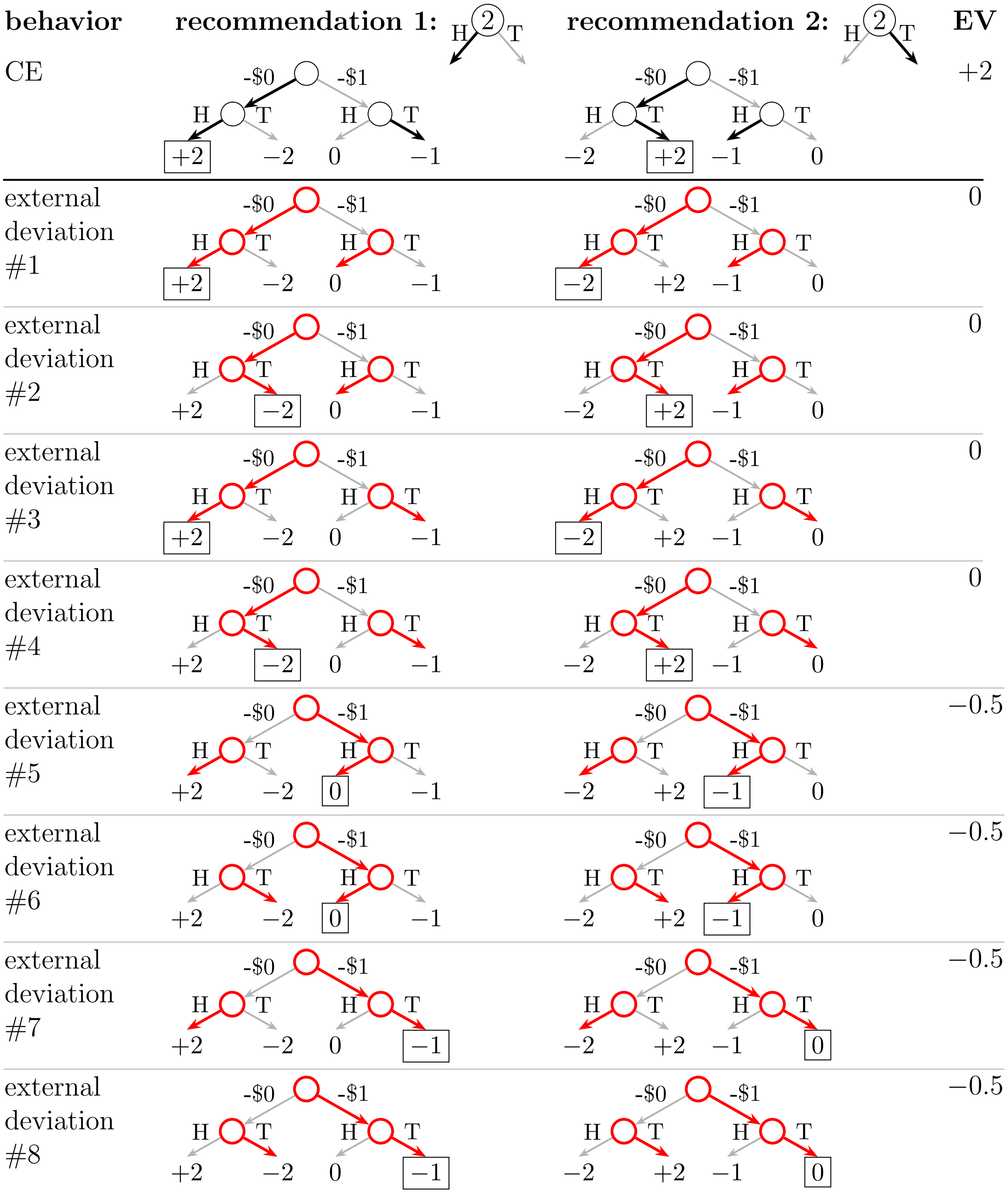


Figure 6.E.15 (Cont.): All external deviations in the gambling matching pennies example of a CE is not an observable sequential CCE.

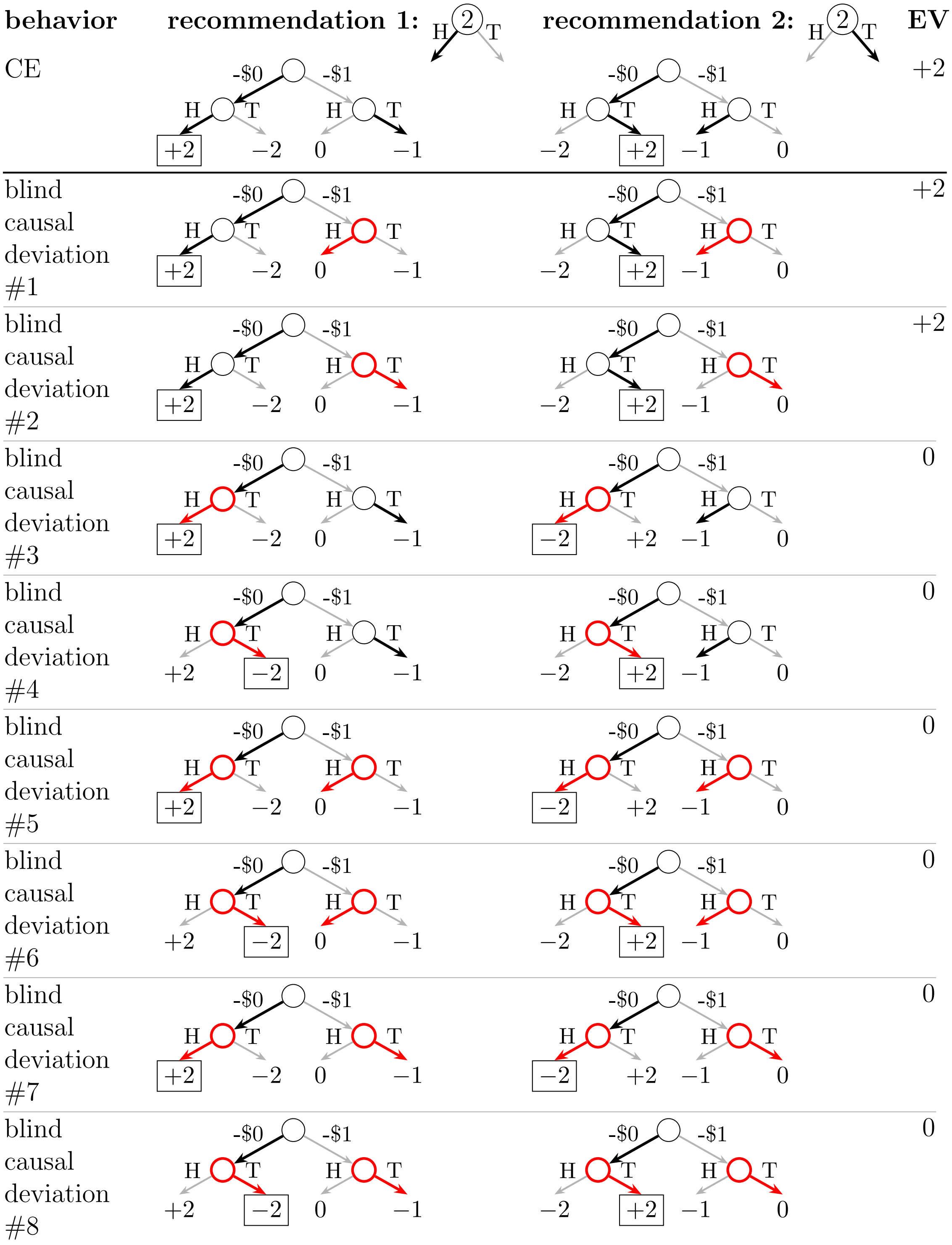

behavior
recommendation 1:
recommendation 2:
EV
H
2
T
CE
+2
blind causal deviation #1
+2
blind causal deviation #2
+2
blind causal deviation #3
0
blind causal deviation #4
0
blind causal deviation #5
0
blind causal deviation #6
0
blind causal deviation #7
0
blind causal deviation #8
0
-$0
-$1
H
T
+2
−2
0
−1
−2
+2
−1
0

Figure 6.E.15 (Cont.): All non-external blind causal deviations in the gambling matching pennies example of an CE that is not a observable sequential CCE.

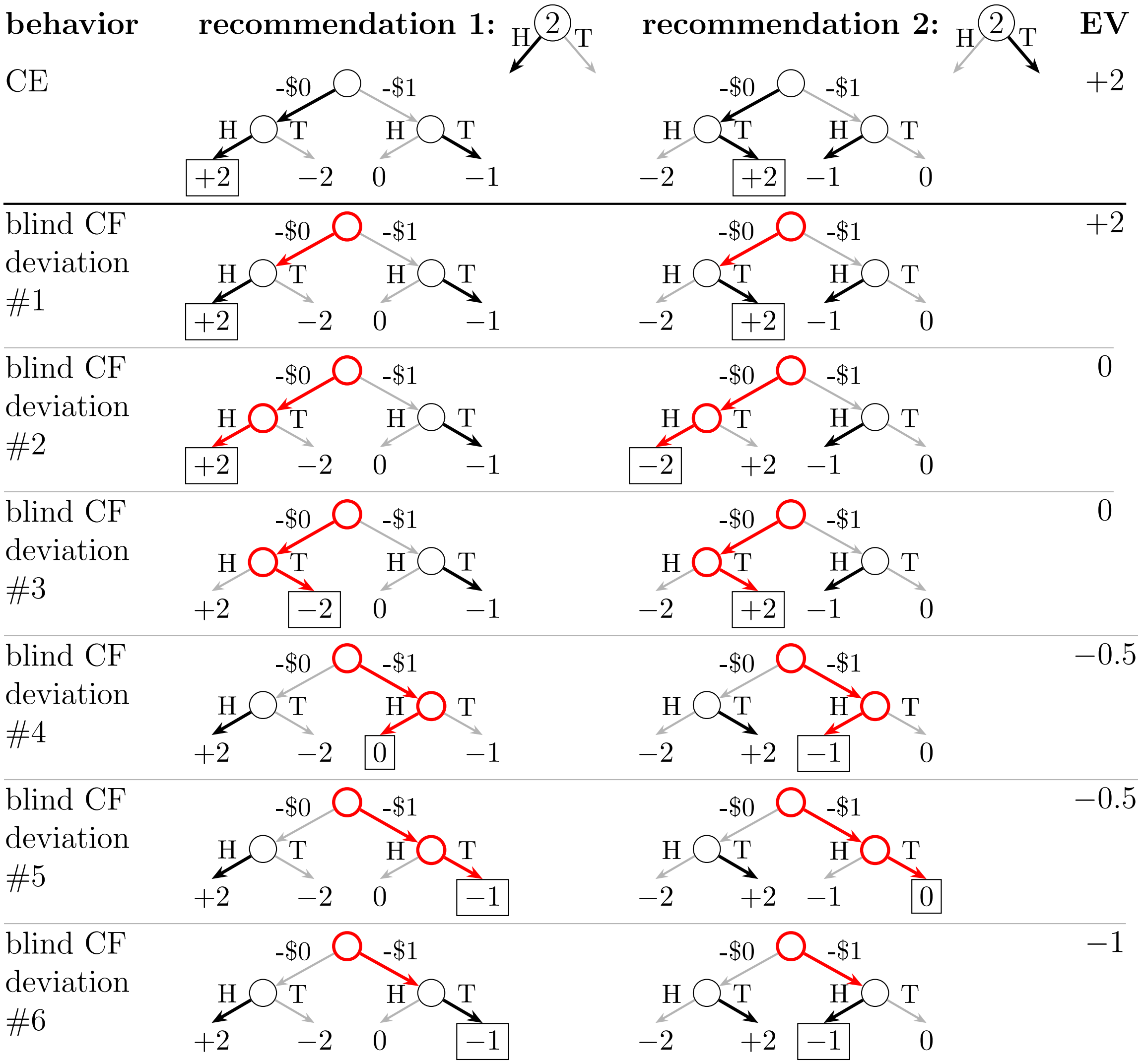


Figure 6.E.15 (Cont.): All blind counterfactual deviations in the gambling matching pennies example of an CE that is not a observable sequential CCE.

Chapter 7

Extensive-Form Regret Minimization

7.1 Introduction

This chapter develops the *extensive-form regret minimization* (*EFR*), a general and extensible algorithm that is observably sequentially (OS) hindsight rational for any given set of behavioral deviations. Its computational requirements and regret bound scale closely with the number of realizable memory states (see Table 6.3).

The key insight that leads to EFR is that each of the deviation player's memory states at a given active agent state s corresponds to a different weighting scheme of the rewards accumulated from s and its successors, and the weights of this scheme change on every round. Weighted regrets can be accumulated for each memory state to summarize the incentives that each deviation has to play through s. If immediate strategies are chosen so that all of these incentives decrease on average, then the total benefit for any deviation also decreases on average, which is the OS hindsight rationality condition. In this way, we reduce the problem of achieving OS hindsight rationality to minimizing immediate regret across each memory state and active agent state simultaneously, which we further reduce to time selection regret minimization (Section 2.3.4).

7.2 Immediate Regret Minimization for Behavioral Deviations

Section 4.5 shows that the growth of cumulative immediate and full regret can be controlled, in principle. However, if the deviation set Φ is insufficiently constrained, this procedure may be intractable because of circular dependencies between immediate strategies at different agent states or because the number of time selection functions grows exponentially with the number of agent states. This challenge motivates the development of an efficient procedure for competing with behavioral deviations specifically. The behavioral deviations are the ideal target for designing a general hindsight rationality algorithm in POHPs because they are just restrictive enough to rule out the possibility of circular dependencies between immediate strategies at different agent states.[1]

For behavioral deviations, we can generate time selection functions that correspond to deviations and memory states. Each time selection function captures the joint probability of reaching an agent state with a particular memory state.

Imagine that a POHP agent declares a behavioral strategy π but a deviation player, executing behavioral deviation ϕ, actually plays actions for the agent. At each agent state s, the agent samples an action A from $\pi(s)$ and the deviation player transforms A into another action A' and plays A' on the agent's behalf. The action transformation that the deviation player uses, $\phi_{s,\lambda}$, depends on their memory state, λ, and s. If $\phi_{s,\lambda}$ is external, *i.e.*, $\phi_{s,\lambda}(a) = a'$ for some $a' \in \mathcal{A}(s)$ and all $a \in \mathcal{A}(s)$, then the deviation player appends a "*" character to their memory state and otherwise appends A. Formally, the deviation player partially observes the action through the function

$$\omega^{\text{DEV}} : a; \phi_{s,\lambda} \mapsto \begin{cases} * & \text{if } \exists a', \forall \bar{a} \phi_{s,\lambda}(\bar{a}) = a' \\ a & \text{o.w.} \end{cases}$$

so that the next memory state is $\Lambda' = \lambda \omega^{\text{DEV}}(A; \phi_{s,\lambda})$.

Since action transformations are deterministic, the probability of a deviation player observation b given an action a, memory state λ, and agent state s is $\mathbb{P}_{\phi,\pi}[b \,|\, a, \lambda, s] = \mathbb{1}\{b = \omega^{\text{DEV}}(a; \phi_{s,\lambda})\}$. The product $\mathbb{P}_{\phi,\pi}[b, a \,|\, \lambda, s] = \mathbb{1}\{b = \omega^{\text{DEV}}(a; \phi_{s,\lambda})\}\pi(a \,|\, s)$ is the corresponding joint probability. By the chain rule of probability, $\mathbb{P}_{\phi,\pi}[\lambda b, s, a] = \mathbb{P}_{\phi,\pi}[b, a \,|\, \lambda, s]\mathbb{P}_{\phi,\pi}[\lambda, s]$.

Under perfect recall, the joint probability of λ and s where $|\lambda| = |\eta_{\mathcal{A}}(s)|$ is the product of the joint probabilities along the path to s, *i.e.*, $\mathbb{P}_{\phi,\pi}[\lambda, s] = \prod_{i=1}^{|\lambda|} \mathbb{P}_{\phi,\pi}[\lambda_i, \eta_{\mathcal{A}}(s)_i \,|\, \lambda_{1:i-1}, \eta_{\mathcal{H}_{\mathcal{A}}}(s)_i]$, where $\lambda_{1:0} = \varnothing$. $\mathbb{P}_{\phi,\pi}[\lambda, s]$ is the *memory probability* of λ at s. Under pure strategies, the memory probability expresses memory state realizability. We overload

$$\mathcal{G}_{\phi}(s) = \{\lambda \in \mathcal{G} \mid \exists x \in \mathcal{X}, \mathbb{P}_{\phi,x}[\lambda, s] = 1\}$$

as the set of memory states that ϕ can realize at s and $\mathcal{G}_{\Phi}(s) = \bigcup_{\phi \in \Phi} \mathcal{G}_{\phi}(s)$ is the set of all memory states that all deviations in $\Phi \subseteq \Phi^{\text{SW}}_{\mathcal{S}_{\mathcal{A}}}$.

Conditioned on memory state λ, we can define the counterfactual value that behavioral deviation ϕ achieves from agent state s as

$$v^{\text{CF}}_{s,\lambda}(\phi(\pi); \sigma) = \sum_{h \in I(s)} \mathbb{P}_{\sigma}[h]\mathbb{E}[G_h(\phi_{s\preceq,\lambda\sqsubseteq}(\pi); \sigma)],$$

[1] There are non-behavioral deviations that do not induce such circular dependencies, but they rely on arbitrary asymmetries. The action transformation at agent state s can depend on play at $s' \not\preceq s$ but the action transformation at s' cannot depend on play at s or that at any other agent state $\bar{s}$ that informs the action transformation at s.

where $\phi_{s\preceq,\lambda\sqsubseteq}$ is the deviation that applies ϕ only at s and its successors with memory states that are prefixed by λ. This deviation's counterfactual value across memory states is naturally the expected memory-state specific counterfactual value $v_s^{\text{CF}}(\phi(\pi);\sigma) = \mathbb{E}_{\Lambda\sim\mathbb{P}_{\phi,\pi}[\cdot\,|\,s]}\big[v_{s,\Lambda}^{\text{CF}}(\phi(\pi);\sigma)\big]$, where $\mathbb{P}_{\phi,\pi}[\cdot\,|\,s] : \lambda \mapsto \mathbb{P}_{\phi,\pi}[\lambda, s]/\mathbb{P}_{\phi(\pi)}[s]$ is the conditional distribution over memory states. The counterfactual regret is likewise the expected memory-specific counterfactual regret

$$\rho_s^{\text{CF}}(\phi,\pi;\sigma) = \mathbb{E}_{\Lambda\sim\mathbb{P}_{\phi,\pi}[\cdot\,|\,s]}\underbrace{\left[v_{s,\Lambda}^{\text{CF}}(\phi(\pi);\sigma) - v_{s,\Lambda}^{\text{CF}}(\pi;\sigma)\right]}_{\doteq\,\rho_{s,\Lambda}^{\text{CF}}(\phi,\pi;\sigma)}.$$

The realization weighted expected return of behavioral deviation ϕ is therefore the sum of realization weighted conditional returns across memory states. That is, starting from Eq. (4.2),

$$\begin{aligned}
v_s(\phi(\pi);\sigma) &= \mathbb{P}_{\phi(\pi)}[s]v_s^{\text{CF}}(\phi(\pi);\sigma)\\
&= \mathbb{P}_{\phi(\pi)}[s]\sum_{\lambda\in\mathcal{G}_\phi(s)}\mathbb{P}_{\phi,\pi}[\lambda\,|\,s]v_{s,\lambda}^{\text{CF}}(\phi(\pi);\sigma)\\
&= \sum_{\lambda\in\mathcal{G}_\phi(s)}\underbrace{\mathbb{P}_{\phi,\pi}[\lambda, s]v_{s,\lambda}^{\text{CF}}(\phi(\pi);\sigma).}_{\doteq\, v_{s,\lambda}(\phi(\pi);\sigma).}
\end{aligned}$$

Consequently, full regret can also be written as the sum of memory-state-specific full regrets

$$\begin{aligned}
\rho_s(\phi,\pi;\sigma) &= v_s(\phi(\pi);\sigma) - v_s(\pi;\sigma) &\quad (7.1)\\
&= \sum_{\lambda\in\mathcal{G}_\phi(s)}\underbrace{v_{s,\lambda}(\phi(\pi);\sigma) - v_{s,\lambda}(\phi_{\prec s,\sqsubset\lambda}(\pi);\sigma)}_{\doteq\,\rho_{s,\lambda}(\phi,\pi;\sigma)}. &\quad (7.2)
\end{aligned}$$

Furthermore, immediate regret is a special case of full regret so we can write

$$\rho_s(\phi_{\preceq s},\pi;\sigma) = \sum_{\lambda\in\mathcal{G}_\phi(s)}\rho_{s,\lambda}(\phi_{\preceq s,\sqsubseteq\lambda},\pi;\sigma). \quad (7.3)$$

Can we use Eq. (7.3) to design an algorithm for minimizing immediate regret with respect to a strictly truncated behavioral deviation $\phi_{\prec s}$? Yes, by reducing the problem to time selection regret minimization. If we define time selection functions with memory probabilities as $\mathcal{W}_s(\phi) = \{w_\lambda : t \mapsto \mathbb{P}_{\phi,\pi^t}[\lambda, s]\}_{\lambda\in\mathcal{G}_\phi(s)} = \mathcal{W}_s(\phi_{\prec s})$ where π^t is the strategy that the agent plays on round t, then instantaneous immediate regret can be written as

$$\begin{aligned}
\rho_s(\phi_{\preceq s}, \pi^t; \sigma^t) &= \sum_{\lambda \in \mathcal{G}_{\phi_{\prec s}}(s)} \mathbb{P}_{\phi, \pi^t}[\lambda, s]\big(v^{\text{CF}}_{s,\lambda}(\phi_{\preceq s, \sqsubseteq \lambda}(\pi^t); \sigma^t) - v^{\text{CF}}_{s,\lambda}(\pi^t; \sigma^t)\big) && (7.4)\\
&= \sum_{\substack{w_\lambda \in \mathcal{W}_s(\phi_{\prec s}),\\ \phi'_s \in \Phi_{\mathcal{A}(s)}}} \mathbb{1}\{\phi'_s = \phi_{s,\lambda}\} w^t_\lambda \big(v^{\text{CF}}_{s,\lambda}(\phi'_s(\pi^t); \sigma^t) - v^{\text{CF}}_{s,\lambda}(\pi^t; \sigma^t)\big) && (7.5)\\
&= \sum_{\substack{w_\lambda \in \mathcal{W}_s(\phi_{\prec s}),\\ \phi'_s \in \Phi_{\mathcal{A}(s)}}} \mathbb{1}\{\phi'_s = \phi_{s,\lambda}\} w^t_\lambda \rho^{\text{CF}}_{s,\lambda}(\phi'_s, \pi^t; \sigma^t). && (7.6)
\end{aligned}$$

Since we do not know which set of action transformations will yield the best truncated behavioral deviation in hindsight, we must minimize weighted counterfactual regret with respect to all valid pairs of time selection functions and action transformations simultaneously. That is, consider splitting the best truncated deviation in hindsight similarly to Eq. (7.6) as

$$\begin{aligned}
\max_{\phi_{\preceq s} \in \Phi} \sum_{t=1}^{T} \rho_s(\phi_{\preceq s}, \pi^t; \sigma^t) &= \max_{\phi_{\prec s} \in \Phi} \sum_{t=1}^{T} \sum_{\substack{w_\lambda \in \mathcal{W}_s(\phi_{\prec s}),\\ \phi'_s \in \Phi_{\mathcal{A}(s)}}} \mathbb{1}\{\phi'_s = \phi_{s,\lambda}\} w^t_\lambda \rho^{\text{CF}}_{s,\lambda}(\phi'_s, \pi^t; \sigma^t) && (7.7)\\
&= \max_{\phi_{\prec s} \in \Phi} \sum_{w_\lambda \in \mathcal{W}_s(\phi_{\prec s})} \max_{\{\phi'_{s,\lambda}\}_{\phi' \in \Phi}} \sum_{t=1}^{T} w^t_\lambda \rho^{\text{CF}}_{s,\lambda}(\phi'_{s,\lambda}, \pi^t; \sigma^t) && (7.8)
\end{aligned}$$

for any set of behavioral deviations $\Phi \subseteq \Phi^{\text{SW}}_{\mathcal{S}_\mathcal{A}}$. Finally, if we overload $\mathcal{W}_s(\Phi) = \bigcup_{\phi \in \Phi} \mathcal{W}_s(\phi)$, $\Phi_{s,\lambda} = \{\phi_{s,\lambda}\}_{\phi \in \Phi}$, and $\Phi_s = \bigcup_{\lambda \in \mathcal{G}_\phi(s)} \Phi_{s,\lambda}$, then we can further bound

$$\begin{aligned}
&\max_{\phi_{\prec s} \in \Phi} \sum_{w_\lambda \in \mathcal{W}_s(\phi_{\prec s})} \max_{\{\phi'_{s,\lambda}\}_{\phi' \in \Phi}} \sum_{t=1}^{T} w^t_\lambda \rho^{\text{CF}}_{s,\lambda}(\phi'_{s,\lambda}, \pi^t; \sigma^t) && (7.9)\\
&= \max_{\phi_{\prec s} \in \Phi} \sum_{w_\lambda \in \mathcal{W}_s(\phi_{\prec s})} \max_{\phi'_s \in \Phi_s} \mathbb{1}\{\phi'_s \in \Phi_{s,\lambda}\} \sum_{t=1}^{T} w^t_\lambda \rho^{\text{CF}}_{s,\lambda}(\phi'_s, \pi^t; \sigma^t) && (7.10)\\
&\leq \sum_{w_\lambda \in \mathcal{W}_s(\Phi)} \max_{\phi_s \in \Phi_s} \mathbb{1}\{\phi_s \in \Phi_{s,\lambda}\} \left[\sum_{t=1}^{T} w^t_\lambda \rho^{\text{CF}}_{s,\lambda}(\phi_s, \pi^t; \sigma^t)\right]_+. && (7.11)
\end{aligned}$$

Eq. (7.11) completes the formal reduction to time selection regret minimization. This is a slightly more onerous condition than the basic OTSDP objective as performance is measured by the sum of positive regrets across time selection functions rather than the maximum regret on any single one. Given a bound on this maximum, we could of course bound the sum by multiplying by the maximum number of time selection functions. However, time selection regret matching, the algorithm developed in Section 7.4, directly bounds Eq. (7.11), leading to a sublinear dependence on the number of time selection functions associated with any single action transformation!

7.3 Extensive-Form Regret Minimization

The *extensive-form regret minimization* (*EFR*) algorithm takes a set of behavioral deviations, $\Phi \subseteq \Phi^{\text{sw}}_{\mathcal{S}_{\mathcal{A}}}$, as an argument, and chooses its immediate strategy at each agent state on each round so as to minimize Eq. (7.11) with respect to consolidated action transformation sets $\{\Phi_s\}_{s \in \mathcal{S}_{\mathcal{A}}}$. As long as a sublinear bound on Eq. (7.11) is achieved with respect to the round number, then Theorem 5 implies a full-regret bound at each agent state and observable sequential rationality with respect to Φ.

Notice that as a matter of practical implementation, EFR only requires Φ_s, $\mathcal{W}_s(\Phi)$, and predicates $\{\mathbb{1}\{\phi_s \in \Phi_{s,\lambda}\}\}_{\phi_s \in \Phi_s,\, \lambda \in \mathcal{G}_{\Phi}(s)}$ connecting the two for all active agent states $s \in \mathcal{S}_{\mathcal{A}}$, which are often easier to specify than Φ itself. Table 7.1 shows how to set these parameters for a range of deviation types. In addition to implementation simplicity, this feature ensures that EFR always implicitly transforms deviations from its nominal deviation set, Φ, into single-target deviations that re-correlate. This both potentially improves EFR's performance and ensures that learning is efficient even for some exponentially large deviation sets, like the external, blind causal, and informed causal deviations.

For example, it is equivalent to instantiate EFR with the blind causal deviations or the BPS deviations. Likewise for the informed causal deviations and the CSPS deviations, where EFR reduces to a variation of ICFR (Celli et al. 2020). To be precise, ICFR is pure EFR (analogous to pure CFR) instantiated with the CSPS deviations except that the external and internal action transformation learners at separate memory states within an agent state are sampled and updated independently in ICFR. EFR therefore improves on this algorithm (beyond its generality) because EFR's action transformation learners share all experience, potentially leading to faster learning, and EFR enjoys a deterministic finite time regret bound.

Crucially, EFR's generality does not come at a computational cost. EFR reduces to the CFR algorithms previously described to handle counterfactual and action deviations (Morrill, D'Orazio, Sarfati, Lanctot, Wright, A. R. Greenwald, et al. 2021; Zinkevich, Johanson, et al. 2007b). Furthermore, EFR inherits CFR's flexibility as it can be used with Monte Carlo sampling (Burch, Lanctot, et al. 2012; Gibson et al. 2012; Johanson, Bard, Lanctot, et al. 2012; Lanctot, Waugh, et al. 2009), function approximation (Brown, Lerer, et al. 2019; D'Orazio 2020; D'Orazio, Morrill, et al. 2020; Morrill 2016; Steinberger et al. 2020; Waugh, Morrill, et al. 2015), variance reduction (Davis et al. 2020; Schmid, Burch, et al. 2019), and predictions (D'Orazio and R. Huang 2021; Farina, Kroer, Brown, et al. 2019; Farina, Kroer, and Sandholm 2021; Rakhlin et al. 2013).

Table 7.1: EFR parameters and regret bound constants for different deviation types. We use $\Phi^{\text{IN}\setminus 1}_{\mathcal{A}(s)} = \Phi^{\text{IN}}_{\mathcal{A}(s)} \setminus \{\phi^1\}$ to denote the non-identity internal transformations.

type	Φ_s for each $s \in \mathcal{S}_\mathcal{A}$	$\mathcal{W}_s(\Phi)$ for each $s \in \mathcal{S}_\mathcal{A}$	$\mathbb{1}\{\phi_s \in \Phi_{s,\lambda}\}$ for each $\phi_s \in \Phi_s,\, \lambda \in \mathcal{G}_\Phi(s)$	$\alpha(\Phi)$
BHV	$\Phi^{\text{IN}\setminus 1}_{\mathcal{A}(s)}$	$\{t \mapsto 1\} \cup$ $\left\{t \mapsto \prod_{\bar{s} \preceq \bar{s}'} \pi^t(a_{\bar{s}} \mid \bar{s})\right\}_{\bar{s}' \prec s,\, \forall \bar{s} \preceq \bar{s}', a_{\bar{s}} \in \mathcal{A}(\bar{s})}$	1	$n_\mathcal{A}^{d_*}(n_\mathcal{A}^2 - n_\mathcal{A})$
TIPS	$\Phi^{\text{IN}\setminus 1}_{\mathcal{A}(s)}$	$\{t \mapsto 1\} \cup$ $\left\{t \mapsto \mathbb{P}_{\pi^t}[s^!]\pi^t(a^! \mid s^!)\right\}_{s^! \prec s, a^! \in \mathcal{A}(s^!)}$	1	$(d_* n_\mathcal{A} + 1)$ $(n_\mathcal{A}^2 - n_\mathcal{A})$
CSPS / in. causal	$\Phi^{\text{EX}}_{\mathcal{A}(s)} \cup \Phi^{\text{IN}\setminus 1}_{\mathcal{A}(s)}$	$\{t \mapsto 1\} \cup$ $\left\{t \mapsto \mathbb{P}_{\pi^t}[s^!]\pi^t(a^! \mid s^!)\right\}_{s^! \prec s, a^! \in \mathcal{A}(s^!)}$	$\mathbb{1}\left\{\begin{matrix}\phi_s \in \Phi^{\text{IN}}_{\mathcal{A}(s)}, \\ \lambda = \eta_\mathcal{A}(s)\end{matrix}\right\}$ $+\mathbb{1}\left\{\begin{matrix}\phi_s \in \Phi^{\text{EX}}_{\mathcal{A}(s)}, \\ \lambda \neq \eta_\mathcal{A}(s)\end{matrix}\right\}$	$d_* n_\mathcal{A}(n_\mathcal{A}^2 - 2)$
CFPS	$\Phi^{\text{IN}\setminus 1}_{\mathcal{A}(s)}$	$\{t \mapsto 1\} \cup$ $\left\{t \mapsto \mathbb{P}_{\pi^t}[s^!]\right\}_{s^! \preceq s}$	1	$(d_* + 1)$ $(n_\mathcal{A}^2 - n_\mathcal{A})$
BPS / blind causal	$\Phi^{\text{EX}}_{\mathcal{A}(s)}$	$\{t \mapsto 1\} \cup$ $\left\{t \mapsto \mathbb{P}_{\pi^t}[s^!]\right\}_{s^! \preceq s}$	1	$(d_* + 1)$ $(n_\mathcal{A} - 1)$
in. action	$\Phi^{\text{IN}\setminus 1}_{\mathcal{A}(s)}$	$\{t \mapsto \mathbb{P}_{\pi^t}[s]\}$	1	$n_\mathcal{A}^2 - n_\mathcal{A}$
in. CF	$\Phi^{\text{IN}\setminus 1}_{\mathcal{A}(s)}$	$\{t \mapsto 1\}$	1	$n_\mathcal{A}^2 - n_\mathcal{A}$
blind action	$\Phi^{\text{EX}}_{\mathcal{A}(s)}$	$\{t \mapsto \mathbb{P}_{\pi^t}[s]\}$	1	$n_\mathcal{A} - 1$
blind CF / external	$\Phi^{\text{EX}}_{\mathcal{A}(s)}$	$\{t \mapsto 1\}$	1	$n_\mathcal{A} - 1$

7.4 Time Selection Regret Matching

We now consider a time selection regret minimization algorithm for EFR.

7.4.1 A Failed Attempt: Regret Matching++

Kash et al. (2020) presents the regret matching++ algorithm and claims that it is no-external-regret. This algorithm's proposed regret bound implies a sublinear bound on cumulative positive regret, which would further imply that it has the same bound with respect to *all possible* time selection functions. The surprising aspect of this result is that the algorithm does not require any information about any of the possible time selection functions and requires no more computation or storage than basic regret matching. However, here we show that there is actually no algorithm that can achieve a sublinear bound on cumulative positive regret. This result proves that regret matching++ cannot be no-external-regret as claimed.

Theorem 14. *The worst-case maximum cumulative positive regret over T rounds,*

$$Q^T = \max_{x \in \mathcal{X}} \sum_{t=1}^{T} \big[\upsilon^t(x) - \upsilon^t(\pi^t) \big]_+,$$

of any algorithm that chooses mixed strategy $\pi^t \in \Delta(\mathcal{X})$ in an ODP where payoffs are in $[0, 1]$, is at least $T/4$.

Proof. Without loss of generality, consider a two pure strategy environment, $\mathcal{X} = (x, x')$, and any learning algorithm that deterministically chooses a distribution, $\pi^t \in \Delta(\mathcal{X})$, over them on each round t. The environment gets to see the agent's strategy before presenting a utility function. If the agent weights one pure strategy more than the other, the environment gives a payoff of zero for the pure strategy with the larger weight and one to the pure strategy with the smaller weight. Formally, if $\pi^t(x) \geq 0.5$, then $\upsilon^t(x) = 0$, $\upsilon^t(x') = 1$, and vice-versa otherwise.

Let $x_{\text{low}} = x'$ if $\pi^t(x) \geq 0.5$ and $x_{\text{low}} = x$ otherwise. The positive regrets on any round t are $[1 - \pi^t(x_{\text{low}})]_+ \geq 0.5$ and $[0 - (1 - \pi^t(x_{\text{low}}))]_+ = 0$. So the agent is forced to suffer at least 0.5 positive regret on each round for one of the pure strategies. Since there are only two pure strategies, then over T rounds one of the strategies must have accumulated a regret of 0.5 on at least $T/2$ rounds. The cumulative positive regret for this pure strategy must then be $T/4$. Therefore, the maximum cumulative positive regret of any deterministic algorithm in this environment must be at least $T/4$.

To extend this result to include algorithms that stochastically choose π^t, we simply need to consider the expected cumulative positive regret and notice that the ramp function is convex. By Jensen's inequality and the fact that the max of an expectation is no larger than the expectation of the max, the expected cumulative positive regret is lower bounded by the cumulative positive regret under the agent's expected distributions, $\mathbb{E}[\pi^t]$, *i.e.*, $\mathbb{E}[Q^T] \geq \max_{x \in \mathcal{X}} \sum_{t=1}^{T} [\upsilon^t(x) - \upsilon^t(\mathbb{E}[\pi^t])]_+ \geq T/4$. Since $\mathbb{E}[\pi^t]$ is a mixed strategy, we have reduced the stochastic case to the deterministic case, thereby showing they have the same regret lower-bound. □

I now identify the mistake in the regret matching++ external regret bound proof. Define the cumulative positive regret of pure strategy $x \in \mathcal{X}$ after T rounds as $Q_x^T = \sum_{t=1}^{T} [\rho(\phi^{\to x}, \pi^t; \upsilon^t)]_+$. Kash et al. (2020) bounds

$$(\max_{x \in \mathcal{X}} Q_x^T)^2 \leq \sum_x (Q_x^T)^2 = \sum_{x \in \mathcal{X}} \big(Q_x^{T-1} + [\rho(\phi^{\to x}, \pi^t; \upsilon^t)]_+\big)^2.$$

They then state that

$$\big(Q_x^{T-1} + [\rho(\phi^{\to x}, \pi^T; \upsilon^T)]_+\big)^2 \leq \big(Q_x^{T-1} + \rho(\phi^{\to x}, \pi^T; \upsilon^T)\big)^2 + (2U)^2,$$

where U is the maximum payoff magnitude. This is false in general:

$$\big(Q_x^{T-1} + [\rho(\phi^{\to x}, \pi^T; \upsilon^T)]_+\big)^2 \tag{7.12}$$

$$= \big(Q_x^{T-1}\big)^2 + \big(\rho(\phi^{\to x}, \pi^T; \upsilon^T)\big)^2 + 2Q_x^{T-1}[\rho(\phi^{\to x}, \pi^T; \upsilon^T)]_+ \tag{7.13}$$

$$\leq \big(Q_x^{T-1}\big)^2 + \big(\rho(\phi^{\to x}, \pi^T; \upsilon^T)\big)^2 + 2Q_x^{T-1}\rho(\phi^{\to x}, \pi^T; \upsilon^T) + 2Q_x^{T-1}(2U) \tag{7.14}$$

$$= \big(Q_x^{T-1} + \rho(\phi^{\to x}, \pi^T; \upsilon^T)\big)^2 + 2Q_x^{T-1}(2U), \tag{7.15}$$

where $2Q_x^{T-1}(2U) > (2U)^2$ if $Q_x^{T-1} > (2U)/2$. There are scenarios where Eq. (7.15) is tight so it is unclear how this bound could be improved. Attempting the rest of the proof, we get

$$\sum_x \big(Q_x^{T-1} + [\rho(\phi^{\to x}, \pi^T; \upsilon^T)]_+\big)^2 \leq |\mathcal{X}|(2U)^2 + \sum_x \big(Q_x^{T-1}\big)^2 + 2(2U) \sum_x Q_x^{T-1}.$$

Unrolling the recursion exactly is messy, but the extra $2(2U) \sum_x Q_x^{T-1}$ term ensures that the bound will be no smaller than $\sum_x Q_x^{T-1} + [\rho(\phi^{\to x}, \pi^T; \upsilon^T)]_+ \leq 2^{\frac{T}{2}} |\mathcal{X}|^{\frac{T-1}{2}} (2U)^T$.

7.4.2 Time Selection Regret Matching

To give EFR the same implementation flexibility as CFR (which we will see is a special case of EFR) we develop regret matching for time selection in full generality. Namely, our time selection regret matching algorithm allows us to use a link function that leads to hyperparameter-free learning, and it allows regret approximations and predictions.

This section defines time selection functions in an OTSDP for each deviation individually, *i.e.*, $\mathcal{W}(\phi)$ is the finite set of time selection functions for deviation ϕ. I overload $\mathcal{W} = \bigcup_{\phi \in \Phi} \mathcal{W}(\phi)$ and work with $|\Phi| \times |\mathcal{W}|$ matrices where entries corresponding to incompatible

(ϕ, w)-pairs ($i.e.$, $w \notin \mathcal{W}(\phi)$) are always zero. I refer to instantaneous regrets with the matrix $\rho^t \in \mathbb{R}^{|\Phi| \times |\mathcal{W}|}$ where $\rho^t_{\phi,w} = w^t \rho(\phi, \pi^t; \sigma^t)$, cumulative regrets with $\rho^{1:T} = \sum_{t=1}^T \rho^t$, and regret matching$^+$ pseudo regrets with $q^{1:T} = [q^{1:T-1} + \rho^T]_+$ where $q^{1:0} = \mathbf{0}$. An algorithm is no-regret for all time selection functions in $\mathcal{W}$ as long as every entry of $\rho^{1:T}$ grows sublinearly with T.

The first step in the usual regret matching procedure is to construct non-negative preferences for each deviation by applying a link function to each cumulative regret or pseudo regret. This is actually the only step we need to modify to generalize regret matching to the time selection setting. The preferences in our algorithm are constructed by passing each cumulative weighted regret or pseudo regret through the link function as usual, but now the link outputs are weighted by time selection weights on the current round and summed across the time selection function dimension. From here, we apply the remainder of the regret matching procedure without modification. Treating each deviation as a matrix, we construct the average deviation matrix according to the normalized preferences and play a strategy that is a fixed point under this average deviation.

Our algorithm is a generalization of optimistic regret matching (D'Orazio and R. Huang 2021) that, after $t-1$ rounds, uses preferences $y^t_\phi = \sum_{w \in \mathcal{W}(\phi)} w^t f(x^t_{\phi,w} + m^t_{\phi,w})$, where either $x^t = \rho^{1:t-1}$ or $x^t = q^{1:t-1}$ for regret matching$^+$ with $x^1 = \mathbf{0}$, and m^t is a matrix of arbitrary predictions or approximation errors. For the rest of our analysis, we assume the ramp link function but the arguments involved in all proofs apply more generally to link functions that are subgradients of convex potential functions.[2] Only the final bounds would change. Notice that $x^t + m^t$ can be generated from a function approximator instead of storing either term in a table. Denoting the weighted sum of the preferences as $z^t = \sum_{\phi \in \Phi} y^t_\phi$ and representing each deviation as an $\mathcal{X} \times \mathcal{X}$ matrix, the average deviation is constructed as usual: $\bar{\phi}^t = \frac{1}{z^t} \sum_{\phi \in \Phi} y^t_\phi \phi$. Time selection regret matching chooses $\pi^t \in \Delta(\mathcal{X})$ to be a fixed point of the linear transformation $\bar{\phi}^t : \pi \mapsto \bar{\phi}^t \pi$, where π is represented as a $|\mathcal{X}|$-length column vector. If z^t is zero so that $\bar{\phi}^t$ is undefined, then there are no positive regrets and an arbitrary strategy can be played.

Note that if all deviations are external, then $\bar{\phi}^t$ is a matrix where each column is identical and forms a probability distribution. This distribution is a fixed point of $\bar{\phi}^t$, so the next strategy π^t can be chosen as the first column of $\bar{\phi}^t$ to avoid any extra computation.

[2] For regret matching$^+$ to be no-regret, the potential function must also be positive invariant.

7.4.3 Analysis

Theorem 15. *After T rounds, $(\mathcal{W}, \Phi, [\cdot]_+)$-optimistic regret matching or regret matching$^+$ ensures that*

$$\rho^{1:T}_{\phi,w} \leq \sqrt{\sum_{t=1}^{T} \sum_{\substack{\phi' \in \Phi, \\ \bar{w} \in \mathcal{W}(\phi')}} \left(\rho^t_{\phi',\bar{w}} - m^t_{\phi',\bar{w}}\right)^2}$$

for every deviation $\phi \in \Phi \subseteq \Phi^{\text{SW}}_{\mathcal{X}}$ and time selection function $w \in \mathcal{W}$.

Proof. Let $M_{\cdot,w} = [M_{\phi,w}]_{\phi \in \Phi}$ for any matrix $M \in \mathbb{R}^{|\Phi| \times |\mathcal{W}|}$ and time selection function $w \in \mathcal{W}$. The quadratic potential function, $G(\cdot) = \frac{1}{2}\|[\cdot]_+\|_2^2$ is convex, positive invariant (with equality), its gradient, $\nabla G(\cdot) = [\cdot]_+$, is the ramp function, and G is smooth with respect to $\gamma(\cdot) = \frac{1}{2}\|\cdot\|_2^2$. Altogether, these properties imply that

$$\begin{aligned}
&G\Big(\big[x^t_{\cdot,w} + \rho^t_{\cdot,w}\big]_+\Big) \\
&= G\Big(\big[x^t_{\cdot,w} + m^t_{\cdot,w} + \rho^t_{\cdot,w} - m^t_{\cdot,w}\big]_+\Big) && (7.16)\\
&= G\big(x^t_{\cdot,w} + m^t_{\cdot,w} + \rho^t_{\cdot,w} - m^t_{\cdot,w}\big) && (7.17)\\
&\leq G\big(x^t_{\cdot,w} + m^t_{\cdot,w}\big) + \langle \rho^t_{\cdot,w} - m^t_{\cdot,w},\ \big[x^t_{\cdot,w} + m^t_{\cdot,w}\big]_+\rangle + \gamma\big(\rho^t_{\cdot,w} - m^t_{\cdot,w}\big). && (7.18)
\end{aligned}$$

By convexity, $G(a) - G(b) \leq \langle \nabla G(a),\, a - b\rangle$, for any vectors a and b, so we substitute $a = x^t_{\cdot,w} + m^t_{\cdot,w}$ and $b = x^t_{\cdot,w}$ to bound

$$G(x^t_{\cdot,w} + m^t_{\cdot,w}) - \langle m^t_{\cdot,w},\, [x^t_{\cdot,w} + m^t_{\cdot,w}]_+\rangle \leq G(x^t_{\cdot,w}).$$

Therefore,

$$\begin{aligned}
&G\Big(\big[x^t_{\cdot,w} + \rho^t_{\cdot,w}\big]_+\Big) \\
&\leq G(x^t_{\cdot,w}) + \langle \rho^t_{\cdot,w},\ \big[x^t_{\cdot,w} + m^t_{\cdot,w}\big]_+\rangle + \gamma\big(\rho^t_{\cdot,w} - m^t_{\cdot,w}\big) && (7.19)\\
&= G\Big(\big[x^t_{\cdot,w}\big]_+\Big) + \langle \rho^t_{\cdot,w},\ \big[x^t_{\cdot,w} + m^t_{\cdot,w}\big]_+\rangle + \gamma\big(\rho^t_{\cdot,w} - m^t_{\cdot,w}\big). && (7.20)
\end{aligned}$$

Summing the potentials across time selection functions,

$$\begin{aligned}
&\sum_{w \in \mathcal{W}} G\Big(\big[x^t_{\cdot,w} + \rho^t_{\cdot,w}\big]_+\Big) \\
&\leq \sum_{w \in \mathcal{W}} G\Big(\big[x^t_{\cdot,w}\big]_+\Big) + \langle \rho^t_{\cdot,w},\ \big[x^t_{\cdot,w} + m^t_{\cdot,w}\big]_+\rangle + \gamma\big(\rho^t_{\cdot,w} - m^t_{\cdot,w}\big). && (7.21)
\end{aligned}$$

With some algebra, we can rewrite the sum of inner products:

$$\sum_{w\in\mathcal{W}} \langle \rho^t_{\cdot,w},\, [x^t_{\cdot,w} + m^t_{\cdot,w}]_+\rangle = \sum_{w\in\mathcal{W}}\sum_{\phi\in\Phi} w^t \rho(\phi;\pi^t,\sigma^t)\big[x^t_{\phi,w} + m^t_{\phi,w}\big]_+ \tag{7.22}$$

$$= \sum_{\phi\in\Phi} \rho(\phi;\pi^t,\sigma^t) \sum_{w\in\mathcal{W}(\phi)} w^t \big[x^t_{\phi,w} + m^t_{\phi,w}\big]_+ \tag{7.23}$$

$$= \sum_{\phi\in\Phi} \rho(\phi;\pi^t,\sigma^t) y^t_\phi. \tag{7.24}$$

Since the strategy π^t is the fixed point of $\bar{\phi}^t$ generated from preferences y^t, the Blackwell condition $\sum_{\phi\in\Phi}\rho(\phi;\pi^t,\sigma^t)y^t_\phi \le 0$ is satisfied with equality. For proof, see, for example, A. Greenwald, Z. Li, and Marks (2006a). The sum of potential functions after T rounds are then bounded as

$$\sum_{w\in\mathcal{W}} G\Big(\big[x^T_{\cdot,w} + \rho^T_{\cdot,w}\big]_+\Big) \le \sum_{w\in\mathcal{W}} G\Big(\big[x^T_{\cdot,w}\big]_+\Big) + \gamma(\rho^T_{\cdot,w} - m^T_{\cdot,w}). \tag{7.25}$$

Expanding the definition of γ,

$$\sum_{w\in\mathcal{W}} G\left(\underbrace{\big[x^T_{\cdot,w} + \rho^T_{\cdot,w}\big]_+}_{\left[x^{T+1}_{\cdot,w}\right]_+}\right) \le \sum_{w\in\mathcal{W}} G\Big(\big[x^T_{\cdot,w}\big]_+\Big) + \frac{1}{2}\sum_{w\in\mathcal{W}}\sum_{\phi\in\Phi}\big(\rho^T_{\phi,w} - m^T_{\phi,w}\big)^2 \tag{7.26}$$

$$= \sum_{w\in\mathcal{W}} G\Big(\big[x^T_{\cdot,w}\big]_+\Big) + \frac{1}{2}\sum_{\substack{\phi\in\Phi,\\ w\in\mathcal{W}(\phi)}}\big(\rho^T_{\phi,w} - m^T_{\phi,w}\big)^2. \tag{7.27}$$

Unrolling this potential function recursion across rounds,

$$\sum_{w\in\mathcal{W}} G\Big(\big[x^{T+1}_{\cdot,w}\big]_+\Big) \le \frac{1}{2}\sum_{t=1}^{T}\sum_{\substack{\phi\in\Phi,\\ w\in\mathcal{W}(\phi)}}\big(\rho^t_{\phi,w} - m^t_{\phi,w}\big)^2. \tag{7.28}$$

We lower bound

$$\sum_{w\in\mathcal{W}} G\Big(\big[x^{T+1}_{\cdot,w}\big]_+\Big) = \frac{1}{2}\sum_{w\in\mathcal{W}}\sum_{\phi\in\Phi}\Big(\big[x^{T+1}_{\phi,w}\big]_+\Big)^2 \tag{7.29}$$

$$\ge \frac{1}{2}\max_{\substack{\phi\in\Phi,\\ w\in\mathcal{W}(\phi)}}\Big(\big[x^{T+1}_{\phi,w}\big]_+\Big)^2 \tag{7.30}$$

so that

$$\frac{1}{2}\max_{\substack{\phi\in\Phi,\\ w\in\mathcal{W}(\phi)}}\Big(\big[x^{T+1}_{\phi,w}\big]_+\Big)^2 \le \frac{1}{2}\sum_{t=1}^{T}\sum_{\substack{\phi\in\Phi,\\ w\in\mathcal{W}(\phi)}}\big(\rho^t_{\phi,w} - m^t_{\phi,w}\big)^2. \tag{7.31}$$

Multiplying both sides by two, taking the square root, and applying $\rho^{1:T}_{\phi,w} \leq \left[x^{T+1}_{\phi,w}\right]_+$, we arrive at the final bound,

$$\max_{\substack{\phi\in\Phi,\\ w\in\mathcal{W}(\phi)}} \rho^{1:T}_{\phi,w} \leq \sqrt{\sum_{t=1}^{T} \sum_{\substack{\phi\in\Phi,\\ w\in\mathcal{W}(\phi)}} \left(\rho^t_{\phi,w} - m^t_{\phi,w}\right)^2}. \tag{7.32}$$

Since the bound is true of the worst-case $\phi \in \Phi$ and $w \in \mathcal{W}$, it is true of each pair, thereby proving the claim. □

Let the size of the largest time selection function set be $m^* = \max_{\phi\in\Phi} m(\phi)$. If all of the predictions m^t are zero, then we arrive at a simple bound as a function of m^* for ordinary regret matching.

Corollary 1. $(\mathcal{W}, \Phi, [\cdot]_+)$*-regret matching or regret matching*$^+$ *ensures that* $\rho^{1:T}_{\phi,w} \leq 2U\sqrt{m^*\alpha(\Phi)T}$ *for any deviation* $\phi \in \Phi \subseteq \Phi^{\mathrm{SW}}_{\mathcal{X}}$ *and time selection function* $w \in \mathcal{W}$*, where* $\alpha(\Phi) = \max_{x\in\mathcal{X}} \sum_{\phi\in\Phi} \mathbb{1}\{\phi(x) \neq x\}$ *is the maximal activation of* Φ*.*

Proof. Since $m^t = \mathbf{0}$ on every round t, we know from Theorem 15 that

$$\rho^{1:T}_{\phi,w} \leq \sqrt{\sum_{t=1}^{T} \sum_{\substack{\phi'\in\Phi,\\ \bar{w}\in\mathcal{W}(\phi')}} \left(\bar{w}^t\rho(\phi', \pi^t; \sigma^t)\right)^2} \tag{7.33}$$

$$= \sqrt{\sum_{t=1}^{T} \sum_{\phi'\in\Phi} \left(\rho(\phi', \pi^t; \sigma^t)\right)^2 \sum_{\bar{w}\in\mathcal{W}(\phi')} (\bar{w}^t)^2}. \tag{7.34}$$

Since $0 \leq \bar{w}^t \leq 1$,

$$\rho^{1:T}_{\phi,w} \leq \sqrt{m^* \sum_{t=1}^{T} \sum_{\phi'\in\Phi} \left(\rho(\phi', \pi^t; \sigma^t)\right)^2}. \tag{7.35}$$

Since $\sum_{\phi'\in\Phi} \left(\rho(\phi', \pi^t; \sigma^t)\right)^2 \leq (2U)^2\alpha(\Phi)$ (see A. Greenwald, Z. Li, and Marks (2006a)),

$$\rho^{1:T}_{\phi,w} \leq \sqrt{m^*(2U)^2\alpha(\Phi)T} \tag{7.36}$$

$$= 2U\sqrt{m^*\alpha(\Phi)T}. \tag{7.37}$$

This result completes the argument. □

The predictions m^t can alternatively be interpreted as errors in approximating the exact link inputs x^t for ordinary regret matching. In this case, Theorem 15 shows that as long as these errors are small or appear similar to the regret on the next round, then an approximate regret matching algorithm will have small regret. This is an alternative to the more complicated approximate regret matching bounds given by D'Orazio (2020) and D'Orazio, Morrill, et al. (2020).

Algorithm 2 EFR update for player i with exact regret matching.

```
Input: agent strategy, $\pi^t \in \Pi$,
    daimon strategy, $\sigma^t \in \Sigma$, and
    behavioral deviations, $\Phi \subseteq \Phi^{\text{SW}}_{\mathcal{S}_{\mathcal{A}}}$.
initialize table $\rho^{1:0}_{\cdot,\cdot}(\cdot) = 0$.
▷ Update cumulative immediate regrets:
for $s \in \mathcal{S}_{\mathcal{A}}$, $\phi_s \in \Phi_s$, $w_\lambda \in \mathcal{W}_s(\Phi)$ do
    $\rho^{1:t}_{s,\lambda}(\phi_s) \leftarrow \rho^{1:t-1}_{s,\lambda}(\phi_s) + \mathbb{1}\{\phi_s \in \Phi_{s,\lambda}\} w^t_\lambda \rho^{\text{CF}}_s(\phi_s, \pi^t; \sigma^t)$
▷ Construct $\pi^{t+1}$ with regret matching:
for $s \in \mathcal{S}_{\mathcal{A}}$ from the start of the game to the end do
    for $\phi_s \in \Phi_s$ do
        ▷ $\pi^{t+1}$ need only be defined at $\bar{s} \prec s$ for $w^{t+1}_\lambda$ to be well-defined.
        $y^{t+1}_{\phi_s} \leftarrow \sum_{w_\lambda \in \mathcal{W}_s(\Phi)} \mathbb{1}\{\phi_s \in \Phi_{s,\lambda}\} w^{t+1}_\lambda \left[\rho^{1:t}_{s,\lambda}(\phi_s)\right]_+$
    $z^{t+1} \leftarrow \sum_{\phi_s \in \Phi_s} y^{t+1}_{\phi_s}$
    $\bar{\phi}^{t+1}_s \leftarrow \frac{1}{z^{t+1}} \sum_{\phi_s \in \Phi_s} y^{t+1}_{\phi_s} \phi_s$ if $z^{t+1} > 0$ else $I$
    $\pi^{t+1}(s) \leftarrow$ a fixed point of $\bar{\phi}^{t+1}_s$
return $\pi^{t+1}$
```

7.4.4 Use in EFR

When exact time selection regret matching with the ramp link function is applied to EFR, we arrive at the following concrete regret bound:

Theorem 16. *Instantiate EFR with exact ramp regret matching and a set of behavioral deviations $\Phi \subseteq \Phi^{\text{SW}}_{\mathcal{S}_{\mathcal{A}}}$. Overload*

$$\alpha : \Phi \mapsto \max_{I \in \mathcal{S}_{\mathcal{A}}, \phi_s \in \Phi_s} \alpha(\Phi_I) \sum_{w_\lambda \in \mathcal{W}_I(\phi)} \mathbb{1}\{\phi_s \in \Phi_{s,\lambda}\}$$

as the maximal activation for behavioral deviations. EFR's cumulative full regret at any active agent state after T rounds with respect to Φ and the set of single-target deviations generated from Φ, $\Phi_{\preceq\odot}$, is no more than $2 d_* U |\mathcal{S}_{\mathcal{A}}| \sqrt{\alpha(\Phi) T}$. In addition, this implies that EFR is OS hindsight rational with respect to $\Phi \cup \Phi_{\preceq\odot}$.

Proof. EFR's immediate strategies at each agent state s on each round are chosen according to time selection regret matching on the cumulative memory-state-specific immediate regrets and memory state probabilities there. The number of time selection functions for a given action transformation ϕ_s at s is $\sum_{w_\lambda \in \mathcal{W}_s(\Phi)} \mathbb{1}\{\phi_s \in \Phi_{s,\lambda}\}$ so the maximal activation of Φ, $\alpha(\Phi)$, is the largest product of $m^*_s \alpha(\Phi_s)$ across all agent states s. Exact ramp regret matching thus ensures that cumulative immediate regret is no larger than $2U\sqrt{\alpha(\Phi)T}$ according to Corollary 1 and Eqs. (7.8) and (7.11). Cumulative full regret is therefore no larger than $2 d_* U |\mathcal{S}_{\mathcal{A}}| \sqrt{\alpha(\Phi) T}$ according to Theorem 5. □

See Table 7.1 for the maximal activation value for each deviation type. Algorithm 2 provides an implementation of EFR with exact regret matching.

7.5 Experiments

Our theoretical results show that EFR variants utilizing more powerful deviation types are pushed to accumulate higher payoffs during learning in worst-case environments. Do these deviation types make a practical difference outside of the worst case?

We investigate the performance of EFR with different deviation types in nine benchmark game instances from *OpenSpiel* (Lanctot, Lockhart, et al. 2019). We evaluate each EFR variant by the expected payoffs accumulated over the course of playing each game in each seat over 1000 rounds under two different regimes for selecting the other players. In the “fixed regime”, other players play their parts of the fixed sequence of strategy profiles generated with self-play before the start of the experiment using one of the EFR variants under evaluation. In the “simultaneous regime”, the other players are EFR instances themselves. In games with more than two players, all other players share the same EFR variant and we only record the score for the solo EFR instance. The fixed regime provides a test of how well each EFR variant adapts when the other players are gradually changing in an oblivious way where comparison is simple, while the simultaneous regime is a possibly more realistic test of dynamic adaptation where it is more difficult to draw definitive conclusions about relative effectiveness.

Since we evaluate expected payoff, use expected EFR updates, and use exact regret matching, all results are deterministic and hyperparameter-free. To compute the regret matching fixed point when internal transformations are used, we solve a linear system with the Jacobi singular value algorithm implemented by the `jacobiSvd` method from the Eigen C++ library (Guennebaud et al. 2010). Experimental data and code for generating both the data and final results are available on GitHub.[3] Experiments took roughly 20 hours to complete on a 2.10GHz Intel® Xeon® CPU E5-2683 v4 processor with 10 GB of RAM.

Section 7.C hosts the full set of results but a representative summary from two variants of imperfect information goofspiel (Lanctot 2013; Ross 1971) (a two-player and a three-player version denoted as $\mathrm{g}_{2,\,5,\,\uparrow}$ and $\mathrm{g}_{3,\,4,\,\uparrow}$, respectively, both zero-sum) and Sheriff (two-player, non-zero-sum) is presented in Table 7.2. See Section 7.A for descriptions of all games.

Stronger deviations consistently lead to better performance in both the fixed and the simultaneous regime. The behavioral deviations (BHV) and the informed action deviations ($\mathrm{ACT}_{\textsc{in}}$) often lead to the best and worst performance, respectively, and this is true of each scenario in Table 7.2. In many cases however, TIPS or CSPS yield similar performance

[3] https://github.com/dmorrill10/hr_edl_experiments

Table 7.2: The payoff of each EFR instance averaged across both 1000 rounds and each instance pairing (eight pairs in total) in two-player and three-player goofspiel (measured in win frequency between zero and one), and Sheriff (measured in points between -6 and $+6$). The top group of algorithms use weak deviation types (ACT$_{\text{IN}}$ → informed action deviations, CF → blind counterfactual, and CF$_{\text{IN}}$ → informed counterfactual) and the middle group use partial sequence deviation types. The BHV instance uses the full set of behavioral deviations.

	fixed			simultaneous		
	$g_{2,\,5,\,\uparrow}$	$g_{3,\,4,\,\uparrow}$	Sheriff	$g_{2,\,5,\,\uparrow}$	$g_{3,\,4,\,\uparrow}$[†]	Sheriff
ACT$_{\text{IN}}$	0.51	0.48	0.28	0.45	0.86	0.00
CF	0.56	0.51	0.48	0.50	0.88	0.34
CF$_{\text{IN}}$	0.57	0.51	0.60	0.50	0.92	0.37
BPS	0.58	0.51	0.58	0.50	0.85	0.34
CF	0.58	0.52	0.70	0.51	0.84	0.37
CSPS	0.59	0.52	0.61	0.51	0.91	0.37
TIPS	0.60	0.53	0.82	0.51	0.87	0.38
BHV	0.63	0.53	0.91	0.51	0.92	0.38

[†] In three-player goofspiel, players who tend to play the same actions perform worse. Since the game is symmetric across player seats, two players who use the same (deterministic) algorithm will always employ the same strategies and often play the same actions, giving the third player a substantial advantage. The win percentage for all variants in the simultaneous regime tends to be high because we only record the score for each variant when they are instantiated in a single seat. The relative comparison is still informative.

to BHV. A notable outlier from the scenarios in Table 7.2 is three-player goofspiel with a descending point deck. Here, blind counterfactual (CF) and BPS deviations lead to better performance in the first few rounds before all variants quickly converge to play that achieves essentially the same payoff (see Figs. 7.C.1 to 7.C.4).

7.6 Conclusion

I introduced EFR, an algorithm that is OS hindsight rational for any given set of behavioral deviations. While the full set of behavioral deviations leads to generally intractable computational requirements, the four partial sequence deviation types are both tractable and powerful in games with moderate lengths when combined with OSR.

An important tradeoff within EFR is that using stronger deviation types generally leads to slower strategy updates, demonstrated by Figs. 7.C.5 and 7.C.6 where learning curves are plotted according to runtime. Often in a tournament setting, the number of rounds and computational budget may be fixed so that running faster cannot lead to more reward for the learner, but it can be beneficial to have faster updates in other scenarios. Quantifying the potential benefit of using a stronger deviation type in particular games could aid in navigating this tradeoff.

Alternatively, one could hope that the learner could navigate this tradeoff on their own. Algorithms like the fixed-share forecaster (Herbster et al. 1998) or context tree weighting (Willems et al. 1993) efficiently minimize regret across large structured sets of experts, effectively avoiding a similar tradeoff. Unfortunately, this efficiency is entirely dependent on multiplicative weight updates that cannot be applied to time selection.

A second tradeoff is that stronger deviation types lead to EFR regret bounds with larger constant factors even if the best deviation is part of a "simpler" class, *e.g.*, the regret bound that TIPS EFR has with respect to counterfactual deviations is larger than that of CFR even though a TIPS EFR instance might often accumulate more reward in order to compete with the larger TIPS deviations. A simple case of this can be studied in NFGs where regret matching on internal regret has a worse external regret bound than regret matching on external regret. Perhaps an EFR variant can be designed that would compete with large sets of behavioral deviations, but its regret bound would scale with the "complexity" (in a sense that has yet to be rigorously defined) of the best deviation rather than the size of the whole deviation set.

My analysis of general immediate regret minimization for POHPs in Section 4.5 and the impossibility result of Theorem 3 brings up questions about how far this procedure can be generalized. The EFR regret decomposition is based on a perfect-recall and a realization-weighted variant of Kakade (2003)'s performance difference lemma (Lemma 5.2.1). This observation is used in Chapter 9 to show how CFR can be applied to continuing, discounted MDPs with reward uncertainty, but it is less clear how deviations other than the counterfactual and action deviations could be used in this setting. The POHP formalism can perhaps allow us to better understand how and when EFR can be applied without perfect recall by considering Lanctot, Burch, et al. (2012)'s well-formed-game conditions allowing efficient full counterfactual regret minimization in imperfect recall EFGs together with the analysis from Chapter 9.

References

Brown, N., A. Lerer, S. Gross, and T. Sandholm (2019). "Deep Counterfactual Regret Minimization". In: *36th International Conference on Machine Learning (ICML 2019)*, pp. 793–802.

Burch, N., M. Lanctot, D. Szafron, and R. Gibson (2012). “Efficient Monte Carlo counterfactual regret minimization in games with many player actions”. In: *Advances in Neural Information Processing Systems*, pp. 1880–1888.

Celli, A., A. Marchesi, G. Farina, and N. Gatti (2020). “No-regret learning dynamics for extensive-form correlated equilibrium”. In: *Advances in Neural Information Processing Systems* 33.

D’Orazio, R. (2020). “Regret Minimization with Function Approximation in Extensive-Form Games”. Master’s thesis. University of Alberta.

D’Orazio, R. and R. Huang (2021). “Optimistic and Adaptive Lagrangian Hedging”. In: *AAAI Reinforcement Learning in Games Workshop*.

D’Orazio, R., D. Morrill, J. R. Wright, and M. Bowling (May 2020). “Alternative Function Approximation Parameterizations for Solving Games: An Analysis of f-Regression Counterfactual Regret Minimization”. In: *19th International Conference on Autonomous Agents and Multi-Agent Systems*. International Foundation for Autonomous Agents and Multiagent Systems.

Davis, T., M. Schmid, and M. Bowling (2020). “Low-Variance and Zero-Variance Baselines for Extensive-Form Games”. In: *International Conference on Machine Learning*. PMLR, pp. 2392–2401.

Farina, G., C. Kroer, N. Brown, and T. Sandholm (2019). “Stable-Predictive Optimistic Counterfactual Regret Minimization”. In: *International Conference on Machine Learning*, pp. 1853–1862.

Farina, G., C. Kroer, and T. Sandholm (2021). “Faster Game Solving via Predictive Blackwell Approachability: Connecting Regret Matching and Mirror Descent”. In: *35th AAAI Conference on Artificial Intelligence*. Vol. 35. 6, pp. 5363–5371.

Farina, G., C. K. Ling, F. Fang, and T. Sandholm (2019). “Correlation in Extensive-Form Games: Saddle-Point Formulation and Benchmarks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Foerster, J., F. Song, E. Hughes, N. Burch, I. Dunning, S. Whiteson, M. Botvinick, and M. Bowling (2019). “Bayesian action decoder for deep multi-agent reinforcement learning”. In: *International Conference on Machine Learning*. PMLR, pp. 1942–1951.

Gibson, R., M. Lanctot, N. Burch, D. Szafron, and M. Bowling (2012). “Generalized Sampling and Variance in Counterfactual Regret Minimization”. In: *26th Conference on Artificial Intelligence (AAAI-12)*, pp. 1355–1361.

Greenwald, A., Z. Li, and C. Marks (Jan. 2006a). “Bounds for Regret-Matching Algorithms”. In: *International Symposium on Artificial Intelligence and Mathematics (ISAIM 2006)*. Fort Lauderdale, Florida, USA.

Guennebaud, G., B. Jacob, et al. (2010). “Eigen”. In: *URl: http://eigen. tuxfamily. org*.

Herbster, M. and M. K. Warmuth (1998). “Tracking the best expert”. In: *Machine learning* 32.2, pp. 151–178.

Johanson, M., N. Bard, M. Lanctot, R. Gibson, and M. Bowling (2012). “Efficient Nash Equilibrium Approximation through Monte Carlo Counterfactual Regret Minimization”. In: *11th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Kakade, S. M. (2003). “On the sample complexity of reinforcement learning”. PhD thesis. UCL (University College London).

Kash, I. A., M. Sullins, and K. Hofmann (May 2020). “Combining no-regret and Q-learning”. In: *19th International Conference on Autonomous Agents and Multi-Agent Systems*. International Foundation for Autonomous Agents and Multiagent Systems.

Lanctot, M. (June 2013). “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.

Lanctot, M., N. Burch, M. Zinkevich, M. Bowling, and R. G. Gibson (2012). “No-Regret Learning in Extensive-Form Games with Imperfect Recall”. In: *29th International Conference on Machine Learning (ICML 2012)*, pp. 65–72.

Lanctot, M., E. Lockhart, et al. (2019). “OpenSpiel: A Framework for Reinforcement Learning in Games”. In: *CoRR* abs/1908.09453. arXiv: 1908.09453 `[cs.LG]`.

Lanctot, M., K. Waugh, M. Zinkevich, and M. Bowling (2009). “Monte Carlo Sampling for Regret Minimization in Extensive Games”. In: *Advances in Neural Information Processing Systems 22*. Ed. by Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, pp. 1078–1086.

Morrill, D. (2016). “Using Regret Estimation to Solve Games Compactly”. Master’s thesis. University of Alberta.

Morrill, D., R. D’Orazio, R. Sarfati, M. Lanctot, J. R. Wright, A. R. Greenwald, and M. Bowling (Feb. 2021). “Hindsight and Sequential Rationality of Correlated Play”. In: *35th AAAI Conference on Artificial Intelligence*. Vol. 35. 6. virtual, pp. 5584–5594.

Rakhlin, S. and K. Sridharan (2013). “Optimization, learning, and games with predictable sequences”. In: *Advances in Neural Information Processing Systems*, pp. 3066–3074.

Ross, S. M. (1971). “Goofspiel — The game of pure strategy”. In: *Journal of Applied Probability* 8.3, pp. 621–625.

Schmid, M., N. Burch, M. Lanctot, M. Moravčík, R. Kadlec, and M. Bowling (2019). “Variance Reduction in Monte Carlo Counterfactual Regret Minimization (VR-MCCFR) for Extensive Form Games using Baselines”. In: *33rd AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 2157–2164.

Southey, F., M. Bowling, B. Larson, C. Piccione, N. Burch, D. Billings, and D. C. Rayner (July 2005). “Bayes’ Bluff: Opponent Modelling in Poker”. In: *21st Conference in Uncertainty in Artificial Intelligence (UAI 2005)*. Edinburgh, Scotland, pp. 550–558.

Steinberger, E., A. Lerer, and N. Brown (2020). “DREAM: Deep regret minimization with advantage baselines and model-free learning”. In: *arXiv preprint arXiv:2006.10410*.

Waugh, K., D. Morrill, J. A. Bagnell, and M. Bowling (2015). “Solving Games with Functional Regret Estimation”. In: *29th AAAI Conference on Artificial Intelligence (AAAI-15)*. Vol. 29. 1, pp. 2138–2144.

Willems, F. M., Y. M. Shtarkov, and T. J. Tjalkens (1993). “Context tree weighting: a sequential universal source coding procedure for FSMX sources”. In: *1993 IEEE International Symposium on Information Theory*. Institute of Electrical and Electronics Engineers, p. 59.

Zinkevich, M., M. Johanson, M. Bowling, and C. Piccione (Dec. 2007b). “Regret Minimization in Games with Incomplete Information”. In: *Advances in Neural Information Processing Systems (NeurIPS 2007)*. Vancouver, British Columbia, pp. 1729–1736.

7.A Games

The OpenSpiel (Lanctot, Lockhart, et al. 2019) implementation of each game is used for experiments.

7.A.1 Leduc Hold'em Poker

Leduc hold'em poker (Southey et al. 2005) is a two-player poker game with a deck of six cards (two suits and three ranks). At the start of the game, both players ante one chip and receive one private card. There are two betting rounds and there is a maximum of two raises on each round. Bet sizes are limited to two chips in the first round and four in the second. If one player folds, the other wins. At the start of the second round, a public card is revealed. A showdown occurs at the end of the second round if no player folds. The strongest hand in a showdown is a pair (using the public card), and if no player pairs, players compare the ranks of their private cards. The player with the stronger hand takes all chips in the pot or players split the pot if their hands have the same strength. Payoffs are reported in milli-big blinds (mbb) (where the ante is considered a big blind) for consistency with the way performance is reported in other poker games.

7.A.2 Imperfect Information Goofspiel

Imperfect information goofspiel (Lanctot 2013; Ross 1971) is a bidding game for N players. Each player is given a hand of n ranks that they play to bid on n point cards. On each round, one point card is revealed and each player simultaneously bids on the point card. The point cards might be sorted in ascending order ($\uparrow$), descending order ($\downarrow$), or they might be shuffled (R). If there is one bid that is greater than all the others, the player who made that bid wins the point card. If there is a draw, the bid card is instead discarded. The player with the most points wins so payoffs are reported in win percentage. We use five goofspiel variants:

- two-player, 5-ranks, ascending (goofspiel$(5, \uparrow, N = 2)$, denoted as $\mathrm{g}_{2,\,5,\,\uparrow}$ in the main paper),
- two-player, 5-ranks, descending (goofspiel$(5, \downarrow, N = 2)$),
- two-player, 4-ranks, random (goofspiel$(4, R, N = 2)$),
- three-player, 4-ranks, ascending (goofspiel$(4, \uparrow, N = 3)$, denoted as $\mathrm{g}_{3,\,4,\,\uparrow}$ in the main paper), and
- three-player, 4-ranks, descending (goofspiel$(4, \downarrow, N = 3)$).

7.A.3 Sheriff

Sheriff is a two-player, non-zero-sum negotiation game resembling the Sheriff of Nottingham board game and it was introduced by Farina, Ling, et al. (2019). At the beginning of the game, the “smuggler” player chooses zero or more illegal items (maximum of three) to add to their cargo. The rest of the game proceeds over four rounds.

At the beginning of each round, the smuggler signals how much they would be willing to pay the “sheriff” player to bribe them into not inspecting the smuggler’s cargo, between zero and three. The sheriff responds by signalling whether or not they would inspect the cargo. On the last round, the bribe amount chosen by the smuggler and the sheriff’s decision about whether or not to inspect the cargo are binding.

If the cargo is not inspected, then the smuggler receives a payoff equal to the number of illegal items included within, minus their bribe amount, and the sheriff receives the bribe amount. Otherwise, the sheriff inspects the cargo. If the sheriff finds an illegal item, then the sheriff forces the smuggler to pay them two times the number of illegal items. Otherwise, the sheriff compensates the smuggler by paying them three.

7.A.4 Tiny Bridge

A miniature version of bridge created by Edward Lockhart, inspired by a research project at University of Alberta by Michael Bowling, Kate Davison, and Nathan Sturtevant. We use the smaller two-player rather than the full four-player version. See the implementation from Lanctot, Lockhart, et al. (2019) for more details.

7.A.5 Tiny Hanabi

A miniature two-player version of Hanabi described by J. Foerster, Song, et al. (2019). The game is fully cooperative and the optimal score is ten. Both players take only one action so all EFR instances collapse except when they differ in their choice of Φ_I.

7.B Alternative Φ_I Choices

When implementing EFR for deviations that set the action transformations at each agent state to the internal transformations, we have the option of implementing these variants by using the union of the internal and external transformations without substantially changing the variant’s theoretical properties. We test how this impacts practical performance within EFR variants for informed counterfactual deviations, CFPS deviations, and TIPS deviations. These variants have an “EX+ IN” subscript.

7.C Results

We present four sets of figures to summarize the performance of each EFR variant in the fixed and simultaneous regimes described in Section 7.

The first three sets of figures illustrate how each variant performs on average in each round individually. Figs. 7.C.1 and 7.C.3 show the running average expected payoff of each variant over rounds, averaged over play with all EFR variants (including itself). These figures summarize the progress that each variant makes over rounds to adapt to and correlate with its companion variant, on average. Figs. 7.C.2 and 7.C.4 show the instantaneous expected payoff of each variant over rounds, averaged over play with all EFR variants. Figs. 7.C.5 and 7.C.6 show the same data as in Figs. 7.C.1 and 7.C.3 except according to runtime rather than rounds. Tiny Hanabi is omitted because it is too small to make meaningful runtime comparisons between EFR variants.

Fig. 7.C.7 show the average expected payoff of each variant paired with each other variant (including itself) after 1000 rounds. These figures summarize how well each variant works with each other variant.

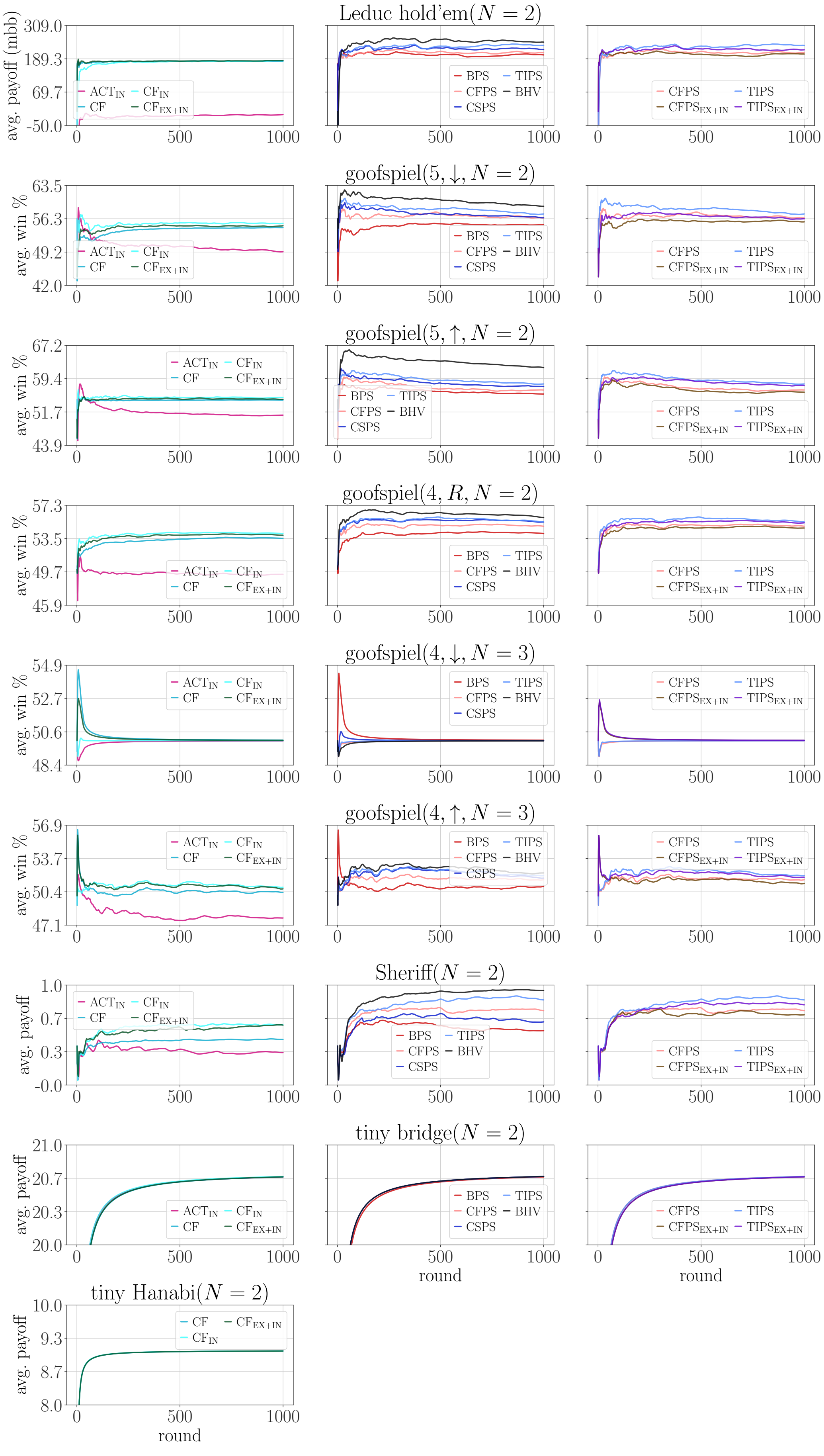


Figure 7.C.1: The expected payoff accumulated by each EFR variant over rounds averaged over play with all EFR variants in each game in the fixed regime.

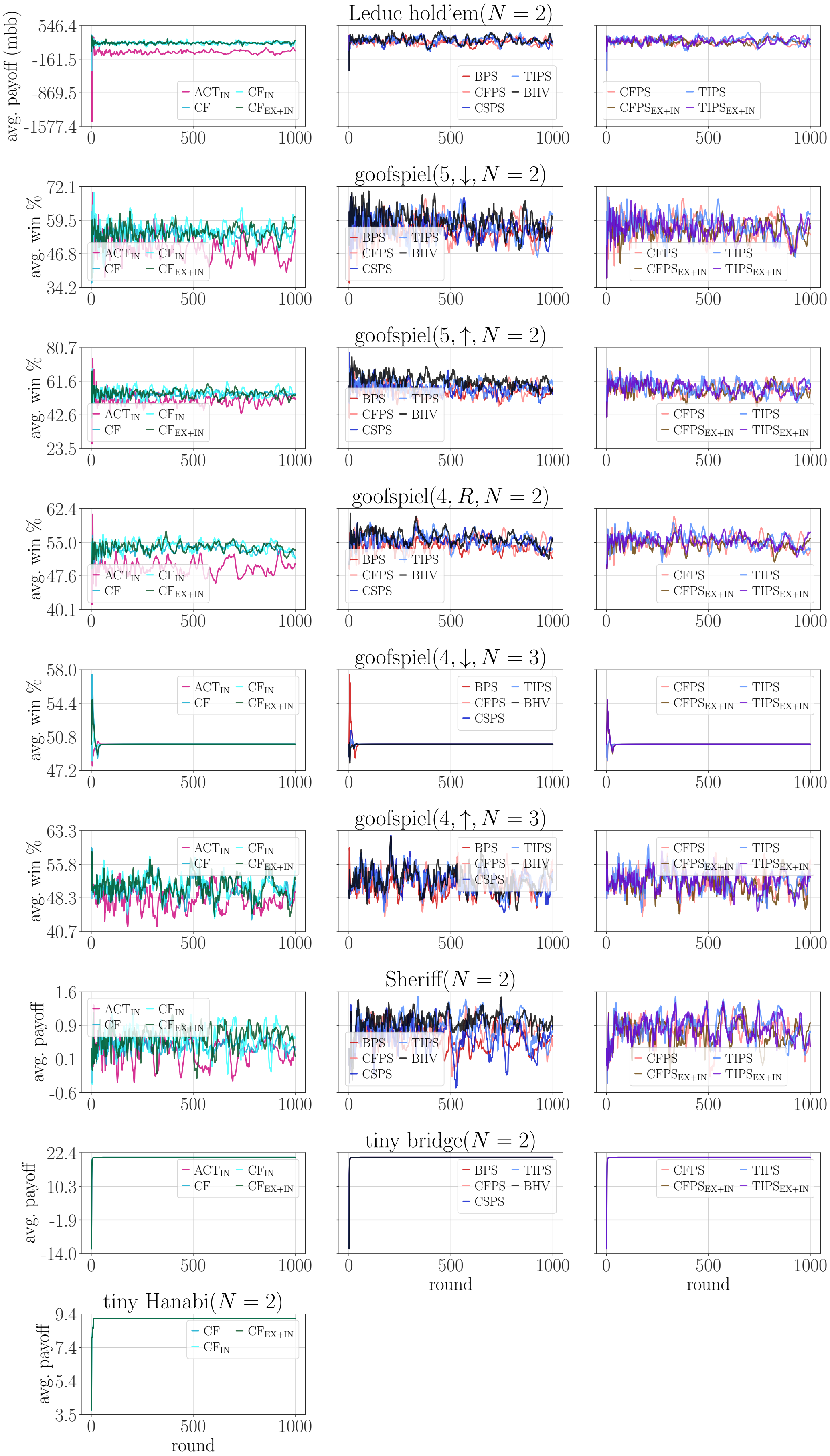


Figure 7.C.2: The instantaneous payoff achieved by each EFR variant on each round averaged over play with all EFR variants in each game in the fixed regime.

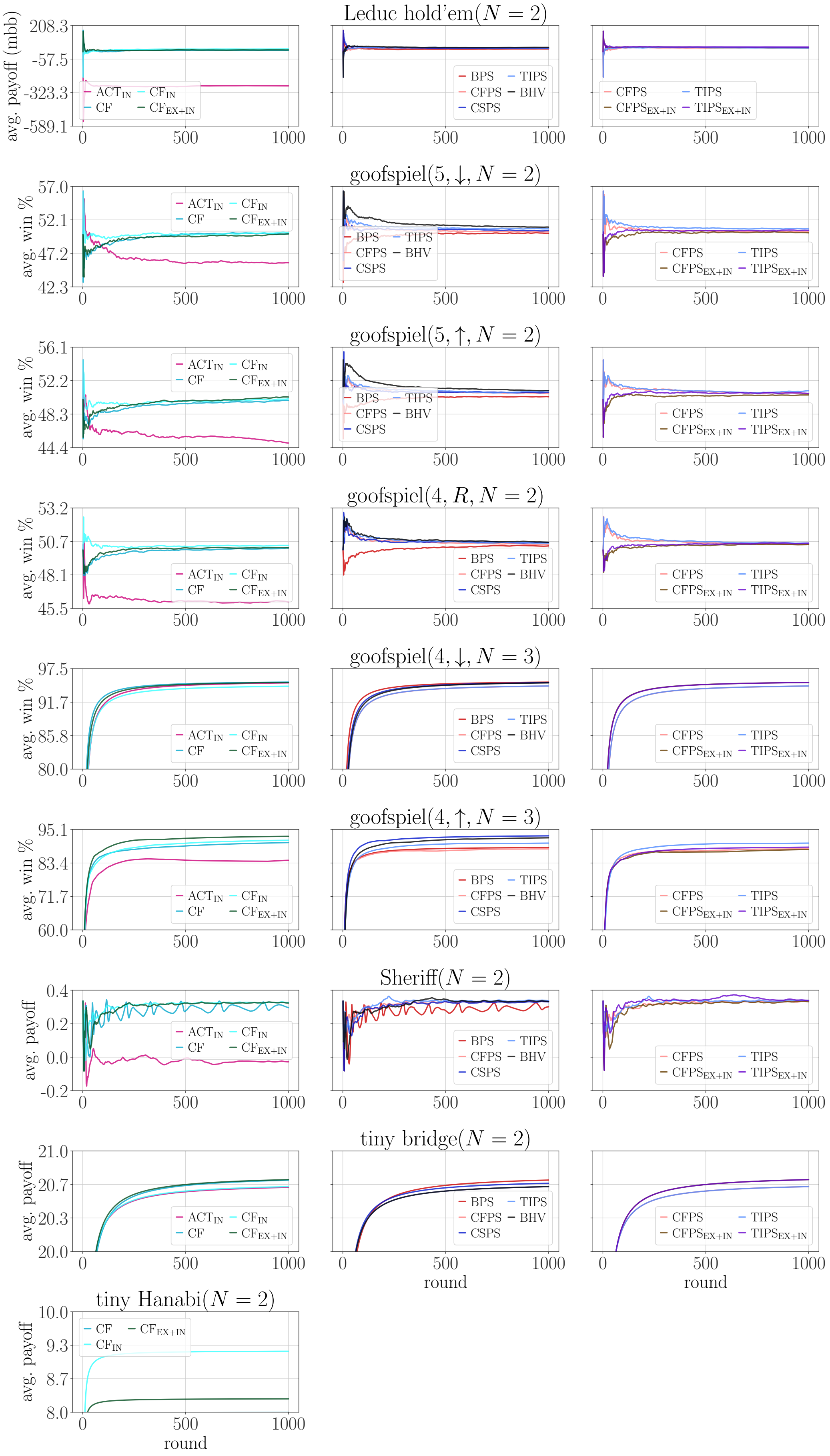


Figure 7.C.3: The expected payoff accumulated by each EFR variant over rounds averaged over play with all EFR variants in each game in the simultaneous regime.

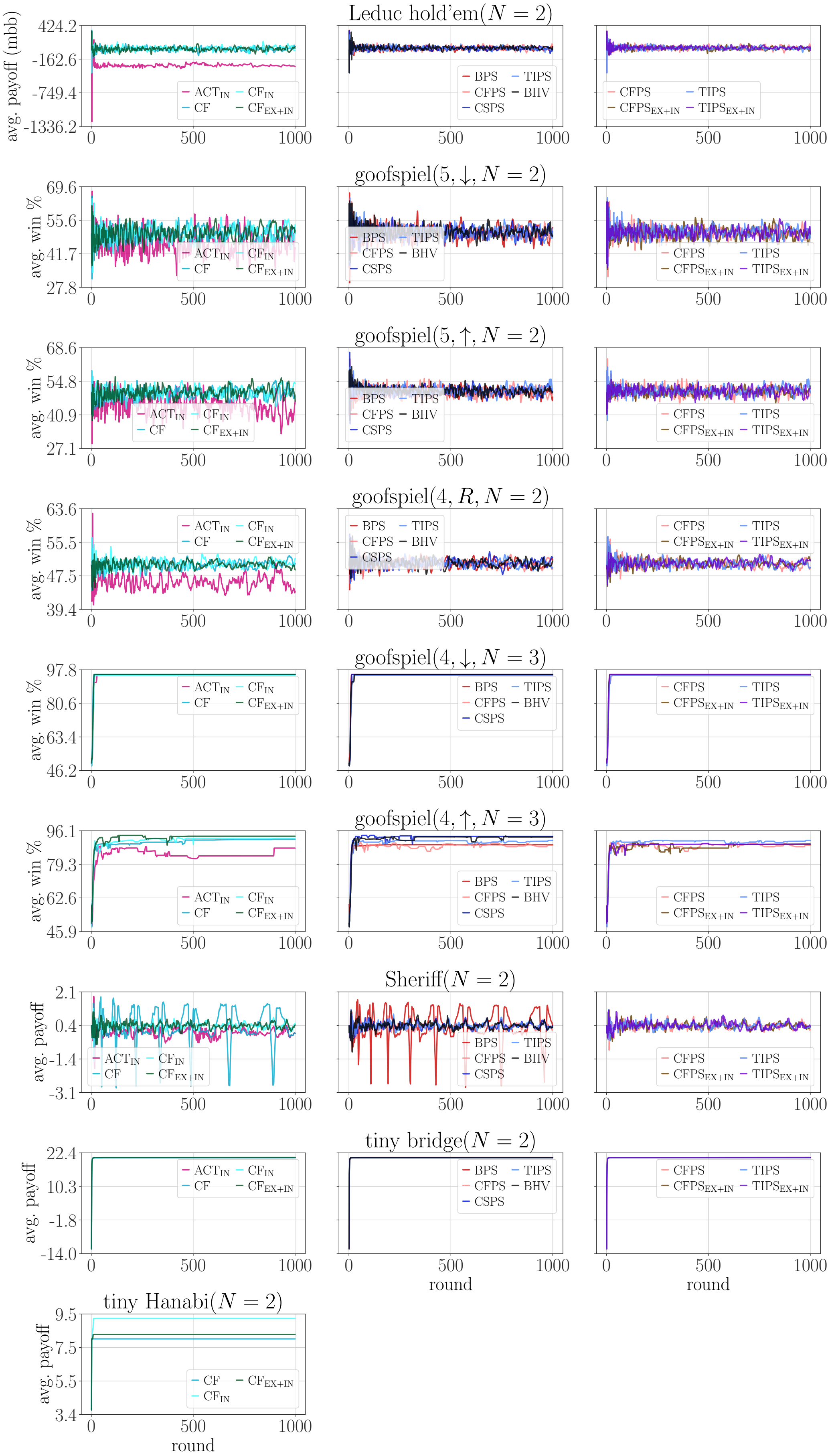


Figure 7.C.4: The instantaneous payoff achieved by each EFR variant on each round averaged over play with all EFR variants in each game in the simultaneous regime.

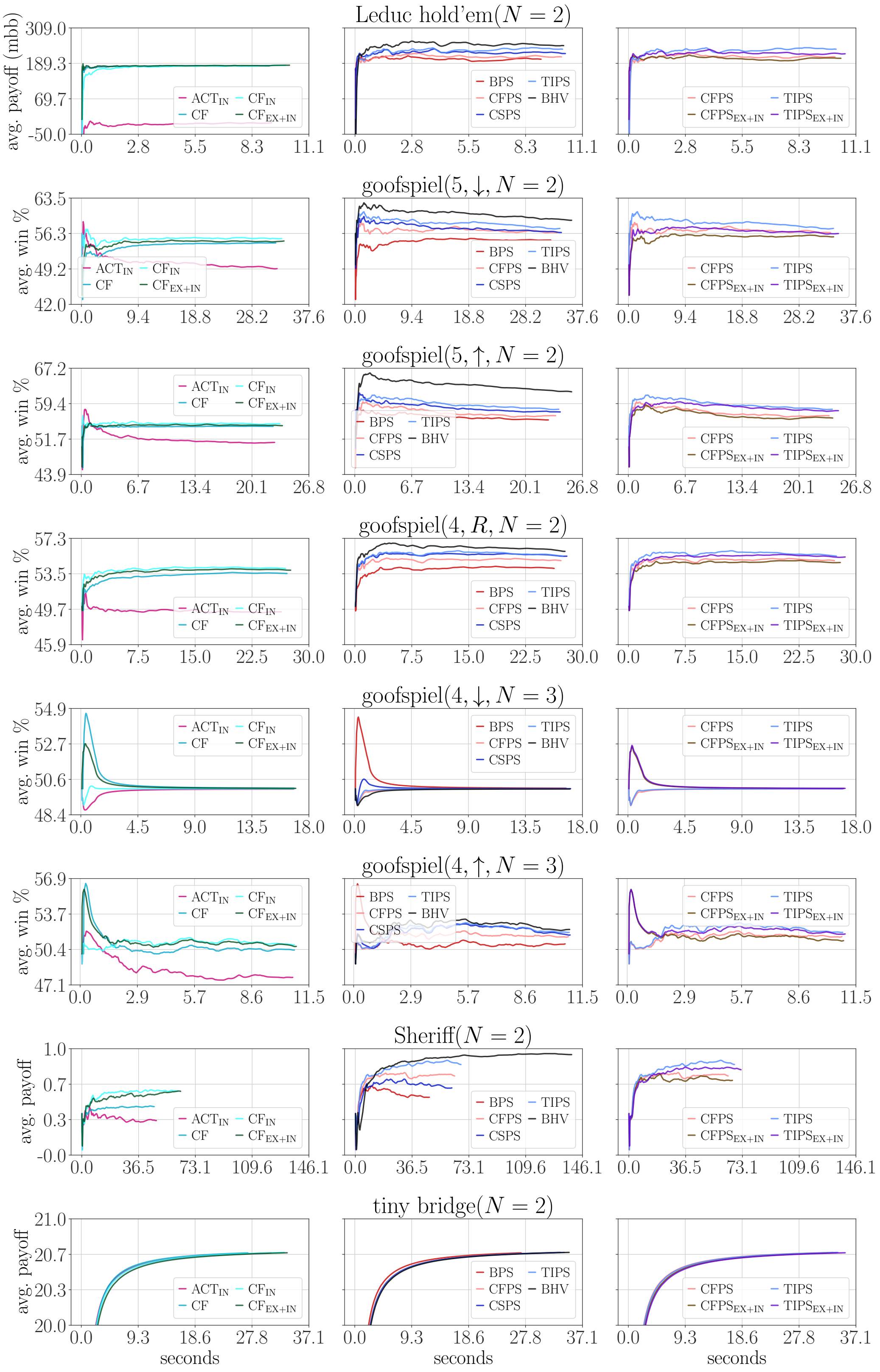


Figure 7.C.5: The expected payoff accumulated by each EFR variant over runtime averaged over play with all EFR variants in each game in the fixed regime.

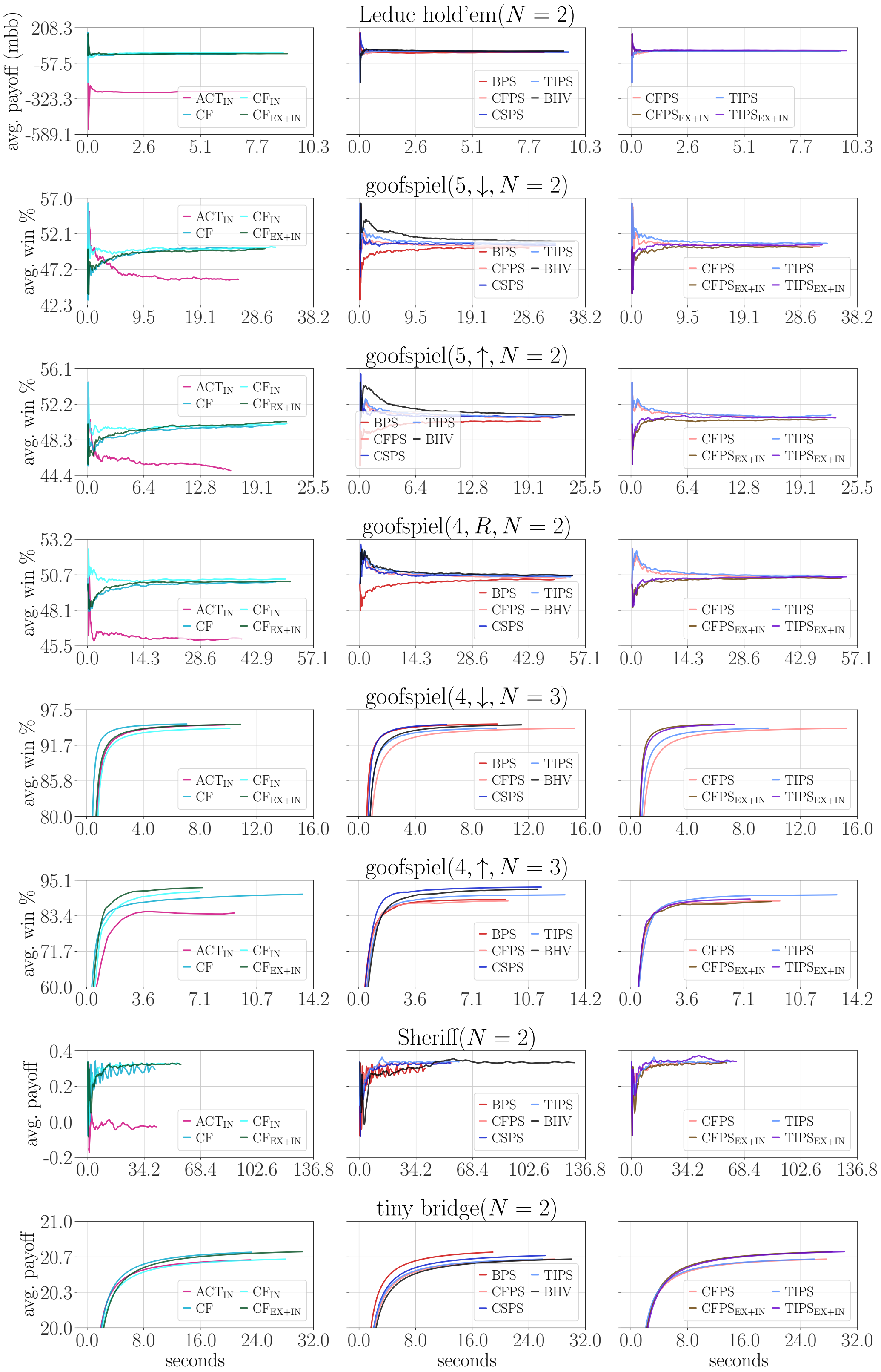


Figure 7.C.6: The expected payoff accumulated by each EFR variant over runtime averaged over play with all EFR variants in each game in the simultaneous regime.

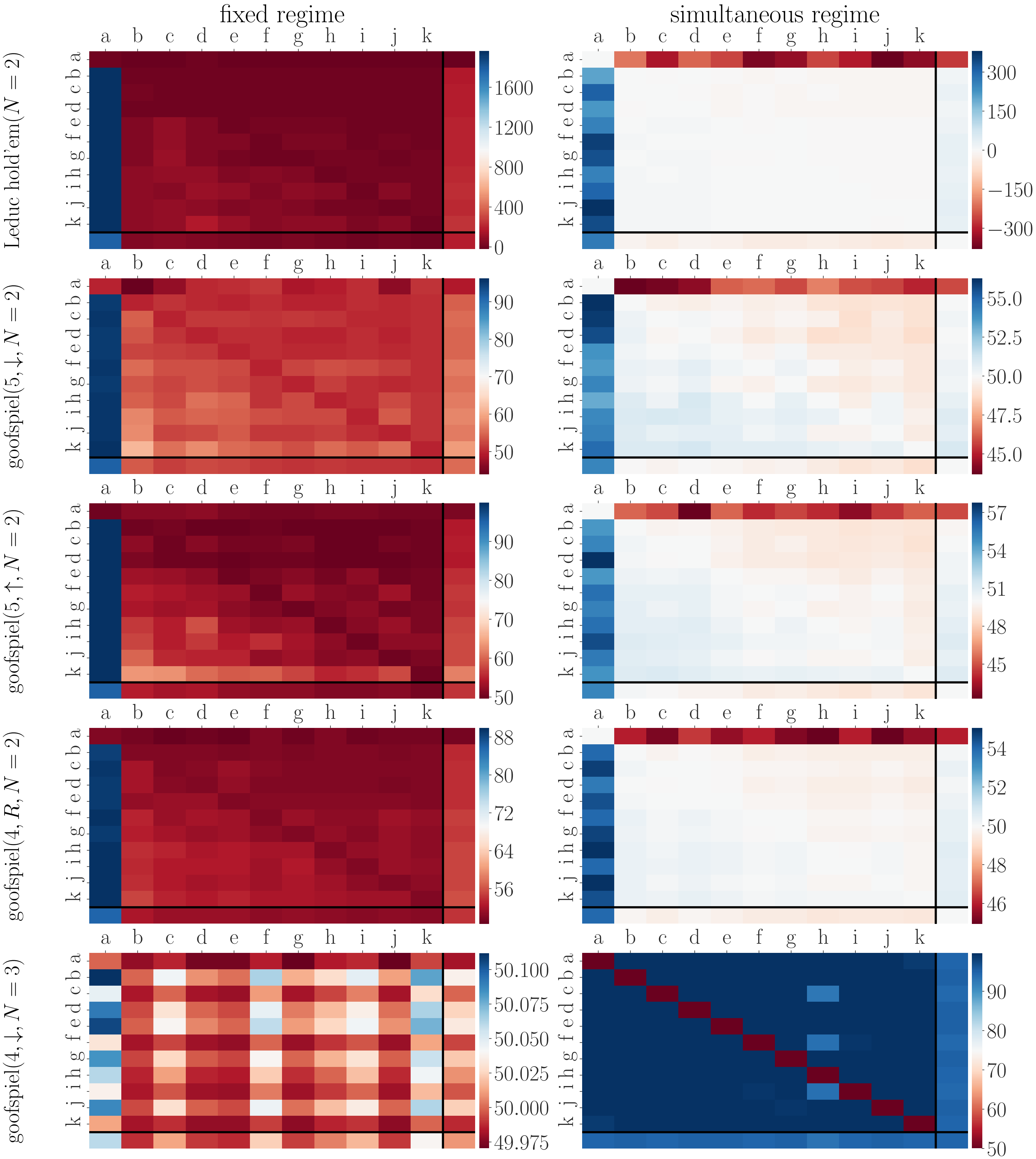


Figure 7.C.7: (1 / 2) The average expected payoff accumulated by each EFR variant (listed by row) from playing with each other EFR variant (listed by column) in each game after 1000 rounds where a → ACT$_{\text{IN}}$, b → CF, c → CF$_{\text{IN}}$, d → CF$_{\text{EX+IN}}$, e → BPS, f → CFPS, g → CFPS$_{\text{EX+IN}}$, h → CSPS, i → TIPS, j → TIPS$_{\text{EX+IN}}$, k → BHV. The bottom rows and farthest right columns represent the column and row averages, respectively.

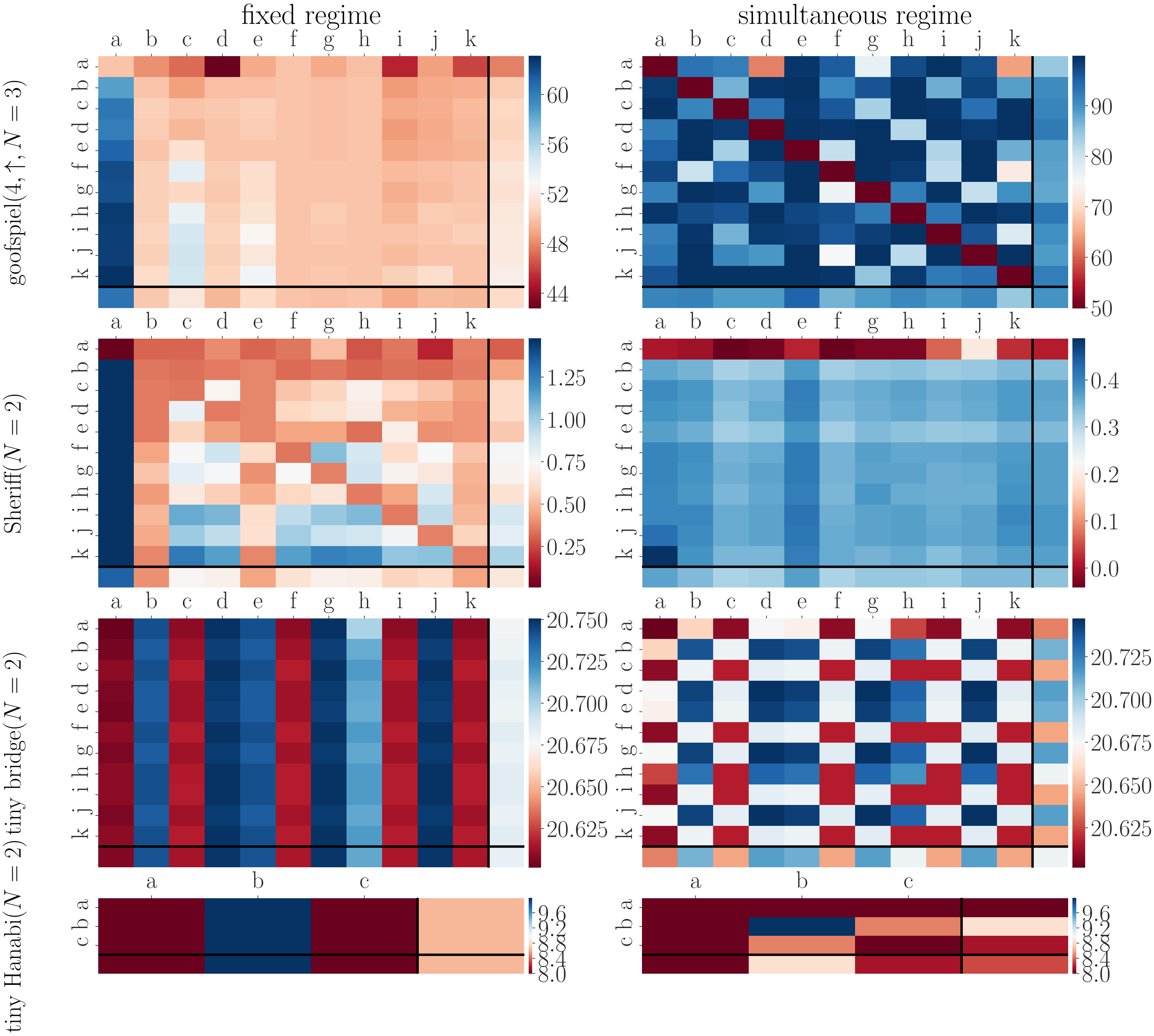


Figure 7.C.8: (2 / 2) The average expected payoff accumulated by each EFR variant (listed by row) from playing with each other EFR variant (listed by column) in each game after 1000 rounds where a → ACT$_{\text{IN}}$, b → CF, c → CF$_{\text{IN}}$, d → CF$_{\text{EX+IN}}$, e → BPS, f → CFPS, g → CFPS$_{\text{EX+IN}}$, h → CSPS, i → TIPS, j → TIPS$_{\text{EX+IN}}$, k → BHV. The bottom rows and farthest right columns represent the column and row averages, respectively.

Part III

Extensions

Chapter 8

Background

8.1 Introduction

This Part of the thesis presents two extensions of the ideas developed in the previous chapters, one on applying regret minimization algorithms to solve robust optimization and AI safety problems in POHPs, and another analyzing CFR with alternative function approximation parameterizations. The background for this Part introduces basic concepts related to learning with function approximation in a POHP, the regression CFR (RCFR) framework for using CFR with function approximation, and k-of-N CFR for robust policy optimization.

8.2 Learning with Function Approximation

POHP models for games that humans are interested in playing, or for problems of practical importance, typically generate an immense number of agent states under perfect recall. In these cases, relationships between environment elements like symmetries and redundancies can be utilized to make the POHP more manageable. To automatically detect such relationships, we look to the field of supervised learning, which specializes in algorithms for learning generalized mappings from input–output pairs.

Supervised learning is “supervised” because training examples are labeled, *i.e.*, every input example has an associated target output value. After completing the training procedure, a supervised learning algorithm returns a function approximator that maps the space of inputs to the space of outputs. The goal is not to naïvely reproduce the input–output pattern from the training data, but to generalize to unseen testing data, effectively predicting the results of the target generating process. The inputs of supervised learning applied to POHPs can represent agent state features, thereby forming the basis of compact mappings from agent states to task-specific values that can generalize across states. Such functions are useful for dealing with large POHPs because these functions allow experience from one part of the

POHP to improve decisions in a separate but related part of the POHP.

When the input space is complicated, *e.g.*, agent states, we can define a feature function, $\varphi : \mathcal{S} \to \mathbb{R}^d$, $d > 0$, that maps inputs to feature vectors. A feature vector lists salient elements of the input that may be important in capturing the mechanics of the target generating process. We will only study "regression problems", where the output is a continuous real value.

8.2.1 Regression CFR

Regression CFR (RCFR; Waugh, Morrill, et al. 2015) uses a function approximator to estimate cumulative counterfactual regrets at each agent state and generates immediate strategies with a normalized ramp transformation to approximate ramp regret matching. That is, if $\rho_s^{1:t-1,\textsc{imm},\textsc{cf}} \in \mathbb{R}^{|\mathcal{A}(s)|}$ is a vector of cumulative immediate counterfactual regrets for the external action transformations at agent state s, then RCFR's function approximator maps s to approximate regrets, $\widetilde{\rho}_s^{1:t-1,\textsc{imm},\textsc{cf}} \in \mathbb{R}^{|\mathcal{A}(s)|}$, which generate's RCFR's immediate strategy at s as $\pi^t(s) = [\widetilde{\rho}_s^{1:t-1,\textsc{imm},\textsc{cf}}]_+ / \langle \mathbf{1}, [\widetilde{\rho}_s^{1:t-1,\textsc{imm},\textsc{cf}}]_+ \rangle$ or uniform random if none of the regret estimates are positive. Fig. 8.1 provides an illustration of the RCFR pipeline from agent state to immediate strategy.

RCFR function approximators are usually trained with regression to one of three types of targets at each agent state s: (i) exact targets, (ii) estimated targets, or (iii) bootstrapped targets. To implement method (i), immediate counterfactual regrets are accumulated exactly in a table and the RCFR function approximator is trained to minimize a loss like the *mean-squared error* (*MSE*) $\|\rho_s^{1:t,\textsc{imm},\textsc{cf}} - \widetilde{\rho}_s^{1:t,\textsc{imm},\textsc{cf}}\|_2^2$. Method (ii) is implemented by constructing unbiased estimates of the cumulative immediate counterfactual regrets, $\hat{\rho}_s^{1:t,\textsc{imm},\textsc{cf}} \in \mathbb{R}^{|\mathcal{A}(s)|}$, *e.g.*, by keeping a reservoir buffer (Vitter 1985) of instantaneous regrets, and minimizing a loss like MSE with respect to these estimated targets. Finally, method (iii) constructs an estimate of the next cumulative immediate counterfactual regrets by bootstrapping off of the current function approximator predictions, *i.e.*, after t rounds, the RCFR function approximator is trained to minimize $\|\widetilde{\rho}_s^{1:t-1,\textsc{imm},\textsc{cf}} + \rho_s^{\textsc{cf}}(\cdot, \pi^t; \sigma^t) - \widetilde{\rho}_s^{1:t,\textsc{imm},\textsc{cf}}\|_2^2$.

The overall RCFR regression objective is the sum or average of agent-state local regression objectives, so any one of these three methods can, in principle, be applied to each agent-state independently. Typically, however, the same method is applied to each agent state. Method (i) is typically only applied to each agent state in small test games since it requires an exact table of regrets spanning the entire agent-state space, in addition to the RCFR function approximator. Method (ii) can be applied practically to each agent state, though the reservoir buffer may have to be large to accurately reproduce the true cumulative regrets and perform well. Method (iii) is the simplest approach as it does not require a supplementary table or buffer, but it places a heavy burden on the expressive power of the function approximator since RCFR can perform poorly if the cumulative regrets are not accurately estimated on each and every round.

To give a bound on the regret of RCFR, Waugh, Morrill, et al. (2015) and my M.Sc. thesis (Morrill 2016) first provide an external regret bound for regression regret matching where a function approximator estimates regrets for each action in a stateless ODP setting. Then we apply Theorem 8 to give an external regret bound when regression regret matching is used at each agent state within CFR, and this is exactly the RCFR algorithm. In both cases, the bound gets smaller as the function approximator more closely approximates the exact cumulative immediate counterfactual regrets.

Theorem 17 (Morrill (ibid., Theorem 3.0.4))**.** *Ramp regression regret matching is an ODP algorithm that chooses an arbitrary strategy on the first round and, on each subsequent round t, chooses $\pi^t \propto [\widetilde{\rho}^{1:t-1}]_+$ according to regret estimates $\widetilde{\rho}^{1:t-1} \in \mathbb{R}^{|\mathcal{A}|}$. This algorithm ensures that, after T rounds, the cumulative regret with respect to each external deviation $\phi \in \Phi^{\text{EX}}_{\mathcal{X}}$ is upper bounded as*

$$\rho^{1:T}(\phi) \leq 2U \sqrt{T|\mathcal{X}| + \underbrace{4U\sqrt{|\mathcal{X}|}\sum_{t=2}^{T}\|[\rho^{1:t-1}]_+ - [\widetilde{\rho}^{1:t-1}]_+\|_1}_{\textit{Slack induced by approximation errors.}}}. \tag{8.1}$$

Corollary 2 (Morrill (ibid., Corollary 3.0.5))**.** *RCFR is CFR with ramp regression regret matching as its local learning algorithm so that its strategy at active agent state $s \in \mathcal{S}_{\mathcal{A}}$ is $\pi^t(s) \propto [\widetilde{\rho}^{1:t-1}(s)]_+$ on round t. Denote the cumulative approximation error in s as $\epsilon^T_s = \sum_{t=2}^{T}\|[\rho^{1:t-1}(s)]_+ - [\widetilde{\rho}^{1:t-1}(s)]_+\|_1$. After T rounds, this algorithm guarantees that cumulative regret with respect to each external deviation $\phi \in \Phi^{\text{EX}}_{\mathcal{X}}$ is upper bounded as*

$$\begin{aligned}\rho^{1:T}(\phi) &\leq \sum_{s \in \mathcal{S}_{\mathcal{A}}} 2U\sqrt{T|\mathcal{A}(s)| + 4U\sqrt{|\mathcal{A}(s)|}\epsilon^T_s} && (8.2)\\ &\leq 2U|\mathcal{S}_{\mathcal{A}}|\sqrt{|n_{\mathcal{A}}| + \underbrace{4U\sqrt{|n_{\mathcal{A}}|}\epsilon^*}_{\textit{Slack induced by approximation errors.}}}, && (8.3)\end{aligned}$$

where $\epsilon^* = \max_{s \in \mathcal{S}_{\mathcal{A}}} \epsilon^T_s$.

The ramp regression regret matching bound of Eq. (8.1) differs from that of exact ramp regret matching only by an additive cumulative approximation error term.[1] The bound of 8.3 likewise differs from CFR's in the same way.

[1] With the exception that $T(|\mathcal{A}| - 1)$ has been replaced with $T|\mathcal{A}|$ since Morrill (2016) did not use the analysis of A. Greenwald, Z. Li, and Marks (2006a).

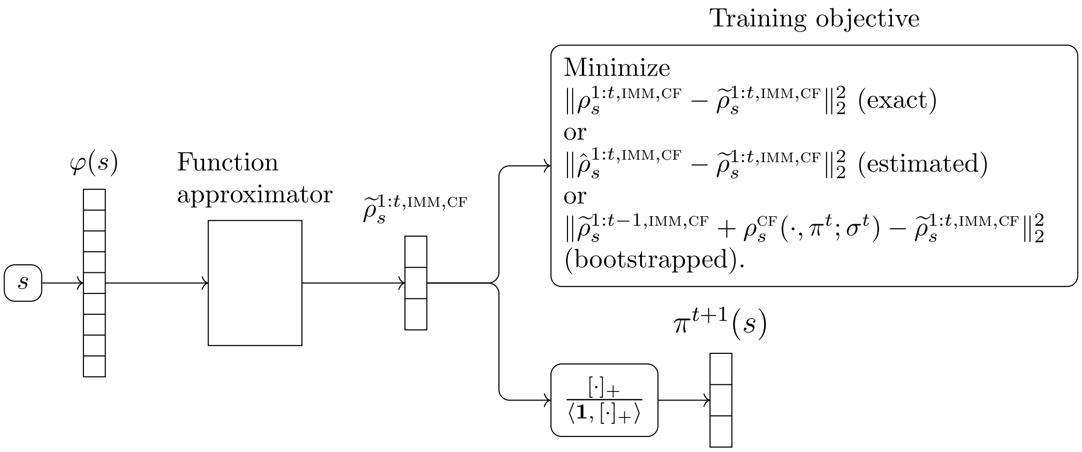

Training objective
Minimize
$\|\rho_s^{1:t,\text{IMM},\text{CF}} - \widetilde{\rho}_s^{1:t,\text{IMM},\text{CF}}\|_2^2$ (exact)
or
$\|\hat{\rho}_s^{1:t,\text{IMM},\text{CF}} - \widetilde{\rho}_s^{1:t,\text{IMM},\text{CF}}\|_2^2$ (estimated)
or
$\|\widetilde{\rho}_s^{1:t-1,\text{IMM},\text{CF}} + \rho_s^{\text{CF}}(\cdot, \pi^t; \sigma^t) - \widetilde{\rho}_s^{1:t,\text{IMM},\text{CF}}\|_2^2$
(bootstrapped).
$\varphi(s)$
Function
approximator
$\widetilde{\rho}_s^{1:t,\text{IMM},\text{CF}}$
s
$\pi^{t+1}(s)$
$\frac{[\cdot]_+}{\langle \mathbf{1}, [\cdot]_+ \rangle}$

Figure 8.1: The RCFR pipeline from agent state to immediate strategy.

The RCFR training objectives presented in this section require exact instantaneous regrets to be computed on each round to construct targets, even if those targets are estimates of the cumulative regret. Another reasonable approach is to compute Monte Carlo estimates of instantaneous regrets and construct RCFR targets on these estimates. Brown, Lerer, et al. (2019), for example, uses a reservoir buffer with Monte Carlo instantaneous regrets.

8.3 Policy Gradient in a POHP

Function approximation has been an integrated into policy gradient since its inception (R. S. Sutton et al. 2000; Williams 1992). With function approximation, policy gradient is an end-to-end learning procedure where function parameters determine action preference outputs and these parameters are trained with backpropagated gradients of accumulated rewards (see Fig. 8.2). Various popular deep reinforcement learning algorithms are based on policy gradient (Espeholt et al. 2018; Lillicrap et al. 2015; Mnih et al. 2016; Schulman, Levine, et al. 2015; Schulman, Wolski, et al. 2017). Policy gradient is also popular as a multi-agent learning algorithm (*e.g.*, Baker et al. (2019), Bansal et al. (2018), J. Foerster, R. Y. Chen,

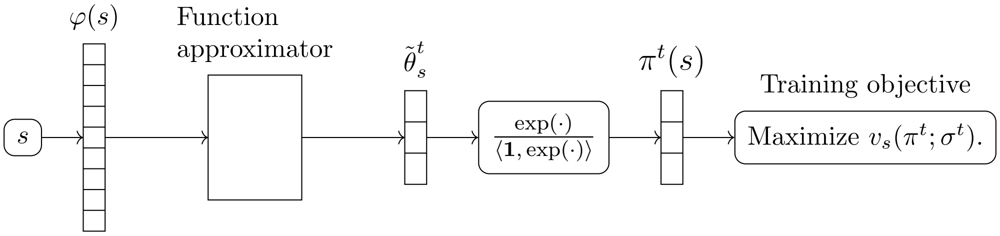

$\varphi(s)$
Function
approximator
$\tilde{\theta}_s^t$
$\pi^t(s)$
Training objective
s
$\frac{\exp(\cdot)}{\langle \mathbf{1}, \exp(\cdot) \rangle}$
Maximize $v_s(\pi^t; \sigma^t)$.

Figure 8.2: The softmax policy gradient pipeline from agent state to immediate strategy in a finite-horizon POHP with timed updates.

et al. (2018), J. N. Foerster et al. (2018), and Lowe et al. (2017)).

In Section 10.4, softmax policy gradient is compared algorithmically and experimentally with a version of RCFR that also uses a softmax policy.

8.4 Uncertain MDPs and Robust Optimization

An *uncertain MDP* or MDP with parameter uncertainty (see, *e.g.*, K. Chen et al. (2012)) is an MDP where the reward function or the transition probability distribution (or both) is a priori unknown to the agent. Model-free reinforcement learning control (see, *e.g.*, R. Sutton et al. (2018)) addresses this problem by having the agent learn a good policy gradually from direct experience with the MDP. If it is difficult for the agent to gain experience in the MDP, *e.g.*, if mistakes from trial and error are costly or dangerous, as in the autonomous driving or medical treatment domains, then such an approach may be infeasible or at least insufficient.

Instead, robust policy optimization constructs policies that are likely to be effective in the MDP without a perfect simulator. The idea is to characterize a belief (probability distribution) about how likely each possible parameterization accurately reproduces the real MDP. Each candidate parameterization can be simulated so we can construct a policy that is effective across this belief. The more likely the true parameterization is under the belief, the more confidence we can have that the policy will also be effective in the real MDP.

To account for belief weight on incorrect parameterizations, robust policy optimization requires a risk measure to induce robustness. For example, the risk measure might dictate that the policy should maximize its performance on the bottom 10% of parameterizations. That is, the policy should perform well on the most challenging 10% of parameterizations sampled from the belief and selected by an adversary.

We can represent an uncertain MDP with a POHP-form MDP except that at the start of the POHP, the (non-chance player's) daimon chooses the parameters that determine the rest of the MDP simulation.

8.5 k-of-N CFR

The *k-of-N CFR* algorithm computes an approximate $\mu_{k\text{-of-}N}$-robust policy, which is a policy that approximately minimizes the k-of-N risk measure, $\mu_{k\text{-of-}N}$ (K. Chen et al. 2012). This Bayesian risk measure is closely related to the classic conditional value at risk (CVaR) measure. By tuning the $k > 0$ and $N \geq k$ parameters, the algorithm designer can set a desired robustness level between worst-case ($k = 1$ and N large) and average-case ($k = N$). As N is increases, $\mu_{k\text{-of-}N}$ approximates the CVaR measure at the k/N percentile. See K. Chen et al. (ibid.) for more details on this risk measure.

k-of-N CFR works by iteratively sampling N parameterizations from a belief and updating the current policy to improve its value under the k-worst reward functions. Here we presuppose that the daimon's expected return is the negative of the agent's so they are incentivized to choose a parameterization on each round that is difficult for the agent's current policy to handle. At the beginning of each round, N parameterizations are sampled from the belief. Then the daimon chooses k of these parameterizations that maximize their value and minimize the agent's. Finally, a single parameterization is sampled uniformly from the k chosen by the daimon to determine the rest of the simulation for this round. The agent cannot observe these preliminary actions. The result is the following optimality guarantee for k-of-N CFR with perfect-recall updates:

Theorem 18. *With probability $1 - p$ for $0 < p \leq 1$, a uniformly sampled policy from $(\pi^t)_{t=1}^T$ k-of-N CFR is an $\varepsilon(T, p)$-approximation to a $\mu_{k\text{-}of\text{-}N}$-robust policy where*

$$\varepsilon(T, p) = 8\left(1 + \frac{2}{\sqrt{p}}\right) d_* U |\mathcal{S}_\mathcal{A}| \sqrt{\frac{n_\mathcal{A}}{pT}}.$$

If there is only reward function uncertainty and non-zero rewards are only provided to the agent at terminal agent states, then only timed updates are required. In both this specific case and in the general case, the k-of-N CFR algorithm is limited by CFR's restriction to finite-horizon POHPs.

References

Baker, B., I. Kanitscheider, T. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch (2019). "Emergent Tool Use From Multi-Agent Autocurricula". In: *International Conference on Learning Representations.*

Bansal, T., J. Pachocki, S. Sidor, I. Sutskever, and I. Mordatch (2018). "Emergent Complexity via Multi-Agent Competition". In: *6th International Conference on Learning Representations.*

Brown, N., A. Lerer, S. Gross, and T. Sandholm (2019). "Deep Counterfactual Regret Minimization". In: *36th International Conference on Machine Learning (ICML 2019)*, pp. 793–802.

Chen, K. and M. Bowling (2012). "Tractable Objectives for Robust Policy Optimization". In: *Advances in Neural Information Processing Systems*, pp. 2069–2077.

Espeholt, L., H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, et al. (2018). "IMPALA: Scalable distributed Deep-RL with importance weighted actor-learner architectures". In: *ICML.*

Foerster, J., R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch (2018). "Learning with opponent-learning awareness". In: *17th International Conference on Autonomous Agents and MultiAgent Systems.* International Foundation for Autonomous Agents and Multiagent Systems, pp. 122–130.

[1] This detail makes k-of-N CFR an instance of CFR-BR (Johanson, Bard, Burch, et al. 2012).

Foerster, J. N., G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson (2018). "Counterfactual multi-agent policy gradients". In: *32nd AAAI Conference on Artificial Intelligence*.

Greenwald, A., Z. Li, and C. Marks (Jan. 2006a). "Bounds for Regret-Matching Algorithms". In: *International Symposium on Artificial Intelligence and Mathematics (ISAIM 2006)*. Fort Lauderdale, Florida, USA.

Johanson, M., N. Bard, N. Burch, and M. Bowling (2012). "Finding Optimal Abstract Strategies in Extensive Form Games". In: *26th AAAI Conference on Artificial Intelligence (AAAI-12)*.

Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra (2015). "Continuous control with deep reinforcement learning". In: *arXiv preprint arXiv:1509.02971*.

Lowe, R., Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch (2017). "Multi-agent actor-critic for mixed cooperative-competitive environments". In: *Advances in Neural Information Processing Systems*, pp. 6379–6390.

Mnih, V., A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu (2016). "Asynchronous Methods for Deep Reinforcement Learning". In: *33rd International Conference on Machine Learning (ICML)*, pp. 1928–1937.

Morrill, D. (2016). "Using Regret Estimation to Solve Games Compactly". Master's thesis. University of Alberta.

Schulman, J., S. Levine, P. Abbeel, M. Jordan, and P. Moritz (2015). "Trust region policy optimization". In: *International Conference on Machine Learning*, pp. 1889–1897.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov (2017). "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347*.

Sutton, R. and A. Barto (2018). *Reinforcement Learning: An Introduction*. 2nd. MIT Press.

Sutton, R. S., D. McAllester, S. Singh, and Y. Mansour (2000). "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Advances in Neural Information Processing Systems 12*. MIT Press, pp. 1057–1063.

Vitter, J. S. (1985). "Random sampling With a Reservoir". In: *ACM Transactions on Mathematical Software (TOMS)* 11.1, pp. 37–57.

Waugh, K., D. Morrill, J. A. Bagnell, and M. Bowling (2015). "Solving Games with Functional Regret Estimation". In: *29th AAAI Conference on Artificial Intelligence (AAAI-15)*. Vol. 29. 1, pp. 2138–2144.

Williams, R. J. (May 1992). "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine Learning* 8.3, pp. 229–256. ISSN: 1573-0565. DOI: 10.1007/BF00992696. URL: https://doi.org/10.1007/BF00992696.

Chapter 9

k-of-N CFR in a Discounted, Continuing MDP

9.1 Introduction

This chapter shows how the k-of-N CFR procedure can be efficiently applied to discounted, continuing MDPs with reward uncertainty. This version of k-of-N CFR does not require a finite horizon and operates in stationary-policy space so its memory requirements and regret bound scale with the number of MDP states rather than the number of agent states under perfect-recall or timed updates. I then show how this procedure can be used to construct policies that automatically behave cautiously in previously unseen agent states in various elementary safety problems.

9.2 Breaking the Limitations of K. Chen et al. (2012)

The key property of CFR is that regret is minimized across all agent states jointly even though CFR only directly minimizes regret at each agent state in isolation (Zinkevich, Johanson, et al. 2007b). K. Chen et al. (2012) prove as a consequence that if the state transition distribution and reward function on each round of a repeated MDP are sampled uniformly from the k-worst of N candidates sampled from a belief, then CFR approximates an optimal policy under the k-of-N robustness measure with respect to that belief. However, the CFR procedure generally requires a finite POHP and perfect-recall updates. K. Chen et al. (ibid.) argue that as long as there is no transition uncertainty (so there may still be reward uncertainty), then it is safe to instead use only timed updates. While this allows for a reduction in the number of necessary agent states, timed updates still lead to an inflation in the number of agent states compared to the number of nominal MDP states (which need not be timed) and all histories must still deterministically terminate.

The first step in avoiding these requirements is to define and examine realization-weighted expected returns for counterfactual deviations in a POHP-form MDP with a finite number of states and an uncertain reward function. Since only the reward function is uncertain, the daimon's only freedom is in their choice of reward function at the beginning of the POHP. In addition, recall that in a POHP-form MDP, the agent's active agent states match the nominal MDP states one-to-one by construction. Thus, we replace the daimon's strategy σ with the reward function chosen by the daimon, r, and the nominal MDP's state transition distribution, $\mathbb{P}[s' \,|\, s, a]$, as a function of active agent states $s', s \in \mathcal{S}_{\mathcal{A}}$ and action $a \in \mathcal{A}(s)$.

The expected return from every history associated with an active agent state given reward function r (provided these expectations exist) are all equal since each active agent state is Markovian. This is the traditional *state value* used in RL literature (*e.g.*, see R. Sutton et al. (2018)). Rather than defining a separate action value function, we use the state value function of transformed policies to capture the same information, *i.e.*, the value of action a from state s is the state value of policy $\phi_s^{\to a}(\pi)$ at s. We denote the state value function of policy π with reward function r as $q_s(\pi; r) = \mathbb{E}[G_h(\pi; r)]$ for an arbitrary history $h \in I(s)$.

The Markovian assumption can be used to construct reach probabilities without the finite history or timed update assumptions. The probability of reaching a given agent state s can be defined by marginalizing the probability of transitioning to s in k actions, *i.e.*,[1]

$$\mathbb{P}_{\pi,\mathbb{P}}[s] = \sum_{\bar{s} \in \mathcal{S}} \underbrace{\sum_{k=0}^{\infty} \mathbb{P}_{\pi,\mathbb{P}}[s, k \,|\, \bar{s}]}_{\mathbb{P}_{\pi,\mathbb{P}}[s \,|\, \bar{s}]} \mathbb{P}_{\pi,\mathbb{P}}[\bar{s} \,|\, s_{\varnothing}]. \tag{9.1}$$

The realization-weighted expected return from each realizable active agent state s is well defined as long as the state value from s is well defined, *i.e.*,

$$\begin{aligned} v_s(\pi; \sigma) &= \mathbb{P}_{\pi,\mathbb{P}}[s] \sum_{h \in I(s)} \frac{\mathbb{P}_{\pi,\mathbb{P}}[h]}{\mathbb{P}_{\pi,\mathbb{P}}[s]} q_s(\pi; r) && (9.2) \\ &= q_s(\pi; r) \underbrace{\sum_{h \in I(s)} \mathbb{P}_{\pi,\mathbb{P}}[h]}_{\mathbb{P}_{\pi,\mathbb{P}}[s]} && (9.3) \\ &= \mathbb{P}_{\pi,\mathbb{P}}[s] q_s(\pi; r). && (9.4) \end{aligned}$$

Without perfect-recall, there may not be a unique counterfactual value because there could be many ways to play to reach a given agent state, each with their own transition probabilities, that lead to a different realization weight for each counterfactual deviation. However, the cumulative regret for each counterfactual deviation can be simultaneously minimized by ignoring these realization weights, thanks to the fact that states are Markovian.

[1] The k-step transition distribution is derived from the agent's policy and the state transition distribution.

Theorem 19. *Let* $\rho^q_s(\phi, \pi; r) = q_s(\phi(\pi); r) - q_s(\pi; r)$. *be the* advantage *(alternatively, regret with respect to the state value function) of deviation* ϕ *over policy* π *under reward function* r *at state* s *in an MDP. The cumulative advantage of action transformation* $\phi_s \in \Phi^{\text{sw}}_{\mathcal{A}(s)}$ *in a repeated uncertain reward MDP upper bounds the cumulative regret of every counterfactual deviation that plays to reach state* s *and employs* ϕ_s *there with respect to the realization-weighted expected return from* s*. Formally,* $\rho^{1:T}_s(\phi^{\to s}) \leq \rho^{1:T,q}_s(\phi_s)$ *for each counterfactual deviation* $\phi^{\to s}$ *where* $\phi^{\to s}_s = \phi_s$*. Consequently, if an agent is hindsight rational for a set of action transformations* $\Phi_s \subseteq \Phi^{\text{sw}}_{\mathcal{A}(s)}$ *with respect to the state value function in each state* s*, then the agent is hindsight rational for the set of* Φ_s*-counterfactual deviations.*

Proof. Let $\phi^{\to s}$ be a counterfactual deviation that plays to reach s and employs ϕ_s once there. A counterfactual deviation is constructed with external transformations leading to s, which implies that every policy is transformed in the same way leading up to s, *i.e.*, $[\phi^{\to s}_{\prec s}\pi](\bar{s}) = [\phi^{\to s}_{\prec s}\pi'](\bar{s})$ for all pairs of policies π, π' and active predecessors $\bar{s} \prec s$. This fact further implies that $\phi^{\to s}(\pi)$ and $\phi^{\to s}(\pi')$ share the same probability of reaching s, *i.e.*, $\mathbb{P}_{\phi^{\to s}(\pi),\mathbb{P}}[s] = \mathbb{P}_{\phi^{\to s}(\pi'),\mathbb{P}}[s]$. The cumulative full regret of $\phi^{\to s}$ from s is therefore

$$\rho^{1:T}_s(\phi^{\to s}) = \sum_{t=1}^{T} \mathbb{P}_{\phi^{\to s}(\pi^t),\mathbb{P}}[s] q_s(\phi_s(\pi^t); r^t) - \mathbb{P}_{\phi^{\to s}(\pi^t),\mathbb{P}}[s] q_s(\pi^t; r^t) \tag{9.5}$$

$$= \mathbb{P}_{\phi^{\to s}(\pi),\mathbb{P}}[s] \sum_{t=1}^{T} \underbrace{q_s(\phi_s(\pi^t); r^t) - q_s(\pi^t; r^t)}_{\rho^q_s(\phi_s, \pi^t; r^t)} \tag{9.6}$$

$$\leq \sum_{t=1}^{T} \rho^q_s(\phi_s, \pi^t; r^t) = \rho^{1:T,q}_s(\phi_s). \tag{9.7}$$

The right-hand-size of Eq. (9.7) is exactly the cumulative advantage of action transformation ϕ_s, which proves the claim. □

Theorem 19 motivates a definition of CFR for continuing MDPs where a learner is deployed at each state to enforce hindsight rationality under the state value function with respect to action transformations, which guarantees hindsight rationality with respect to counterfactual deviations. The next step is to show that there is an analog to the counterfactual deviation specialization of Lemma 1 in uncertain reward MDPs so that CFR is still hindsight rational for the external deviations. Because CFR in this setting is learning with state values, we can see that the undiscounted half of Kakade (2003)'s performance difference lemma (Lemma 5.2.1) is actually an analog of the counterfactual deviation specialization of Lemma 1 for finite-horizon MDPs with reward uncertainty. The other half of the performance difference lemma provides an analogous statement for discounted continuing MDPs. Even-Dar et al. (2005) uses an average reward version of the performance difference lemma to describe and analyze what is effectively CFR for the average reward objective in continuing MDPs.[2] Our analysis instead considers the discounted return objective and directly addresses the regret and robustness of k-of-N CFR in specific.[3]

Define

$$d_s : s'; \pi \mapsto (1-\gamma)\mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i \mathbb{1}\{S_i = s'\} \mid S_0 = s\right],$$

where $A_i \sim \pi(S_{i-1})$ and $S_i \sim \mathbb{P}[\cdot | S_{i-1}, A_i]$ for $i \geq 1$, to be the γ-discounted future state distribution induced by π from initial state s. Kakade (2003)'s performance difference lemma for this setting is:

Lemma 7 (Kakade (ibid.)'s performance difference lemma (discounted))**.** *The regret of external deviation $\phi^{\to\pi'}$ and policy π from state s in a γ-discounted MDP under reward function r is*

$$\rho^q_s(\phi^{\to\pi'}, \pi; r) = \frac{1}{1-\gamma}\mathbb{E}_{S \sim d_s(\cdot;\pi')}\left[\rho^q_S(\phi^{\to\pi'}_S, \pi; r)\right].$$

From Lemma 7, we derive a new regret and optimality bound for CFR and k-of-N CFR, respectively, in continuing discounted MDPs with reward uncertainty. Given a sequence of reward functions, $(r^t)_{t=1}^T$, CFR produces a sequence of policies, $(\pi^t)_{t=1}^T$, and ensures that the cumulative advantage of each action a at each state s grows sublinearly, *i.e.*, $\sum_{t=1}^T \rho^q_s(\phi^{\to a}_s(\pi^t); r^t) \leq f(T) \in \mathrm{o}(T)$ for bound $f(T)$ that depends on the state-local learning algorithm used. For example, using ramp regret matching yields $f(T) = 2\frac{U}{1-\gamma}\sqrt{(|\mathcal{A}|-1)T}$. The $\frac{U}{1-\gamma}$ factor comes from the fact that rewards have a maximum magnitude of U and the unnormalized expected return can be magnified by a factor of $\frac{1}{1-\gamma}$ according to γ-discounting. Combining this with Lemma 7, we arrive at CFR's cumulative external regret bound

Theorem 20. *CFR bounds the cumulative regret of each external deviation $\phi^{\to\pi}$ as $\rho^{1:T}(\phi^{\to\pi}) = \dfrac{f(T)}{1-\gamma}$.*

Proof. Let $S \sim d_s(\cdot; \pi)$ and $A \sim \pi(S)$. By Lemma 7, the linearity of expectation, and CFR's

[2] Actually, since Even-Dar et al. (2005) predates the original CFR work (Zinkevich, Johanson, et al. 2007b), one could say that CFR is a modification of Even-Dar et al.'s MDP experts algorithm to extensive-form games.

[3] The similarity between CFR for the discounted return objective which we analyze here and Even-Dar et al. (2005) analysis for the average reward objective also implies that our analysis of k-of-N CFR algorithm could easily be repeated for the average reward objective, achieving similar regret and robustness guarantees.

definition, the cumulative advantage from each active agent state s is

$$\rho_s^{1:T,q}(\phi^{\to\pi}) = \sum_{t=1}^{T} \rho_s^q(\phi^{\to\pi}, \pi^t; r^t) \tag{9.8}$$

$$= \frac{1}{1-\gamma}\mathbb{E}\left[\sum_{t=1}^{T} \rho_S^q(\phi_S^{\to A}, \pi^t; r^t)\right] \tag{9.9}$$

$$\leq \frac{1}{1-\gamma}\mathbb{E}[f(T)] = \frac{f(T)}{1-\gamma}. \tag{9.10}$$

Since this bound holds for all states simultaneously, it holds from the root state $s_\varnothing$. The advantage in the root state of playing π is equal to the full regret for not playing π, *i.e.*, $\rho_{s_\varnothing}^{1:T,q}(\phi^{\to\pi}) = \rho^{1:T}(\phi^{\to\pi})$, which completes the argument. □

Taking into account Monte Carlo reward function sampling, k-of-N CFR inherits the following regret bound from Monte Carlo CFR.

Theorem 21. *With probability* $1-p$, $p > 0$, *k-of-N CFR's regret of each external deviation* $\phi^{\to\pi}$ *is upper bounded as*

$$\rho^{1:T}(\phi^{\to\pi}) \leq \frac{f(T) + 4U\sqrt{2T\log {}^1\!/\!_p}}{1-\gamma}.$$

Proof. Let the reward function distribution be $\mathcal{R}$. On each iteration, t, of the k-of-N CFR algorithm, N reward functions, $(R_j^t \sim \mathcal{R})_{j=1}^N$ are sampled and the worst k reward functions for the algorithm's current policy, π^t, are mixed into $\bar{R}^t : o \mapsto \frac{1}{k}\sum_{j=1}^k R_{(j)}^t(o)$ for observation $o \in \mathcal{O}$ where $R_{(j)}^t$ is the j^{th}-worst reward function. The k-element average reward function $\bar{R}^t$ is a sample from $\mathcal{R}$ according to the k-of-N probability measure, $\mu_{k\text{-of-}N}$ (see Proposition 1 of K. Chen et al. (2012) for a formal description of this measure's density) where the reward functions are ranked with respect to π^t.

Let $\mu_{k\text{-of-}N}(\cdot; \pi^t)$ denote the k-of-N reward function distribution with respect to policy π^t. The expected return of π^t under the k-of-N robustness objective is then just the expectation of its return under $\bar{R}^t \sim \mu_{k\text{-of-}N}(\cdot; \pi^t)$, *i.e.*, $\mathbb{E}[q_{s_\varnothing}(\pi^t; \bar{R}^t)]$. Then,

$$\rho^{1:T}(\phi^{\to\pi}) = \sum_{t=1}^{T} \mathbb{E}[q_{s_\varnothing}(\pi; \bar{R}^t)] - \mathbb{E}[q_{s_\varnothing}(\pi^t; \bar{R}^t)] \tag{9.11}$$

$$= \sum_{t=1}^{T} \mathbb{E}[\underbrace{q_{s_\varnothing}(\pi; \bar{R}^t) - q_{s_\varnothing}(\pi^t; \bar{R}^t)}_{\rho(\phi^{\to\pi}, \pi^t; \bar{R}^t)}], \tag{9.12}$$

The rest of the proof largely follows the proof of Farina, Kroer, and Sandholm (2020)'s Proposition 1. Since $\mathbb{E}[\rho(\phi^{\to\pi}, \pi^t; \bar{R}^t)]$ is the expectation of $\rho(\phi^{\to\pi}, \pi^t; \bar{R}^t)$, the sequence of differences,

$$\left(\mathbb{E}\Big[\rho(\phi^{\to\pi}, \pi^t; \bar{R}^t)\Big] - \rho(\phi^{\to\pi}, \pi^t; \bar{R}^t)\right)_{t=1}^{T},$$

is a martingale difference sequence. Furthermore,

$$\left|\mathbb{E}[\rho(\phi^{\to\pi}, \pi^t; \bar{R}^t)] - \rho(\phi^{\to\pi}, \pi^t; \bar{R}^t)\right| \leq \frac{4U}{1-\gamma}$$

since a difference in regret can only be four times as large as the largest return and returns are bounded by the largest reward divided by $1-\gamma$.

The probability that the cumulative regret, $\rho^{1:T}(\phi^{\to\pi})$, is bounded by the cumulative sampled regret plus slack $\tau \geq 0$ is bounded according to the Azuma-Hoeffding inequality (Proposition 5 in Section 9.A):

$$\mathbb{P}\left[\rho^{1:T}(\phi^{\to\pi}) = \sum_{t=1}^{T} \rho(\phi^{\to\pi}, \pi^t; \bar{R}^t) + \tau\right] \tag{9.13}$$

$$\leq \mathbb{P}\left[\sum_{t=1}^{T} \mathbb{E}[\rho(\phi^{\to\pi}, \pi^t; \bar{R}^t)] - \rho(\phi^{\to\pi}, \pi^t; \bar{R}^t) \leq \tau\right] \tag{9.14}$$

$$= 1 - \mathbb{P}\left[\sum_{t=1}^{T} \mathbb{E}[\rho(\phi^{\to\pi}, \pi^t; \bar{R}^t)] - \rho(\phi^{\to\pi}, \pi^t; \bar{R}^t) \geq \tau\right] \tag{9.15}$$

$$\leq 1 - \exp\left(\frac{2\tau^2}{4T\left(\frac{4U}{1-\gamma}\right)^2}\right). \tag{9.16}$$

Setting $\tau = \frac{4U}{1-\gamma}\sqrt{2T\log(1/p)}$ ensures that

$$\rho^{1:T}(\phi^{\to\pi}) \leq \sum_{t=1}^{T} \rho(\phi^{\to\pi}, \pi^t; \bar{R}^t) + \frac{4U}{1-\gamma}\sqrt{2T\log 1/p}$$

with probability $1-p$. Since $\sum_{t=1}^{T} \rho(\phi^{\to\pi}, \pi^t; \bar{R}^t) \leq f(T)/(1-\gamma)$,

$$\rho^{1:T}(\phi^{\to\pi}) \leq \frac{f(T) + 4U\sqrt{2T\log 1/p}}{1-\gamma}$$

with probability $1-p$, as required. □

To achieve an optimality approximation bound, we first provide a general result about no-external-regret learning in an arbitrary ODP with an optimal, adversarial daimon:[4]

Lemma 8. *The best strategy in sequence $(\pi^t)_{t=1}^T$ with external regret of $\varepsilon(T)$ when the daimon plays a strategy on each round that minimizes the agent's expected return is an $\varepsilon(T)$-maximin strategy.*

[4] I originally presented this as Lemma 2 in the work of Lockhart et al. (2019a,b).

Proof. Let σ^{π} be the daimon strategy that minimizes the expected return for agent strategy π. Then the average difference between the payoff of a maximin strategy, π, and that of $(\pi^t)_{t=1}^T$ is

$$\begin{aligned} & v_{s_\varnothing}(\pi; \sigma^{\pi}) - \frac{1}{T} \sum_{t=1} v_{s_\varnothing}\left(\pi^t; \sigma^{\pi^t}\right) && (9.17) \\ & = \frac{1}{T}\left(\sum_{t=1}^{T} v_{s_\varnothing}(\pi; \sigma^{\pi}) - \sum_{t=1} v_{s_\varnothing}\left(\pi^t; \sigma^{\pi^t}\right)\right) && (9.18) \\ & \leq \frac{1}{T}\left(\sum_{t=1}^{T} v_{s_\varnothing}(\pi; \sigma^{\pi^t}) - \sum_{t=1} v_{s_\varnothing}\left(\pi^t; \sigma^{\pi^t}\right)\right) && (9.19) \\ & = \varepsilon(T). && (9.20) \end{aligned}$$

Comparing this to the payoff of one of the best strategies in the sequence, $\pi^{t^*} \in \arg\max_{1 \leq t \leq T} v_{s_\varnothing}\left(\pi^t; \sigma^{\pi^t}\right)$, we can see that

$$\begin{aligned} \varepsilon(T) & \geq v_{s_\varnothing}(\pi; \sigma^{\pi}) - \frac{1}{T} \sum_{t=1} v_{s_\varnothing}\left(\pi^t; \sigma^{\pi^t}\right) && (9.21) \\ & \geq v_{s_\varnothing}(\pi; \sigma^{\pi}) - v_{s_\varnothing}\left(\pi^{t^*}; \sigma^{\pi^{t^*}}\right). && (9.22) \end{aligned}$$

Rearranging, we conclude that

$$v_{s_\varnothing}\left(\pi^{t^*}; \sigma^{\pi^{t^*}}\right) \geq v_{s_\varnothing}(\pi; \sigma^{\pi}) - \varepsilon(T), \tag{9.23}$$

which completes the proof. □

Finally, our theoretical inquiry culminates in the following optimality approximation bound for k-of-N CFR policies:

Theorem 22. *With probability $1 - p$, $p > 0$, the best policy in the sequence of policies generated by k-of-N CFR, $(\pi^t)_{t=1}^T$, is an $\varepsilon(T)$-approximation to a $\mu_{k\text{-of-}N}$-robust policy where*

$$\varepsilon(T) = \frac{f(T) + 4U\sqrt{2T \log {}^{1}\!/_{p}}}{(1-\gamma)T}$$

and with probability at least $(1-p)(1-q)$, $q > 0$, a randomly sampled policy from this sequence is an ${}^{\varepsilon(T)}\!/_{q}$-approximation to a $\mu_{k\text{-of-}N}$-robust policy.

Proof. The first half of the proof is essentially that of Lemma 8 except that the daimon can only probabilistically choose a reward function that minimizes the agent's expected payoff. In this case, a maximin policy is a $\mu_{k\text{-of-}N}$-robust policy.

Let $\bar{R}^t \sim \mu_{k\text{-of-}N}(\cdot; \pi^t)$ and let $\bar{R}^{\pi} \sim \mu_{k\text{-of-}N}(\cdot; \pi)$ for any given policy π. Define $q^*_{s_\varnothing} = \max_{\pi^*} \mathbb{E}[q_{s_\varnothing}(\pi^*; \bar{R}^{\pi^*})]$ to be the return of a $\mu_{k\text{-of-}N}$-robust policy, π^*. Since the competitor term of the regret does not depend on the iteration number, we can rewrite the average regret as

$$\varepsilon(T) \geq q^*_{s_\varnothing} - \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[q_{s_\varnothing}(\pi^t; \bar{R}^t)] \tag{9.24}$$

$$\geq q^*_{s_\varnothing} - \max_{1\leq t\leq T} \mathbb{E}[q_{s_\varnothing}(\pi^t; \bar{R}^t)], \tag{9.25}$$

where the last inequality holds with probability $1-p$ according to Theorem 21. Rearranging terms, we see that

$$\max_{1\leq t\leq T} \mathbb{E}[q_{s_\varnothing}(\pi^t; \bar{R}^t)] \geq q^*_{s_\varnothing} - \varepsilon(T).$$

Thus, the best policy in the sequence achieves the optimal k-of-N value minus $\varepsilon(T)$ with probability $1-p$, as required.

The last half of this proof is essentially that of Johanson, Bard, Burch, et al. (2012)'s Theorem 4.

Let $\hat{\pi} \sim \mathrm{Unif}(\{\pi^t\}_{t=1}^T)$ be the random variable representing a uniformly sampled policy. Then,

$$\varepsilon(T) \geq q^*_{s_\varnothing} - \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[q_{s_\varnothing}(\pi^t; \bar{R}^t)] \tag{9.26}$$

$$= \mathbb{E}\big[q^*_{s_\varnothing} - \mathbb{E}[q_{s_\varnothing}(\hat{\pi}; \bar{R}^{\hat{\pi}})]\big], \tag{9.27}$$

with probability $1-p$ according to Theorem 21.

Let $X = q^*_{s_\varnothing} - \mathbb{E}[q_{s_\varnothing}(\hat{\pi}; \bar{R}^{\hat{\pi}})] \geq 0$. By Markov's inequality (Proposition 6 in Section 9.A),

$$\mathbb{E}[X] \geq \frac{\varepsilon(T)}{q}\mathbb{P}\left[X > \frac{\varepsilon(T)}{q} \mid \mathbb{E}[X] \leq \varepsilon(T)\right].$$

Since $\varepsilon(T) > 0$,

$$q \geq \mathbb{P}\left[X > \frac{\varepsilon(T)}{q} \mid \mathbb{E}[X] \leq \varepsilon(T)\right] \tag{9.28}$$

$$= 1 - \mathbb{P}\left[X \leq \frac{\varepsilon(T)}{q} \mid \mathbb{E}[X] \leq \varepsilon(T)\right] \tag{9.29}$$

and thus $\mathbb{P}\left[X \leq \frac{\varepsilon(T)}{q} \mid \mathbb{E}[X] \leq \varepsilon(T)\right] \geq 1-q$. Finally,

$$\mathbb{P}\left[\mathbb{E}[X] \leq \varepsilon(T), X \leq \frac{\varepsilon(T)}{q}\right] = \mathbb{P}\left[X \leq \frac{\varepsilon(T)}{q} \mid \mathbb{E}[X] \leq \varepsilon(T)\right]\mathbb{P}[\mathbb{E}[X] \leq \varepsilon(T)] \tag{9.30}$$

$$= (1-q)(1-p), \tag{9.31}$$

proving that $\hat{\pi}$ is an $\frac{\varepsilon(T)}{q}$-approximation to a $\mu_{k\text{-of-}N}$-robust policy with probability $(1-p)(1-q)$. □

Theorem 22 shows that k-of-N CFR with no-external-regret local learners, *e.g.*, ramp regret matching instances, generates a $\mu_{k\text{-of-}N}$-robust policy with high probability.

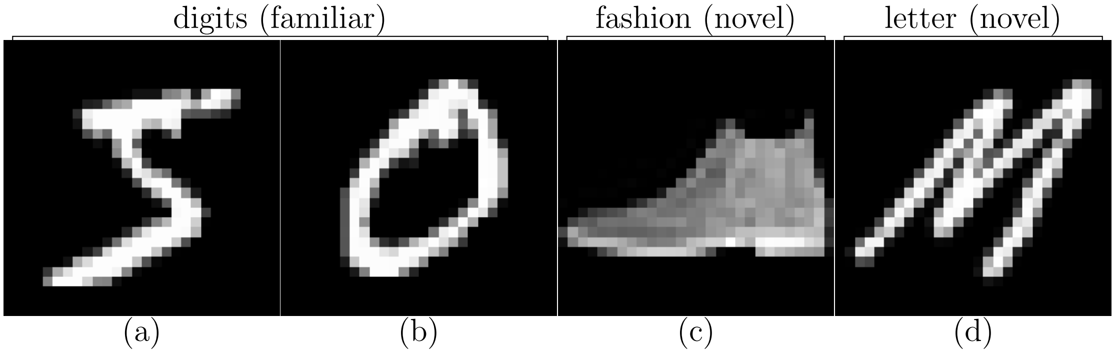


Figure 9.1: From left to right, image of the numbers five and zero from the MNIST digit dataset, an ankle boot from the MNIST fashion dataset, and a capital "M" from the EMNIST letters dataset. In our motivating example, the two images on the left are similar to the ones you have seen before while the two on the right are novel.

9.3 Learning to Be Cautious

9.3.1 A Motivating Example

Consider a decision-making task where you are shown an image and must choose one of eleven actions. The images are hand-drawn digits from MNIST (LeCun et al. 1998, *e.g.*, Fig. 9.1a and b) and you observe a reward of +1 for selecting the action with the index matching the portrayed digit and zero otherwise, except for the eleventh action, which always yields a small reward, +0.25. Now, what do you do when the image is not of a familiar digit but is instead a novel image of a piece of clothing from MNIST fashion (Xiao et al. 2017, *e.g.*, Fig. 9.1c) or a letter from EMNIST letters (Cohen et al. 2017, *e.g.*, Fig. 9.1d)? A natural choice is action eleven, which has always given a reward regardless of the image, while every other action often gave no reward at all. But is this choice common for current AI algorithms?

One obvious approach for choosing the next action is to guess what reward each action will yield with the new image and choose the action with the largest estimate. For example, a nearest-neighbor approach would select a previous image that resembles the new image in some way and use the rewards from the previous image as the reward estimates, effectively extrapolating from familiar to novel images. Considering that the new image looks very different from all the previous ones, an extrapolative approach relies on a questionable premise. Algorithms like this are unlikely to choose action eleven, since its reward was always small and only one of the ten other extrapolations need to look promising for action eleven to be overlooked.

A conventional RL approach like Q-learning (Watkins 1989) or policy gradient (R. S. Sutton et al. 2000; Williams 1992) with function approximation also employs extrapolative guessing and fails to behave cautiously in this task. After training on MNIST digits, a greedy policy with respect to a single neural network model of the reward function (effectively Q-learning) chooses action eleven less than 2% of the time when presented images from MNIST fashion.

A common approach to achieve caution is to provide prior knowledge about what behaviors are safe. For example, we could designate action eleven as a "safe action" and encourage the agent to choose it when observing a non-digit image or when the agent has no strong preference for any other action. Thomas et al. (2019) outlines a general methodology for algorithms of this sort and Kahn et al. (2017) provides a more sophisticated example. Embedding prior knowledge about safety into an algorithm would be easy and effective in this particular task but it is problematic as a general approach because safety is highly task-specific and the design burden becomes worse for complicated tasks where safety guarantees would be most useful. In this vein, we present variations on our MNIST task such that cautious behavior becomes increasingly non-obvious.

An alternative to explicitly specifying cautious behavior or safety incentives is risk-sensitive RL. Broadly, these methods characterize an agent's uncertainty about future rewards of different behaviors, and then choose *robust* behaviors, *i.e.*, those that maximize the agent's reward assuming unfavorable conditions (often with a formal risk measure). There are two types of uncertainty that might be present in a decision-making task, (i) *aleatoric* uncertainty that is stochasticity inherent in the environment, *e.g.*, the agent may be uncertain about the the number that a die will show before it is rolled, and (ii) *epistemic* uncertainty that stems from the agent's lack of certainty about the specific environment, *e.g.*, the agent may be uncertain about a die's probability distribution, not just its outcome.

There are various methods for learning policies that are robust to aleatoric uncertainty (Chow, Ghavamzadeh, et al. 2017; Clements et al. 2019; Tang et al. 2020), but since the mapping from images and actions to rewards is deterministic in our MNIST example task, there is no aleatoric uncertainty to be robust to. Consequently, these methods do not behave differently from extrapolative systems in tasks like this.

Alternatively, if the agent is certain about action eleven's reward and less certain about the rewards of the other actions, then a robust policy would choose action eleven, provided the level of uncertainty is great enough. Thus, epistemic uncertainty has the potential to induce caution.[5] In this case, the agent's beliefs are crafted with the domain in mind to achieve the desired behavior in much the same way as the previously discussed prior knowledge approaches. There are many more sophisticated variations on this idea (Chow, Tamar, et al. 2015; Ghavamzadeh et al. 2016; Petrik et al. 2014; Rigter et al. 2021; Zahavy et al. 2020),

[5] One might be tempted to think that cautious behavior and robustness to epistemic uncertainty are the same. However, as noted, whether robust policies produce cautious behavior is critically dependent on the uncertainty distribution.

but they share similar downsides as prior knowledge approaches.

Our approach, that we detail for the remainder of the chapter, uses robust optimization with a *learned* belief without imposing any task-specific safety information into either component to automatically construct cautious policies. This algorithm learns autonomously to identify and choose cautious behavior that is unique to each task. We evaluate in a sequence of tasks where cautious behavior is increasingly complex. This sequence begins with the MNIST example task described here and escalates to a gridworld driving task that requires sequential decision-making.

9.3.2 The Problem Setting

An agent that interacts with the world and learns from experience will inevitably encounter both familiar and novel situations. We believe that such agents can and should use their previous experience to automatically respond cautiously in novel situations. Our tasks use a simplified formulation of the learning-to-be-cautious problem. The agent's world is separated into the familiar and the novel, each represented as a discounted MDP with a finite set of states.

Extrapolation. Since we are primarily interested in examining the agent's behavior in novel situations about which they have never received feedback, we do not define a reward function for the novel MDP. We assess the agent's behavior in the novel MDP qualitatively. Here we focus on only reward uncertainty; investigating caution with transition uncertainty needs further investigation both theoretically and practically.

A straightforward approach for the agent to formulate goals for the novel MDP is to extrapolate the familiar reward function. Ordinary RL planning algorithms can then be applied to generate a policy that will perform well if the novel and familiar MDPs are very similar. Extrapolation can be done with conventional regression methods, *e.g.*, we can model the reward function as a neural network and train its parameters by applying an optimization algorithm like stochastic gradient descent to minimize the network's mean squared error. A natural approach, given an extrapolated reward function model, $\hat{r}$, is then to behave according to an optimal policy, *e.g.*, in each state s, set $\pi^{\mathrm{Optim}(\hat{r})}(a \mid s) = 1$, where action a is the first action (under an arbitrary ordering) that maximizes $q_s(\cdot, \pi^{\mathrm{Optim}(\hat{r})}; \hat{r})$. This approach will represent a simple non-cautious baseline in our experiments.

Inference. A fundamental problem with extrapolation is that there are typically multiple reward models that match the familiar reward function but differ in novel situations from the novel MDP (*i.e.*, state, action, next state triples not present in the familiar MDP), even within a restricted model class. To address this issue, we can infer a posterior belief (a probability distribution) about which reward models are more reasonable, given a prior belief that describes what it means for a reward model to be "reasonable". Exact Bayesian inference is typically intractable for high dimensional data, but a convenient approximation is to train an ensemble of neural networks, each with unique initialization parameters and trained on independently shuffled familiar examples. Each neural network acts like a sample from a posterior with an implicit prior so that the entire ensemble implicitly characterizes a posterior-like belief. Various previous works (*e.g.*, Heskes et al. (1997), Lakshminarayanan et al. (2017), Lu et al. (2017), Osband et al. (2019), Pearce et al. (2018), and Tibshirani (1996)) have used neural networks in similar ways to characterize uncertainty with connections to proper Bayesian inference.

Robust Optimization. An inference approach characterizes the agent's uncertainty about what reward functions are reasonable in the novel MDP given the familiar MDP, but ordinary RL algorithms cannot make use of this information beyond optimizing for a single reward function generated from the belief (*e.g.*, a sample, the expected posterior, or the maximum a posteriori reward function). Robust policy optimization algorithms however, are designed to learn policies that are robust to such uncertainty.

As described in Section 9.3.1, it is critical to pair robust optimization with an appropriate belief for cautious behavior to emerge. Often this is achieved by manually tailoring the belief to specific aspects of a task, but can we instead use a generic neural network ensemble to induce caution? Consider the belief that such an ensemble would learn from training data where the reward for one action is a constant, as in the example from Section 9.3.1. If, as is common, the training procedure has any preference for neural networks with small weights, then all of the last layer weights corresponding to the constant reward action in all of the neural networks will converge toward zero and their bias terms will converge toward the constant. Since all neural networks agree about the reward for this action in all states, the ensemble belief is always nearly certain about the reward of this action. Uncertainty about the rewards for other actions caused by disagreement between neural networks in the ensemble pushes a robust policy into choosing the constant reward action.

The experiments in the next section show that k-of-N CFR under a neural network ensemble belief can effectively learn to be cautious in various tasks.[6]

9.3.3 Experiments

We now present a sequence of tasks that require agents to automatically learn cautious behavior. Tasks vary in difficulty from one that requires no sequential reasoning and includes a universal cautious action, to one that requires sequential reasoning and where the return

[6] Other algorithms that are robust to epistemic uncertainty, *e.g.*, Chow, Tamar, et al. (2015), Ghavamzadeh et al. (2016), Petrik et al. (2014), and Zahavy et al. (2020) could potentially be used instead of k-of-N CFR.

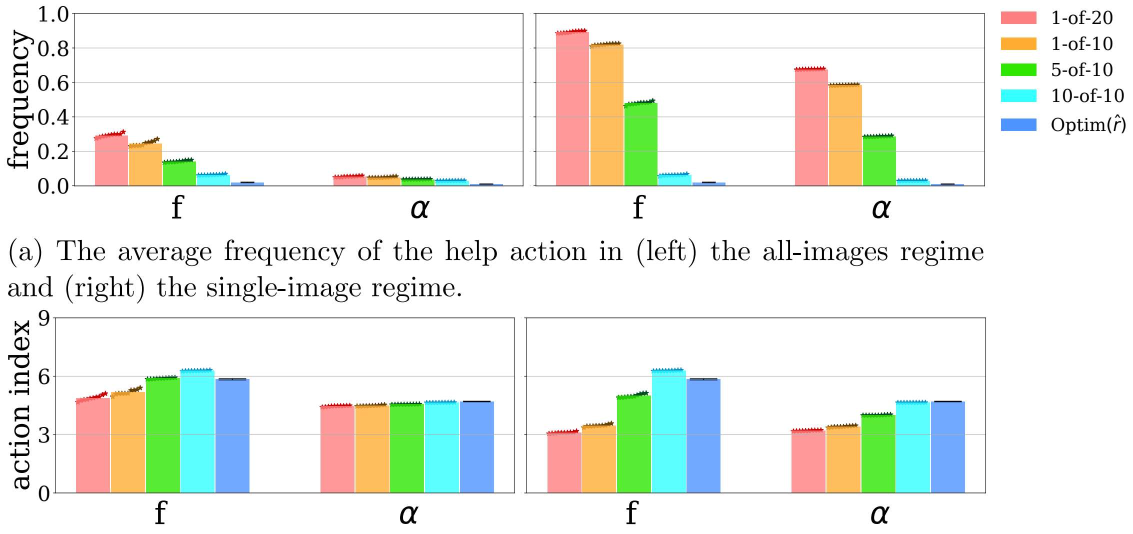
from each action is context dependent, with a natural progression in-between. Experimental design details and hyperparameters for the algorithms tested are provided in Section 9.B.

Learning to Ask for Help. Our first task is the previously described decision making task with MNIST images. The familiar states are the 60,000 training images in the MNIST digit dataset, where the initial state and each next state is sampled uniformly at random. Ten of the actions correspond to a digit label and a reward of +1 is given when the label matches the image and zero otherwise. The eleventh action can be thought of as an "ask for help" option that always receives a reward of +0.25. All action labels are solely to aid our discussion whereas the agent only observes action indices. The discount factor is zero so the agent's return is simply their reward, making this a contextual bandit task.

The k-of-N CFR procedure iteratively improves an approximately robust policy by evaluating it on N samples from a belief updating the policy according to the k-worst samples. Thus, for T iterations, we need to train NT neural networks. We train 2000 reward models on the familiar MDP so that we can run 100 CFR iterations with a maximum $N = 20$. These models also provide the basis for the Optim($\hat{r}$) baseline, where each neural network in the ensemble represents an extrapolated reward model, $\hat{r}$. We set $N = 20$ $k = 1$ for the most robustness, $N = 10$ $k = 1$ for moderate robustness, and $N = 10$ $k = 5$ for marginal robustness. We represent each k-of-N CFR instance with the last policy generated after 100 iterations.

We construct novel MDPs with 10,000 images from the MNIST fashion (Xiao et al. 2017) test set and 20,800 images from EMNIST letters (Cohen et al. 2017) test set (lower and uppercase). Using the set of images as a set of states, we construct two novel MDPs with two different state distribution schemes representing two evaluation scenarios. The first scenario replicates the dynamics of the familiar MDP in that each image is always sampled uniformly. This describes a task where the agent must come up with a policy that works well on all novel images, without emphasizing the performance given any particular one. Our second scenario uses a point-mass initial state distribution and identity transition distribution. This scenario corresponds to a decision-making task where a single crucial novel state is given instead of a distribution over multiple possible novel states. In this scenario, the impact of robustness is exaggerated because the k-of-N CFR policy trains on the k-worst reward functions specifically targeted to a single state rather than the k-worst averaged across many states. Fig. 9.2a shows the results of both experiments.

In both state distribution regimes, the classification accuracy of all policies, including the most robust k-of-N policies, on the 10,000 images in the MNIST digit test set, ranges from 96% to 99%. The two most robust policies, 1-of-20 and 1-of-10, choose the help action 2% and 1% of the time respectively in the single-image regime, but the rest of the policies across both regimes almost never choose the help action. This uniformity in behavior is caused



(a) The average frequency of the help action in (left) the all-images regime and (right) the single-image regime.

(b) Average action index chosen in (left) the all-images regime and (right) the single-image regime.

Figure 9.2: Results for the "learning to ask for help" and "discovering non-obvious cautious actions" tasks in each novel environment ("f" for fashion and "α" for letters).

by the fact that our neural networks effectively generalize to all MNIST digit test images, making the ensemble belief accurate and confident on these images.

The "all fashion images" scenario replicates our motivating example and shows that the help action is utilized more on the fashion images by k-of-N policies as k is decreased (*i.e.*, with more risk aversion), up to 29% of the time for 1-of-20. The Optim($\hat{r}$) baseline is the least likely to use the help action on each novel dataset, and this propensity does not change substantially with the dataset. Decreasing the k/N ratio causes the k-of-N policies to increase the help action frequency on the letter images from 3% to 6%.

In the single-image regime, 1-of-20 selects the help action 89% of the time on the fashion images and 68% on the letter images—46 and 69 times more often, respectively, than the Optim($\hat{r}$) baseline. And when 1-of-20 does not select the help action with the letter images, it does so for letters that resemble digits, *e.g.*, o, s, i, l, j, and z resemble 0, 5, 1, and 2. See Section 9.B for more details, including confusion matrices of selected actions.

Discovering Non-Obvious Cautious Actions. Our next task is to discover non-obvious cautious actions where the value of each action is input-dependent. This time, there are only ten actions and the reward for action indexed as $a \in \{0, \ldots, 9\}$ is $(a + 1)$ if a is the correct label for a given digit image or $^{-(a+2)}/_9$ otherwise. The reward for a correct classification scales with the action index, but so does the cost of misclassification. This reward function also ensures that always choosing action zero has the same expected value as guessing the digit at uniform random assuming a balanced distribution of digit images. Thus, policies that choose lower index actions are more cautious.

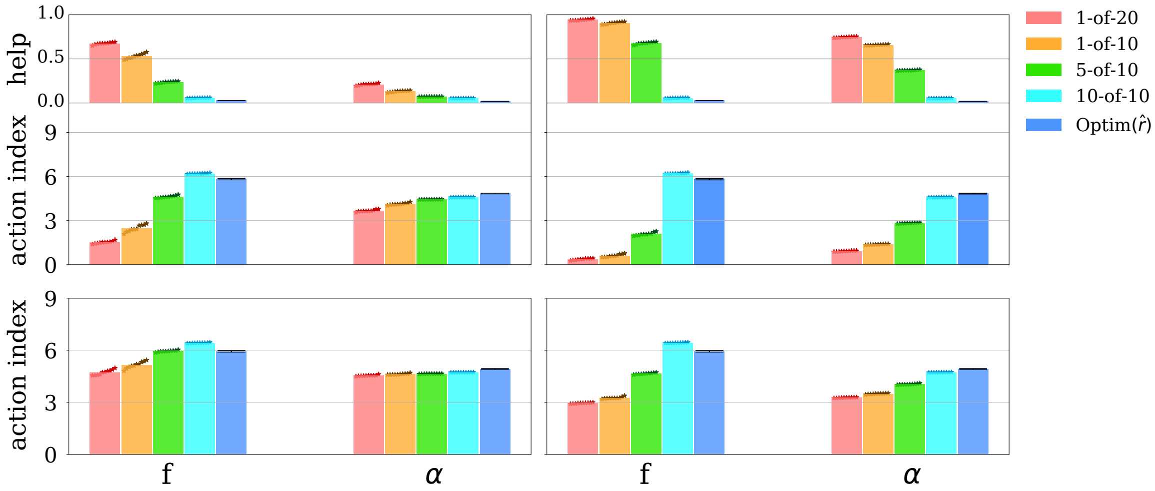


Figure 9.3: Average action index and help action frequency chosen by each method in each novel environment in the “ask for help only when it is available” task. (top row) Help is available, (bottom row) help is unavailable, (left column) the all-images regime, (right column) the single-image regime. All methods essentially never choose the help action when help is unavailable.

Again, we evaluate our approach in two regimes, one where the set of novel states is an MNIST test set and another where evaluation is done on each of these images individually. Fig. 9.2b shows the average action index chosen by each algorithm in each novel environment.

Again, all polices correctly label nearly all test digit images. In both regimes, evaluating on the fashion images, we see that 1-of-20 and 1-of-10 systematically choose smaller actions at lower indices than non-robust algorithms, and 5-of-10 is intermediate between 1-of-20 and 10-of-10 along this metric. The differences are smaller on the letter images in the all-images regime, likely due to many similarities between letter and digit images, but the ordering of methods according to robustness is preserved in both regimes.

Ask for Help Only When it is Available. In the previous scenarios, cautious actions could be identified without taking features of the input into account. Here we modify the previous task where lower index actions are generally more cautious to have an extra action, as in the “learning to ask for help” task, but the value of this action changes depending on an input feature. This feature is a signal that help is available, in which case the “ask for help” action receives a reward of $+{}^{1}/_{20}$. The “ask for help” action is therefore better than any incorrect classification and worse than correctly classifying even the least valuable digit (zero) if there is help available. If help is unavailable, the “ask for help” action is the worst action as it always receives a reward of $-{}^{11}/_{9}$. Fig. 9.3 show the results for each method in the all-images and single-image regimes (left and right, respectively).

Evaluating on fashion images, we see that the robust methods with $k < N$ select the help action much more than the non-robust methods when help is available. When help

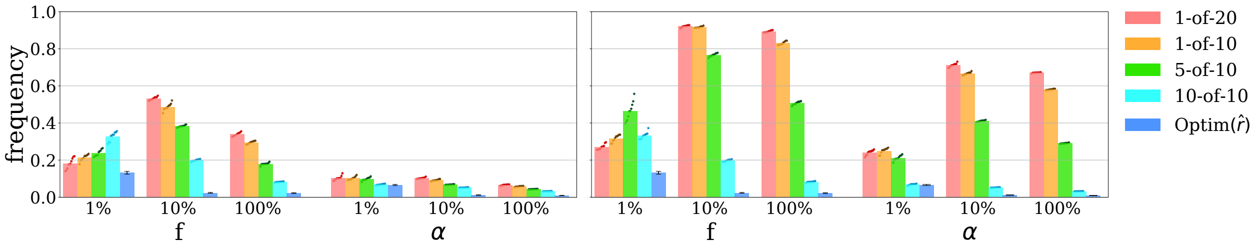


Figure 9.4: The average frequency of the help action in each novel environment on the "learning to ask for help" task with perturbed rewards, where reward models are trained on only 1%, 10%, or 100% of the digit dataset. (left) The all-images regime, (right) the single-image regime.

is unavailable, these methods never select the help action and instead choose actions with smaller indices. The average action index decreases much more when help becomes available because policies switch from choosing actions with high indices to choosing the help action.

How Caution Depends on the Extent of Training Data. Do the k-of-N policies really *learn* to be cautious? Here we investigate how cautious algorithms behave depending on the extent of training data. We repeat the "learning to ask for help" task except that rewards are perturbed by white noise (with standard deviation 0.1) once before neural network training, and the neural network training data varies between 1%, 10%, and 100% of the full digit dataset. Noise is added so that it takes more than a single training example to learn that the expected reward of the help action is constant across training images. Results are shown in Fig. 9.4.

When reward models are trained with 1% of the digit images, we observe that decreasing k to increase robustness does not induce caution. Effectively, the neural network ensemble belief has not seen enough data to infer that the "ask for help" action yields a small but consistent reward. Increasing the training set size to 10%, the correlation between robustness and caution returns and is even stronger than when the full digit dataset is used for training. This shows that caution requires enough training data for the agent to accurately infer the training reward function, and once achieved, the robust agents find cautious behavior.

Driving Gridworld. For a sequential decision making task, we introduce a gridworld driving environment (see Fig. 9.5 for an example frame) in the spirit of the AI safety gridworlds (Leike et al. 2017). A state is a five column image, where the first and second columns represent a two-lane road, the outer two columns represent a ditch, and the last column represents a speedometer. The agent's car is on the bottom row of the image and the world shifts down as the car drives forward. The height of the image represents how far ahead the driver can see. An obstacle randomly appears on the new parts of the road revealed when the car moves forward. To keep the number of states in the gridworld small, only one obstacle can be present on both the left and right halves of the gridworld at a time, and we use a vision

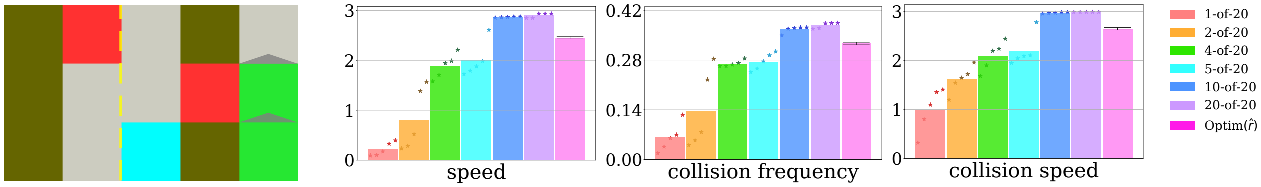


Figure 9.5: Left: a frame from the driving gridworld environment. The cyan square is the car, the red squares are obstacles, and the rightmost column is the car's speedometer. Right: normalized γ-discounted safety statistics for each algorithm in the driving gridworld.

range of two. The car's speed limit is the vision range plus one so that they can "overdrive" their vision by one unit.

The agent has five actions: change lane left, change lane right, accelerate, brake, and cruise. Accelerate and brake increases or decreases the car's speed by one unit, respectively. The car always moves according to its current speed, so the impact of accelerating or braking on the distance the car travels is only felt in later time steps. The car changes lanes one space at a time and changing lanes requires momentum so the car must not be stopped and it travels forward by one fewer space than it would otherwise. The car's speed and lane does not change if the agent chooses to cruise.

The agent's goal is to drive as far from their starting location as possible. As there is no fixed destination, the task is naturally represented as a continuing MDP. The agent receives a reward of $+1$ for each space it moves forward, -2 for each ditch space it moves over, and -2 times the current speed of the car when it drives over an obstacle.

We investigate how our algorithm reacts to novel situations by restricting obstacles to the two ditches in the familiar MDP and allowing them to appear on the road in the novel MDP. We build our ensemble belief by training 2000 neural networks to mimic the familiar MDP's reward function and each k-of-N CFR instance (where $k \in \{1, 2, 4, 5, 10, 20\}$ and $N = 20$) is represented by the last policy generated after 100 iterations.

Fig. 9.5 shows that the more robust policies drive slower, and drive over obstacles both less frequently and at slower speeds in the novel MDP, reflecting intuitively cautious driving behavior. The non-robust policies in contrast almost always drive at full speed.

Why do we see this difference? Since obstacles are never observed on the road in the familiar MDP, there is no clear signal that driving over these obstacles will cause a bad outcome. There is a clear signal however that driving fast on the road yields larger rewards, so the non-robust policies optimize their behavior around this signal, which is reflected in the belief's average reward function. The robust policies instead take the belief's uncertainty about what could happen when the car drives over an obstacle on the road into account. Since some of the reward functions in the ensemble belief generalize from collisions in the ditch to those on the road, the agent learns to avoid collisions altogether in the novel MDP.

9.4 Conclusion

This chapter showed how CFR and k-of-N CFR can be applied to continuing MDPs with uncertain rewards without expanding the MDP state space. This extension allows k-of-N CFR to be applied in more complex robust optimization tasks than previously feasible. We showed how this extension could be utilized in AI safety problems to construct an algorithm that learns to be cautious in unforeseen circumstances.

Our proof of concept algorithm based on a neural network ensemble and k-of-N CFR shows that algorithms can learn to be cautious. Our testbeds are simple, they capture key aspects of AI safety, and they facilitate experimental comparisons. Our hope is that algorithms that learn to be cautious can improve the safety of, and our confidence in, deployed AI systems. However, this level of automated safety is meant to enhance, *not replace*, human judgement and safety planning.

Transition certainty is a strong assumption that will need to be relaxed for most practical applications of these ideas. The increased difficulty of computing robust policies or even minimizing regret with transition uncertainty is discussed by K. Chen et al. (2012) and Even-Dar et al. (2005). It appears an algorithm must search through policies that condition on the entire state history to be sound, which makes policies infeasibly complex in typical environments. Both theoretical and experimental work is required to overcome this hurdle.

Critical limitations of our k-of-N CFR implementation are that it is tabular and requires exact policy evaluation on each iteration to determine the worst-k reward functions. CFR has been used with function approximation (Brown, Lerer, et al. 2019; D'Orazio 2020; D'Orazio, Morrill, et al. 2020; Morrill 2016; Steinberger et al. 2020; Waugh, Morrill, et al. 2015) and approximate worst-case policy evaluation (Davis 2015), so applying these enhancements can allow our approach to scale to more complicated environments.

References

Brown, N., A. Lerer, S. Gross, and T. Sandholm (2019). "Deep Counterfactual Regret Minimization". In: *36th International Conference on Machine Learning (ICML 2019)*, pp. 793–802.

Chen, K. and M. Bowling (2012). "Tractable Objectives for Robust Policy Optimization". In: *Advances in Neural Information Processing Systems*, pp. 2069–2077.

Chow, Y., M. Ghavamzadeh, L. Janson, and M. Pavone (2017). "Risk-constrained reinforcement learning with percentile risk criteria". In: *The Journal of Machine Learning Research* 18.1, pp. 6070–6120.

Chow, Y., A. Tamar, S. Mannor, and M. Pavone (2015). "Risk-sensitive and robust decision-making: a CVaR optimization approach". In: *Neural Information Processing Systems* 28, pp. 1522–1530.

Clements, W. R., B. Van Delft, B.-M. Robaglia, R. B. Slaoui, and S. Toth (2019). “Estimating risk and uncertainty in deep reinforcement learning”. In: *Workshop on Uncertainty and Robustness in Deep Learning at International Conference on Machine Learning*.

Cohen, G., S. Afshar, J. Tapson, and A. Van Schaik (2017). “EMNIST: Extending MNIST to handwritten letters”. In: *International Joint Conference on Neural Networks*, pp. 2921–2926.

D’Orazio, R. (2020). “Regret Minimization with Function Approximation in Extensive-Form Games”. Master’s thesis. University of Alberta.

D’Orazio, R., D. Morrill, J. R. Wright, and M. Bowling (May 2020). “Alternative Function Approximation Parameterizations for Solving Games: An Analysis of f-Regression Counterfactual Regret Minimization”. In: *19th International Conference on Autonomous Agents and Multi-Agent Systems*. International Foundation for Autonomous Agents and Multiagent Systems.

Davis, T. (2015). “Using Response Functions for Strategy Training and Evaluation”. Master’s thesis. University of Alberta.

Dayan, P. (1993). “Improving generalization for temporal difference learning: The successor representation”. In: *Neural Computation* 5.4, pp. 613–624.

Even-Dar, E., S. M. Kakade, and Y. Mansour (2005). “Experts in a Markov decision process”. In: *Advances in Neural Information Processing Systems*, pp. 401–408.

Farina, G., C. Kroer, and T. Sandholm (2020). “Stochastic regret minimization in extensive-form games”. In: *International Conference on Machine Learning*, pp. 3018–3028.

Ghavamzadeh, M., M. Petrik, and Y. Chow (2016). “Safe policy improvement by minimizing robust baseline regret”. In: *Neural Information Processing Systems* 29, pp. 2298–2306.

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

Heskes, T., W. Wiegerinck, and H. Kappen (1997). “Practical confidence and prediction intervals for prediction tasks”. In: *Progress in Neural Processing*, pp. 128–135.

Johanson, M., N. Bard, N. Burch, and M. Bowling (2012). “Finding Optimal Abstract Strategies in Extensive Form Games”. In: *26th AAAI Conference on Artificial Intelligence (AAAI-12)*.

Kahn, G., A. Villaflor, V. Pong, P. Abbeel, and S. Levine (2017). “Uncertainty-aware reinforcement learning for collision avoidance”. In: *arXiv preprint arXiv:1702.01182*.

Kakade, S. M. (2003). “On the sample complexity of reinforcement learning”. PhD thesis. UCL (University College London).

Kingma, D. P. and J. Ba (2014). “Adam: A Method for Stochastic Optimization”. In: *CoRR* abs/1412.6980. URL: http://arxiv.org/abs/1412.6980.

Lakshminarayanan, B., A. Pritzel, and C. Blundell (2017). “Simple and scalable predictive uncertainty estimation using deep ensembles”. In: *Neural Information Processing Systems*, pp. 6405–6416.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). “Gradient-based learning applied to document recognition”. In: *IEEE* 86.11, pp. 2278–2324.

Leike, J., M. Martic, V. Krakovna, P. A. Ortega, T. Everitt, A. Lefrancq, L. Orseau, and S. Legg (2017). “AI safety gridworlds”. In: *arXiv preprint arXiv:1711.09883*.

Lockhart, E., M. Lanctot, J. Pérolat, J.-B. Lespiau, D. Morrill, F. Timbers, and K. Tuyls (2019a). “Computing Approximate Equilibria in Sequential Adversarial Games by Exploitability Descent". In: *28th International Joint Conference on Artificial Intelligence (IJCAI 2019)*.

Lockhart, E., M. Lanctot, J. Pérolat, J.-B. Lespiau, D. Morrill, F. Timbers, and K. Tuyls (2019b). "Computing approximate equilibria in sequential adversarial games by exploitability descent". In: *arXiv preprint arXiv:1903.05614*.

Lu, X. and B. Van Roy (2017). "Ensemble sampling". In: *Neural Information Processing Systems*, pp. 3260–3268.

McDiarmid, C. (1998). "Concentration". In: *Probabilistic methods for algorithmic discrete mathematics*, pp. 195–248.

Morrill, D. (2016). "Using Regret Estimation to Solve Games Compactly". Master's thesis. University of Alberta.

Osband, I., B. Van Roy, D. J. Russo, and Z. Wen (2019). "Deep exploration via randomized value functions". In: *Journal of Machine Learning Research* 20.124, pp. 1–62.

Paszke, A. et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*, pp. 8024–8035.

Pearce, T., M. Zaki, and A. Neely (2018). "Bayesian neural network ensembles". In: *Workshop on Bayesian Deep Learning, Neural Information Processing Systems*.

Petrik, M. and D. Subramanian (2014). "RAAM: The benefits of robustness in approximating aggregated MDPs in reinforcement learning". In: *Advances in Neural Information Processing Systems* 27, pp. 1979–1987.

Reddi, S. J., S. Kale, and S. Kumar (2018). "On the Convergence of Adam and Beyond". In: *International Conference on Learning Representations*.

Rigter, M., B. Lacerda, and N. Hawes (2021). "Risk-averse Bayes-adaptive reinforcement learning". In: *arXiv preprint arXiv:2102.05762*.

Steinberger, E., A. Lerer, and N. Brown (2020). "DREAM: Deep regret minimization with advantage baselines and model-free learning". In: *arXiv preprint arXiv:2006.10410*.

Sutton, R. and A. Barto (2018). *Reinforcement Learning: An Introduction*. 2nd. MIT Press.

Sutton, R. S., D. McAllester, S. Singh, and Y. Mansour (2000). "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Advances in Neural Information Processing Systems 12*. MIT Press, pp. 1057–1063.

Tang, Y. C., J. Zhang, and R. Salakhutdinov (2020). "Worst cases policy gradients". In: *Conference on Robot Learning*, pp. 1078–1093.

Thomas, P. S., B. C. da Silva, A. G. Barto, S. Giguere, Y. Brun, and E. Brunskill (2019). "Preventing undesirable behavior of intelligent machines". In: *Science* 366.6468, pp. 999–1004.

Tibshirani, R. (1996). "A comparison of some error estimates for neural network models". In: *Neural Computation* 8.1, pp. 152–163.

Watkins, C. J. C. H. (1989). "Learning from delayed rewards". In:

Waugh, K., D. Morrill, J. A. Bagnell, and M. Bowling (2015). "Solving Games with Functional Regret Estimation". In: *29th AAAI Conference on Artificial Intelligence (AAAI-15)*. Vol. 29. 1, pp. 2138–2144.

Williams, R. J. (May 1992). "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine Learning* 8.3, pp. 229–256. ISSN: 1573-0565. DOI: 10.1007/BF00992696. URL: https://doi.org/10.1007/BF00992696.

Xiao, H., K. Rasul, and R. Vollgraf (2017). “Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms”. In: *arXiv preprint arXiv:1708.07747*. MIT license.

Zahavy, T., A. Barreto, D. J. Mankowitz, S. Hou, B. O’Donoghue, I. Kemaev, and S. Singh (2020). “Discovering a set of policies for the worst case reward”. In: *International Conference on Learning Representations*.

Zinkevich, M., M. Johanson, M. Bowling, and C. Piccione (Dec. 2007b). “Regret Minimization in Games with Incomplete Information”. In: *Advances in Neural Information Processing Systems (NeurIPS 2007)*. Vancouver, British Columbia, pp. 1729–1736.

9.A Elementary Supplementary Propositions

We make use of the Azuma-Hoeffding inequality in the k-of-N CFR regret bound so it is restated here for completeness:

Proposition 5 (Azuma-Hoeffding inequality). *For constants $(c^t)_{t=1}^T$, martingale difference sequence $(Y^t)_{t=1}^T$ where $|Y^t| \leq c^t$ for each t, and $\tau \geq 0$,*

$$\mathbb{P}\left[\left|\sum_{t=1}^{T} Y^t\right| \geq \tau\right] \leq 2\exp\left(\frac{-\tau^2}{2\sum_{t=1}^{T}(c^t)^2}\right).$$

For proof, see that of Theorem 3.14 by McDiarmid (1998).

The k-of-N CFR optimality bound makes use of another elementary result, Markov's inequality:

Proposition 6 (Markov's inequality). *The probability that non-negative random variable $X \geq 0$ is at least $a > 0$ is upper bounded by X's expectation divided by a,* i.e., $\mathbb{P}[X > a] \leq {}^{\mathbb{E}[X]}/_{a}$.

Proof. By the law of total expectation,

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X \mid X \leq a]] \tag{9.32}$$

$$= \mathbb{P}[X \leq a]\mathbb{E}[X \mid X \leq a] + \mathbb{P}[X > a]\mathbb{E}[X \mid X > a]. \tag{9.33}$$

Since X is non-negative and that $\mathbb{E}[X \mid X > a]$ conditions on X being no-smaller than a,

$$\geq \mathbb{P}[X \leq a]0 + \mathbb{P}[X > a]a \tag{9.34}$$

$$= \mathbb{P}[X > a]a. \tag{9.35}$$

Dividing both sides by a yields the desired statement. □

9.B Supplementary Experimental Details and Results

In all experiments, k-of-N CFR is implemented with regret matching (Hart et al. 2000), which is deterministic and hyperparameter-free. However, since k-of-N CFR requires sampling N reward functions, its output policy is random. Each of the 2000 trained neural network reward function models represents a single sample from an implicit belief, so sampling N of them consists of pulling N of these reward function models out of a queue. To account for the random variation caused by the ordering of the reward function models in the queue, we run multiple repetitions of k-of-N CFR by shuffling the order of the reward function models in the queue before the start of each run. We run ten repetitions in each MNIST experiment and five in the driving gridworld experiment.

Table 9.1: The batch size and number of epochs used to train the neural network reward models for each setting in the "how caution depends on the extent of training data" experiment. All other MNIST experiments use the same settings as in the 100% case.

training data fraction	batch size	# of epochs
1%	64	10,000
10%	128	1,000
100%	512	100

The progress of each k-of-N CFR policy, measured in terms of expected return on the N reward functions used on each CFR iteration, is given for each MNIST experiment in Figs. 9.B.6, 9.B.9, 9.B.12 and 9.B.16 to 9.B.18. The progress of k-of-N CFR in the driving gridworld is similar. The value always plateaus relatively quickly with little variation between runs, indicating that running more iterations or more repetitions would not change the results substantially. Because each run uses different sets of N reward functions on each iteration by design, the value would still show some variation even if the policies for different runs were identical.

PyTorch (Paszke et al. 2019) is used to build and train all neural networks.

MNIST Experiments. For MNIST experiments, we tested three neural network architectures. One used four fully connected layers separated by ramp/rectified linear unit (ReLU) activations and a second used two convolutional layers each with one output channel followed by two fully connected layers. These architectures were outperformed by one that begins with two convolutional layers and ends three fully connected layers, all separated by ReLU activations. The first convolutional layer has a single input channel and 64 output channels with 4×4 kernel followed by 2×2 max-pooling. The second convolutional layer is the same except it has only 16 output channels. The fully connected layers have 50, 15, and 10 outputs, respectively. All results use this architecture.

Networks are trained to minimize the mean-squared error (MSE) between reward predictions and target rewards with the Adam optimizer Kingma et al. 2014 using a learning rate of 0.0016 (we also try 0.01, and 0.001). The remaining parameters for Adam in PyTorch (β_1, β_2, ϵ, and weight decay) are set to their defaults (0.9, 0.999, 10^{-8}, 0) without the AMSGrad (Reddi et al. 2018) modification. In the "discovering non-obvious cautious actions" and "ask for help only when it is available" experiment, we weight the loss on each output index $a \in \{0, \ldots, 9\}$ according to $1/(a+1)^2$ and weight the help action by one. See Table 9.1 for the batch sizes and the number of epochs run in each MNIST experiment.

For all MNIST experiments, we used an NVIDIA Tesla V100 GPU and a 2.2GHz Intel®

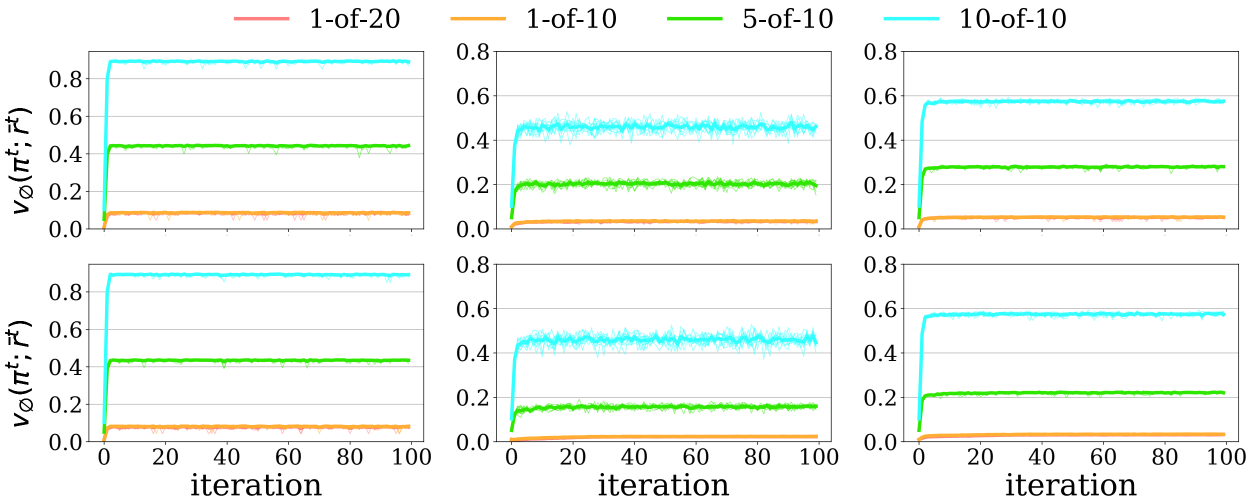


Figure 9.B.6: Expected return of each k-of-N policy on each iteration t given the sampled k-of-N reward function, $\bar{r}^t$, in the "learning to ask for help" task. A single bold line shows the average across all ten runs while the values from individual runs are given by thinner lines. (top row) All-images regime, (bottom row) single-image regime, (left column) digits, (middle column) fashion, (right column) letter.

Xeon® CPU with 100 GB memory. Since we use a neural network ensemble with 2,000 models for each experiment it takes about 50 GPU hours for each experiment, which makes a total of 300 GPU hours for all of our MNIST experiments.

The heatmaps Figs. 9.B.8, 9.B.11, 9.B.15, 9.B.19, 9.B.23 and 9.B.24 show where policies reasonably conflate some letters with digits.

The most cautious policy (1-of-20) in the all-images regime selects the help action upon observing most letters except for those similar to digits (*e.g.*, I/i, L/l, O/o, S/s and Z/z) as shown in Fig. 9.B.8. The effect is exaggerated in the single-image regime where 1-of-20 selects the help action with very high probability except for letters similar to digits. The baseline Optim($\hat{r}$) does not select the help action at all.

The most cautious policy (1-of-20) in the all-images regime picks the help action when help is available and otherwise selects less risky actions with small indices upon observing most letters except for those similar to digits, as shown in Fig. 9.B.15. The effect is exaggerated in the single-image regime.

The Driving Gridworld Experiment. For the driving gridworld experiment, the training data for our neural networks are generated by enumerating all of the (state, next state, reward)-tuples in the familiar driving gridworld where obstacles can only appear in either of the two ditch lanes on the far left or far right column. The action is omitted because the familiar driving gridworld's reward function depends only on the initial and next state. Each driving gridworld state image is pre-processed into a four channel image where each

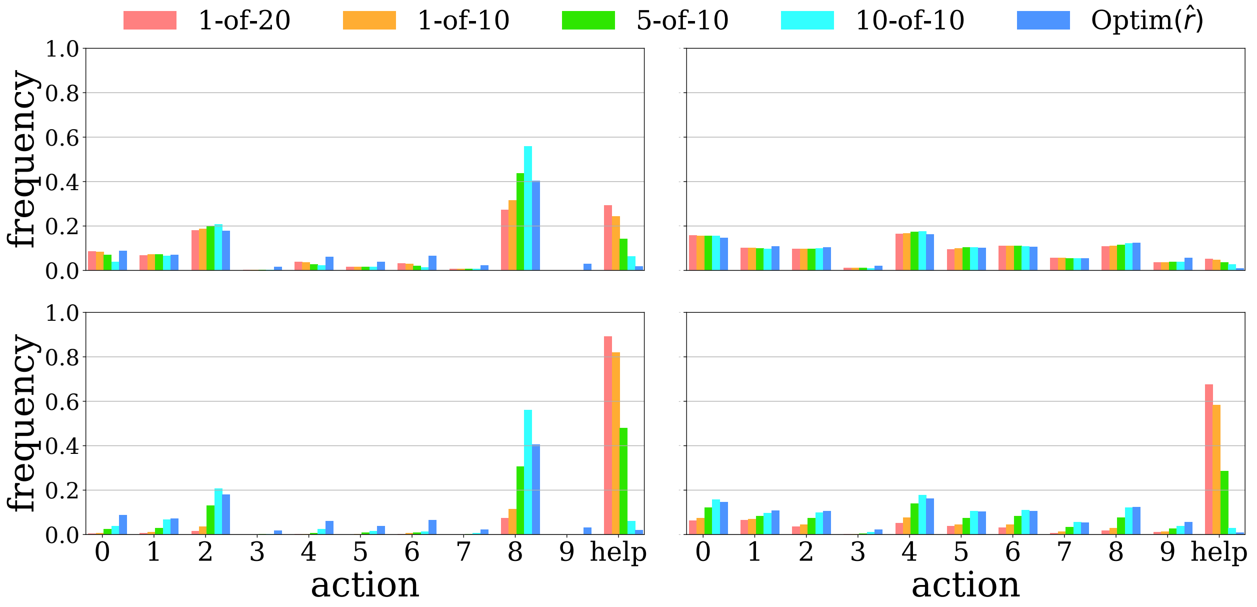
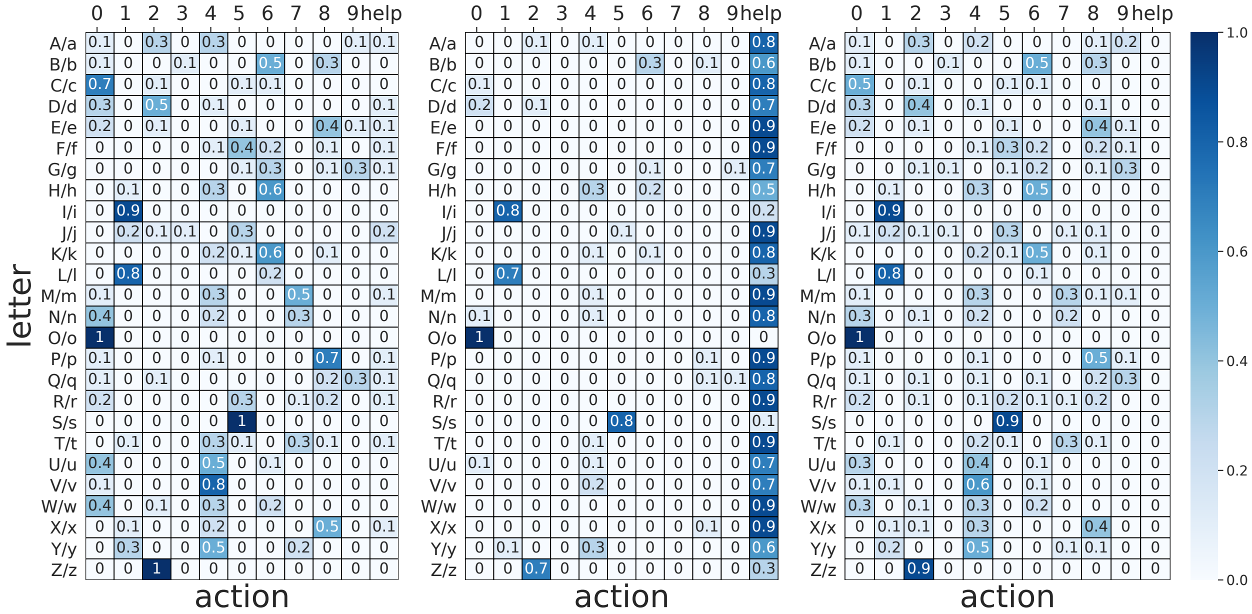


Figure 9.B.7: Action distribution of each k-of-N policy and baseline in the "learning to ask for help" task. (top row) All-images regime, (bottom row) single-image regime, (left column) fashion, (right column) letter.



letter	0	1	2	3	4	5	6	7	8	9	help
A/a	0.1	0	0.3	0	0.3	0	0	0	0	0.1	0.1
B/b	0.1	0	0	0.1	0	0	0.5	0	0.3	0	0
C/c	0.7	0	0.1	0	0	0.1	0.1	0	0	0	0
D/d	0.3	0	0.5	0	0.1	0	0	0	0	0	0.1
E/e	0.2	0	0.1	0	0	0.1	0	0	0.4	0.1	0.1
F/f	0	0	0	0	0.1	0.4	0.2	0	0.1	0	0.1
G/g	0	0	0	0	0	0.1	0.3	0	0.1	0.3	0.1
H/h	0	0.1	0	0	0.3	0	0.6	0	0	0	0
I/i	0	0.9	0	0	0	0	0	0	0	0	0
J/j	0	0.2	0.1	0.1	0	0.3	0	0	0	0	0.2
K/k	0	0	0	0	0.2	0.1	0.6	0	0.1	0	0
L/l	0	0.8	0	0	0	0	0.2	0	0	0	0
M/m	0.1	0	0	0	0.3	0	0	0.5	0	0	0.1
N/n	0.4	0	0	0	0.2	0	0	0.3	0	0	0
O/o	1	0	0	0	0	0	0	0	0	0	0
P/p	0.1	0	0	0	0.1	0	0	0	0.7	0	0.1
Q/q	0.1	0	0.1	0	0	0	0	0	0.2	0.3	0.1
R/r	0.2	0	0	0	0	0.3	0	0.1	0.2	0	0.1
S/s	0	0	0	0	0	1	0	0	0	0	0
T/t	0	0.1	0	0	0.3	0.1	0	0.3	0.1	0	0.1
U/u	0.4	0	0	0	0.5	0	0.1	0	0	0	0
V/v	0.1	0	0	0	0.8	0	0	0	0	0	0
W/w	0.4	0	0.1	0	0.3	0	0.2	0	0	0	0
X/x	0	0.1	0	0	0.2	0	0	0	0.5	0	0.1
Y/y	0	0.3	0	0	0.5	0	0	0.2	0	0	0
Z/z	0	0	1	0	0	0	0	0	0	0	0

letter	0	1	2	3	4	5	6	7	8	9	help
A/a	0	0	0.1	0	0.1	0	0	0	0	0	0.8
B/b	0	0	0	0	0	0	0.3	0	0.1	0	0.6
C/c	0.1	0	0	0	0	0	0	0	0	0	0.8
D/d	0.2	0	0.1	0	0	0	0	0	0	0	0.7
E/e	0	0	0	0	0	0	0	0	0	0	0.9
F/f	0	0	0	0	0	0	0	0	0	0	0.9
G/g	0	0	0	0	0	0	0.1	0	0	0.1	0.7
H/h	0	0	0	0	0.3	0	0.2	0	0	0	0.5
I/i	0	0.8	0	0	0	0	0	0	0	0	0.2
J/j	0	0	0	0	0	0.1	0	0	0	0	0.9
K/k	0	0	0	0	0.1	0	0.1	0	0	0	0.8
L/l	0	0.7	0	0	0	0	0	0	0	0	0.3
M/m	0	0	0	0	0.1	0	0	0	0	0	0.9
N/n	0.1	0	0	0	0.1	0	0	0	0	0	0.8
O/o	1	0	0	0	0	0	0	0	0	0	0
P/p	0	0	0	0	0	0	0	0	0.1	0	0.9
Q/q	0	0	0	0	0	0	0	0	0.1	0.1	0.8
R/r	0	0	0	0	0	0	0	0	0	0	0.9
S/s	0	0	0	0	0	0.8	0	0	0	0	0.1
T/t	0	0	0	0	0.1	0	0	0	0	0	0.9
U/u	0.1	0	0	0	0.1	0	0	0	0	0	0.7
V/v	0	0	0	0	0.2	0	0	0	0	0	0.7
W/w	0	0	0	0	0	0	0	0	0	0	0.9
X/x	0	0	0	0	0	0	0	0	0.1	0	0.9
Y/y	0	0.1	0	0	0.3	0	0	0	0	0	0.6
Z/z	0	0	0.7	0	0	0	0	0	0	0	0.3

letter	0	1	2	3	4	5	6	7	8	9	help
A/a	0.1	0	0.3	0	0.2	0	0	0	0.1	0.2	0
B/b	0.1	0	0	0.1	0	0	0.5	0	0.3	0	0
C/c	0.5	0	0.1	0	0	0.1	0.1	0	0	0	0
D/d	0.3	0	0.4	0	0.1	0	0	0	0.1	0	0
E/e	0.2	0	0.1	0	0	0.1	0	0	0.4	0.1	0
F/f	0	0	0	0	0.1	0.3	0.2	0	0.2	0.1	0
G/g	0	0	0.1	0.1	0	0.1	0.2	0	0.1	0.3	0
H/h	0	0.1	0	0	0.3	0	0.5	0	0	0	0
I/i	0	0.9	0	0	0	0	0	0	0	0	0
J/j	0.1	0.2	0.1	0.1	0	0.3	0	0.1	0.1	0	0
K/k	0	0	0	0	0.2	0.1	0.5	0	0.1	0	0
L/l	0	0.8	0	0	0	0	0.1	0	0	0	0
M/m	0.1	0	0	0	0.3	0	0	0.3	0.1	0.1	0
N/n	0.3	0	0.1	0	0.2	0	0	0.2	0	0	0
O/o	1	0	0	0	0	0	0	0	0	0	0
P/p	0.1	0	0	0	0.1	0	0	0	0.5	0.1	0
Q/q	0.1	0	0.1	0	0.1	0	0.1	0	0.2	0.3	0
R/r	0.2	0	0.1	0	0.1	0.2	0.1	0.1	0.2	0	0
S/s	0	0	0	0	0	0.9	0	0	0	0	0
T/t	0	0.1	0	0	0.2	0.1	0	0.3	0.1	0	0
U/u	0.3	0	0	0	0.4	0	0.1	0	0	0	0
V/v	0.1	0.1	0	0	0.6	0	0.1	0	0	0	0
W/w	0.3	0	0.1	0	0.3	0	0.2	0	0	0	0
X/x	0	0.1	0.1	0	0.3	0	0	0	0.4	0	0
Y/y	0	0.2	0	0	0.5	0	0	0.1	0.1	0	0
Z/z	0	0	0.9	0	0	0	0	0	0	0	0

Figure 9.B.8: The average frequency of each action on each letter, averaged over lowercase and uppercase images, in the "learning to ask for help" task. (left) 1-of-20 all-images regime, (middle) 1-of-20 single-image regime, (right) Optim($\hat{r}$).

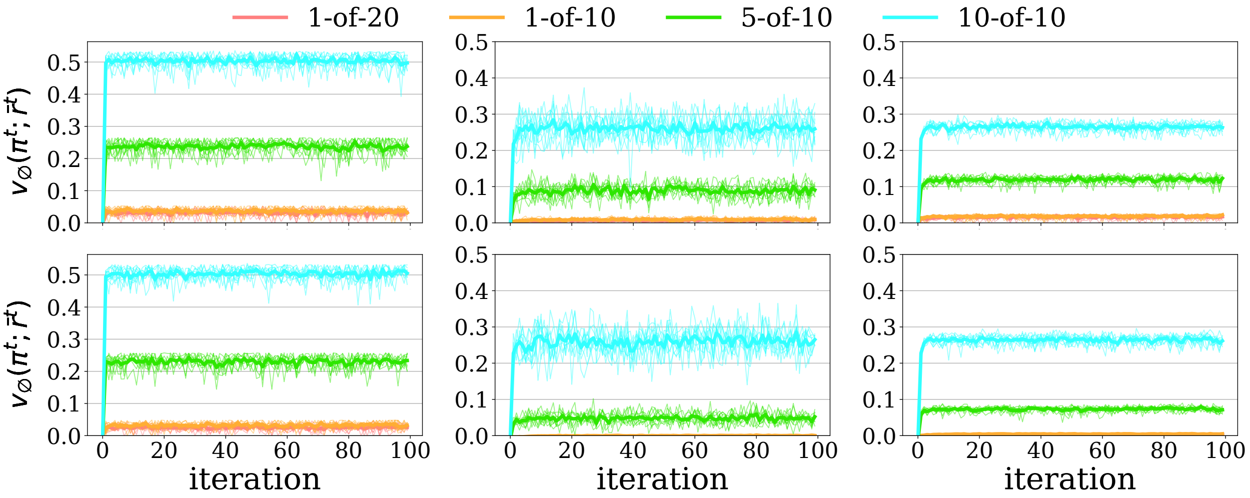


Figure 9.B.9: Expected return of each k-of-N policy on each iteration t given the sampled k-of-N reward function, $\bar{r}^t$, in the "discovering non-obvious cautious actions" task. (top row) All-images regime, (bottom row) single-image regime, (left column) digits, (middle column) fashion, (right column) letter.

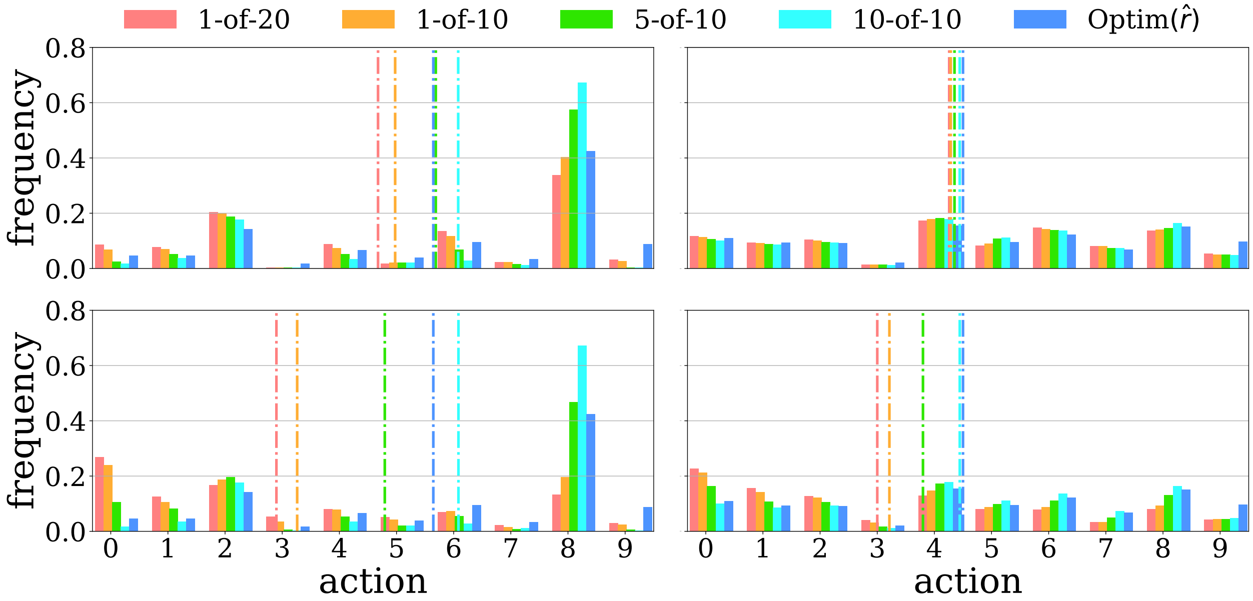


Figure 9.B.10: Action distribution of each k-of-N policy and baseline in the "discovering non-obvious cautious actions" task, dotted lines represent average action taken by each policy. (top row) All-images regime, (bottom row) single-image regime, (left column) fashion, (right column) letter.

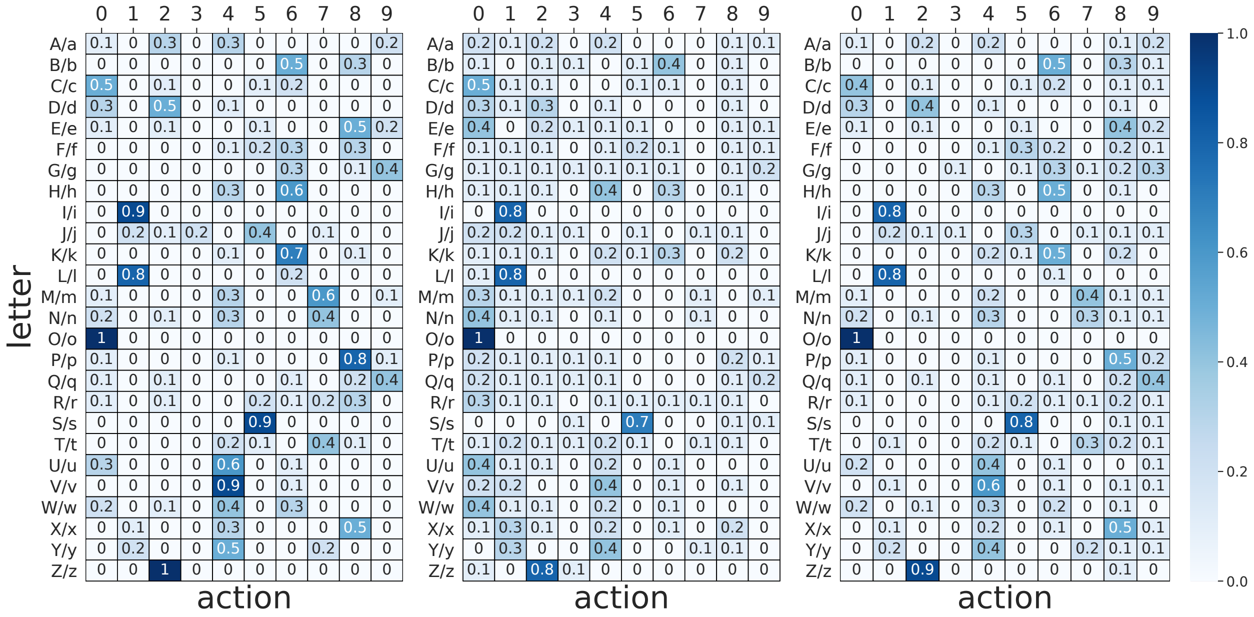


Figure 9.B.11: The average frequency of each action on each letter, averaged over lowercase and uppercase images, in the "discovering non-obvious cautious actions" task. (left) 1-of-20 all-images regime, (middle) 1-of-20 single-image regime, (right) Optim($\hat{r}$).

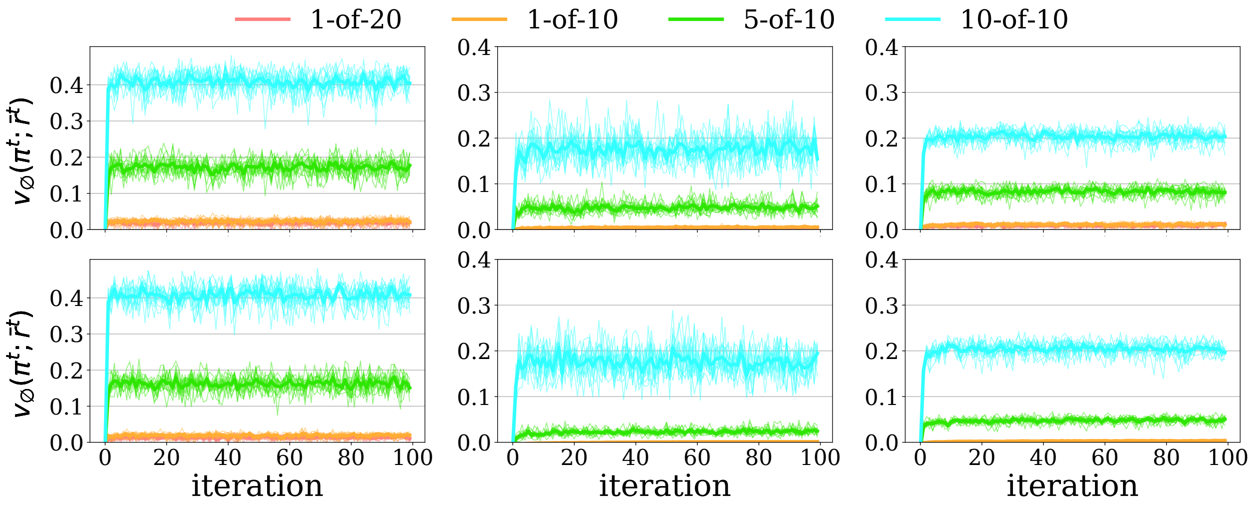


Figure 9.B.12: Expected return of each k-of-N policy on each iteration t given the sampled k-of-N reward function, $\bar{r}^t$, in the "ask for help only when it is available" task. (top row) All-images regime, (bottom row) single-image regime, (left column) digits, (middle column) fashion, (right column) letter.

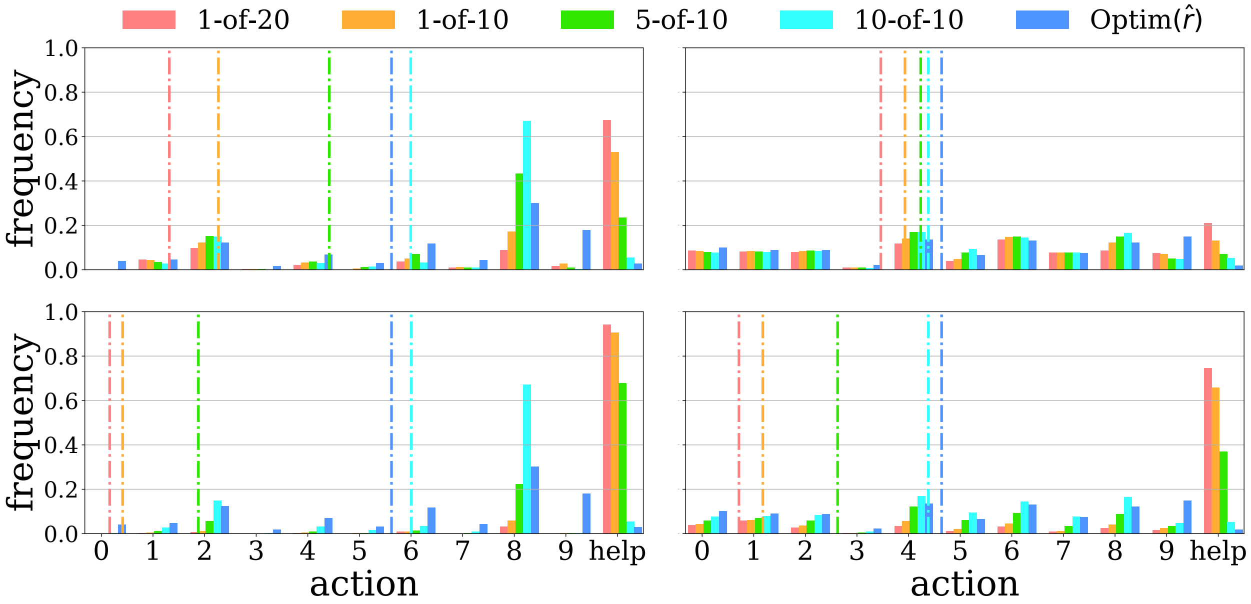


Figure 9.B.13: Actions distribution for each k-of-N policy and baseline in the “ask for help only when it is available” task in case that **help is available**. dotted lines represent average action taken by each policy. (top row) All-images regime, (bottom row) single-image regime, (left column) fashion, (right column) letter.

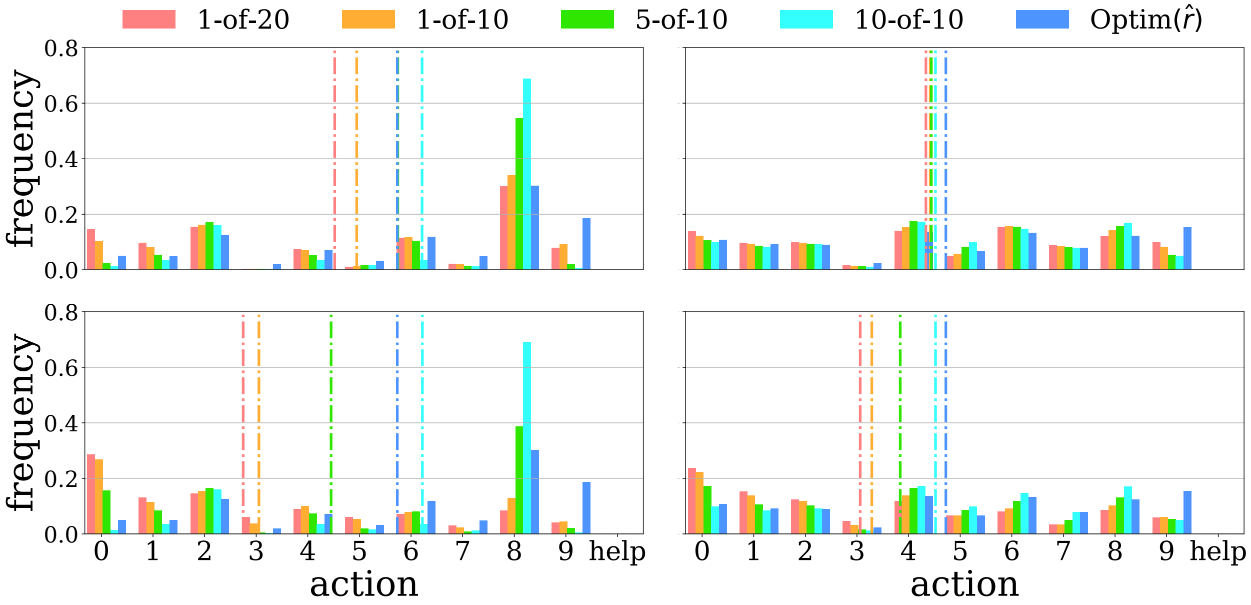


Figure 9.B.14: Actions distribution for each k-of-N policy and baseline in the “ask for help only when it is available” task in case that **help is unavailable**. dotted lines represent average action taken by each policy. (top row) All-images regime, (bottom row) single-image regime, (left column) fashion, (right column) letter.

Table 9.2: Frequency of the correct label action index and the help action across the 10,000 MNIST test images in the "ask for help only when it is available" all-images regime.

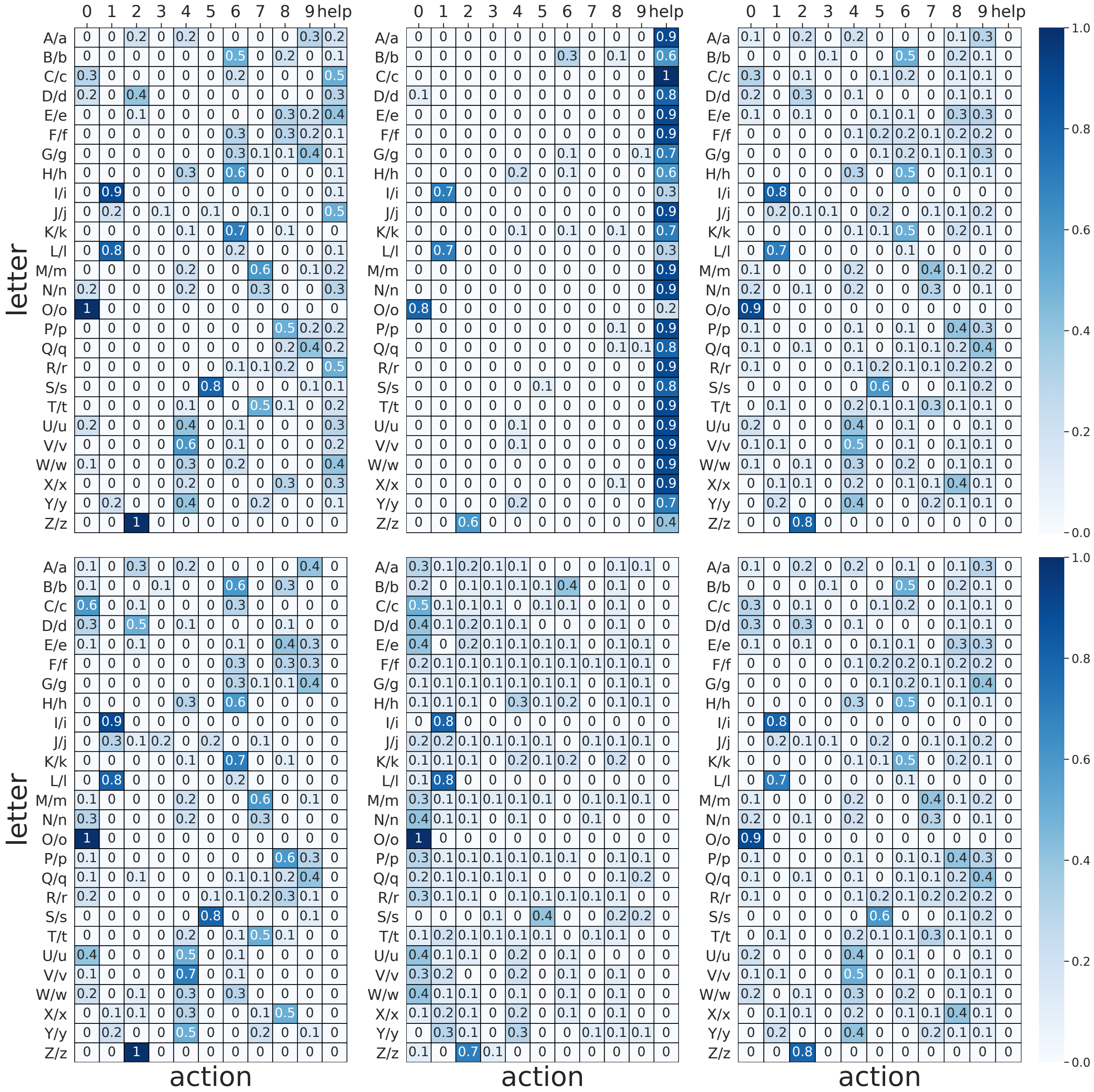
		1-of-20	1-of-10	5-of-10	10-of-10	Optim($\hat{r}$)
help is available	correct	97.50±3.11	98.57±1.00	99.27±0.01	99.27±0.01	89.43±4.16
	help	0.39±0.48	0.14±0.02	0.09±0.00	0.07±0.00	1.16±0.00
help is unavailable	correct	97.64±3.05	98.67±0.99	99.32±0.01	99.31±0.01	94.96±4.01
	help	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.23±3.87

Table 9.3: Frequency of the correct label action index and the help action across the 10,000 MNIST test images in the "ask for help only when it is available" single-image regime.

		1-of-20	1-of-10	5-of-10	10-of-10	Optim($\hat{r}$)
help is available	correct	84.13±1.07	90.68±1.54	98.80±0.01	99.28±0.01	89.43±4.16
	help	10.42±0.51	3.25±0.47	0.05±0.01	0.07±0.01	1.16±0.00
help is unavailable	correct	88.54±1.14	92.18±1.59	99.06±0.02	99.31±0.01	94.96±4.01
	help	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.23±3.87

Table 9.4: Frequency of the correct label action index and the help action across the 10,000 MNIST test images in the "learning to ask for help" task with perturbed rewards in the all-images regime, where reward models are trained on 1%, 10%, or 100% of the digit dataset.

		1-of-20	1-of-10	5-of-10	10-of-10	Optim($\hat{r}$)
1%	correct	95.81±0.14	95.82±0.18	95.97±0.08	96.10±0.06	89.43±4.16
	help	0.39±0.03	0.41±0.07	0.34±0.02	0.30±0.03	1.16±0.93
10%	correct	98.63±0.07	98.62±0.08	98.67±0.04	98.67±0.03	94.96±4.01
	help	0.13±0.01	0.12±0.01	0.12±0.01	0.10±0.01	0.23±3.87
100%	correct	99.34±0.04	99.36±0.04	99.38±0.02	99.38±0.02	98.23±3.57
	help	0.04±0.00	0.03±0.01	0.03±0.00	0.02±0.00	0.08±0.73



	0	1	2	3	4	5	6	7	8	9	help
A/a	0	0	0.2	0	0.2	0	0	0	0	0.3	0.2
B/b	0	0	0	0	0	0	0.5	0	0.2	0	0.1
C/c	0.3	0	0	0	0	0	0.2	0	0	0	0.5
D/d	0.2	0	0.4	0	0	0	0	0	0	0	0.3
E/e	0	0	0.1	0	0	0	0	0	0.3	0.2	0.4
F/f	0	0	0	0	0	0	0.3	0	0.3	0.2	0.1
G/g	0	0	0	0	0	0	0.3	0.1	0.1	0.4	0.1
H/h	0	0	0	0	0.3	0	0.6	0	0	0	0.1
I/i	0	0.9	0	0	0	0	0	0	0	0	0.1
J/j	0	0.2	0	0.1	0	0.1	0	0.1	0	0	0.5
K/k	0	0	0	0	0.1	0	0.7	0	0.1	0	0
L/l	0	0.8	0	0	0	0	0.2	0	0	0	0.1
M/m	0	0	0	0	0.2	0	0	0.6	0	0.1	0.2
N/n	0.2	0	0	0	0.2	0	0	0.3	0	0	0.3
O/o	1	0	0	0	0	0	0	0	0	0	0
P/p	0	0	0	0	0	0	0	0	0.5	0.2	0.2
Q/q	0	0	0	0	0	0	0	0	0.2	0.4	0.2
R/r	0	0	0	0	0	0	0.1	0.1	0.2	0	0.5
S/s	0	0	0	0	0	0.8	0	0	0	0.1	0.1
T/t	0	0	0	0	0.1	0	0	0.5	0.1	0	0.2
U/u	0.2	0	0	0	0.4	0	0.1	0	0	0	0.3
V/v	0	0	0	0	0.6	0	0.1	0	0	0	0.2
W/w	0.1	0	0	0	0.3	0	0.2	0	0	0	0.4
X/x	0	0	0	0	0.2	0	0	0	0.3	0	0.3
Y/y	0	0.2	0	0	0.4	0	0	0.2	0	0	0.1
Z/z	0	0	1	0	0	0	0	0	0	0	0

	0	1	2	3	4	5	6	7	8	9	help
A/a	0	0	0	0	0	0	0	0	0	0	0.9
B/b	0	0	0	0	0	0	0.3	0	0.1	0	0.6
C/c	0	0	0	0	0	0	0	0	0	0	1
D/d	0.1	0	0	0	0	0	0	0	0	0	0.8
E/e	0	0	0	0	0	0	0	0	0	0	0.9
F/f	0	0	0	0	0	0	0	0	0	0	0.9
G/g	0	0	0	0	0	0	0.1	0	0	0.1	0.7
H/h	0	0	0	0	0.2	0	0.1	0	0	0	0.6
I/i	0	0.7	0	0	0	0	0	0	0	0	0.3
J/j	0	0	0	0	0	0	0	0	0	0	0.9
K/k	0	0	0	0	0.1	0	0.1	0	0.1	0	0.7
L/l	0	0.7	0	0	0	0	0	0	0	0	0.3
M/m	0	0	0	0	0	0	0	0	0	0	0.9
N/n	0	0	0	0	0	0	0	0	0	0	0.9
O/o	0.8	0	0	0	0	0	0	0	0	0	0.2
P/p	0	0	0	0	0	0	0	0	0.1	0	0.9
Q/q	0	0	0	0	0	0	0	0	0.1	0.1	0.8
R/r	0	0	0	0	0	0	0	0	0	0	0.9
S/s	0	0	0	0	0	0.1	0	0	0	0	0.8
T/t	0	0	0	0	0	0	0	0	0	0	0.9
U/u	0	0	0	0	0.1	0	0	0	0	0	0.9
V/v	0	0	0	0	0.1	0	0	0	0	0	0.9
W/w	0	0	0	0	0	0	0	0	0	0	0.9
X/x	0	0	0	0	0	0	0	0	0.1	0	0.9
Y/y	0	0	0	0	0.2	0	0	0	0	0	0.7
Z/z	0	0	0.6	0	0	0	0	0	0	0	0.4

	0	1	2	3	4	5	6	7	8	9	help
A/a	0.1	0	0.2	0	0.2	0	0	0	0.1	0.3	0
B/b	0	0	0	0.1	0	0	0.5	0	0.2	0.1	0
C/c	0.3	0	0.1	0	0	0.1	0.2	0	0.1	0.1	0
D/d	0.2	0	0.3	0	0.1	0	0	0	0.1	0.1	0
E/e	0.1	0	0.1	0	0	0.1	0.1	0	0.3	0.3	0
F/f	0	0	0	0	0.1	0.2	0.2	0.1	0.2	0.2	0
G/g	0	0	0	0	0	0.1	0.2	0.1	0.1	0.3	0
H/h	0	0	0	0	0.3	0	0.5	0	0.1	0.1	0
I/i	0	0.8	0	0	0	0	0	0	0	0	0
J/j	0	0.2	0.1	0.1	0	0.2	0	0.1	0.1	0.2	0
K/k	0	0	0	0	0.1	0.1	0.5	0	0.2	0.1	0
L/l	0	0.7	0	0	0	0	0.1	0	0	0	0
M/m	0.1	0	0	0	0.2	0	0	0.4	0.1	0.2	0
N/n	0.2	0	0.1	0	0.2	0	0	0.3	0	0.1	0
O/o	0.9	0	0	0	0	0	0	0	0	0	0
P/p	0.1	0	0	0	0.1	0	0.1	0	0.4	0.3	0
Q/q	0.1	0	0.1	0	0.1	0	0.1	0.1	0.2	0.4	0
R/r	0.1	0	0	0	0.1	0.2	0.1	0.1	0.2	0.2	0
S/s	0	0	0	0	0	0.6	0	0	0.1	0.2	0
T/t	0	0.1	0	0	0.2	0.1	0.1	0.3	0.1	0.1	0
U/u	0.2	0	0	0	0.4	0	0.1	0	0	0.1	0
V/v	0.1	0.1	0	0	0.5	0	0.1	0	0.1	0.1	0
W/w	0.1	0	0.1	0	0.3	0	0.2	0	0.1	0.1	0
X/x	0	0.1	0.1	0	0.2	0	0.1	0.1	0.4	0.1	0
Y/y	0	0.2	0	0	0.4	0	0	0.2	0.1	0.1	0
Z/z	0	0	0.8	0	0	0	0	0	0	0	0

	0	1	2	3	4	5	6	7	8	9	help
A/a	0.1	0	0.3	0	0.2	0	0	0	0	0.4	0
B/b	0.1	0	0	0.1	0	0	0.6	0	0.3	0	0
C/c	0.6	0	0.1	0	0	0	0.3	0	0	0	0
D/d	0.3	0	0.5	0	0.1	0	0	0	0.1	0	0
E/e	0.1	0	0.1	0	0	0	0.1	0	0.4	0.3	0
F/f	0	0	0	0	0	0	0.3	0	0.3	0.3	0
G/g	0	0	0	0	0	0	0.3	0.1	0.1	0.4	0
H/h	0	0	0	0	0.3	0	0.6	0	0	0	0
I/i	0	0.9	0	0	0	0	0	0	0	0	0
J/j	0	0.3	0.1	0.2	0	0.2	0	0.1	0	0	0
K/k	0	0	0	0	0.1	0	0.7	0	0.1	0	0
L/l	0	0.8	0	0	0	0	0.2	0	0	0	0
M/m	0.1	0	0	0	0.2	0	0	0.6	0	0.1	0
N/n	0.3	0	0	0	0.2	0	0	0.3	0	0	0
O/o	1	0	0	0	0	0	0	0	0	0	0
P/p	0.1	0	0	0	0	0	0	0	0.6	0.3	0
Q/q	0.1	0	0.1	0	0	0	0.1	0.1	0.2	0.4	0
R/r	0.2	0	0	0	0	0.1	0.1	0.2	0.3	0.1	0
S/s	0	0	0	0	0	0.8	0	0	0	0.1	0
T/t	0	0	0	0	0.2	0	0.1	0.5	0.1	0	0
U/u	0.4	0	0	0	0.5	0	0.1	0	0	0	0
V/v	0.1	0	0	0	0.7	0	0.1	0	0	0	0
W/w	0.2	0	0.1	0	0.3	0	0.3	0	0	0	0
X/x	0	0.1	0.1	0	0.3	0	0	0.1	0.5	0	0
Y/y	0	0.2	0	0	0.5	0	0	0.2	0	0.1	0
Z/z	0	0	1	0	0	0	0	0	0	0	0

	0	1	2	3	4	5	6	7	8	9	help
A/a	0.3	0.1	0.2	0.1	0.1	0	0	0	0.1	0.1	0
B/b	0.2	0	0.1	0.1	0.1	0.1	0.4	0	0.1	0	0
C/c	0.5	0.1	0.1	0.1	0	0.1	0.1	0	0.1	0	0
D/d	0.4	0.1	0.2	0.1	0.1	0	0	0	0.1	0	0
E/e	0.4	0	0.2	0.1	0.1	0.1	0.1	0	0.1	0.1	0
F/f	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0
G/g	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0	0.1	0.1	0
H/h	0.1	0.1	0.1	0	0.3	0.1	0.2	0	0.1	0.1	0
I/i	0	0.8	0	0	0	0	0	0	0	0	0
J/j	0.2	0.2	0.1	0.1	0.1	0.1	0	0.1	0.1	0.1	0
K/k	0.1	0.1	0.1	0	0.2	0.1	0.2	0	0.2	0	0
L/l	0.1	0.8	0	0	0	0	0	0	0	0	0
M/m	0.3	0.1	0.1	0.1	0.1	0.1	0	0.1	0.1	0.1	0
N/n	0.4	0.1	0.1	0	0.1	0	0	0.1	0	0	0
O/o	1	0	0	0	0	0	0	0	0	0	0
P/p	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0	0.1	0.1	0
Q/q	0.2	0.1	0.1	0.1	0.1	0	0	0	0.1	0.2	0
R/r	0.3	0.1	0.1	0	0.1	0.1	0.1	0.1	0.1	0	0
S/s	0	0	0	0.1	0	0.4	0	0	0.2	0.2	0
T/t	0.1	0.2	0.1	0.1	0.1	0.1	0	0.1	0.1	0	0
U/u	0.4	0.1	0.1	0	0.2	0	0.1	0	0	0	0
V/v	0.3	0.2	0	0	0.2	0	0.1	0	0.1	0	0
W/w	0.4	0.1	0.1	0	0.1	0	0.1	0	0.1	0	0
X/x	0.1	0.2	0.1	0	0.2	0	0.1	0	0.1	0	0
Y/y	0	0.3	0.1	0	0.3	0	0	0.1	0.1	0.1	0
Z/z	0.1	0	0.7	0.1	0	0	0	0	0	0	0

	0	1	2	3	4	5	6	7	8	9	help
A/a	0.1	0	0.2	0	0.2	0	0.1	0	0.1	0.3	0
B/b	0	0	0	0.1	0	0	0.5	0	0.2	0.1	0
C/c	0.3	0	0.1	0	0	0.1	0.2	0	0.1	0.1	0
D/d	0.3	0	0.3	0	0.1	0	0	0	0.1	0.1	0
E/e	0.1	0	0.1	0	0	0.1	0.1	0	0.3	0.3	0
F/f	0	0	0	0	0.1	0.2	0.2	0.1	0.2	0.2	0
G/g	0	0	0	0	0	0.1	0.2	0.1	0.1	0.4	0
H/h	0	0	0	0	0.3	0	0.5	0	0.1	0.1	0
I/i	0	0.8	0	0	0	0	0	0	0	0	0
J/j	0	0.2	0.1	0.1	0	0.2	0	0.1	0.1	0.2	0
K/k	0	0	0	0	0.1	0.1	0.5	0	0.2	0.1	0
L/l	0	0.7	0	0	0	0	0.1	0	0	0	0
M/m	0.1	0	0	0	0.2	0	0	0.4	0.1	0.2	0
N/n	0.2	0	0.1	0	0.2	0	0	0.3	0	0.1	0
O/o	0.9	0	0	0	0	0	0	0	0	0	0
P/p	0.1	0	0	0	0.1	0	0.1	0.1	0.4	0.3	0
Q/q	0.1	0	0.1	0	0.1	0	0.1	0.1	0.2	0.4	0
R/r	0.1	0	0	0	0.1	0.2	0.1	0.2	0.2	0.2	0
S/s	0	0	0	0	0	0.6	0	0	0.1	0.2	0
T/t	0	0.1	0	0	0.2	0.1	0.1	0.3	0.1	0.1	0
U/u	0.2	0	0	0	0.4	0	0.1	0	0	0.1	0
V/v	0.1	0.1	0	0	0.5	0	0.1	0	0.1	0.1	0
W/w	0.2	0	0.1	0	0.3	0	0.2	0	0.1	0.1	0
X/x	0	0.1	0.1	0	0.2	0	0.1	0.1	0.4	0.1	0
Y/y	0	0.2	0	0	0.4	0	0	0.2	0.1	0.1	0
Z/z	0	0	0.8	0	0	0	0	0	0	0	0

Figure 9.B.15: The average frequency of each action on each letter, averaged over lowercase and uppercase images, in the "ask for help only when it is available" task. (top row) Help is available, (bottom row) help is unavailable. (Left column) 1-of-20 all-images regime, (middle column) 1-of-20 single-image regime, (right column) Optim($\hat{r}$).

Table 9.5: Frequency of the correct label action index and the help action across the 10,000 MNIST test images in the "learning to ask for help" task with perturbed rewards in the single-image regime, where reward models are trained on 1%, 10%, or 100% of the digit dataset.

		1-of-20	1-of-10	5-of-10	10-of-10	Optim($\hat{r}$)
1%	correct	86.96±0.41	89.75±0.34	94.56±0.12	96.11±0.06	89.43±4.16
	help	5.80±0.26	4.49±0.26	1.61±0.06	0.29±0.02	1.16±0.93
10%	correct	88.78±0.16	92.51±0.13	97.50±0.06	98.68±0.03	94.96±4.0
	help	9.08±0.10	6.61±0.09	1.16±0.02	0.11±0.01	0.23±3.87
100%	correct	96.61±0.07	97.67±0.06	99.07±0.03	99.37±0.02	98.23±3.57
	help	2.47±0.02	1.52±0.03	0.36±0.01	0.02±0.00	0.08±0.73

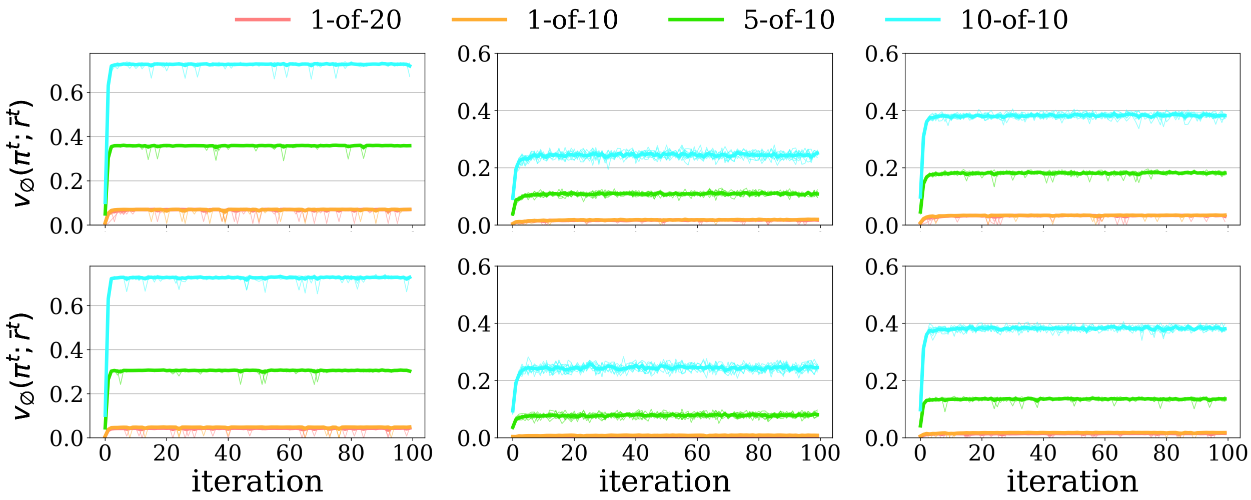


Figure 9.B.16: Expected return of each k-of-N policy on each iteration t given the sampled k-of-N reward function, $\bar{r}^t$, in the "learning to ask for help" task with perturbed rewards, where reward models are trained on 1% of the digit dataset. (top row) All-images regime, (bottom row) single-image regime, (left column) digits, (middle column) fashion, (right column) letter.

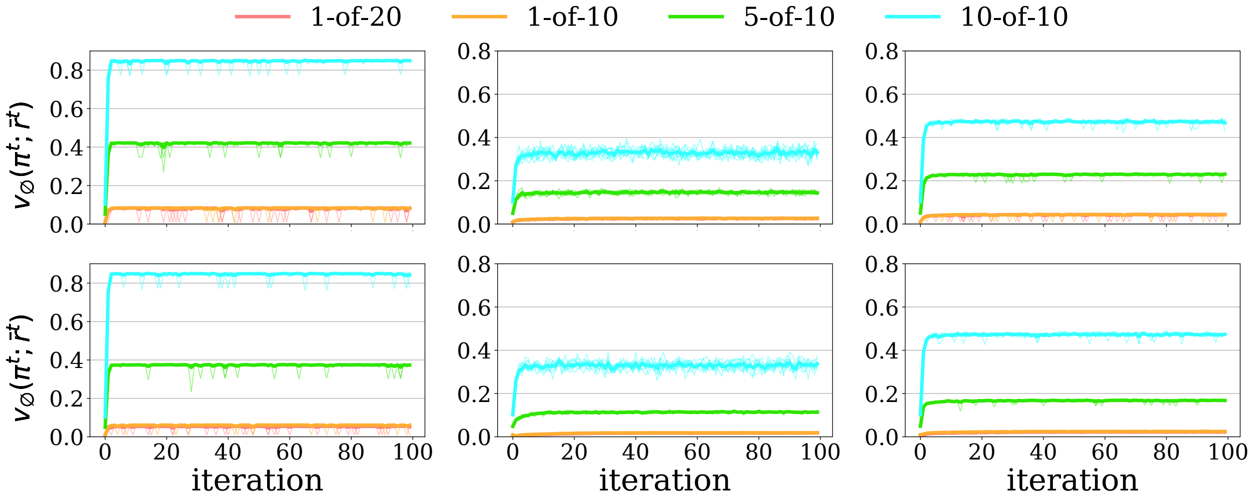


Figure 9.B.17: Expected return of each k-of-N policy on each iteration t given the sampled k-of-N reward function, $\bar{r}^t$, in the "learning to ask for help" task with perturbed rewards, where reward models are trained on 10% of the digit dataset. (top row) All-images regime, (bottom row) single-image regime, (left column) digits, (middle column) fashion, (right column) letter.

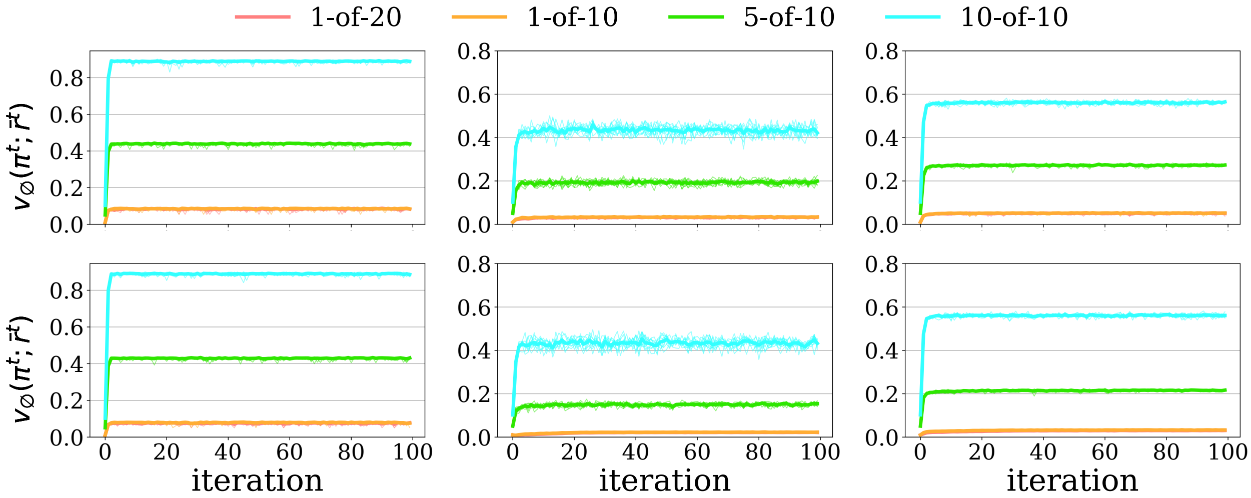
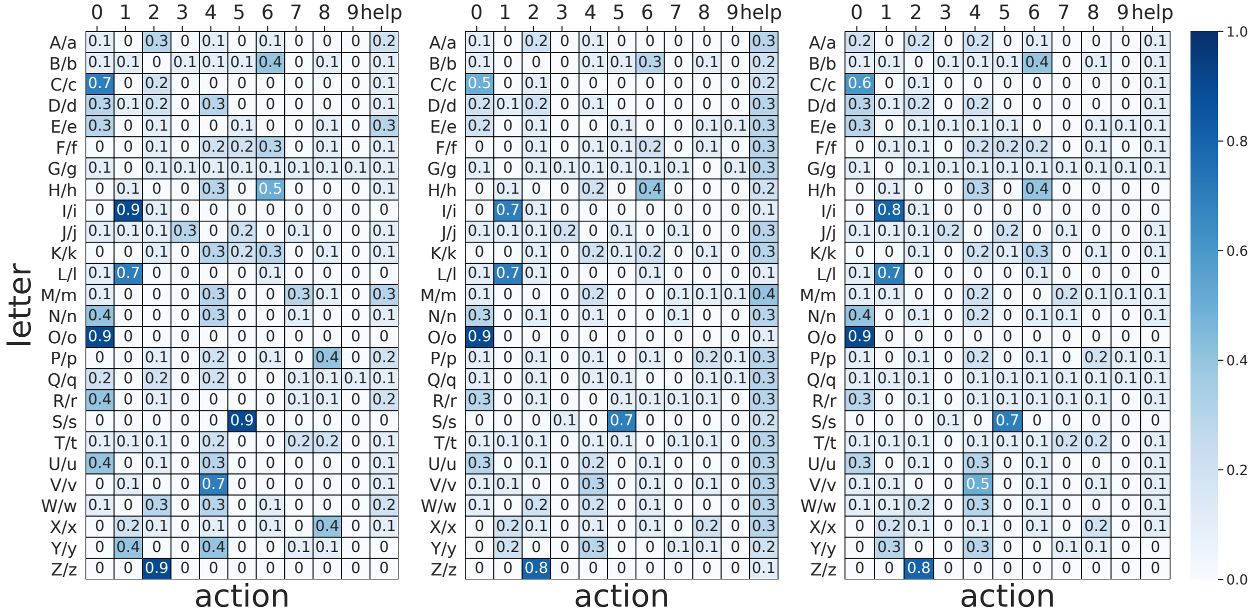


Figure 9.B.18: Expected return of each k-of-N policy on each iteration t given the sampled k-of-N reward function, $\bar{r}^t$, in the "learning to ask for help" task with perturbed rewards, where reward models are trained on 100% of the digit dataset. (top row) All-images regime, (bottom row) single-image regime, (left column) digits, (middle column) fashion, (right column) letter.



letter	0	1	2	3	4	5	6	7	8	9	help
A/a	0.1	0	0.3	0	0.1	0	0.1	0	0	0	0.2
B/b	0.1	0.1	0	0.1	0.1	0.1	0.4	0	0.1	0	0.1
C/c	0.7	0	0.2	0	0	0	0	0	0	0	0.1
D/d	0.3	0.1	0.2	0	0.3	0	0	0	0	0	0.1
E/e	0.3	0	0.1	0	0	0.1	0	0	0.1	0	0.3
F/f	0	0	0.1	0	0.2	0.2	0.3	0	0.1	0	0.1
G/g	0.1	0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
H/h	0	0.1	0	0	0.3	0	0.5	0	0	0	0.1
I/i	0	0.9	0.1	0	0	0	0	0	0	0	0
J/j	0.1	0.1	0.1	0.3	0	0.2	0	0.1	0	0	0.1
K/k	0	0	0.1	0	0.3	0.2	0.3	0	0.1	0	0.1
L/l	0.1	0.7	0	0	0	0	0.1	0	0	0	0
M/m	0.1	0	0	0	0.3	0	0	0.3	0.1	0	0.3
N/n	0.4	0	0	0	0.3	0	0	0.1	0	0	0.1
O/o	0.9	0	0	0	0	0	0	0	0	0	0
P/p	0	0	0.1	0	0.2	0	0.1	0	0.4	0	0.2
Q/q	0.2	0	0.2	0	0.2	0	0	0.1	0.1	0.1	0.1
R/r	0.4	0	0.1	0	0	0	0	0.1	0.1	0	0.2
S/s	0	0	0	0	0	0.9	0	0	0	0	0
T/t	0.1	0.1	0.1	0	0.2	0	0	0.2	0.2	0	0.1
U/u	0.4	0	0.1	0	0.3	0	0	0	0	0	0.1
V/v	0	0.1	0	0	0.7	0	0	0	0	0	0.1
W/w	0.1	0	0.3	0	0.3	0	0.1	0	0	0	0.2
X/x	0	0.2	0.1	0	0.1	0	0.1	0	0.4	0	0.1
Y/y	0	0.4	0	0	0.4	0	0	0.1	0.1	0	0
Z/z	0	0	0.9	0	0	0	0	0	0	0	0

letter	0	1	2	3	4	5	6	7	8	9	help
A/a	0.1	0	0.2	0	0.1	0	0	0	0	0	0.3
B/b	0.1	0	0	0	0.1	0.1	0.3	0	0.1	0	0.2
C/c	0.5	0	0.1	0	0	0	0	0	0	0	0.2
D/d	0.2	0.1	0.2	0	0.1	0	0	0	0	0	0.3
E/e	0.2	0	0.1	0	0	0.1	0	0	0.1	0.1	0.3
F/f	0	0	0.1	0	0.1	0.1	0.2	0	0.1	0	0.3
G/g	0.1	0	0.1	0.1	0.1	0.1	0.1	0.1	0	0.1	0.3
H/h	0	0.1	0	0	0.2	0	0.4	0	0	0	0.2
I/i	0	0.7	0.1	0	0	0	0	0	0	0	0.1
J/j	0.1	0.1	0.1	0.2	0	0.1	0	0.1	0	0	0.3
K/k	0	0	0.1	0	0.2	0.1	0.2	0	0.1	0	0.3
L/l	0.1	0.7	0.1	0	0	0	0.1	0	0	0	0.1
M/m	0.1	0	0	0	0.2	0	0	0.1	0.1	0.1	0.4
N/n	0.3	0	0.1	0	0.1	0	0	0.1	0	0	0.3
O/o	0.9	0	0	0	0	0	0	0	0	0	0.1
P/p	0.1	0	0.1	0	0.1	0	0.1	0	0.2	0.1	0.3
Q/q	0.1	0	0.1	0	0.1	0.1	0	0	0.1	0.1	0.3
R/r	0.3	0	0.1	0	0	0.1	0.1	0.1	0.1	0	0.3
S/s	0	0	0	0.1	0	0.7	0	0	0	0	0.2
T/t	0.1	0.1	0.1	0	0.1	0.1	0	0.1	0.1	0	0.3
U/u	0.3	0	0.1	0	0.2	0	0.1	0	0	0	0.3
V/v	0.1	0.1	0	0	0.3	0	0.1	0	0.1	0	0.3
W/w	0.1	0	0.2	0	0.2	0	0.1	0	0	0	0.3
X/x	0	0.2	0.1	0	0.1	0	0.1	0	0.2	0	0.3
Y/y	0	0.2	0	0	0.3	0	0	0.1	0.1	0	0.2
Z/z	0	0	0.8	0	0	0	0	0	0	0	0.1

letter	0	1	2	3	4	5	6	7	8	9	help
A/a	0.2	0	0.2	0	0.2	0	0.1	0	0	0	0.1
B/b	0.1	0.1	0	0.1	0.1	0.1	0.4	0	0.1	0	0.1
C/c	0.6	0	0.1	0	0	0	0	0	0	0	0.1
D/d	0.3	0.1	0.2	0	0.2	0	0	0	0	0	0.1
E/e	0.3	0	0.1	0.1	0.1	0.1	0	0	0.1	0.1	0.1
F/f	0	0.1	0.1	0	0.2	0.2	0.2	0	0.1	0	0.1
G/g	0.1	0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
H/h	0	0.1	0	0	0.3	0	0.4	0	0	0	0
I/i	0	0.8	0.1	0	0	0	0	0	0	0	0
J/j	0.1	0.1	0.1	0.2	0	0.2	0	0.1	0	0	0.1
K/k	0	0	0.1	0	0.2	0.1	0.3	0	0.1	0	0.1
L/l	0.1	0.7	0	0	0	0	0.1	0	0	0	0
M/m	0.1	0.1	0	0	0.2	0	0	0.2	0.1	0.1	0.1
N/n	0.4	0	0.1	0	0.2	0	0.1	0.1	0	0	0.1
O/o	0.9	0	0	0	0	0	0	0	0	0	0
P/p	0.1	0	0.1	0	0.2	0	0.1	0	0.2	0.1	0.1
Q/q	0.1	0.1	0.1	0	0.1	0.1	0.1	0.1	0.1	0.1	0.1
R/r	0.3	0	0.1	0	0.1	0.1	0.1	0.1	0.1	0	0.1
S/s	0	0	0	0.1	0	0.7	0	0	0	0	0
T/t	0.1	0.1	0.1	0	0.1	0.1	0.1	0.2	0.2	0	0.1
U/u	0.3	0	0.1	0	0.3	0	0.1	0	0	0	0.1
V/v	0.1	0.1	0	0	0.5	0	0.1	0	0.1	0	0.1
W/w	0.1	0.1	0.2	0	0.3	0	0.1	0	0	0	0.1
X/x	0	0.2	0.1	0	0.1	0	0.1	0	0.2	0	0.1
Y/y	0	0.3	0	0	0.3	0	0	0.1	0.1	0	0
Z/z	0	0	0.8	0	0	0	0	0	0	0	0

Figure 9.B.19: The average frequency of each action on each letter, averaged over lowercase and uppercase images, in the "learning to ask for help" task with perturbed rewards, where reward models are trained on 1% of the digit dataset. (left) 1-of-20 all-images regime, (middle) 1-of-20 single-image regime, (right) Optim($\hat{r}$).

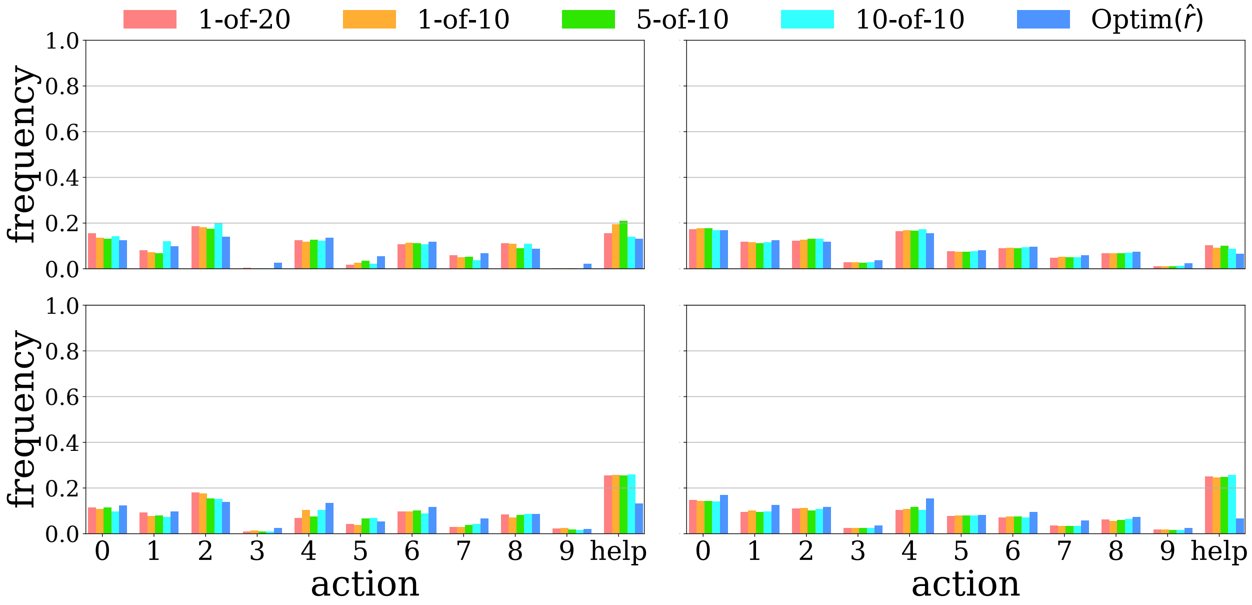


Figure 9.B.20: Action distribution of each k-of-N policy and baseline in the "learning to ask for help" task with perturbed rewards, where reward models are trained on 1% of the digit dataset. (top row) All-images regime, (bottom row) single-image regime, (left column) fashion, (right column) letter.

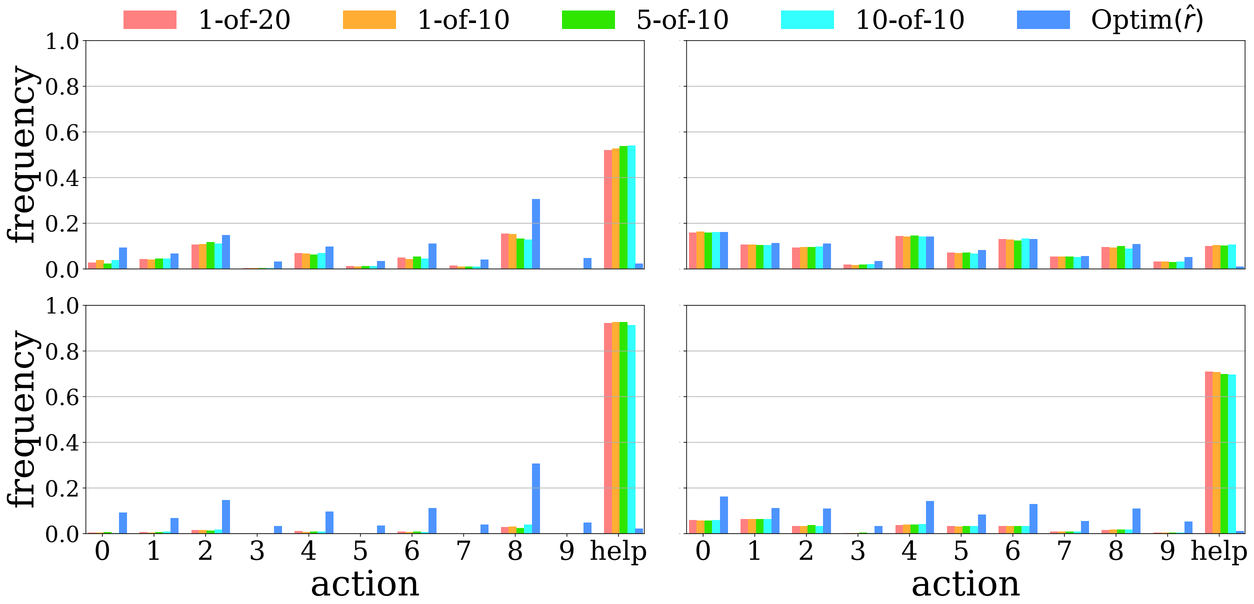


Figure 9.B.21: Action distribution of each k-of-N policy and baseline in the "learning to ask for help" task with perturbed rewards, where reward models are trained on 10% of the digit dataset. (top row) All-images regime, (bottom row) single-image regime, (left column) fashion, (right column) letter.

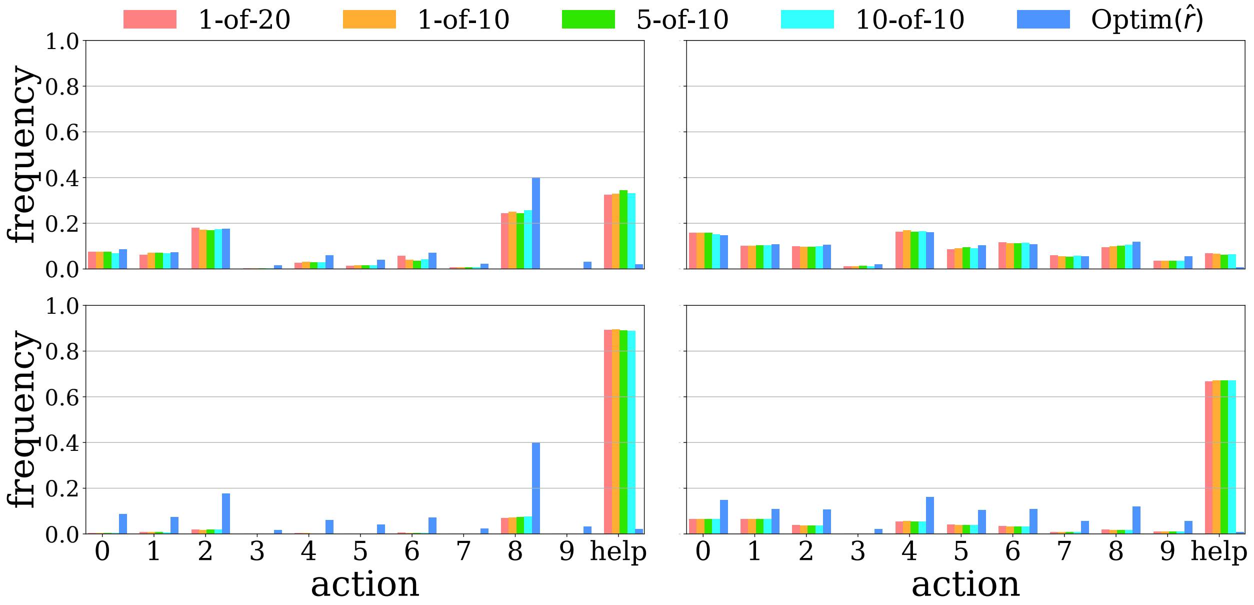


Figure 9.B.22: Action distribution of each k-of-N policy and baseline in the "learning to ask for help" task with perturbed rewards, where reward models are trained on 100% of the digit dataset. (top row) All-images regime, (bottom row) single-image regime, (left column) fashion, (right column) letter.

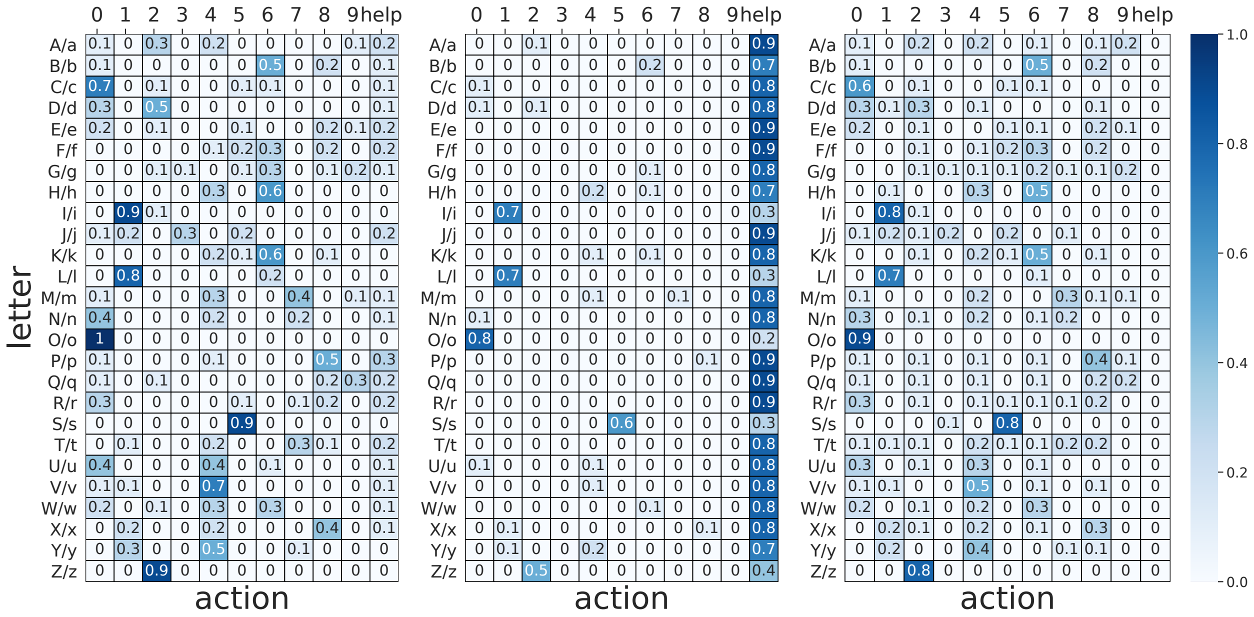


Figure 9.B.23: The average frequency of each action on each letter, averaged over lowercase and uppercase images, in the "learning to ask for help" task with perturbed rewards, where reward models are trained on 10% of the digit dataset. (left) 1-of-20 all-images regime, (middle) 1-of-20 single-image regime, (right) Optim($\hat{r}$).

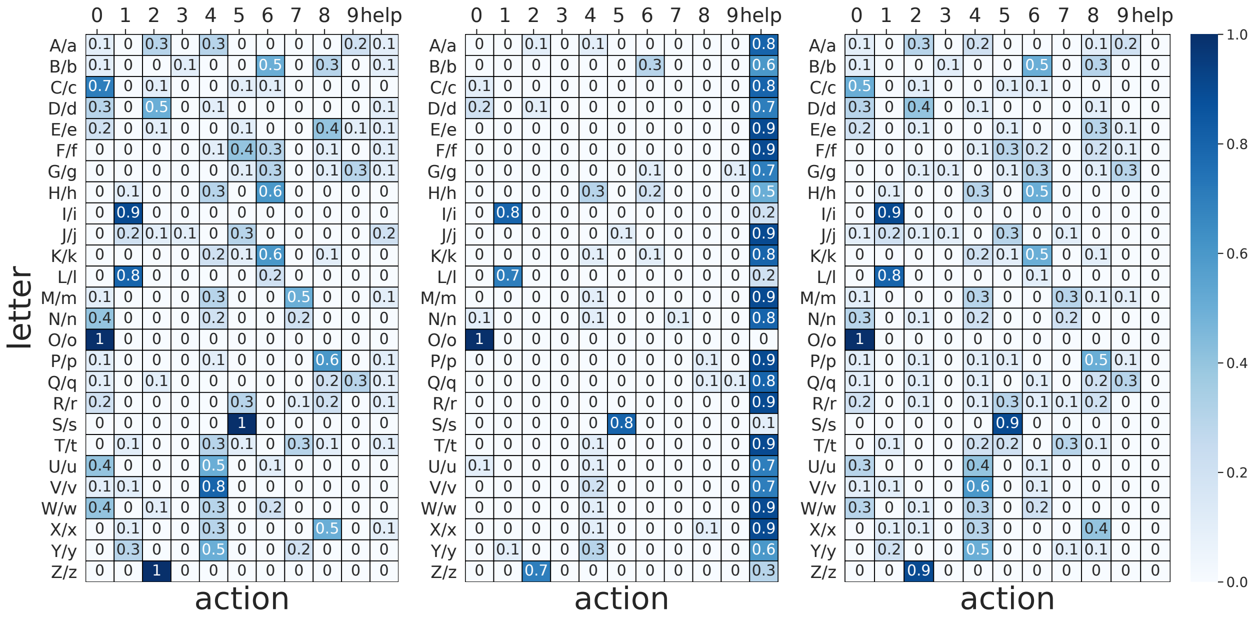


Figure 9.B.24: The average frequency of each action on each letter, averaged over lowercase and uppercase images, in the "learning to ask for help" task with perturbed rewards, where reward models are trained on 100% of the digit dataset. (left) 1-of-20 all-images regime, (middle) 1-of-20 single-image regime, (right) Optim($\hat{r}$).

channel encodes the positions of different aspects of the environment: pavement, ditch, car, and obstacle.

Our networks have two parallel convolutional layers with four output channels and 2×2 filters, each followed by a ReLU transformation. The outputs from these layers are flattened, concatenated together, and that result is concatenated with a one-hot encoding of the car's speed in the next state. Next, we apply two fully connected layers separated by a ReLU function, the first with 32 outputs and the second with a single output, respectively. For each possible speed the car could have in the initial state, we manage a different pair of fully connected layers with the same shapes. To compute the expected reward for a given action in a given state, we query the neural network with every possible state that could follow from the given action and use the transition probabilities to combine these state–next state reward estimates.

Networks are trained over 51,200 epochs using a batch size of 800 to minimize the MSE using Adam with a learning rate of 0.0001 and weight decay of 10^{-5}. The remaining parameters for Adam in PyTorch (β_1, β_2, and ϵ) are set to their defaults (0.9, 0.999, 10^{-8}) without the AMSGrad (Reddi et al. 2018) modification.

Policies are evaluated by iterative dynamic programming according with the Bellman operator. An approximation of the γ-discounted expected return function is updated simultaneously for each state and action until the maximum absolute change is smaller than 10^{-6}. We use a discount factor of $\gamma = 0.99$. On every k-of-N CFR iteration, this process is run N times given the current CFR policy to evaluate it under each of the N reward functions sampled at the start of the iteration.[7]

This experiment was run on a 3.60GHz Intel® Core™ i9-9900K CPU with 7.7 GB of memory without a GPU. Model training took roughly 25 minutes per random initialization, so 833 CPU hours for all 2000 models. 100 iterations of k-of-N CFR took roughly three minutes, so for all six settings of k and N, and all five random repetitions, it took about 100 minutes. In total, our experiment used about 835 CPU hours of computation.

[7] Alternatively, the successor representation (Dayan 1993) could be used to fully characterize the current CFR policy for essentially the same cost as a single policy evaluation. Given this information, computing the expected return given a reward function can be computed with a single pass over each state and action, thereby reducing the amount of computation that scales with N.

Chapter 10

Alternative Function Approximation Parameterizations for RCFR

10.1 Introduction

The RCFR algorithm incorporates function approximation into CFR but both the theory and experiments for RCFR were originally specific to the ramp link function or a normalized ramp policy parameterization. In contrast, many algorithms like Hedge and softmax policy gradient, use a softmax link function/parameterization. In the case of Hedge (*i.e.*, softmax regret matching), this parameterization even gives it a better regret bound than that known for ramp regret matching with respect to the number of actions.

This chapter presents regret bounds for RCFR with the softmax and polynomial link functions, and examines their performance differences in small two-player, zero-sum, imperfect-information games from OpenSpiel (Lanctot, Lockhart, et al. 2019). These regret bounds are derived from a generic analysis of f-RCFR parameterized by link function f, and approximate (Φ, f)-regret matching, which is additionally parameterized by a deviation set Φ. Notably, the softmax link function retains its theoretical advantage in approximate regret matching and that approximation errors impact the bound differently than for polynomial link functions. In experiments with f-RCFR, the softmax link function exhibits better performance when approximation errors are large, suggesting that softmax RCFR may scale better than ramp RCFR.

Softmax RCFR is not just superficially similar to softmax policy gradient; by choosing a particular training setup using "bootstrapped targets" (Morrill 2016), softmax RCFR becomes a "one line change" of softmax policy gradient. This algorithm is *neural replicator dynamics* (*NeuRD*) as it can also be derived as a discrete, approximate version of the replicator dynamics of evolutionary game theory. Experiments show that NeuRD outperforms softmax policy gradient in games and non-stationary environments.

10.2 Approximate (Φ, f)-Regret Matching

A (Φ, f)-regret matching algorithm is an ODP algorithm that chooses the strategy on round t that is a fixed point of the best average of the deviations $\Phi \subseteq \Phi^{\text{SW}}_{\mathcal{X}}$ in hindsight, $\bar{\phi}^t$, defined by Eq. (2.24). The (Φ, f)-regret matching theory is based on reasoning about the growth of a potential function, $G : \mathbb{R}^n \to \mathbb{R}$, applied to cumulative regrets. The key relationship between f and G is that there is a function $g : \mathbb{R}^n \to \mathbb{R}^n_+$ where $g(\cdot) = cf(\cdot)$ for a positive scaling factor $c > 0$ such that $G(x + x') \leq G(x) + \langle g(x),\, x' \rangle + \gamma(x')$ for any $x, x' \in \mathbb{R}^n$, where $\gamma : \mathbb{R}^n \to \mathbb{R}_+$. Together, (G, g, γ) is called a Gordon triple. The function g is present to utilize the scale invariance of $\bar{\phi}^t$. Since each $\bar{\phi}^t$ is normalized by the sum of link outputs from the cumulative regret vector, *i.e.*, $\langle \mathbf{1},\, f(\rho^{1:t-1}) \rangle$, all link functions that are proportional to f produce the same strategies. This allows us to reason about link function $g \propto f$ instead of f when convenient.

Our new approximate (Φ, f)-regret matching extension provides an analogous framework for agents who use an approximate average deviation in hindsight,

$$\widetilde{\phi}^t = \begin{cases} \dfrac{1}{\langle \mathbf{1},\, \widetilde{y}^t \rangle} \sum_{\phi \in \Phi} \widetilde{y}^t_\phi \phi & \text{if } \langle \mathbf{1},\, \widetilde{y}^t \rangle > 0 \\ I & \text{o.w.}, \end{cases} \tag{10.1}$$

where $\widetilde{y}^t = f(\widetilde{\rho}^{1:t-1})$ are the link outputs of approximate cumulative regrets $\widetilde{\rho}^{1:t-1} \in \mathbb{R}^{|\Phi|}$ and I is the identity matrix.

Theorem 23. *Given Gordon triple (G, g, γ), an approximate (Φ, g)-regret-matching algorithm has, after T rounds, a bounded regret potential*

$$G(\rho^{1:T}) \leq G(\mathbf{0}) + \sum_{t=1}^{T} \gamma(\rho^t) + \underbrace{2U \big\| g(\rho^{1:t-1}) - g\big(\widetilde{\rho}^{1:t-1}\big) \big\|_1}_{\text{Slack induced by approximation errors.}}.$$

Proof. Starting from the Gordon triple smoothness condition,

$$\begin{aligned} G(\rho^{1:T}) &\leq G(\rho^{1:T-1}) + \langle g(\rho^{1:T-1}),\, \rho^T \rangle + \gamma(\rho^T) && (10.2) \\ &= G(\rho^{1:T-1}) + \langle g(\rho^{1:T-1}),\, \rho^T \rangle - \overbrace{\langle g(\widetilde{\rho}^{1:T-1}),\, \rho^T \rangle}^{0} + \gamma(\rho^T) && (10.3) \\ &= G(\rho^{1:T-1}) + \langle g(\rho^{1:T-1}) - g(\widetilde{\rho}^{1:T-1}),\, \rho^T \rangle + \gamma(\rho^T). && (10.4) \end{aligned}$$

Applying Cauchy-Schwarz and using the payoff bound,

$$\begin{aligned} G(\rho^{1:T}) &\leq G(\rho^{1:T-1}) + \big\| g(\rho^{1:T-1}) - g(\widetilde{\rho}^{1:T-1}) \big\|_1 \big\| \rho^T \big\|_\infty + \gamma(\rho^T) && (10.5) \\ &\leq G(\rho^{1:T-1}) + 2U \big\| g(\rho^{1:T-1}) - g(\widetilde{\rho}^{1:T-1}) \big\|_1 + \gamma(\rho^T). && (10.6) \end{aligned}$$

Unrolling the recursion where $\rho^{1:0} = \mathbf{0}$, we arrive at the desired result,

$$G(\rho^{1:T}) \leq G(\mathbf{0}) + \sum_{t=1}^{T} \gamma(\rho^t) + 2U\big\|g(\rho^{1:t-1}) - g(\widetilde{\rho}^{1:t-1})\big\|_1. \tag{10.7}$$

□

Theorem 23 differs from the exact (Φ, g)-regret-matching bound (A. Greenwald, Z. Li, and Marks 2006a, Corollary 7) only by an additive error term.

Theorem 23 leads to the following bounds for polynomial and exponential link functions.

Theorem 24. *Let* $\wedge$ *be binary minimum,* i.e., $x \wedge y = \min\{x, y\}$*, and* $\vee$ *be binary maximum,* i.e., $x \vee y = \max\{x, y\}$*. The polynomial link function with power* $p-1$*,* $p > 1$ *is* $f(\cdot) = [\cdot]_+^{p-1}$*. Define the scaled link function* g *pointwise as* $g_i(x) = 0$ *if* $x_i \leq 0$ *and* $g_i(x) = \frac{p\wedge 2}{\|[x]_+\|_p^{(p-2)\vee 0}} x_i^{p-1}$ *otherwise. Approximate* (Φ, f)*-regret matching ensures that, for each deviation* $\phi \in \Phi$*, after* T *rounds,*

$$\rho^{1:T}(\phi) \leq 2U\sqrt[p]{\alpha(\Phi)}\sqrt[p\wedge 2]{((p-1)\vee 1)T + \frac{1}{2U\alpha^{1\wedge 2/p}(\Phi)}\sum_{t=1}^{T}\big\|g(\rho^{1:t-1}) - g(\widetilde{\rho}^{1:t-1})\big\|_1}.$$

Proof. Let $G(\cdot) = \|[\cdot]_+\|_p^{p\wedge 2}$ and $\gamma(\cdot) = ((p-1)\vee 1)\|\cdot\|_p^{p\wedge 2}$. Lemma 10 and 12 of A. Greenwald, Z. Li, and Marks (2006b) show that (G, g, γ) is a Gordon triple. Since $G(\mathbf{0}) = 0$, Theorem 23 becomes

$$\Big\|\big[\rho^{1:T}\big]_+\Big\|_p^{p\wedge 2} \leq \sum_{t=1}^{T}((p-1)\vee 1)\|\rho^t\|_p^{p\wedge 2} + 2U\big\|g(\rho^{1:t-1}) - g(\widetilde{\rho}^{1:t-1})\big\|_1 \tag{10.8}$$

$$\leq ((p-1)\vee 1)(2U)^{p\wedge 2}\alpha^{(p\wedge 2)/p}(\Phi)T + 2U\sum_{t=1}^{T}\big\|g(\rho^{1:t-1}) - g(\widetilde{\rho}^{1:t-1})\big\|_1, \tag{10.9}$$

where the second inequality results from the application of Lemma 9 from A. Greenwald, Z. Li, and Marks (ibid.) and the payoff magnitude bound.

The polynomial potential upper bounds the cumulative regret as $\Big\|\big[\rho^{1:T}\big]_+\Big\|_p^{p\wedge 2} \geq \Big\|\big[\rho^{1:T}\big]_+\Big\|_\infty^{p\wedge 2} \geq \big[\rho^{1:T}(\phi)\big]_+^{p\wedge 2}$ for each deviation $\phi \in \Phi$. Therefore,

$$\rho^{1:T}(\phi) \leq \big[\rho^{1:T}(\phi)\big]_+ \tag{10.10}$$

$$\leq \Big\|\big[\rho^{1:T}\big]_+\Big\|_p \tag{10.11}$$

$$\leq \sqrt[p\wedge 2]{((p-1)\vee 1)(2U)^{p\wedge 2}\alpha(\Phi)^{(p\wedge 2)/p}T + 2U\sum_{t=1}^{T}\big\|g(\rho^{1:t-1}) - g(\widetilde{\rho}^{1:t-1})\big\|_1} \tag{10.12}$$

$$\leq 2U\sqrt[p]{\alpha(\Phi)}\sqrt[p\wedge 2]{((p-1)\vee 1)T + \frac{1}{2U\alpha^{1\wedge 2/p}(\Phi)}\sum_{t=1}^{T}\big\|g(\rho^{1:t-1}) - g(\widetilde{\rho}^{1:t-1})\big\|_1}, \tag{10.13}$$

as desired. □

Theorem 25. *The exponential link function with temperature $\tau > 0$ is $f(\cdot) = \exp(\frac{1}{\tau}\cdot)$. Define the scaled link function g as $g(\cdot) = f(\cdot)/\langle \mathbf{1},\, f(\cdot)\rangle$. Approximate (Φ, f)-regret matching ensures that, for each deviation $\phi \in \Phi$, after T rounds,*

$$\rho^{1:T}(\phi) \le \tau \ln|\Phi| + 2U^2\frac{T}{\tau} + 2U\sum_{t=1}^{T}\left\|g(\rho^{1:t-1}) - g(\widetilde{\rho}^{1:t-1})\right\|_1.$$

Proof. Let $G(\cdot) = \tau \ln\langle \mathbf{1},\, f(\cdot)\rangle$ and $\gamma(\cdot) = \frac{1}{2\tau}\|\cdot\|_\infty^2$. Lemma 14 of A. Greenwald, Z. Li, and Marks (2006b) shows that (G, g, γ) is a Gordon triple. The logsumexp potential upper bounds the cumulative regret as $\tau\ln\langle \mathbf{1},\, f(\rho^{1:T})\rangle \ge \rho^{1:T}(\phi)$ for each deviation $\phi \in \Phi$ and $G(\mathbf{0}) = \tau \ln|\Phi|$, so Theorem 23 becomes

$$\begin{aligned}\rho^{1:T}(\phi) &\le \tau\ln\langle \mathbf{1},\, f(\rho^{1:T})\rangle && (10.14)\\ &\le \tau\ln|\Phi| + \sum_{t=1}^{T}\frac{1}{2\tau}\|\rho^t\|_\infty^2 + 2U\left\|g(\rho^{1:t-1}) - g(\widetilde{\rho}^{1:t-1})\right\|_1 && (10.15)\\ &\le \tau\ln|\Phi| + 2U^2\frac{T}{\tau} + 2U\sum_{t=1}^{T}\left\|g(\rho^{1:t-1}) - g(\widetilde{\rho}^{1:t-1})\right\|_1, && (10.16)\end{aligned}$$

as desired. □

See D'Orazio, Morrill, et al. (2019, Appendix B) for proofs. Note that the polynomial link function with $p = 2$ yields Waugh, Morrill, et al. (2015)'s original regression regret matching algorithm, while the exponential link function yields an approximate version of Hedge.

Theorem 24 improves upon Corollary 3.0.5 of Morrill (2016) by removing the $\sqrt{\mathcal{X}}$-term in the approximation error, due to the use of Theorem B.1 of D'Orazio, Morrill, et al. (2019), which is an improved version of Theorem 3.0.3 from Morrill (2016). Theorem 24 also replaces $|\mathcal{X}|$ with $|\mathcal{X}| - 1$ since $\alpha(\Phi^{\text{EX}}_{\mathcal{X}}) = |\mathcal{X}| - 1$.

10.3 *f*-RCFR

Thanks to our new analysis of approximate regret matching, we can use various link functions and deviation sets to construct different forms of RCFR. The f-RCFR strategy for player i given functional regret estimator $\widetilde{\rho}^{1:t-1}$ is $\pi^t(s) \propto f(\widetilde{\rho}^{1:t-1}(s))$ or it is arbitrary when $\widetilde{\rho}^{1:t-1}(s) \le \mathbf{0}$, for each agent state, $s \in \mathcal{S}_{\mathcal{A}}$. 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 (Brown, Lerer, et al. 2019; H. Li et al. 2019; Morrill 2016; Steinberger 2019;

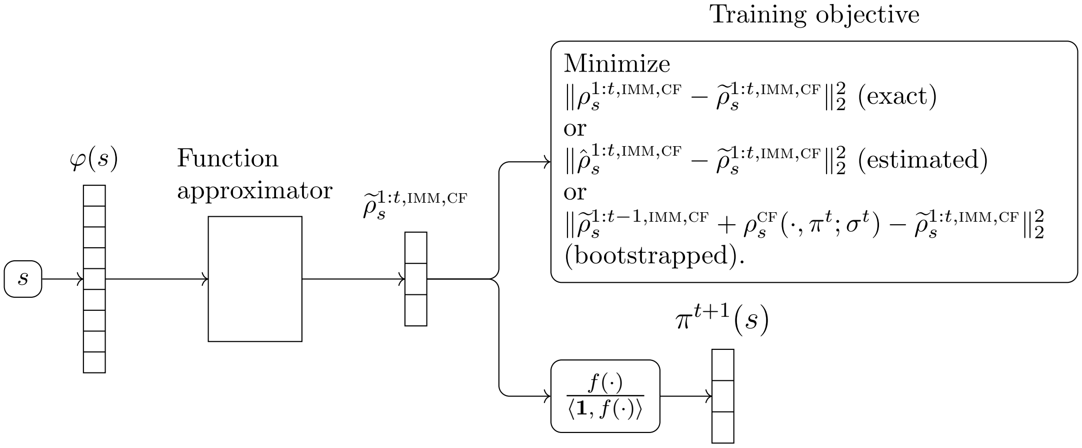

Training objective
Minimize
$\|\rho_s^{1:t,\text{IMM},\text{CF}} - \widetilde{\rho}_s^{1:t,\text{IMM},\text{CF}}\|_2^2$ (exact)
or
$\|\hat{\rho}_s^{1:t,\text{IMM},\text{CF}} - \widetilde{\rho}_s^{1:t,\text{IMM},\text{CF}}\|_2^2$ (estimated)
or
$\|\widetilde{\rho}_s^{1:t-1,\text{IMM},\text{CF}} + \rho_s^{\text{CF}}(\cdot, \pi^t; \sigma^t) - \widetilde{\rho}_s^{1:t,\text{IMM},\text{CF}}\|_2^2$
(bootstrapped).
$\varphi(s)$
Function approximator
$\widetilde{\rho}_s^{1:t,\text{IMM},\text{CF}}$
s
$\pi^{t+1}(s)$
$\frac{f(\cdot)}{\langle \mathbf{1}, f(\cdot)\rangle}$

Figure 10.1: The f-RCFR pipeline from agent state to immediate strategy. The only difference from Fig. 8.1 is the application of a general link function f rather than the specific ramp link function.

Steinberger et al. 2020; Waugh, Morrill, et al. 2015). See Fig. 10.1 for a visualization of the f-RCFR architecture.

Since Theorem 5 is parameterized by an immediate regret bound, the improvements to Theorem 24 over the original regression regret matching bound carry over to improvements over the original RCFR bound. We also achieve a new bound for the softmax parameterization/exponential link function with a better dependence on the number of immediate deviations according to Theorem 25.

Corollary 3. *Given a finite-horizon POHP and perfect-recall updates with maximum depth d_*, instantiate EFR with the blind counterfactual deviations and approximate regret matching with the $(p-1)$-power polynomial link function. Denote cumulative regret estimates at each agent state s on each round t as $\widetilde{\rho}_s^{1:t-1} \in \mathbb{R}^{|\mathcal{A}(s)|}$. This is an instance of f-RCFR where the maximum cumulative link approximation error is $\epsilon : T \mapsto \max_{s \in \mathcal{S}_\mathcal{A}} \sum_{t=1}^T \|g(\rho_s^{1:t-1}) - g(\widetilde{\rho}_s^{1:t-1})\|_1$ where g is defined pointwise as $g_i(x) = 0$ if $x_i \leq 0$ and $g_i(x) = \frac{p \wedge 2}{\|[x]_+\|_p^{(p-2)\vee 0}} x_i^{p-1}$ otherwise. The full blind counterfactual and external regret after T rounds of this instance of f-RCFR is no more than*

$$2d_* U |\mathcal{S}_\mathcal{A}| \sqrt[p-1]{n_\mathcal{A} - 1} \sqrt[p\wedge 2]{((p-1) \vee 1)T + \frac{1}{2U(n_\mathcal{A} - 1)^{1 \wedge 2/p}} \epsilon(T)}.$$

Corollary 4. *Instantiate EFR with the blind counterfactual deviations and approximate regret matching with the τ-exponential link function. Denote cumulative regret estimates at each agent state s on each round t as $\widetilde{\rho}_s^{1:t-1} \in \mathbb{R}^{|\mathcal{A}(s)|}$. This is an instance of f-RCFR where the maximum cumulative link approximation error is $\epsilon : T \mapsto \max_{s \in \mathcal{S}_\mathcal{A}} \sum_{t=1}^T \|g(\rho_s^{1:t-1}) - g(\widetilde{\rho}_s^{1:t-1})\|_1$ $g(\cdot) = \frac{\exp(\frac{1}{\tau} \cdot)}{\langle \mathbf{1}, \exp(\frac{1}{\tau} \cdot)\rangle}$. The full blind counterfactual* *and external regret after T rounds of this instance of f-RCFR is no more than*

$$|\mathcal{S}_{\mathcal{A}}|\left(\tau \ln n_{\mathcal{A}} + 2(d_* U)^2 \frac{T}{\tau} + 2 d_* U \epsilon(T)\right).$$

10.3.1 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, two-player Leduc hold'em poker (Southey et al. 2005) and two-player imperfect information goofspiel (Lanctot 2013) with linear function approximation. See Sections 7.A.1 and 7.A.2 for game details. Our experiments use the *OpenSpiel* (Lanctot, Lockhart, et al. 2019) implementations of these games.

The payoffs in each game are reported so that they have similar scales; milli-big blinds (mbb) for Leduc hold'em and milli-utils (mu) for goofspiel, where one "util" is the payoff for winning a game of goofspiel. The perfect-recall representation of Leduc hold'em contains 936 active agent states across both players. 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". This five rank version of goofspiel is roughly twice as large as Leduc hold'em at 2124 perfect-recall active agent states, while random goofspiel is larger still at 3608 perfect-recall active agent states.

These games are zero-sum, so a natural way to evaluate f-RCFR is to observe the exploitability of its average strategy profile generated during self-play.

Algorithm Implementation

Our regret estimators are independent linear function approximators for each player, $i \in \{1, 2\}$, and action $a \in \bigcup_{s \in \mathcal{S}_{i,\mathcal{A}}} \mathcal{A}(s)$. Our features are based on tug-of-war hashing features (Bellemare et al. 2012).

For each action, we randomly partition the agent states that share that 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 assigned to reduce bias introduced by hash collisions (sampled independently). The expected feature value over all agent states that share a non-zero entry in their feature vectors is, by design, zero. We use the number of partitions, n, to control the severity of approximation in our experiments; the larger n is, the more precisely agent states can be discriminated between, at the cost of more weights in the function approximator.

More formally, consider n random hash functions from a universal family. Each such hash function maps an agent state to indices, *i.e.*, $\{\zeta_i : \mathcal{S}_{i,\mathcal{A}} \to \{1, \ldots, m\}\}_{i=1}^{n}$. The feature representation $\varphi(s) \in \mathbb{R}^{mn}$ is an n-sparse vector with non-zero entries at the indices selected by the hash functions, *i.e.*, $\{\zeta_1(s), m + \zeta_2(s), 2m + \zeta_3(s), \ldots, m(n-1) + \zeta_n(s)\}$. The value of the non-zero entries are all either $+1$ or -1, determined by n additional random hash functions $\{\zeta_i : \mathcal{S}_{i,\mathcal{A}} \to \{-1, +1\}\}_{i=n+1}^{2n}$. This gives us the following feature vector, $\varphi(s) = \sum_{i=1}^{n} \zeta_{n+i}(s) \boldsymbol{e}_{(i-1)m+\zeta_i(s)}$, where $\boldsymbol{e}_j$ is the unit vector in direction j.

We do ridge regression on instantaneous counterfactual regret targets to update our regret estimators. After the first iteration, we simply add this new vector of weights to our previous weights. Since the instantaneous counterfactual regrets are computed for each agent state–action sequence on every iteration, the same feature matrix is used in the regression after each iteration. Therefore, the ridge regression solution for predicting the cumulative counterfactual regrets is just the sum of ridge regression solutions for predicting each of the instantaneous counterfactual regrets. Beyond computing 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 strategies for each instance exactly in a table. While this is less practical than other approaches, such as learning the average strategies from data, it removes another variable from the analysis and allows us to examine the impact of different link functions in relative isolation. Equivalently, we could have saved copies of the regret estimator weights across all iterations and computed the average policy on demand as suggested by Steinberger (2019).

Parameters

The appearance of function approximation error within the f-RCFR regret bounds (Corollaries 3 and 4) 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 test the polynomial link function with $p \in \{1.1, 1.5, 2, 2.5, 3\}$ to test values around the conventional $p = 2$ setting. The exponential link function is 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 approximation error, we examine the empirical exploitability of f-RCFR with different levels of approximation. The degree of approximation is adjusted via the quality of the features. In particular, we vary the number of partitions, n. Increasing n increases discriminative power and reduces approximation error (Fig. 10.2).

The number of buckets in each partition is fixed at $m = 10$. If the number of agent states that share an action is not evenly divisible by ten, a subset of the buckets are assigned one more agent state than the others. Thus, adding a partition adds ten features. Only one feature per partition is non-zero for any given agent state, so the prediction cost grows

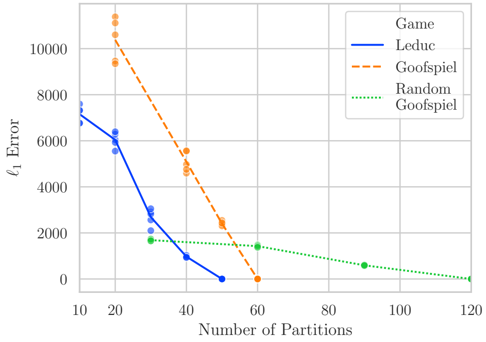


Figure 10.2: The cumulative counterfactual regret estimation error accumulated over iterations and agent states for select f-RCFR instances in Leduc hold'em poker, goofspiel, and random goofspiel. For each game and setting of the number of partitions, we select the link function and the parameter with 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.

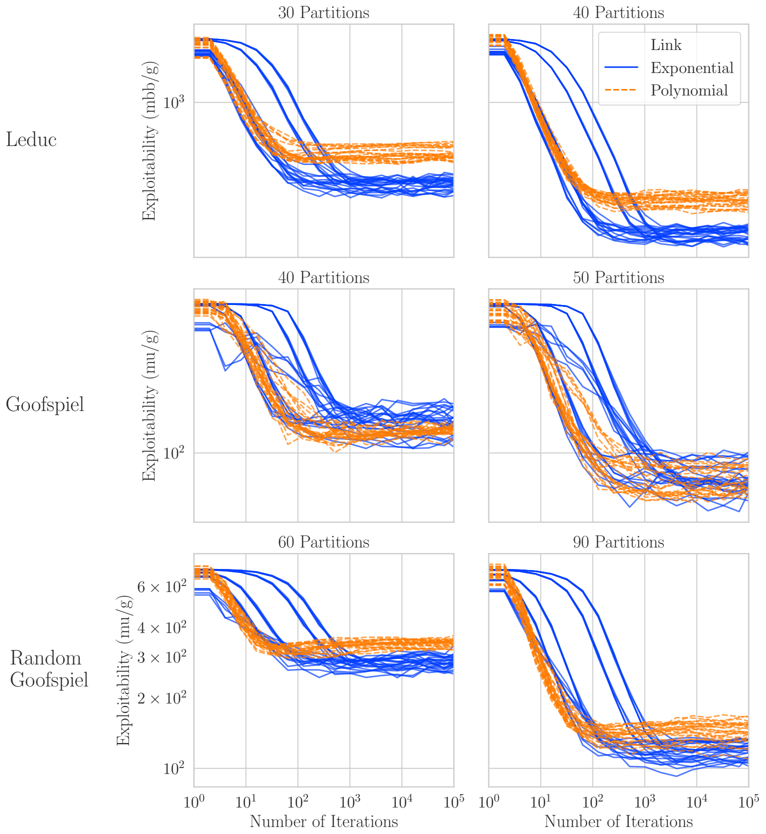


Figure 10.3: Exploitability of the average strategy profile for all configurations and runs with the exponential and polynomial link functions.

linearly with the number of partitions. The ridge regression update cost however, grows quadratically with the total number of features.

Results and Analysis

Figure 10.4 shows the average exploitability of the best link function and hyper-parameter configuration during learning (top) and after 100k-iterations (bottom), where an iteration is a round for each player. Players are updated in an alternating pattern, *i.e.*, player one is updated given player two's strategy and then player two is updated according to player one's updated strategy (Burch, Moravčík, et al. 2019; Tammelin 2014). The best parameterization was selected according to the average final exploitability after 100k-iterations over 5-runs. 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 (Fig. 10.2).

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 including

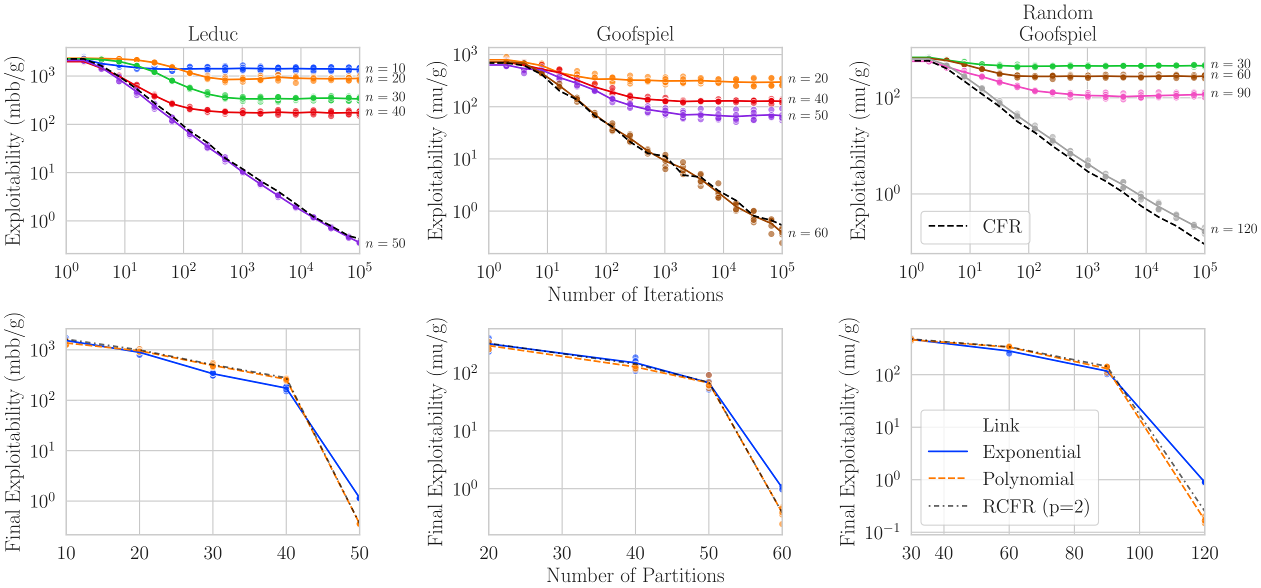


Figure 10.4: (top) The exploitability of the average strategy profile of tabular CFR and f-RCFR instances during the first 100k-iterations in Leduc hold'em (top left), goofspiel (top center), and random goofspiel (top right). For each setting of the number of partitions, we show the performance of the 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).

the original RCFR (polynomial link with $p = 2$; Fig. 10.3, top). The same occurs in random goofspiel with 60 or 90-partitions (Fig. 10.3, bottom). These feature parameters correspond to a moderate amount of function approximation error. This performance difference is noticeable across most configurations of the exponential and polynomial link in Leduc hold'em. Both link functions perform similarly in goofspiel with 40 or 50-partitions (Fig. 10.3, center).

The exponential link function does not outperform the polynomial link function in goofspiel or when the number of partitions is large, however (Fig. 10.3, center, and Fig. 10.4, 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 (Fig. 10.4 (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.

10.4 Neural Replicator Dynamics

Applying the softmax link function to f-RCFR yields a version of CFR with Hedge at each agent state where action transformation preferences are generated by a function approximator rather than a table. As we saw in Section 3.3.1, Hedge and softmax policy gradient (SPG) in an ODP setting are procedurally similar, so what relationship does softmax RCFR have with SPG in the POHP setting? To answer this question, we examine an online softmax RCFR update using bootstrapped targets and compare this update procedure with that of SPG.

10.4.1 The Neural Replicator Dynamics Update

On each round t, f-RCFR generates action transformation preferences $\widetilde{\rho}_s^{1:t-1} = \widetilde{\rho}(s; \theta^t) \in \mathbb{R}^{|\Phi_s|}$ for a given agent state s, determined by d function approximation parameters $\widetilde{\theta}^t \in \mathbb{R}^d$. A bootstrapped RCFR update target is an estimate of the cumulative regret vector generated from the current function approximator parameters plus the next instantaneous regret, *i.e.*, $\widehat{\rho}_s^{1:t} = \widetilde{\rho}_s^{1:t-1} + \rho_s^t \approx \rho_s^{1:t}$. If the vector $\widetilde{\theta}^{t+1}$ is chosen on each round t so that the squared Euclidean distance between $\widehat{\rho}_s^{1:t}$ and $\widetilde{\rho}(s; \theta^{t+1})$ is zero, *i.e.*, $\|\widehat{\rho}_s^{1:t} - \widetilde{\rho}(s; \theta^{t+1})\|_2^2 = 0$, then $\widetilde{\rho}(s; \theta^{t+1}) = \rho_s^{1:t}$. In this case, softmax RCFR exactly reproduces CFR with Hedge.

Rather than trying to ensure that this distance is always zero, which may not be possible according to the structure of $\widetilde{\rho}$, a natural approximation is to update $\widetilde{\rho}^{1:t-1}(s; \theta^t)$ so that it is more like $\widehat{\rho}_s^{1:t}$ after each round. Taking a single step of size $\frac{1}{\tau}$ in the direction of the gradient of the distance results in the online update $\theta^{t+1} = \theta^t + \frac{1}{\tau} \nabla_{\theta^t} \|[\![\widehat{\rho}_s^{1:t}]\!] - \widetilde{\rho}(s; \theta^t)\|_2^2$ at agent state s, where $[\![\cdot]\!]$ is the "stop gradient" operator. This update simplifies to

$$\begin{aligned} \theta^{t+1} &= \theta^t + \frac{1}{\tau} \sum_{\phi_s \in \Phi_s} \widetilde{\rho}_{\phi_s}(s; \theta^t) + \rho_{s,\phi_s}^t - \widetilde{\rho}_{\phi_s}(s; \theta^t) \nabla_{\theta^t} \widetilde{\rho}_{\phi_s}(s; \theta^t) && (10.17) \\ &= \theta^t + \frac{1}{\tau} \sum_{\phi_s \in \Phi_s} \rho_{s,\phi_s}^t \nabla_{\theta^t} \widetilde{\rho}_{\phi_s}(s; \theta^t). && (10.18) \end{aligned}$$

Using this approach, $\widetilde{\rho}$ generates approximations of cumulative regrets, and thus, approximations of the Hedge preferences, with fast incremental updates and without storing instantaneous regrets from previous rounds. Softmax RCFR using this particular update methodology has been dubbed *neural replicator dynamics* (*NeuRD*) for its connection with the replicator dynamics, which is described next. See Fig. 10.5 for a depiction of the NeuRD pipeline from active agent state to immediate strategy.

10.4.2 Connection with Replicator Dynamics

The *replicator dynamics* describe the continuous time evolution of a population via selection and mutation pressures (J. Hofbauer et al. 1998; Taylor 1979; Taylor and Jonker 1978;

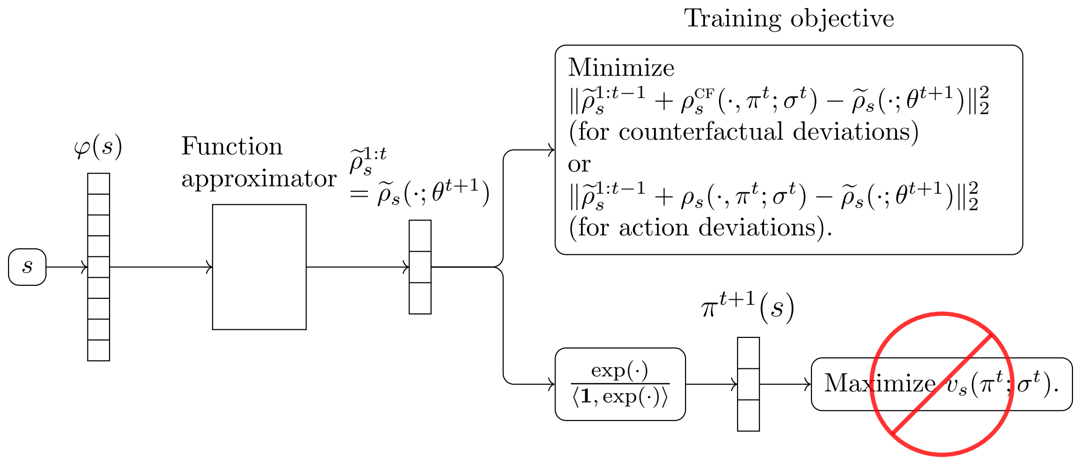

Training objective
Minimize
$\|\widetilde{\rho}_s^{1:t-1} + \rho_s^{\text{CF}}(\cdot, \pi^t; \sigma^t) - \widetilde{\rho}_s(\cdot; \theta^{t+1})\|_2^2$
(for counterfactual deviations)
or
$\|\widetilde{\rho}_s^{1:t-1} + \rho_s(\cdot, \pi^t; \sigma^t) - \widetilde{\rho}_s(\cdot; \theta^{t+1})\|_2^2$
(for action deviations).
$\varphi(s)$
Function approximator
$\widetilde{\rho}_s^{1:t} = \widetilde{\rho}_s(\cdot; \theta^{t+1})$
s
$\pi^{t+1}(s)$
$\frac{\exp(\cdot)}{\langle \mathbf{1}, \exp(\cdot) \rangle}$
Maximize $\upsilon_s(\pi^t; \sigma^t)$.

Figure 10.5: The NeuRD pipeline from agent state to immediate strategy.

Zeeman 1980, 1981). If there is a finite set of different species in a population, $\mathcal{X}$, where the relative proportion of each species is described by a distribution $\pi \in \Delta(\mathcal{X})$ and the fitness of each species within the population is described by a bounded function $\upsilon : \mathcal{X} \times \Delta(\mathcal{X}) \to [-U, U]$, then the replicator dynamic of species x is defined by the differential equation

$$\dot{\pi}(x) = \pi(x)(\upsilon(x; \pi) - \mathbb{E}_{X \sim \pi}[\upsilon(X; \pi)]). \tag{10.19}$$

That is, the time derivative of the prevalence of a species x in the replicator dynamics is proportional to the difference in x's fitness within the current population and the average fitness of the whole population. For asymmetric games where a daimon chooses strategy σ, the replicator dynamics becomes $\dot{\pi}(x) = \pi(x)\rho(\phi^{\to x}, \pi; \sigma)$. Solving this differential equation gives an exponential form, $\pi^T \propto \exp(\int_0^T \rho(\phi^{\to x}, \pi^t; \sigma^t) dt)$, for weights after a duration of T. Discretizing $\int_0^T \rho(\phi^{\to x}, \pi^t; \sigma^t) dt$ with discrete rounds yields the incremental update $\tilde{\theta}_x^T = \tilde{\theta}_x^{T-1} + \frac{1}{\tau}\rho(\phi^{\to x}, \pi^{T-1}; \sigma^{T-1}) = \frac{1}{\tau}\rho^{1:T-1}$, where $\tilde{\theta}^0 = \mathbf{0}$, that results in the policy, $\pi^T \propto \mathrm{e}^{\tilde{\theta}^T} = \exp(\frac{1}{\tau}\rho^{1:T-1})$, which is Hedge. The replicator equation Eq. (10.19) can likewise be obtained by taking the continuous time limit of the Hedge algorithm as the step size is driven to zero (see Section 3.1 of Krichene (2016) for details). Through this connection, we can see that since NeuRD is a function approximation generalization of Hedge, it is also that for the replicator dynamics.

10.4.3 Comparison with Softmax Policy Gradient

Recall the SPG algorithm from Sections 2.5 and 8.3. In a finite-horizon POHP with timed updates, the partial derivative of the expected return of the policy composed of immediate strategies $\pi^t(s) \propto \exp(g(s; \theta^t))$ for each active agent state s, generated with function approximator g and parameters $\theta^t \in \mathbb{R}^d$, with respect to parameter θ^t_i is

$$\frac{\partial v(\pi^t; \sigma^t)}{\partial \theta^t_i} = \sum_{s \in \mathcal{S}_{\mathcal{A}}} \frac{\partial v_s(\pi^t; \sigma^t)}{\partial \theta^t_i}. \tag{10.20}$$

The same variable's partial derivative of the counterfactual value function at active agent state s is

$$\frac{\partial v_s(\pi^t; \sigma^t)}{\partial \theta^t_i} = \sum_{a \in \mathcal{A}(s)} v_s(\phi^{\to a}_s(\pi^t); \sigma^t) \sum_{a' \in \mathcal{A}(s)} \frac{\partial \pi^t(a)}{\partial g_{a'}(s; \theta^t)} \frac{\partial g_{a'}(s; \theta^t)}{\partial \theta^t_i}. \tag{10.21}$$

By Section 2.8 of R. Sutton et al. (2018),

$$\frac{\partial v_s(\pi^t; \sigma^t)}{\partial \theta^t_i} = \sum_{a \in \mathcal{A}(s)} v_s(\phi^{\to a}_s(\pi^t); \sigma^t) \sum_{a' \in \mathcal{A}(s)} \pi^t(a)\big(\mathbb{1}\{a' = a\} - \pi^t(a')\big) \frac{\partial g_{a'}(s; \theta^t)}{\partial \theta^t_i}. \tag{10.22}$$

Swapping the order of the summations and repeating the reasoning described in Section 3.3.1 leads SPG to reduce to Eq. (3.6) in the ODP setting,

$$\frac{\partial v_s(\pi^t; \sigma^t)}{\partial \theta^t_i} = \sum_{a \in \mathcal{A}(s)} \pi^t(a \,|\, s) \rho_s(\phi^{\to a}_s, \pi^t; \sigma^t) \frac{\partial g_a(s; \theta^t)}{\partial \theta^t_i}. \tag{10.23}$$

The SPG update at s is therefore

$$\theta^{t+1} = \theta^t + \frac{1}{\tau} \sum_{a \in \mathcal{A}(s)} \underbrace{\pi^t(a \,|\, s)}_{\text{Omitted in Eq. (10.18).}} \rho_s(\phi^{\to a}_s, \pi^t; \sigma^t) \frac{\partial g_a(s; \theta^t)}{\partial \theta^t_i}. \tag{10.24}$$

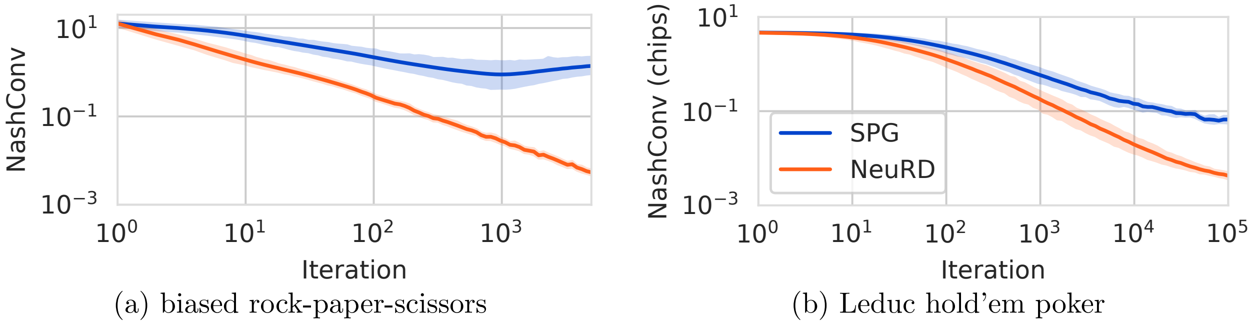
The NeuRD update, Eq. (10.18), on the external action transformations differs from that of SPG, Eq. (10.24), *only* in that NeuRD does not multiply the action deviation regret of action a by $\pi^t(a \,|\, s)$.

I contribute Corollaries 3.1 and 3.2 of Omidshafiei et al. (2019), giving regret and equilibrium approximation guarantees for applying tabular NeuRD (*i.e.*, Hedge) as the local learner in CFR, which are restated here.

Corollary 5. *Tabular NeuRD on the blind counterfactual deviations,* i.e., $\widetilde{\rho}^{1:t}_s = \widetilde{\rho}^{1:t-1}_s + \rho^{\text{CF}}_s(\cdot, \pi^t; \sigma^t)$ *and* $\widetilde{\rho}(s; \theta^{t+1}) = \rho^{1:t,\text{IMM},\text{CF}}_s$*, is CFR with local Hedge learners and setting the step size in active agent state s to* $\frac{1}{\tau_s} = \sqrt{2 \ln(|\mathcal{A}(s)|) T^{-1}}$ *on each round in a finite-horizon POHP with perfect-recall updates, ensures that NeuRD has a cumulative blind counterfactual and external deviation regret upper bound of* $2Ud_*|\mathcal{S}_{\mathcal{A}}|\sqrt{2 \ln n_{\mathcal{A}} T}$ *after T rounds.*

Proof. This follows from the definitions of NeuRD, blind counterfactual deviations, CFR, and Hedge, and by substituting the Hedge regret bound (Cesa-Bianchi et al. 2006; Freund et al. 1997) into CFR's abstract regret bounds (Theorem 10 and Lemma 5). □

Corollary 6. *In a two-player, zero-sum game, tabular NeuRD on the blind counterfactual deviations in self-play generates an average strategy that is an ε-Nash equilibrium, with ε no larger than the sum over players of the average (collected over iterations) external regret.*



(a) biased rock-paper-scissors

(b) Leduc hold'em poker

Figure 10.6: a NashConv of the average NeuRD and SPG policies in biased rock-paper-scissors (roshambo) (see Section 10.A.1). b Average policy NashConv in Leduc hold'em. Algorithms are tabular NeuRD on the blind counterfactual deviations and CFR using SPG as a local learner (basically, SPG on the blind counterfactual deviations).

Proof. Corollary 5 and Proposition 3. □

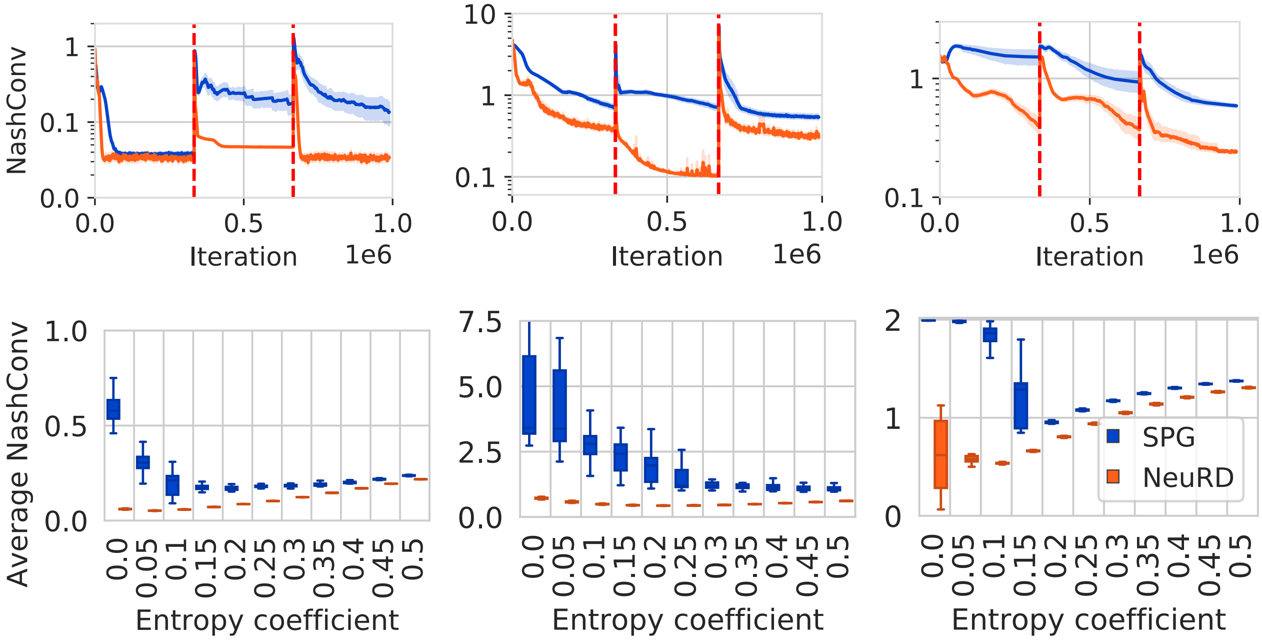
In addition, my coauthors derive non-obvious connections to SPG and natural policy gradient (Kakade 2002). See Omidshafiei et al. (2019, Appendices A.3 and A.4) for proofs.

Theorem 26. *Setting the NeuRD difference function to be the Kullback-Leibler divergence instead of the squared Euclidean distance converts NeuRD into SPG.*

Theorem 27. *The NeuRD update rule (Eq.* (10.18)*), is a naturalized policy gradient rule, in the sense that NeuRD applies a natural gradient only at the policy output level of softmax function over preferences, and backpropagates the standard gradient otherwise.*

10.4.4 Experiments

How does NeuRD compare to SPG in practice? Figure 10.6 shows that tabular NeuRD on the blind counterfactual deviations reduces NashConv substantially faster and more consistently than CFR using SPG as its local learning algorithm (basically, SPG on the counterfactual deviations) in games with and without sequential decision-making. Figure 10.7 compares scalable versions of NeuRD on the blind action deviations and SPG with deep neural network function approximation, outcome sampling, variance reduction, and entropy reward bonuses in non-stationary versions of three sequential imperfect-information card games. See Section 10.A for more experiment details. In these cases as well, NeuRD more quickly reaches better equilibrium approximations than SPG, and more gracefully adapts to utility function changes.



(a) Kuhn poker

(b) Leduc hold'em

(c) goofspiel

Figure 10.7: The NashConv (top) and average NashConv across all iterations (bottom) of the current policies of outcome sampled NeuRD on the blind action deviations and SPG with entropy reward bonuses (Perolat et al. 2021), across non-stationary modifications of Kuhn poker (see Section 10.A.1), Leduc hold'em, and goofspiel. The vertical dashed lines show where the utility function is negated. The first and third phases of each game have the same utility functions. These results summarize forty independent runs. (top) The averages across forty independent runs are shown as solid lines and the shaded region depicts the 95% confidence interval.

10.5 Conclusion

In this chapter, regression regret matching theory was generalized in two dimensions, the link function to include the polynomial and exponential link functions and deviation functions to include external and internal regret. The generalization to different link functions allowed us to construct regret bounds for a general f-RCFR algorithm. This chapter showed how f-RCFR is observably sequentially hindsight rational for the blind counterfactual deviations with the polynomial and exponential link functions as long its function approximator accurately reproduces tabular cumulative counterfactual regrets.

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

Generalizing RCFR to allow for the use of the softmax link function reveals connections between RCFR and traditional online learning, evolutionary game theory, and RL algorithms. This chapter showed how softmax RCFR with online gradient updates and bootstrapped targets results in the NeuRD algorithm, a generalization of Hedge and replicator dynamics, as well as a minimal change of the SPG algorithm. NeuRD is a theoretically grounded algorithm for achieving hindsight rationality and is implemented almost identically to SPG. Experiments compared the performance of NeuRD and SPG in games and non-stationary environments, which showed that NeuRD substantially outperforms SPG in these environments.

References

Bellemare, M., J. Veness, and M. Bowling (2012). “Sketch-based linear value function approximation”. In: *Advances in Neural Information Processing Systems*, pp. 2213–2221.

Brown, N., A. Lerer, S. Gross, and T. Sandholm (2019). “Deep Counterfactual Regret Minimization”. In: *36th International Conference on Machine Learning (ICML 2019)*, pp. 793–802.

Burch, N., M. Moravčík, and M. Schmid (2019). “Revisiting CFR$^+$ and Alternating Updates”. In: *Journal of Artificial Intelligence Research* 64, pp. 429–443.

Cesa-Bianchi, N. and G. Lugosi (2006). *Prediction, Learning, and Games.* Cambridge University Press.

D’Orazio, R., D. Morrill, J. R. Wright, and M. Bowling (2019). “Alternative Function Approximation Parameterizations for Solving Games: An Analysis of f-Regression Counterfactual Regret Minimization”. In: *arXiv preprint arXiv:1912.02967.*

Freund, Y. and R. E. Schapire (1997). “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”. In: *Journal of Computer and System Sciences* 55.1, pp. 119–139.

Greenwald, A., Z. Li, and C. Marks (Jan. 2006a). “Bounds for Regret-Matching Algorithms”. In: *International Symposium on Artificial Intelligence and Mathematics (ISAIM 2006).* Fort Lauderdale, Florida, USA.

— (June 2006b). *Bounds for Regret-Matching Algorithms.* Tech. rep. CS-06-10. Brown University, Department of Computer Science.

J. Hofbauer, J. and K. Sigmund (1998). “Evolutionary games and population dynamics”. In: *Cambridge University Press.*

Kakade, S. M. (2002). “A natural policy gradient”. In: *Advances in Neural Information Processing Systems*, pp. 1531–1538.

Krichene, W. (2016). “Continuous and discrete dynamics for online learning and convex optimization”. In: *Ph. D. Dissertation.*

Lanctot, M. (June 2013). “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.

Lanctot, M., E. Lockhart, et al. (2019). "OpenSpiel: A Framework for Reinforcement Learning in Games". In: *CoRR* abs/1908.09453. arXiv: 1908.09453 `[cs.LG]`.

Li, H., K. Hu, S. Zhang, Y. Qi, and L. Song (2019). "Double Neural Counterfactual Regret Minimization". In: *International Conference on Learning Representations*.

Morrill, D. (2016). "Using Regret Estimation to Solve Games Compactly". Master's thesis. University of Alberta.

Omidshafiei, S., D. Hennes, D. Morrill, R. Munos, J. Pérolat, M. Lanctot, A. Gruslys, J.-B. Lespiau, and K. Tuyls (2019). "Neural Replicator Dynamics". In: *CoRR* abs/1906.00190. arXiv: 1906.00190.

Perolat, J., R. Munos, J.-B. Lespiau, S. Omidshafiei, M. Rowland, P. Ortega, N. Burch, T. Anthony, D. Balduzzi, B. De Vylder, et al. (2021). "From Poincaré recurrence to convergence in imperfect information games: Finding equilibrium via regularization". In: *International Conference on Machine Learning*. PMLR, pp. 8525–8535.

Southey, F., M. Bowling, B. Larson, C. Piccione, N. Burch, D. Billings, and D. C. Rayner (July 2005). "Bayes' Bluff: Opponent Modelling in Poker". In: *21st Conference in Uncertainty in Artificial Intelligence (UAI 2005)*. Edinburgh, Scotland, pp. 550–558.

Steinberger, E. (2019). "Single Deep Counterfactual Regret Minimization". In: *arXiv preprint arXiv:1901.07621*.

Steinberger, E., A. Lerer, and N. Brown (2020). "DREAM: Deep regret minimization with advantage baselines and model-free learning". In: *arXiv preprint arXiv:2006.10410*.

Sutton, R. and A. Barto (2018). *Reinforcement Learning: An Introduction*. 2nd. MIT Press.

Tammelin, O. (2014). "Solving Large Imperfect Information Games Using CFR+". In: *arXiv preprint arXiv:1407.5042*.

Taylor, P. (1979). "Evolutionarily Stable Strategies with Two Types of Players". In: *Journal of Applied Probability* 16, pp. 76–83.

Taylor, P. and L. Jonker (1978). "Evolutionarily Stable Strategies and Game Dynamics". In: *Mathematical Biosciences* 40, pp. 145–156.

Waugh, K., D. Morrill, J. A. Bagnell, and M. Bowling (2015). "Solving Games with Functional Regret Estimation". In: *29th AAAI Conference on Artificial Intelligence (AAAI-15)*. Vol. 29. 1, pp. 2138–2144.

Zeeman, E. (1980). "Population Dynamics from Game Theory". In: *Lecture Notes in Mathematics, Global Theory of Dynamical Systems* 819.

— (1981). "Dynamics of the evolution of animal conflicts". In: *Theoretical Biology* 89, pp. 249–270.

10.A NeuRD Experiments

10.A.1 Additional Games

Table 10.1: Player one's biased rock-paper-scissors payoffs.

	R	P	S
R	0	-1	20
P	1	0	-1
S	-20	1	0

Biased rock-paper-scissors uses the familiar rock-paper-scissors payoff matrix, except the reward for winning by playing scissors is twenty instead of one (see Table 10.1).

Kuhn poker is a one round poker game with a three card deck, each with a different rank. Both players ante a single chip into the pot before the round begins and are then dealt a single private card. In turn, the players can choose to bet or call the opponent's bet with another chip. Players facing a bet can also fold and forfeit their ante. If no player bets or the player facing a bet calls, their cards are revealed and the player with the highest ranking card wins the pot. This game has 12 perfect-recall active agent states.

10.A.2 Parameters and Training Regimes

For the tabular CFR experiments in Leduc hold'em displayed in Fig. 10.6b, the set of constant step sizes/inverse temperatures tried were the same for both algorithms: $\alpha \in \{0.5, 0.9, 1, 1.5, 2, 2.5, 3, 3.5, 4\}$. The shaded area corresponds to the 95% interval that would result from a uniform sampling of the step size from this set. An "iteration" consists of one (full tree walk) alternating CFR update for both players (Burch, Moravčík, et al. 2019).

Neural network experiments displayed in Fig. 10.7 were completed by coauthors and use two-layer neural networks with 128 hidden units initialized randomly. A conditional expected action value function with the same architecture is used to reduce variance. The action value network updates from batches of four on-policy sampled trajectories with trajectory lengths of five for Kuhn poker, and eight for Leduc hold'em and goofspiel, and a learning rate of 0.01. The policy network is updated once every four updates of the action value network with batches of 256 on-policy sampled trajectories and a learning rate of 0.002. An "iteration" consists then of four action value network updates and a single policy network update.

The reward function of each game is negated after every $1/3 \times 10^6$ iterations to split learning into three phases, as denoted by red dashed lines in Fig. 10.7 (top). The policy and action value networks are not reset when the reward functions are negated.

Chapter 11

Conclusion

This thesis has presented a critique of the common approach to designing and analyzing RL algorithms for policy optimization in Markov decision processes with the expectation that they will perform well in multi-agent and non-stationary environments. The hindsight rationality objective and the POHP formalism form an alternative framework for designing and analyzing RL algorithms specifically to be effective in multi-agent and non-stationary environments. Chapter 3 and Sections 9.3 and 10.4 show that these alternatives have various benefits in formulating principled goals for complex tasks, tackling new problems in AI safety, and improving reward accumulation performance. Algorithms designed to achieve hindsight rationality goals can perform better in environments with other agents, imperfect information, and non-stationary dynamics, compared to procedurally similar algorithms that are only designed to approximate optimal policies.

This thesis also presented a critique of the dichotomy between single-agent and multi-agent RL algorithms. EFR breaks down this binary by achieving multi-agent RL goals without performing explicit multi-agent reasoning. Explicitly searching joint strategy spaces or explicitly modeling other agents is often computationally difficult. Bard et al. (2013) shows that in the context of opponent modeling, it can be both computationally practical and empirically effective to "implicitly model" other agents with a bandit algorithm to select between various potential counter-strategies. In a similar way, EFR adapts to the play of other agents implicitly with a no-regret algorithm that competes with various deviations, which allows EFR to avoid expensive multi-agent reasoning.

The POHP formalism facilitates the analysis and development of single-agent RL algorithms like EFR that achieve multi-agent goals. Traditionally, algorithms like EFR would be analyzed in the extensive-form game (EFG) formalism, which causes friction for discussing algorithm properties in single-agent tasks, or those where the number of agents is unknown or changing. The EFG formalism also uses different objects and mechanisms to describe multi-agent interactions than traditional RL models like Markov decision processes or partially observable Markov games, which prevents algorithms from being straightforwardly compared, and prevents advances designed in one formalism from being shared to another. Work in this thesis, particularly the analysis of correlated equilibria and behavioral deviations (Chapter 6), the analysis and application of CFR in uncertain reward MDPs (Chapter 9), and the comparison between NeuRD and SPG (Section 10.4), shows that the POHP formalism can bridge these differences and serve as a common language for algorithmic game theory, single-agent RL, and multi-agent RL.

In the field of game theory, the causal deviations and their equilibria (EFCCE and EFCE) are the focus of most investigations into equilibria in sequential decision-making settings. Part II shows that there is nothing particularly special about the causal deviations. They are efficient to work with, but are more computationally expensive than counterfactual or action deviations without being clearly stronger than either. Chapter 6 suggests that perhaps the partial sequence deviations are more interesting because they have roughly the same computational cost as the causal deviations and do actually subsume the counterfactual and action deviations under observable sequential rationality. Regardless, Chapter 6 shows that exclusively focusing on any one class of deviations and equilibria is ultimately unjustified and limiting because of the computation–strength tradeoffs spread through the deviation space in sequential decision-making settings.

f-RCFR and NeuRD show that the hindsight rationality approach can be utilized along with the same function approximation architectures and similar training procedures as those used by policy optimization algorithms. In addition to the experiment depicted in Fig. 10.7, various works show that CFR and RCFR can also be used with Monte-Carlo sampling (see, *e.g.*, Brown, Lerer, et al. (2019), Davis et al. (2020), Lanctot, Waugh, et al. (2009), Schmid, Burch, et al. (2019), Steinberger et al. (2020), and Zinkevich, Johanson, et al. (2007b)). There is no technical barrier preventing the application of function approximation, Monte-Carlo sampling, and variance reduction in EFR, enabling EFR to be applied in substantially more complex settings than those investigated in this thesis.

The primary challenge in scaling EFR is to consistently generalize well across agent states and enable fast learning with generalization. The time selection regret matching theory (Theorem 15) rewards generalization as long the estimated cumulative regrets happen to be more like the cumulative regrets *after* observing the *next* utility function. Unfortunately, this theory does not suggest an obvious method to achieve such “predictive generalization”. Perhaps a step in this direction would be to have an algorithm that begins learning by aggressively generalizing to minimize regret quickly if nearly all agent states are in fact strategically identical, and gradually differentiating between different agent states that are unlikely to be strategically similar.

Hindsight rationality is an appealing objective for AI systems that are continually learning and adapting in complex environments because it requires few assumptions about the environment. However, hindsight rationality does require the imposition of discrete rounds, which leads to two independent dimensions representing time in a repeated POHP: time steps within the POHP and the round number (the number of times the POHP has been executed). This artificial partitioning of time is unnatural and untenable in continual learning tasks where there is no clear notion of rounds. Formulating hindsight rationality without imposing artificial round boundaries, perhaps by allowing agents to recover a sense of repetition from similarity between agent states or constructing a fixed-length, incrementally advancing horizon, is a promising research direction that would enable us to import all of the algorithmic tools developed for CFR and EFR into continual learning tasks.

Bibliography

Aumann, R. J. (1974). "Subjectivity and Correlation in Randomized Strategies". In: *Journal of Mathematical Economics* 1.1, pp. 67–96.

Baird, L. C. (1994). "Reinforcement learning in continuous time: Advantage updating". In: *1994 IEEE International Conference on Neural Networks (ICNN'94).* Vol. 4. IEEE, pp. 2448–2453.

Baker, B., I. Kanitscheider, T. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch (2019). "Emergent Tool Use From Multi-Agent Autocurricula". In: *International Conference on Learning Representations.*

Bansal, T., J. Pachocki, S. Sidor, I. Sutskever, and I. Mordatch (2018). "Emergent Complexity via Multi-Agent Competition". In: *6th International Conference on Learning Representations.*

Bard, N., M. Johanson, N. Burch, and M. Bowling (2013). "Online Implicit Agent Modelling". In: *International Conference on Autonomous Agents and Multiagent Systems (AAMAS).*

Bellemare, M., J. Veness, and M. Bowling (2012). "Sketch-based linear value function approximation". In: *Advances in Neural Information Processing Systems*, pp. 2213–2221.

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

Blum, A. and Y. Mansour (2007). "From External to Internal Regret". In: *Journal of Machine Learning Research* 8.6, pp. 1307–1324.

Bowling, M., N. Burch, M. Johanson, and O. Tammelin (2015). "Heads-up Limit Hold'em Poker is Solved". In: *Science* 347.6218, pp. 145–149.

Breitmoser, Y., J. H. Tan, and D. J. Zizzo (2010). "On the beliefs off the path: Equilibrium refinement due to quantal response and level-k". In: *Nottingham University Business School Research Paper* 2010-07.

Brown, N., A. Lerer, S. Gross, and T. Sandholm (2019). "Deep Counterfactual Regret Minimization". In: *36th International Conference on Machine Learning (ICML 2019)*, pp. 793–802.

Brown, N. and T. Sandholm (2018). "Superhuman AI for Heads-Up No-Limit Poker: Libratus Beats Top Professionals". In: *Science* 359.6374, pp. 418–424.

— (2019). "Superhuman AI for Multiplayer Poker". In: *Science* 365.6456, pp. 885–890.

Burch, N. (2017). "Time and space: Why imperfect information games are hard". PhD thesis. University of Alberta.

Burch, N., M. Lanctot, D. Szafron, and R. Gibson (2012). "Efficient Monte Carlo counterfactual regret minimization in games with many player actions". In: *Advances in Neural Information Processing Systems*, pp. 1880–1888.

Burch, N., M. Moravčík, and M. Schmid (2019). "Revisiting CFR$^+$ and Alternating Updates". In: *Journal of Artificial Intelligence Research* 64, pp. 429–443.

Celli, A., A. Marchesi, G. Farina, and N. Gatti (2020). "No-regret learning dynamics for extensive-form correlated equilibrium". In: *Advances in Neural Information Processing Systems* 33.

Cesa-Bianchi, N. and G. Lugosi (2006). *Prediction, Learning, and Games*. Cambridge University Press.

Chen, K. and M. Bowling (2012). "Tractable Objectives for Robust Policy Optimization". In: *Advances in Neural Information Processing Systems*, pp. 2069–2077.

Chow, Y., M. Ghavamzadeh, L. Janson, and M. Pavone (2017). "Risk-constrained reinforcement learning with percentile risk criteria". In: *The Journal of Machine Learning Research* 18.1, pp. 6070–6120.

Chow, Y., A. Tamar, S. Mannor, and M. Pavone (2015). "Risk-sensitive and robust decision-making: a CVaR optimization approach". In: *Neural Information Processing Systems* 28, pp. 1522–1530.

Clements, W. R., B. Van Delft, B.-M. Robaglia, R. B. Slaoui, and S. Toth (2019). "Estimating risk and uncertainty in deep reinforcement learning". In: *Workshop on Uncertainty and Robustness in Deep Learning at International Conference on Machine Learning*.

Cohen, G., S. Afshar, J. Tapson, and A. Van Schaik (2017). "EMNIST: Extending MNIST to handwritten letters". In: *International Joint Conference on Neural Networks*, pp. 2921–2926.

D'Orazio, R. (2020). "Regret Minimization with Function Approximation in Extensive-Form Games". Master's thesis. University of Alberta.

D'Orazio, R. and R. Huang (2021). "Optimistic and Adaptive Lagrangian Hedging". In: *AAAI Reinforcement Learning in Games Workshop*.

D'Orazio, R., D. Morrill, J. R. Wright, and M. Bowling (2019). "Alternative Function Approximation Parameterizations for Solving Games: An Analysis of f-Regression Counterfactual Regret Minimization". In: *arXiv preprint arXiv:1912.02967*.

— (May 2020). "Alternative Function Approximation Parameterizations for Solving Games: An Analysis of f-Regression Counterfactual Regret Minimization". In: *19th International Conference on Autonomous Agents and Multi-Agent Systems*. International Foundation for Autonomous Agents and Multiagent Systems.

Davis, T. (2015). "Using Response Functions for Strategy Training and Evaluation". Master's thesis. University of Alberta.

Davis, T., M. Schmid, and M. Bowling (2020). "Low-Variance and Zero-Variance Baselines for Extensive-Form Games". In: *International Conference on Machine Learning*. PMLR, pp. 2392–2401.

Dayan, P. (1993). "Improving generalization for temporal difference learning: The successor representation". In: *Neural Computation* 5.4, pp. 613–624.

Dekel, E. and M. M. Siniscalchi (2015). "Epistemic Game Theory". In: *Handbook of Game Theory With Economic Applications* 4, pp. 619–702.

Dong, S., B. Van Roy, and Z. Zhou (2021). "Simple Agent, Complex Environment: Efficient Reinforcement Learning with Agent States". In: *CoRR* abs/2102.05261.

Dudík, M. and G. J. Gordon (2009). "A Sampling-Based Approach to Computing Equilibria in Succinct Extensive-Form Games". In: *25th Conference on Uncertainty in Artificial Intelligence (UAI-2009)*.

Espeholt, L., H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, et al. (2018). "IMPALA: Scalable distributed Deep-RL with importance weighted actor-learner architectures". In: *ICML*.

Even-Dar, E., S. M. Kakade, and Y. Mansour (2005). "Experts in a Markov decision process". In: *Advances in Neural Information Processing Systems*, pp. 401–408.

Farina, G., T. Bianchi, and T. Sandholm (Feb. 2020). "Coarse Correlation in Extensive-Form Games". In: *34th AAAI Conference on Artificial Intelligence*. New York, New York, USA.

Farina, G., C. Kroer, N. Brown, and T. Sandholm (2019). "Stable-Predictive Optimistic Counterfactual Regret Minimization". In: *International Conference on Machine Learning*, pp. 1853–1862.

Farina, G., C. Kroer, and T. Sandholm (2019). "Online convex optimization for sequential decision processes and extensive-form games". In: *33rd AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 1917–1925.

— (2020). "Stochastic regret minimization in extensive-form games". In: *International Conference on Machine Learning*, pp. 3018–3028.

— (2021). "Faster Game Solving via Predictive Blackwell Approachability: Connecting Regret Matching and Mirror Descent". In: *35th AAAI Conference on Artificial Intelligence*. Vol. 35. 6, pp. 5363–5371.

Farina, G., C. K. Ling, F. Fang, and T. Sandholm (2019). "Correlation in Extensive-Form Games: Saddle-Point Formulation and Benchmarks". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Foerster, J., R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch (2018). "Learning with opponent-learning awareness". In: *17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, pp. 122–130.

Foerster, J., F. Song, E. Hughes, N. Burch, I. Dunning, S. Whiteson, M. Botvinick, and M. Bowling (2019). "Bayesian action decoder for deep multi-agent reinforcement learning". In: *International Conference on Machine Learning*. PMLR, pp. 1942–1951.

Foerster, J. N., G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson (2018). "Counterfactual multi-agent policy gradients". In: *32nd AAAI Conference on Artificial Intelligence*.

Forges, F. (1986). "Correlated equilibria in repeated games with lack of information on one side: a model with verifiable types". In: *International Journal of Game Theory* 15.2, pp. 65–82.

Foster, D. P. and R. Vohra (1999). "Regret in the On-Line Decision Problem". In: *Games and Economic Behavior* 29.1-2, pp. 7–35.

Freund, Y. and R. E. Schapire (1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1, pp. 119–139.

Ghavamzadeh, M., M. Petrik, and Y. Chow (2016). "Safe policy improvement by minimizing robust baseline regret". In: *Neural Information Processing Systems* 29, pp. 2298–2306.

Gibson, R. (2014). "Regret Minimization in Games and the Development of Champion Multiplayer Computer Poker-Playing Agents". PhD thesis. University of Alberta.

Gibson, R., M. Lanctot, N. Burch, D. Szafron, and M. Bowling (2012). “Generalized Sampling and Variance in Counterfactual Regret Minimization”. In: *26th Conference on Artificial Intelligence (AAAI-12)*, pp. 1355–1361.

Gordon, G. J. (2005). *No-Regret Algorithms for Structured Prediction Problems.* Tech. rep. CMU-CALD-05-112. Carnegie Mellon University.

Gordon, G. J., A. Greenwald, and C. Marks (2008). “No-Regret Learning in Convex Games”. In: *25th international conference on Machine learning*, pp. 360–367.

Graeber, D. (Dec. 2019). *From Managerial Feudalism to the Revolt of the Caring Classes.* Chaos Computer Club (via media.ccc.de).

Greenwald, A., A. Jafari, and C. Marks (Aug. 2003). “A general class of no-regret learning algorithms and game-theoretic equilibria”. In: *2003 Computational Learning Theory Conference*, pp. 1–11.

Greenwald, A., J. Li, and E. Sodomka (2017). “Solving for Best Responses and Equilibria in Extensive-Form Games with Reinforcement Learning Methods”. In: *Rohit Parikh on Logic, Language and Society.* Ed. by C. Başkent, L. S. Moss, and R. Ramanujam. Cham: Springer International Publishing, pp. 185–226. ISBN: 978-3-319-47843-2. DOI: 10.1007/978-3-319-47843-2_11.

Greenwald, A., Z. Li, and C. Marks (Jan. 2006a). “Bounds for Regret-Matching Algorithms”. In: *International Symposium on Artificial Intelligence and Mathematics (ISAIM 2006).* Fort Lauderdale, Florida, USA.

— (June 2006b). *Bounds for Regret-Matching Algorithms.* Tech. rep. CS-06-10. Brown University, Department of Computer Science.

Greenwald, A., Z. Li, and W. Schudy (2008). “More Efficient Internal-Regret-Minimizing Algorithms.” In: *COLT*, pp. 239–250.

Guennebaud, G., B. Jacob, et al. (2010). “Eigen”. In: *URl: http://eigen. tuxfamily. org.*

Hansen, E. A., D. S. Bernstein, and S. Zilberstein (2004). “Dynamic programming for partially observable stochastic games”. In: *20th AAAI Conference on Artificial Intelligence (AAAI-04).* Vol. 4, pp. 709–715.

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

Hennes, D., D. Morrill, S. Omidshafiei, R. Munos, J. Perolat, M. Lanctot, A. Gruslys, J.-B. Lespiau, P. Parmas, E. Duéñez-Guzmán, et al. (2020). “Neural replicator dynamics: Multiagent learning via hedging policy gradients”. In: *19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2020)*, pp. 492–501.

Herbster, M. and M. K. Warmuth (1998). “Tracking the best expert”. In: *Machine learning* 32.2, pp. 151–178.

Heskes, T., W. Wiegerinck, and H. Kappen (1997). “Practical confidence and prediction intervals for prediction tasks”. In: *Progress in Neural Processing*, pp. 128–135.

Hillas, J. (1987). *Sequential equilibria and stable sets of beliefs.* Institute for Mathematical Studies in the Social Sciences, Stanford University.

Huang, W. (2011). “Equilibrium Computation for Extensive Games”. PhD thesis. London School of Economics and Political Science.

J. Hofbauer, J. and K. Sigmund (1998). “Evolutionary games and population dynamics”. In: *Cambridge University Press.*

Johanson, M., N. Bard, N. Burch, and M. Bowling (2012). "Finding Optimal Abstract Strategies in Extensive Form Games". In: *26th AAAI Conference on Artificial Intelligence (AAAI-12)*.

Johanson, M., N. Bard, M. Lanctot, R. Gibson, and M. Bowling (2012). "Efficient Nash Equilibrium Approximation through Monte Carlo Counterfactual Regret Minimization". In: *11th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Kahn, G., A. Villaflor, V. Pong, P. Abbeel, and S. Levine (2017). "Uncertainty-aware reinforcement learning for collision avoidance". In: *arXiv preprint arXiv:1702.01182*.

Kakade, S. M. (2002). "A natural policy gradient". In: *Advances in Neural Information Processing Systems*, pp. 1531–1538.

Kakade, S. M. (2003). "On the sample complexity of reinforcement learning". PhD thesis. UCL (University College London).

Kash, I. A., M. Sullins, and K. Hofmann (May 2020). "Combining no-regret and Q-learning". In: *19th International Conference on Autonomous Agents and Multi-Agent Systems*. International Foundation for Autonomous Agents and Multiagent Systems.

Kierkegaard, S. (1967). "Søren Kierkegaard's Journals and Papers. Volume 1, A-E". In: 01. Ed. by H. V. Hong, E. H. Hong, and G. Malantschuk.

Kingma, D. P. and J. Ba (2014). "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980. URL: http://arxiv.org/abs/1412.6980.

Kovařík, V., M. Schmid, N. Burch, M. Bowling, and V. Lisỳ (2019). "Rethinking formal models of partially observable multiagent decision making". In: *arXiv preprint arXiv:1906.11110*.

Kreps, D. M. and R. Wilson (1982). "Sequential equilibria". In: *Econometrica* 50.4, pp. 863–894.

Krichene, W. (2016). "Continuous and discrete dynamics for online learning and convex optimization". In: *Ph. D. Dissertation*.

Kuhn, H. W. (1953). "Extensive Games and the Problem of Information". In: *Contributions to the Theory of Games* 2. Ed. by H. W. Kuhn and A. W. Tucker, pp. 193–216.

Lakshminarayanan, B., A. Pritzel, and C. Blundell (2017). "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Neural Information Processing Systems*, pp. 6405–6416.

Lanctot, M. (June 2013). "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.

Lanctot, M., N. Burch, M. Zinkevich, M. Bowling, and R. G. Gibson (2012). "No-Regret Learning in Extensive-Form Games with Imperfect Recall". In: *29th International Conference on Machine Learning (ICML 2012)*, pp. 65–72.

Lanctot, M., E. Lockhart, et al. (2019). "OpenSpiel: A Framework for Reinforcement Learning in Games". In: *CoRR* abs/1908.09453. arXiv: 1908.09453 [cs.LG].

Lanctot, M., K. Waugh, M. Zinkevich, and M. Bowling (2009). "Monte Carlo Sampling for Regret Minimization in Extensive Games". In: *Advances in Neural Information Processing Systems 22*. Ed. by Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, pp. 1078–1086.

Lattimore, T. and C. Szepesvári (2020). *Bandit algorithms*. Cambridge University Press.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). “Gradient-based learning applied to document recognition”. In: *IEEE* 86.11, pp. 2278–2324.

Lehrer, E. (2003). “A Wide Range No-Regret Theorem”. In: *Games and Economic Behavior* 42.1, pp. 101–115.

Leike, J., M. Martic, V. Krakovna, P. A. Ortega, T. Everitt, A. Lefrancq, L. Orseau, and S. Legg (2017). “AI safety gridworlds”. In: *arXiv preprint arXiv:1711.09883*.

Li, H., K. Hu, S. Zhang, Y. Qi, and L. Song (2019). “Double Neural Counterfactual Regret Minimization”. In: *International Conference on Learning Representations*.

Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra (2015). “Continuous control with deep reinforcement learning”. In: *arXiv preprint arXiv:1509.02971*.

Lockhart, E., M. Lanctot, J. Pérolat, J.-B. Lespiau, D. Morrill, F. Timbers, and K. Tuyls (2019a). “Computing Approximate Equilibria in Sequential Adversarial Games by Exploitability Descent”. In: *28th International Joint Conference on Artificial Intelligence (IJCAI 2019)*.

— (2019b). “Computing approximate equilibria in sequential adversarial games by exploitability descent”. In: *arXiv preprint arXiv:1903.05614*.

Lowe, R., Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch (2017). “Multi-agent actor-critic for mixed cooperative-competitive environments”. In: *Advances in Neural Information Processing Systems*, pp. 6379–6390.

Lu, X. and B. Van Roy (2017). “Ensemble sampling”. In: *Neural Information Processing Systems*, pp. 3260–3268.

MacQueen, R. (May 2022). “Personal communication”.

McDiarmid, C. (1998). “Concentration”. In: *Probabilistic methods for algorithmic discrete mathematics*, pp. 195–248.

Mei, J., C. Xiao, B. Dai, L. Li, C. Szepesvári, and D. Schuurmans (2020). “Escaping the Gravitational Pull of Softmax.” In: *NeurIPS*.

Mnih, V., A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu (2016). “Asynchronous Methods for Deep Reinforcement Learning”. In: *33rd International Conference on Machine Learning (ICML)*, pp. 1928–1937.

Mohammedalamen, M., D. Morrill, A. Sieusahai, Y. Satsangi, and M. Bowling (2021). “Learning to Be Cautious”. In: *arXiv preprint arXiv:2110.15907*.

Moravčík, M., M. Schmid, N. Burch, V. Lisỳ, D. Morrill, N. Bard, T. Davis, K. Waugh, M. Johanson, and M. Bowling (2017). “DeepStack: Expert-Level Artificial Intelligence in Heads-Up No-Limit Poker”. In: *Science* 356.6337, pp. 508–513.

Morrill, D. (2016). “Using Regret Estimation to Solve Games Compactly”. Master’s thesis. University of Alberta.

Morrill, D., R. D’Orazio, M. Lanctot, J. R. Wright, M. Bowling, and A. Greenwald (2021). “Efficient Deviation Types and Learning for Hindsight Rationality in Extensive-Form Games”. In: *CoRR* abs/2102.06973.

Morrill, D., R. D’Orazio, M. Lanctot, J. R. Wright, M. Bowling, and A. R. Greenwald (July 2021). “Efficient Deviation Types and Learning for Hindsight Rationality in Extensive-Form Games”. In: *38th International Conference on Machine Learning (ICML 2021)*. Vol. 139. virtual, pp. 7818–7828.

Morrill, D., R. D'Orazio, R. Sarfati, M. Lanctot, J. R. Wright, A. Greenwald, and M. Bowling (2022). "Hindsight and Sequential Rationality of Correlated Play". In: *CoRR* abs/2012.05874.

Morrill, D., R. D'Orazio, R. Sarfati, M. Lanctot, J. R. Wright, A. R. Greenwald, and M. Bowling (Feb. 2021). "Hindsight and Sequential Rationality of Correlated Play". In: *35th AAAI Conference on Artificial Intelligence*. Vol. 35. 6. virtual, pp. 5584–5594.

Morrill, D., A. R. Greenwald, and M. Bowling (2022). "The Partially Observable History Process". In: *AAAI-22 Workshop on Reinforcement Learning and Games*.

Moulin, H. and J.-P. Vial (1978). "Strategically zero-sum games: the class of games whose completely mixed equilibria cannot be improved upon". In: *International Journal of Game Theory* 7.3-4, pp. 201–221.

Myerson, R. B. (1997). *Game Theory: Analysis of Conflict*. Harvard university press.

Nash, J. (1951). "Non-Cooperative Games". In: *The Annals of Mathematics* 54.2, pp. 286–295.

Neumann, J. von and O. Morgenstern (1947). *The Theory of Games and Economic Behavior*. 2nd. Princeton University Press.

Omidshafiei, S., D. Hennes, D. Morrill, R. Munos, J. Pérolat, M. Lanctot, A. Gruslys, J.-B. Lespiau, and K. Tuyls (2019). "Neural Replicator Dynamics". In: *CoRR* abs/1906.00190. arXiv: 1906.00190.

Osband, I., B. Van Roy, D. J. Russo, and Z. Wen (2019). "Deep exploration via randomized value functions". In: *Journal of Machine Learning Research* 20.124, pp. 1–62.

Paszke, A. et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*, pp. 8024–8035.

Pearce, T., M. Zaki, and A. Neely (2018). "Bayesian neural network ensembles". In: *Workshop on Bayesian Deep Learning, Neural Information Processing Systems*.

Perolat, J., R. Munos, J.-B. Lespiau, S. Omidshafiei, M. Rowland, P. Ortega, N. Burch, T. Anthony, D. Balduzzi, B. De Vylder, et al. (2021). "From Poincaré recurrence to convergence in imperfect information games: Finding equilibrium via regularization". In: *International Conference on Machine Learning*. PMLR, pp. 8525–8535.

Petrik, M. and D. Subramanian (2014). "RAAM: The benefits of robustness in approximating aggregated MDPs in reinforcement learning". In: *Advances in Neural Information Processing Systems* 27, pp. 1979–1987.

Rakhlin, S. and K. Sridharan (2013). "Optimization, learning, and games with predictable sequences". In: *Advances in Neural Information Processing Systems*, pp. 3066–3074.

Reddi, S. J., S. Kale, and S. Kumar (2018). "On the Convergence of Adam and Beyond". In: *International Conference on Learning Representations*.

Rigter, M., B. Lacerda, and N. Hawes (2021). "Risk-averse Bayes-adaptive reinforcement learning". In: *arXiv preprint arXiv:2102.05762*.

Ross, S. M. (1971). "Goofspiel — The game of pure strategy". In: *Journal of Applied Probability* 8.3, pp. 621–625.

Schmid, M., N. Burch, M. Lanctot, M. Moravčík, R. Kadlec, and M. Bowling (2019). "Variance Reduction in Monte Carlo Counterfactual Regret Minimization (VR-MCCFR) for Extensive Form Games using Baselines". In: *33rd AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 2157–2164.

Schmid, M., M. Moravčík, et al. (2021). “Player of Games”. In: *arXiv preprint arXiv:2112.03178*.

Schulman, J., S. Levine, P. Abbeel, M. Jordan, and P. Moritz (2015). “Trust region policy optimization”. In: *International Conference on Machine Learning*, pp. 1889–1897.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov (2017). “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347*.

Selten, R. (1974). “Reexamination of the Perfectness Concept for Equilibrium Points in Extensive Games”. In: *Economics*.

Shapley, L. (1964). “Some Topics in Two-Person Games”. In: *Advances in Game Theory.* Princeton University Press.

Shapley, L. S. (1953). “Stochastic games”. In: *national academy of sciences* 39.10, pp. 1095–1100.

Singh, S., M. James, and M. Rudary (2012). “Predictive state representations: A new theory for modeling dynamical systems”. In: *arXiv preprint arXiv:1207.4167*.

Singh, S. P., M. L. Littman, N. K. Jong, D. Pardoe, and P. Stone (2003). “Learning predictive state representations”. In: *20th International Conference on Machine Learning (ICML 2003)*, pp. 712–719.

Smallwood, R. D. and E. J. Sondik (1973). “The optimal control of partially observable Markov processes over a finite horizon”. In: *Operations research* 21.5, pp. 1071–1088.

Southey, F., M. Bowling, B. Larson, C. Piccione, N. Burch, D. Billings, and D. C. Rayner (July 2005). “Bayes’ Bluff: Opponent Modelling in Poker”. In: *21st Conference in Uncertainty in Artificial Intelligence (UAI 2005).* Edinburgh, Scotland, pp. 550–558.

Srinivasan, S., M. Lanctot, V. Zambaldi, J. Pérolat, K. Tuyls, R. Munos, and M. Bowling (2018). “Actor-Critic Policy Optimization in Partially Observable Multiagent Environments”. In: *Advances in Neural Information Processing Systems.*

Steinberger, E. (2019). “Single Deep Counterfactual Regret Minimization”. In: *arXiv preprint arXiv:1901.07621.*

Steinberger, E., A. Lerer, and N. Brown (2020). “DREAM: Deep regret minimization with advantage baselines and model-free learning”. In: *arXiv preprint arXiv:2006.10410.*

Sutton, R. and A. Barto (2018). *Reinforcement Learning: An Introduction.* 2nd. MIT Press.

Sutton, R. S., D. McAllester, S. Singh, and Y. Mansour (2000). “Policy Gradient Methods for Reinforcement Learning with Function Approximation”. In: *Advances in Neural Information Processing Systems 12.* MIT Press, pp. 1057–1063.

Syrgkanis, V., A. Agarwal, H. Luo, and R. E. Schapire (2015). “Fast convergence of regularized learning in games”. In: *Advances in Neural Information Processing Systems*, pp. 2989–2997.

Tammelin, O. (2014). “Solving Large Imperfect Information Games Using CFR+”. In: *arXiv preprint arXiv:1407.5042.*

Tammelin, O., N. Burch, M. Johanson, and M. Bowling (2015). “Solving Heads-up Limit Texas Hold’em”. In: *24th International Joint Conference on Artificial Intelligence (IJCAI 2015).*

Tang, Y. C., J. Zhang, and R. Salakhutdinov (2020). “Worst cases policy gradients”. In: *Conference on Robot Learning*, pp. 1078–1093.

Taylor, P. (1979). “Evolutionarily Stable Strategies with Two Types of Players”. In: *Journal of Applied Probability* 16, pp. 76–83.

Taylor, P. and L. Jonker (1978). “Evolutionarily Stable Strategies and Game Dynamics”. In: *Mathematical Biosciences* 40, pp. 145–156.

Thomas, P. S., B. C. da Silva, A. G. Barto, S. Giguere, Y. Brun, and E. Brunskill (2019). “Preventing undesirable behavior of intelligent machines”. In: *Science* 366.6468, pp. 999–1004.

Tibshirani, R. (1996). “A comparison of some error estimates for neural network models”. In: *Neural Computation* 8.1, pp. 152–163.

Vitter, J. S. (1985). “Random sampling With a Reservoir”. In: *ACM Transactions on Mathematical Software (TOMS)* 11.1, pp. 37–57.

von Stengel, B. and F. Forges (2008). “Extensive-form correlated equilibrium: Definition and computational complexity”. In: *Mathematics of Operations Research* 33.4, pp. 1002–1022.

Watkins, C. J. C. H. (1989). “Learning from delayed rewards”. In:

Waugh, K. and J. A. Bagnell (2015). “A Unified View of Large-Scale Zero-Sum Equilibrium Computation”. In: *Workshops at the 29th AAAI Conference on Artificial Intelligence.*

Waugh, K., D. Morrill, J. A. Bagnell, and M. Bowling (2015). “Solving Games with Functional Regret Estimation”. In: *29th AAAI Conference on Artificial Intelligence (AAAI-15).* Vol. 29. 1, pp. 2138–2144.

Willems, F. M., Y. M. Shtarkov, and T. J. Tjalkens (1993). “Context tree weighting: a sequential universal source coding procedure for FSMX sources”. In: *1993 IEEE International Symposium on Information Theory.* Institute of Electrical and Electronics Engineers, p. 59.

Williams, R. J. (May 1992). “Simple statistical gradient-following algorithms for connectionist reinforcement learning”. In: *Machine Learning* 8.3, pp. 229–256. ISSN: 1573-0565. DOI: 10.1007/BF00992696. URL: https://doi.org/10.1007/BF00992696.

Xiao, H., K. Rasul, and R. Vollgraf (2017). “Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms”. In: *arXiv preprint arXiv:1708.07747.* MIT license.

Zahavy, T., A. Barreto, D. J. Mankowitz, S. Hou, B. O’Donoghue, I. Kemaev, and S. Singh (2020). “Discovering a set of policies for the worst case reward”. In: *International Conference on Learning Representations.*

Zeeman, E. (1980). “Population Dynamics from Game Theory”. In: *Lecture Notes in Mathematics, Global Theory of Dynamical Systems* 819.

— (1981). “Dynamics of the evolution of animal conflicts”. In: *Theoretical Biology* 89, pp. 249–270.

Zinkevich, M. (2003). “Online Convex Programming and Generalized Infinitesimal Gradient Ascent”. In: *20th International Conference on Machine Learning (ICML 2003).*

Zinkevich, M., M. Johanson, M. Bowling, and C. Piccione (Sept. 2007a). *Regret Minimization in Games with Incomplete Information.* Tech. rep. TR07-14. University of Alberta.

Zinkevich, M., M. Johanson, M. Bowling, and C. Piccione (Dec. 2007b). “Regret Minimization in Games with Incomplete Information”. In: *Advances in Neural Information Processing Systems (NeurIPS 2007).* Vancouver, British Columbia, pp. 1729–1736.