Multiagent Learning by Iterative Refinement of Game Models

by

Yongzhao Wang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2023

Doctoral Committee:

Professor Michael P. Wellman, Chair
Professor Tilman Börgers
Assistant Professor Benjamin Fish
Professor Mingyan Liu

There is only one true heroism in the world: to see the world as it is, and to love it.

—Roman Rolland

Yongzhao Wang

wangyzh@umich.edu

ORCID iD: 0000-0002-9452-0511

To Mom and Dad.

ACKNOWLEDGMENTS

I wish to express my deep gratitude to my family, friends, and colleagues who have actively contributed to my academic voyage at the University of Michigan. Their unwavering support and influence have left an indelible mark on my work.

First and foremost, I would like to express my deepest appreciation to my advisor, Prof. Michael Wellman, who has been the beacon of light guiding me through the winding and challenging paths of research, illuminating my understanding and ensuring my efforts bore fruitful results. I am greatly indebted to him for his invaluable guidance, constant encouragement, constructive critiques, and for the countless hours he spent shaping my research acumen. The memories of the time we spent jotting down our research ideas on the whiteboard, engaging in passionate debates about the right path to take, and ultimately achieving fruitful outcomes will stay with me eternally. In addition to our research endeavors, he has been a long-standing friend who has offered me invaluable life advice, and his wit and wisdom never cease to amaze me. Working with Mike has been an extraordinary honor and a fortune to me.

I wish to express my sincere gratitude to my committee members Prof. Tilman Borgers, Prof. Mingyan Liu, and Prof. Ben Fish. Each one of you has added immeasurable value to this dissertation, my intellectual journey, and my future career. I would like to express my gratitude to the administrative staff of the CSE department, especially Ashley Andreae, for their support and assistance during my time at Michigan.

I would also like to extend my sincere thanks to all the members of our research

lab. I am immensely grateful to each one of you for your cooperation, support, and for fostering a lively and inspiring work environment. Of these past and present members of the Strategic Reasoning Group, I am particularly grateful to Arunesh Sinha, Thanh H. Nguyen, Mason Wright, Frank Cheng, Xintong Wang, Megan Shearer, Mithun Chakraborty, Zun Li, Max Smith, Katherine Mayo, Christine Konicki, Madelyn Gatchel, and Austin Nguyen. I am also grateful to the undergrad students I mentored, Qiurui Ma and Sky Wang, who worked closely with me and inspired me at the early stage of my research.

I want to extend my deepest thanks to my friends, both within and outside the University of Michigan. My time here was enriched by your presence, your camaraderie, and your support throughout my Master and Ph.D. journey, and this journey would not have been the same without your friendships. I am particularly grateful to Mengting Li, Tianxiang Lu, and Haoming Shen for your long and firm companion in this journey.

Lastly, I must express my very profound gratitude to my parents Xizhong Wang and Qingli Song whose love and guidance are with me in whatever I pursue. To my father and mother, thank you for instilling in me a passion for learning and a determination to persevere, no matter how challenging the situation. Your unconditional love, unwavering faith, and ceaseless encouragement have been the foundation upon which my journey rests. You have not only taught me the value of education but also the importance of staying true to oneself and one's ideals. Your sacrifices and unwavering belief in me, even in times of doubt, have been the driving forces behind my perseverance. I am deeply grateful for your understanding and patience, especially during those times when my academic pursuits took precedence. Your nurturing has made me who I am today. This thesis is not just the culmination of years of academic toil, but also a testament to the values and strength that you instilled in me. This achievement is as much yours as it is mine. We are family eternally and love you.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

The methodology of *Empirical Game-Theoretic Analysis* (EGTA) offers a comprehensive collection of techniques for game reasoning with models based on simulation data. For multiagent systems not amenable to analytic solution, EGTA provides a simulation-based alternative, where a game model with a selected set of strategies is evaluated, addressing the most important strategic considerations. The challenge of efficiently assembling a suitable collection of strategies for a game model in EGTA is called the *strategy exploration* problem. The clearest formulation of strategy exploration in EGTA is within an iterative process, in which a game model is iteratively refined through the alternation of the creation of new strategies and the assessment and analysis of the current game model. In particular, the *Policy Space Response Oracles* (PSRO) algorithm provides a flexible framework for strategy exploration, with new strategies generated each iteration through a best response to a target other-players profile using reinforcement learning (RL). The component responsible for determining the target profile is called a *meta-strategy solver* (MSS), which takes an empirical game model as input and "solves" it to produce the target. I actively investigate three main research aspects of strategy exploration under the PSRO framework (i.e., iterative EGTA with RL): evaluating strategy exploration, controlling strategy exploration, and extension of strategy exploration to mean field games (MFGs).

First, I investigate some of the methodological considerations in evaluating intermediate game models generated through strategy exploration, proposing and justifying new evaluation methods based on examples and experimental observations. In

particular, I emphasize the fact that empirical games create a space of strategies and evaluation should reflect how well it covers the strategically relevant space. Based on this fact, I propose a new evaluation scheme that measures the strategic coverage of an empirical game. I show that the evaluation scheme reveals the authentic learning performance of different strategy exploration methods compared to previous evaluation methods.

Second, I investigate how to control strategy exploration to build a game model that involves desired solutions (e.g., a Nash equilibrium) of the full game with minimum computational costs (i.e., with fewest strategies required). Specifically, I investigate controlling strategy exploration by setting MSSs. I introduce a novel MSS for PSRO, called regularized replicator dynamics (RRD), which prevents overfitting by terminating replicator dynamics before it reaches an exact Nash equilibrium (NE). I demonstrate the effectiveness of RRD on identifying strategically important strategies and accelerating strategy exploration in games with large strategy spaces. Furthermore, I provide a novel explanation for the effectiveness of regularization in RRD for strategy exploration through experiments.

I investigate an alternative means for the controlling: setting the *response objective* (RO) employed in deriving a strategy for a given target profile. My motivation is that different ROs may steer strategy exploration toward solutions with various desired properties. I perform a study in the domain of sequential bargaining games, comparing the standard RO based on own payoff with others based on social welfare. I find that an RO encoded with Nash product can lead to identifying equilibrium outcomes with significantly higher social welfare than the standard objective. For other proposed ROs, experiments demonstrate that they can differentially affect the makeup and value of solutions for different players. Overall, I find that the choice of MSSs and the response objectives can affect the quality of solutions jointly.

Third, I extend the iterative EGTA framework to MFGs. I first prove the existence

of NE in the empirical MFG, which then serves as the MSS in the framework. Due to the non-linearity of the utility function in the mean field, to represent a game model, I introduce a game model learning approach, which is essentially a form of regression of the utility function based on utility data collected from previous EGTA iterations. A learned utility function can generalize across mean fields and thus completing the definition of a game model. Moreover, querying a learned utility function can save a significant amount of simulations compared to running simulations for all utility queries. I combine the iterative EGTA framework with game model learning and provide an effective and sample efficient EGTA framework for MFGs.

CHAPTER I

Introduction

In the past decades, the study of multiagent systems has been a core research area in Artificial Intelligence (AI). A multiagent system, or a *game*, involves multiple decision-making agents which interact in a shared environment and one agent's optimal decision should take other agents' behavior into consideration. To understand the strategic behavior among these agents, game theory provides a mathematical tool and defines behavioral stability in the form of equilibria for agents. Empowered by modern AI with various learning techniques, game-theoretic study in recent years has been extended from simple settings in classic game theory to much complex real-world scenarios, promoting the study of learning in games, known as *multiagent learning* (Shoham, Powers, and Grenager 2007).

In multiagent learning, a significant topic is game reasoning, often achieved through various learning techniques and the guidance of game theory. This methodology for game reasoning is largely captured by *empirical game-theoretic analysis* (EGTA) (Wellman 2006). EGTA describes a broad set of methods that are building and reasoning about game models based on simulation data. Game models are approximations of the underlying full game, typically induced from simulations run over combinations of a particular set of strategies, thus feasible and less computationally expensive to analyze compared to directly analyzing the full game. By interleav-

ing game model construction, assessment, analysis, and refinement, EGTA achieves strategic reasoning in underlying full games.

Since the accuracy of a game model directly impacts game-theoretic analysis, how to construct a game model is crucial to EGTA. This particularly means selecting a suitable collection of strategies to analyze for a game model. In prior work of EGTA, a game model is mainly constructed through two ways. One way is to build and extend a game model based on heuristics and handcrafted strategies. Despite the simplicity, strategy design requires significant deliberation, which is arduous especially when the game of interest becomes large. Another way of model construction is *strategy exploration*, which iteratively refines the game model by extending the considered set of strategies based on the analysis of current game models. One famous representative of this iterative strategy generation framework is the *double oracle* (DO) method McMahan, Gordon, and Blum (2003), which sequentially extends strategy sets by best-response to a target profile. For DO, the target profile is the Nash equilibrium (NE) of the current game model.

Following the iterative strategy generation in DO, Lanctot, Zambaldi, et al. (2017) proposed a more general framework, called *Policy Space Response Oracles* (PSRO), for analyzing complex game scenarios. PSRO first specifies deep reinforcement learning (RL) as a best response strategy generator for scenarios with large state and action spaces. PSRO then generalizes the best response target in DO by introducing the concept *meta-strategy solver* (MSS), which decides the profile to extract from the current model as target for the next best-response calculation. Since the selection of an MSS determines the generation of new strategies, it determines strategy exploration. Besides using NE as an MSS, the research community has put significant efforts into designing advanced MSSs to find an NE with minimal computational costs (Balduzzi, Garnelo, et al. 2019; Jordan, Schvartzman, and Wellman 2010; Lanctot, Zambaldi, et al. 2017; Muller, Omidshafiei, et al. 2020; Schvartzman and Wellman

2009a).

Despite the success of these MSSs, scaling up to games with a large number of players remains challenging.[1] Inspired by the large economic literature on games with a continuum of players (Aumann 1964; Schmeidler 1973), the notion of mean field games (MFGs) has been introduced by Lasry and Lions (2007) and Huang, Malhamé, Caines, et al. (2006) to model strategic interactions through the distribution of players' states. By considering the limit case of a continuous distribution of identical agents (i.e., anonymous and with symmetric interests), the MFG framework allows the learning problem to be reduced to the characterization of the optimal behavior of a single representative agent in its interactions with the full population, thus making strategic reasoning feasible. Moreover, an NE in MFGs can be proved to be an approximate NE in the corresponding finite games (Bensoussan, Frehse, Yam, et al. 2013; Carmona and Delarue 2018). Therefore, MFGs provide an alternative way of modeling and reasoning about games with a large number of players.

The goal of this thesis is to systematically investigate strategy exploration under the PSRO framework in both finite games and MFGs with large strategy spaces. Leveraging various learning techniques and game theory, my investigation includes novel algorithms for effective and efficient strategy exploration, evaluation and interpretation of strategy exploration, and extension of strategy exploration to solving MFGs.

1.1 Game Theory Foundations

A normal-form game $\mathcal{G} = (N, (S_i), (u_i))$ consists of a finite set of players N indexed by i; a non-empty set of strategies S_i for player $i \in N$; and a utility function u_i :

[1] All MSSs can be applied to games with high-dimensional payoff matrix conceptually. Various techniques, such as player reduction (Ficici, Parkes, and Pfeffer 2008; Wellman et al. 2005; Wiedenbeck and Wellman 2012) and game model learning (Li and Wellman 2021, 2020; Vorobeychik, Wellman, and Singh 2007; Wiedenbeck, Yang, and Wellman 2018), help to reduce the complexity of evaluating the large number of profiles.

$\prod_{j \in N} S_j \to \mathbb{R}$ for player $i \in N$, where $\prod$ is the Cartesian product.

A mixed strategy σ_i is a probability distribution over strategies in S_i, with $\sigma_i(s_i)$ denoting the probability player i plays strategy s_i. I adopt conventional notation for the other-agent profile: $\sigma_{-i} = (\sigma_j)_{j \neq i}$. Let $\Delta(\cdot)$ represent the probability simplex over a set. The mixed strategy space for player i is given by $\Delta(S_i)$. Similarly, $\Delta(S) = \prod_{i \in N} \Delta(S_i)$ is the mixed profile space.

The set of *best responses* of player i to profile σ include any strategy yielding maximum payoff for i, holding the other players' strategies constant:

$$br_i(\sigma_{-i}) = \underset{\sigma_i' \in \Delta(S_i)}{\operatorname{argmax}} u_i(\sigma_i', \sigma_{-i}).$$

Let $br(\sigma) = \prod_{i \in N} br_i(\sigma_{-i})$ be the overall best-response correspondence for a profile σ. A Nash equilibrium is a profile σ^* such that $\sigma^* \in br(\sigma^*)$, that is,

$$\sigma_i^* \in br_i(\sigma_{-i}^*), \forall i \in N$$

Player i's *regret* in profile σ in game $\mathcal{G}$ is given by

$$\rho_i^{\mathcal{G}}(\sigma) = \max_{s_i' \in S_i} u_i(s_i', \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}).$$

Regret captures the maximum player i can gain in expectation by unilaterally deviating from its mixed strategy in σ to an alternative strategy in S_i. I use the superscript $\mathcal{G}$ in $\rho^{\mathcal{G}}$ to make clear which game we are measuring regret with respect to.

An NE strategy profile has zero regret for each player. A profile is said to be an ϵ-Nash equilibrium (ϵ-NE) if no player can gain more than ϵ by unilateral deviation. The regret of a strategy profile σ is defined as the sum over player regrets:

$$\rho^{\mathcal{G}}(\sigma) = \sum_{i \in N} \rho_i^{\mathcal{G}}(\sigma). \tag{1.1}$$

Some treatments employ max instead of sum for this; when necessary to disambiguate I refer to $\rho^{\mathcal{G}}(\sigma)$ defined by Equation (1.1) as *sum-regret*.

Replicator dynamics (RD) describes an evolving trajectory of mixed profiles, inspired by natural selection (Smith and Price 1973; Taylor and Jonker 1978). RD is commonly employed as a heuristic equilibrium search algorithm. We consider a discrete form of RD, where player i's probability of playing each strategy is updated in proportion to its payoff for deviating to that strategy from the current mixture. Mathematically, the replicator equation for player i's strategy s_i in a current profile σ is given by

$$\frac{d\sigma_i(s_i)}{dt} = \sigma_i(s_i)[u_i(s_i, \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i})].$$

At each iteration of RD, player i's mixed strategy σ_i is updated by $\sigma_i \leftarrow P(\sigma_i + \alpha\frac{d\sigma_i}{dt})$, where α is a step size for RD and P is a projection operator to the strategy simplex, namely $P(\sigma_i) = \text{argmin}_{\sigma_i' \in \Delta} ||\sigma_i' - \sigma_i||_2$.

Quantal response equilibrium (QRE) (McKelvey and Palfrey 1995, 1998) is an equilibrium notion that captures bounded rationality. One common specification for QRE is logit equilibrium in which players' strategies take the form

$$\sigma_i(s_i) = \frac{\exp(\tau u_i(s_i, \sigma_{-i}))}{\sum_{s_i' \in S_i} \exp(\tau u_i(s_i', \sigma_{-i}))},$$

where $0 < \tau < \infty$ is a parameter governing the rationality of players. The response strategy becomes a best response as τ approaches infinity while it becomes uniform when τ converges to 0.

A Nash bargaining solution (Nash Jr. 1950b) is a profile that satisfies the following axioms:

1. Invariant to affine transformations: if the utility function is re-scaled on a linear basis, the solution to the game will not change;

2. Symmetry: if the players' utility functions are the same, they should receive

the same outcome;

3. Pareto efficiency: no strategy is available that makes one player better off without making another worse off;

4. Independence of irrelevant alternatives: by removing strategies that none of the players would have chosen, the outcome will not change.

Nash proved that there is a unique solution σ satisfying these axioms maximizes the expression $\prod_{i \in N} u_i(\sigma)$. The product of the utilities is generally referred to as the *Nash product*.

1.2 Empirical Games

An *empirical game* $\hat{\mathcal{G}}$ is a model of true game $\mathcal{G}$ where payoffs are estimated through a *simulator*, a description of the true game. Typically, a simulator will be realized as a program that implements the interaction among the participating agents and the environment, and generates noisy observations of utility from play. Although the full strategy space allowed by a game simulator can be large, empirical game models usually restrict the strategy space to a small number of strategies. To represent such restriction, I use the notation $S \downarrow X$ to denote that players can only choose from restricted strategy sets $X_i \subseteq S_i$. Thus, $\hat{\mathcal{G}}_{S \downarrow X} = (N, (X_i), (\hat{u}_i))$ denotes an empirical game model where players are restricted to X and $\hat{u}$ is an estimated projection of u onto the strategy space X.[2]

The profile in the restricted game closest to being a solution of the full game is the *minimum regret constrained profile* (MRCP) (Jordan, Schvartzman, and Wellman

[2]Because payoffs are estimated through simulation, $\hat{u}$ is also subject to sampling error. This presents additional statistical issues (Tuyls et al. 2020; Vorobeychik 2010; Wiedenbeck, Cassell, and Wellman 2014). In this thesis, we ignore those and focus on the issues that arise from strategy set restriction.

2010). Formally,

$$MRCP(\mathcal{G}_{S\downarrow X}) = \underset{\sigma\in\Delta(X)}{\operatorname{argmin}} \sum_{i\in N} \rho_i^{\mathcal{G}}(\sigma). \tag{1.2}$$

1.3 Mean Field Games

A *multi-population mean field game* (MP-MFG) in normal form is given by $\mathcal{G} = ([N_p], (S_i), (u_i))$. $[N_p] = \{1, \ldots, N_p\}$ is a set of N_p populations indexed by i. Each population corresponds to a conceptually infinite and interchangeable set of agents playing a particular role in the game. The mean field is defined over an underlying state space Z of the game environment. An agent's strategy maps states to actions, with action space A the same for each population. Formally, i's strategy at time t, $s_{i,t}$, maps from state space Z to the space of action distributions $\Delta(A)$. The overall strategy $s_i = (s_{i,t})_{t\in[0,T-1]}$ is a sequence of strategies from time 0 through horizon T. S_i denotes the set of strategies for population $i \in [N_p]$. Utility functions $u_i : S_i \times [\Delta(Z)^T]^{N_p} \to \mathbb{R}$ define the payoff to a representative player of population i playing its strategy against the distributions of all populations. All populations are assumed to share the same state space Z.

A mixed strategy σ_i is a probability distribution over strategies in S_i, with $\sigma_i(s_i)$ denoting the probability the representative player of population i plays strategy s_i. Let σ be the profile of strategies across populations (i.e., $\sigma = (\sigma_1, \ldots, \sigma_{N_p})$). The expected utility of playing a mixed strategy σ_i for the representative player of population i given distributions $\mu = \{\mu_1, \ldots, \mu_{N_p}\}$, where $\mu_i = (\mu_{i,t})_{t\in[0,T]} \in \Delta(Z)^{T+1}$, is

$$u_i(\sigma_i, \mu) = \sum_{s_i\in S_i} \sigma(s_i) u_i(s_i, \mu). \tag{1.3}$$

The set of best responses of the representative player of population i to population distributions μ involves any strategy yielding maximum payoff for the player, holding

the distributions μ constant:

$$br_i(\mu) = \underset{\sigma_i \in \Delta(S_i)}{\operatorname{argmax}}\, u_i(\sigma_i, \mu).$$

Let $br(\mu) = \prod_{i \in N_p} br_i(\mu)$ be the overall best-response correspondence for populations' distributions μ. A Nash equilibrium for an MFG is a profile σ^* such that $\sigma^* \in br(\mu^*)$, where μ^* is induced by σ^*.

A distribution μ_i is said to be induced by s_i, denoted as $\mu_i^{s_i}$, following the *Forward Equation*, that is, given initial distribution $\mu_{i,0}$, for $t \in [0, T-1]$ and all $z_i' \in Z$,

$$\mu_{i,t+1}^{s_i}(z_i') = \sum_{z_i, a_i \in Z, A} \mu_{i,t}^{s_i}(z_i) s_{i,t}(a_i \mid z_i) p(z_i' \mid z_i, a_i), \tag{1.4}$$

where $p : Z \times A \to \Delta(Z)$ is the transition function.

The representative player of population i's *regret* in profile σ given distributions μ in game $\mathcal{G}$ is given by

$$\rho_i^{\mathcal{G}}(\sigma_i, \mu) = \max_{s_i \in S_i} u_i(s_i, \mu) - u_i(\sigma_i, \mu).$$

Regret captures the maximum the representative player of population i can gain in expectation by unilaterally deviating from its mixed strategy in σ to an alternative strategy in S_i, given distributions μ. An NE strategy profile has zero regret for each representative player. A profile is said to be an ϵ-Nash equilibrium if no representative player can gain more than ϵ by unilateral deviation. Similar to the regret definition in finite games, the regret of a strategy profile σ in MFGs is defined as the sum over representative players' regrets:

$$\rho^{\mathcal{G}}(\sigma, \mu) = \sum_{i \in [N_p]} \rho_i^{\mathcal{G}}(\sigma_i, \mu).$$

1.4 Thesis Overview

I focus on the investigation of four research questions on strategy exploration.

Question 1. *How should intermediate game models be evaluated in strategy exploration?*

Based on my AAMAS-22 paper "Evaluating Strategy Exploration in Empirical Game-Theoretic Analysis" (Wang, Ma, and Wellman 2022), Chapter II addresses the evaluation of game models generated by strategy exploration. I first introduce the strategy exploration problem and then explain what makes the evaluation of strategy exploration distinct from evaluating other game learning algorithms. I highlight that in strategy exploration the generated empirical games create a space of strategies and evaluation should reflect how well the space of strategies covers the strategically relevant space of the full game. To capture this fact in evaluation, I introduce a systematic evaluation procedure for strategy exploration and demonstrate it in various game settings.

Question 2. *How should a game model be effectively constructed by setting MSSs in PSRO?*

Based on my paper (under review) "Regularization for Strategy Exploration in Empirical Game-Theoretic Analysis" (Wang and Wellman 2023a), in Chapter III, I describe how to effectively assemble a set of strategies for a game model by setting MSSs. I introduce a novel MSS, called *regularized replicator dynamics* (RRD), which incorporates regularization in game model analysis. I demonstrate the effectiveness of RRD on identifying strategically important strategies in few-player games with large strategy spaces and provide an explanation on the improved performance of RRD based on experimental observations. Besides RRD, I also show the effectiveness of alternative MSSs for strategy exploration including MRCP and QRE.

Question 3. *What is the impact of response objectives in PSRO for strategy exploration?*

Chapter IV is based on my manuscript (under review) "Generalized Responses for Strategy Exploration in Empirical Game-Theoretic Analysis". In Chapter IV, I investigate an alternative means to control strategy exploration: setting the RO employed in deriving a strategy for a given target profile. A natural hypothesis is that the choice of ROs, which are objectives (approximately) solved through RL at each iteration of PSRO, can substantially impact strategy exploration and equilibrium outcomes. To demonstrate this, I introduce PSRO with generalized ROs. Generalized ROs are not limited to optimizing utility against other players' strategies, as in standard PSRO framework, but can incorporate specified preferences. I propose four RO instances for PSRO with various strategy exploration preferences and evaluate them in sequential bargaining games and attack-graph games, comparing solutions found according to various criteria.

Question 4. *Can the EGTA framework for finite games be extended to MFGs?*

Due to the non-linearity of the utility function in the mean field, the utility function cannot be represented explicitly as in finite games. Therefore, a game model for MFGs cannot be defined in terms of usual components. To handle this issue, I introduce a game model learning approach for MFGs in Chapter V, based on my paper "Game Model Learning for Mean Field Games" (Wang and Wellman 2023c). In particular, my approach learns the utility function of MFGs based on neural networks. I develop a *coarse coding* representation for the high-dimensional inputs (i.e., time-dependent strategies and distributions) of MFG utility functions. I also develop a data sampling scheme that effectively samples data in large strategy spaces. I show that the learned game model exhibits the ability of generalization across mean fields and can successfully support game-theoretic analysis.

With well-defined game models for MFGs, I present the PSRO framework for game model construction as well as a proof for the existence of NE in MFG models in Chapter VI. This chapter is based on my AAMAS-23 paper "Empirical Game-Theoretic Analysis for Mean Field Games" (Wang and Wellman 2023b). My experimental results show that the iterative EGTA framework can successfully construct a game model incorporating the NE of MFGs in various configurations. Moreover, I show that compared to running simulation for all utility queries, a learned game model can dramatically reduce the computational effort required to analyze an intermediate game model due to the relative low cost of querying a learned model.

CHAPTER II

Strategy Exploration and Evaluation

2.1 Introduction to Strategy Exploration

In EGTA, game-theoretic analysis is performed by reasoning about game models. Game models are induced from simulations run over combinations of a particular set of strategies. To construct a feasible and effective game model for game analysis, the selection of the strategies is pivotally important. In particular, a game model is expected to contain a much smaller number of strategies than the full game for representation tractability yet still maintain the key strategic information of the full game (Balduzzi, Tuyls, et al. 2018). This challenge of game model construction is described as the *strategy exploration* problem (Jordan, Schvartzman, and Wellman 2010) in EGTA. Strategy exploration is achieved by iteratively extending the considered strategy set, based on the analysis of the current empirical game model. The goal of strategy exploration is to assemble an effective strategy portfolio for a game model with minimum computational cost (i.e., with fewest strategies required).

2.2 An Iterative Framework for Studying Strategy Exploration

I investigate strategy exploration based on an iterative EGTA framework where strategies are sequentially added to the current game model through best responses. To illustrate this framework, I first introduce DO (McMahan, Gordon, and Blum 2003), an instance of iterative game extension, and then PSRO (Lanctot, Zambaldi, et al. 2017), an iterative framework based on EGTA with RL. DO with empirical game estimation can be viewed as an instance of both PSRO and iterative EGTA.

2.2.1 Double Oracle

DO is an iterative algorithm for solving games with a finite number of strategies. The procedure of DO is shown in Algorithm 1. At the beginning, each player i is initialized with a set of strategies X_i. The set of players, the set of strategies and the utilities of the corresponding profiles constitute a restricted game $\mathcal{G}_{S\downarrow X}$. At each iteration, the NE σ of the current restricted game is first computed, which serves as a best response target. Then each player computes a best response strategy $s' \in br(\sigma_{-i})$, and adds it to the player's strategy set X_i if it has not been added before.

When DO terminates, it means no player can deviate unilaterally to gain extra payoff. In other words, the equilibrium in the current restricted game is an NE of the full game. In finite games, DO is guaranteed to converge to NE, though DO will add all strategies of the full game in the worst case, which trivially includes all strategies in the support of NE.

2.2.2 Policy Space Response Oracles

PSRO extends DO by first introducing deep reinforcement learning as a best response strategy generator. Moreover, PSRO generalizes the best response target

Algorithm 1 Double Oracle (McMahan, Gordon, and Blum 2003)

Input: initial strategy sets X_i for each player i
Compute an expected utility for $\sigma \in X$
Initialize a strategy profile $\sigma \leftarrow NE(\mathcal{G}_{S\downarrow X})$
while DO iteration $\tau = 1, 2, \ldots$ **do**
  deviation $\leftarrow$ False
  for player $i \in N$ **do**
    Compute a best response $s_i' \leftarrow br(\sigma_{-i})$
    if $s_i' \notin X_i$ **then**
      deviation $\leftarrow$ True
      Add the new best response to player i's strategy set $X_i \leftarrow X_i \bigcup s_i'$
    end if
  end for
  if $\neg$ deviation **then**
    Return a restricted game $\mathcal{G}_{S\downarrow X}$ and an equilibrium σ of $\mathcal{G}_{S\downarrow X}$
  end if
  Fill in missing utilities for profiles in $\mathcal{G}_{S\downarrow X}$
  Compute a Nash equilibrium $\sigma \leftarrow NE(\mathcal{G}_{S\downarrow X})$
end while

by introducing the concept *meta-strategy solver*, which decides the profile to extract from the current model as target for the next best-response computation. PSRO is presented below as Algorithm 2. Specifically, each player is initialized with a set of strategies X_i and the utilities for profiles in the profile space X are simulated, resulting in an initial empirical game $\hat{\mathcal{G}}_{S\downarrow X}$. At each iteration of PSRO, an MSS extracts a profile from the empirical game $\hat{\mathcal{G}}_{S\downarrow X}$ as the best response target profile σ. Then each player (i.e., the learning player) computes a best response s_i' against other players' strategies σ_{-i} in the profile σ. The best response is computed through RL with other other players' strategies σ_{-i} fixed. The best response s_i' is then added to player i's strategy set X_i. This procedure repeats for a fixed number of iterations.

A *response objective* for player i in PSRO is a function of strategy profiles, denoted as $RO_i(\sigma)$. For example, in standard PSRO described above, the RO can be written as $RO_i(\sigma) = u_i(s_i', \sigma_{-i})$ and maximizing it over s_i' gives player i a best response against σ_{-i}. I investigate two ways to control strategy exploration, by setting MSSs (Chapter III) and setting ROs (Chapter IV), respectively. I refer to the choice of a

Algorithm 2 PSRO, parametrized by solver MSS (Lanctot, Zambaldi, et al. 2017)

Require: initial strategy sets X
Estimate $\hat{\mathcal{G}}_{S\downarrow X}$ by simulating $\sigma \in X$
Initialize target $\sigma \leftarrow MSS(\hat{\mathcal{G}}_{S\downarrow X})$
for PSRO iteration $\tau = 1, 2, \ldots, \mathcal{T}$ **do**
  for player $i \in N$ **do**
    for many RL training episodes **do**
      Sample a profile $s_{-i} \in \sigma_{-i}$
      Train best response oracle s_i' against s_{-i}
    end for
    $X_i \leftarrow X_i \cup \{s_i'\}$
  end for
  Update $\hat{\mathcal{G}}_{S\downarrow X}$ by simulating missing profiles over X
  Compute best-response target $\sigma \leftarrow MSS(\hat{\mathcal{G}}_{S\downarrow X})$
end for
Return $\hat{\mathcal{G}}_{S\downarrow X}$

pair of an MSS and an RO as an MSS-RO combination.

2.3 Evaluating Strategy Exploration

2.3.1 A Key Fact of Evaluating Strategy Exploration

A key fact I highlight for evaluating the performance of strategy exploration methods is that each method (i.e., MSS) essentially generates a distinct sequence of strategies, and thus the empirical game model at any point reflects a distinct strategy space. The relevant comparisons are across different strategy spaces, which may not be faithfully represented by a simple summary such as an interim solution. This key fact has tended to be neglected in prior studies proposing and evaluating new ideas on strategy exploration, and as I demonstrate, this can lead to misleading conclusions on the performance of different approaches.

2.3.2 Literature Review

In the strategy exploration literature, a profile's fitness as solution candidate is typically measured by its regret in the true game. Jordan, Schvartzman, and Wellman (2010) defined MRCP (Eq. 1.2), the regret of which provides a measure of accuracy of an empirical game. Balduzzi, Garnelo, et al. (2019) introduced the term *Gamescape* to refer to the scope of joint strategies covered by the exploration process to a given point. They employed this concept to characterize the effective diversity of an empirical game state, and proposed a new MSS called *rectified Nash* designed to increase diversity of the Gamescape. Finally, I take note of a couple of recent works that characterize Gamescapes in terms of topological features. Omidshafiei, Tuyls, et al. (2020) proposed using spectral analysis of the α-rank best response graph, and Czarnecki et al. (2020) visualized the strategic topography of real-world games as a spinning top wherein layers are transitive and strategies within a layer are cyclic. (Perez-Nieves et al. 2021) introduced a diversity measure defined through a determinantal point process to measure the diversity of empirical games, viewing the payoff vector of each strategy as the feature of that strategy.

2.3.3 Evaluating an Empirical Game Model

From the perspective of strategy exploration, the key feature of an empirical game model is what strategies it incorporates.[1] In EGTA, the restricted strategy set X is typically a small slice of the set of all strategies S, so the question is how well X covers the strategically relevant space. There may be several ways to interpret "strategically relevant", but one natural criterion is whether the empirical game $\hat{\mathcal{G}}_{S\downarrow X}$ covers solutions or approximate solutions to the true and full game $\mathcal{G}$.

The profile in the empirical game closest to being a solution of the full game is

[1] The accuracy of the estimated payoff functions over these strategies is also relevant, but mainly orthogonal to exploration and outside the scope considered here.

the MRCP. As a reminder,

$$MRCP(\mathcal{G}_{S\downarrow X}) = \operatorname*{argmin}_{\sigma \in \Delta(X)} \sum_{i \in N} \rho_i^{\mathcal{G}}(\sigma)$$

The regret of MRCP thus provides a natural measure of how well X covers the strategically relevant space. In the work by Jordan, Schvartzman, and Wellman (2010), MRCP was studied in games with fixed strategy sets rather than a setting where strategy sets are iteratively built. I extend the study of its properties to the strategy exploration setting. I first notice that the regret of MRCP necessarily decreases as the empirical game model is being extended, since adding strategies can only increase the scope of minimization. Moreover, MRCP tracks convergence in that the regret of MRCP reaches zero exactly when an NE of $\mathcal{G}$ is contained in the empirical game, that is, X covers the support of the NE. I claim both properties of MRCP are important and desirable for evaluation purposes.

Unfortunately, direct use of MRCP as a means for evaluating strategy exploration can be computationally challenging. Calculating regret of a profile, the quantity we are minimizing, generally requires a best-response oracle for the full game, which itself can be quite computationally expensive (which is why we often find RL the best available method). And even given an effective way to calculate regret, the search for MRCP is a non-convex optimization problem over the profile space of the empirical game.

2.3.4 Searching for MRCP

In this section, we assume the full game we target to solve is a matrix game. This assumption makes it easier for computing regret, a key step in MRCP computation. For matrix games, MRCP can be approximated by solving the optimization problem in the definition of MRCP (Eq. 1.2) with black-box optimization tools (e.g., the

amoeba method (Nelder and Mead 1965)). When applying the amoeba method to the optimization problem, we have to reconcile the fact that the optimization problem is constrained while the amoeba method is an unconstrained optimization technique. To handle this issue, Jordan, Schvartzman, and Wellman (2010) proposed a binary search (BS) to select the maximum feasible reflection and expansion (i.e., two steps in the amoeba method) scaling parameters (step sizes), respectively. This approach handles infeasibility by choosing the most recent feasible reflected and expanded points (each point is a strategy profile) given by scaling parameters. However, since the optimal solution points are high-dimensional vectors, they may not be reached exactly by fixed scaling parameters. I apply an alternative means to handle infeasibilty by projecting an infeasible point onto the unit strategy simplex.

2.3.4.1 Projected Amoeba Method for MRCP Computation in Matrix Games

I improve the accuracy of MRCP computation based on Jordan, Schvartzman, and Wellman (2010) and show a projected amoeba method in Algorithm 3. To compute MRCP, the primary goal is to find a profile that minimizes the cumulative regret function $f(\sigma) = \sum_{i \in N} \rho_i^{\mathcal{G}}(\sigma)$ shown in Equation (1.2). Denote a projection operator as $P(\sigma_i) = \text{argmin}_{\sigma_i' \in \Delta(S_i)} \|\sigma_i' - \sigma_i\|$ for player $i \in N$. Denote by $P(\sigma)$ the projection operator for each $\sigma_i \in \sigma$. For amoeba, it follows the default values of $\alpha = 1$, $\gamma = 2$, $\rho = 1/2$, $\sigma = 1/2$.

I compare the performance of the amoeba method with two approaches—BS and projection—in two-player Kuhn poker. Empirical games with different sizes are first sampled from the full game and then MRCP is approximated with different approaches. Table 2.1 shows the regret of MRCP given by different approaches. To illustrate the performance of different approaches, I also provide the regret of NE of the empirical game as a benchmark. I observed that for each size of a empirical game,

Algorithm 3 Projected Amoeba Method

Input: A full game model with regret function f and an empirical game model.
Parameter: Amoeba method parameters α, γ , ρ, σ corresponding to the reflection, expansion, contraction and shrink coefficients.
Output: MRCP σ.

while $t = 1, \ldots, \mathcal{T}$ **do**
  Select current test profiles $\sigma^1, \ldots, \sigma^{n+1}$.
  Order according to the regrets at these profiles: $f(\sigma^1) \leq \ldots \leq f(\sigma^{n+1})$.
  Calculate σ^o, the centroid of profiles except σ^{n+1}
  Reflection: Compute reflected point $\sigma^r \leftarrow \sigma^o + \alpha(\sigma^o - \sigma^{n+1})$
  Project σ^r to probability simplex $\sigma^r \leftarrow P(\sigma^r)$
  if $f(\sigma^1) \leq f(\sigma^r) < f(\sigma^n)$ **then**
    $\sigma^{n+1} \leftarrow \sigma^r$
  else
    Continue.
  end if
  Expansion:
  if $f(\sigma^r) < f(\sigma^1)$ **then**
    $\sigma^e \leftarrow \sigma^o + \alpha(\sigma^r - \sigma^o)$
    $\sigma^e \leftarrow P(\sigma^e)$
    if $f(\sigma^e) < f(\sigma^r)$ **then**
      $\sigma^{n+1} \leftarrow \sigma^e$ and Continue.
    else
      $\sigma^{n+1} \leftarrow \sigma^r$ and Continue.
    end if
  end if
  Contraction: $\sigma^c \leftarrow \sigma^o + \alpha(\sigma^{n+1} - \sigma^o)$
  $\sigma^c \leftarrow P(\sigma^c)$
  if $f(\sigma^e) < f(\sigma^r)$ **then**
    $\sigma^{n+1} \leftarrow \sigma^c$ and Continue.
  end if
  Shrink: $\sigma^c \leftarrow \sigma^o + \alpha(\sigma^{n+1} - \sigma^o)$
  $\sigma^c \leftarrow P(\sigma^c)$ and Continue.
end while
return σ

approximating MRCP with projection results in a profile with significantly lower regret, merely with a different infeasibility handling approach. I also noticed that when the size is small, the performance of two approaches is close. As the size increases, the BS approach does not lead to a good MRCP approximation and we even see that NE of the empirical game could have lower regret than the MRCP approximation.

Moreover, to understand the stability of results given by different approaches, I compute the regret of approximated MRCP of a fixed empirical game for multiple times and measure the variance. I found that the variance of regrets given by BS approach from multiple runs is very large while the variance of the projection method is tiny. This improvement in the accuracy and stability of MRCP approximation will benefit the study of evaluating strategy exploration, where MRCP serves as an important evaluation metric.

	Size = 5					**Size = 7**				
Index	1	2	3	4	5	1	2	3	4	5
$\rho(\bar{\sigma})$ **w. BS**	**0.39**	0.36	0.35	0.44	0.19	0.51	0.36	0.44	0.39	0.21
$\rho(\bar{\sigma})$ **w. Proj**	**0.39**	**0.30**	**0.30**	**0.40**	**0.19**	**0.31**	**0.30**	**0.32**	**0.35**	**0.14**
$\rho(\sigma^*)$	0.50	0.39	0.78	0.73	0.49	0.78	0.50	0.33	0.58	0.39

	Size = 9					**Size = 11**				
Index	1	2	3	4	5	1	2	3	4	5
$\rho(\bar{\sigma})$ **w. BS**	0.26	0.40	0.44	0.45	0.83	0.46	0.49	0.45	0.59	0.60
$\rho(\bar{\sigma})$ **w. Proj**	**0.15**	**0.33**	**0.33**	**0.38**	**0.61**	**0.07**	**0.35**	**0.17**	**0.30**	**0.35**
$\rho(\sigma^*)$	0.21	**0.33**	0.42	0.78	0.71	0.29	0.50	0.26	0.33	0.67

	Size = 13					**Size = 15**				
Index	1	2	3	4	5	1	2	3	4	5
$\rho(\bar{\sigma})$ **w. BS**	0.37	0.57	0.60	0.28	0.27	0.37	0.52	0.30	0.27	0.17
$\rho(\bar{\sigma})$ **w. Proj**	**0.13**	**0.33**	**0.28**	**0.12**	**0.18**	**0.08**	**0.17**	**0.08**	**0.16**	**0.07**
$\rho(\sigma^*)$	0.22	0.53	0.50	0.20	0.28	0.50	0.57	0.20	0.27	0.19

Table 2.1: MRCP quality with different infeasibility handling methods.

2.3.4.2 MRCP Approximation in Large Games

Computing MRCP in large games can be computational arduous since it demands a large number of regret queries, each entailing an expensive best-response computation. I therefore seek an affordable way to approximate MRCP in large games. I start by deriving an upper bound for the regret of a mixed-strategy profile through the

	Size = 5				Size = 10				Size = 15			
Index	1	2	3	4	1	2	3	4	1	2	3	4
$\rho(\bar{\sigma})$	0.67	0.25	0.58	0.20	0.09	0.20	0.15	0.33	0.05	0.08	0.07	0.38
$\rho(\tilde{\sigma})$	0.67	0.27	0.63	0.24	0.09	0.20	0.15	0.34	0.07	0.09	0.09	0.40
$\rho(\sigma^*)$	0.83	0.50	0.72	0.42	0.17	0.44	0.26	0.54	0.26	0.21	0.16	0.46

Table 2.2: MRCP quality with two definitions.

deviation payoff of a finite set of pure-strategy profiles. I then approximate MRCP by minimizing the upper regret bound. This approach allows us to focus on pure-strategy deviations which is a more manageable space compared to the search over mixed-strategy profiles.

I derive the upper regret bound as follows:

$$\begin{aligned}
\rho_i^{\mathcal{G}}(\sigma) &= \max_{s_i' \in S_i} u_i(s_i', \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}) \qquad (2.1)\\
&= \max_{s_i' \in S_i} \sum_{s_{-i} \in S_{-i}} \sigma(s_{-i}) u_i(s_i', s_{-i}) - \sum_{s_i \in S_i} \sum_{s_{-i} \in S_{-i}} \sigma(s_i)\sigma(s_{-i}) u_i(s_i, s_{-i})\\
&\leq \sum_{s_{-i} \in S_{-i}} \sigma(s_{-i}) \max_{s_i' \in S_i} u_i(s_i', s_{-i}) - \sum_{s_i \in S_i} \sum_{s_{-i} \in S_{-i}} \sigma(s_i)\sigma(s_{-i}) u_i(s_i, s_{-i}).
\end{aligned}$$

Note that the utility structure of a game may affect the quality of our regret bound. For example, in two-player zero-sum games, since the sum of players' utilities is zero for every profile, the term $\sum_{s_i \in S_i} \sum_{s_{-i} \in S_{-i}} \sigma(s_i)\sigma(s_{-i}) u_i(s_i, s_{-i})$ (i.e., expected utility of playing σ for player i) will be canceled when we sum the regret bound over players. As a result, minimizing the summation of upper bounds always produces a pure strategy profile, possibly yielding a large estimation error.

I handle this issue by replacing sum-regret (1.1) with the maximal regret over players. My approximate MRCP $\tilde{\sigma}$ employs the max-regret variant:

$$\tilde{\sigma}^X = \operatorname*{argmin}_{\sigma \in \Delta(X)} \max_{i \in N} \rho_i^{\mathcal{G}}(\sigma) \qquad (2.2)$$

This modification prevents the expected utility term from being canceled, yielding a more promising approach for minimizing the regret bound.

To verify that using max-regret does not unduly distort results, we can evaluate the sum-regret of the profile produced by minimizing either version. Let $\bar{\sigma}^X$ be the profile minimizing sum-regret with respect to strategy set X, and $\tilde{\sigma}^X$ the corresponding MRCP using max-regret (2.2). Note that for any X, $\rho(\bar{\sigma}^X) \leq \rho(\tilde{\sigma}^X)$. Table 2.2 compares the two MRCP definitions in five instances of Kuhn poker, for each of three sizes of two-player Kuhn poker. As we can see, the MRCP calculated using max-regret is quite close to the actual sum-regret MRCP in minimizing sum-regret.

I measure the quality of the approximation using the upper regret bound (2.1) with the max-regret version of MRCP (2.2). My experiment employs a synthetic two-player zero-sum matrix game with 200 strategies and utilities uniformly sampled from $[-R, R]$, $R = 1000$. Table 2.3 compares the regrets of exact MRCP $\bar{\sigma}$, approximated MRCP $\tilde{\sigma}$ (I overload the notation for convenience), and NE σ^* (i.e., a benchmark). I observed that in some sampled empirical games (e.g., game 2 with size 3, game 2 with size 13 etc.), the approximation results in profiles with very similar regret as that of the true MRCP.

	Size = 3				**Size = 5**				**Size = 7**			
Index	1	2	3	4	1	2	3	4	1	2	3	4
$\rho(\bar{\sigma})$	359	262	232	428	176	124	487	364	95	228	627	103
$\rho(\tilde{\sigma})$	505	275	265	532	253	144	727	365	575	397	794	183
$\rho(\sigma^*)$	615	275	242	554	535	144	806	737	491	514	973	172

	Size = 9				**Size = 11**				**Size = 13**			
Index	1	2	3	4	1	2	3	4	1	2	3	4
$\rho(\bar{\sigma})$	160	121	180	181	247	250	243	68	324	60	209	103
$\rho(\tilde{\sigma})$	249	156	205	230	263	405	378	165	435	60	318	134
$\rho(\sigma^*)$	236	314	759	330	388	596	446	152	705	216	327	479

Table 2.3: MRCP quality with approximation in symmetric zero-sum games.

2.3.5 Evaluation in Practice: Solver-based Regret

Given the general difficulty of computing MRCP beyond matrix games, studies often employ some other method to select a profile from the empirical game to evaluate. Any such method can be viewed as a meta-strategy solver, and so I use the term *solver-based regret* to denote regret in the true game of a strategy profile selected by an MSS from the empirical game. In symbols, the solver-based regret using a particular MSS is given by $\rho^{\mathcal{G}}(MSS(\mathcal{G}_{S\downarrow X}))$. By definition, MRCP is the MSS that minimizes solver-based regret.

An MSS that is commonly employed for solver-based regret is NE. NE-based regret measures the stability in the true game of a profile that is perfectly stable in the empirical game. Whereas any MSS is eligible to play the role of solver, not all are well-suited for evaluating strategy exploration. For example, self-play simply selects the last strategy added, and is completely oblivious to the rest of the strategy set X. This clearly fails to measure how well X as a whole captures the strategically relevant part of S, which is the main requirement of an evaluation measure as described above.

2.3.6 Solver Consistency

The PSRO framework as described to this point employs MSSs in two distinct ways: to direct a strategy exploration process, and to evaluate intermediate results in strategy exploration. It may seem natural to evaluate exploration that employs MSS M in terms of solver-based regret with M as solver. Indeed, much prior work in PSRO exploration has done exactly this (Lanctot, Zambaldi, et al. 2017; Li and Wellman 2021; Muller, Omidshafiei, et al. 2020).[2]

[2]Although the work by Li and Wellman (2021) is not focused on strategy exploration, it does present some plots (Figs. 2 and 3) with multiple curves using different MSSs for evaluating regret. For other works, I verified this by examining the published code and through my own efforts to reproduce the results in these papers. Specifically, I found the code published as part of OpenSpiel (Lanctot, Lockhart, et al. 2019) evaluates progress in exploration by regret of the MSS employed for exploration. I also reproduced the learning performance of PSRO with different MSSs and inferred that the MSS used for evaluation is the same as the one for strategy exploration, which is often apparent

As I demonstrate below, however, evaluating alternative MSSs M and M' for exploration using their respective MSSs as solvers can produce misleading comparisons, caused by neglecting the principle of evaluating the empirical game as a whole. Instead, I argue, one should apply the same solver-based regret measure to evaluate results under M and M'. In other words, the MSS employed in solver-based regret should be fixed and independent of the MSSs employed for exploration. I term this the *consistency* criterion.

To illustrate the necessity of solver consistency, here I offer two examples to demonstrate how a violation of my consistency criterion could lead to a misleading conclusion.

Example 1. *Consider the symmetric zero-sum matrix game of Table 2.4. Starting from the first strategy of each player, we perform PSRO with uniform and NE as MSSs, respectively. Note that using a uniform distribution over current strategies as MSS in PSRO essentially reproduces the classic fictitious play (FP) algorithm (Brown 1951). The first few iterations of PSRO are presented in Table 2.5. Due to symmetry, the two players' strategy sets and MSS-proposed strategies are identical.*

	a_2^1	a_2^2	a_2^3
a_1^1	(0, 0)	(-0.1, 0.1)	(-3, 3)
a_1^2	(0.1, -0.1)	(0, 0)	(2, -2)
a_1^3	(3, -3)	(-2, 2)	(0, 0)

Table 2.4: A symmetric zero-sum game (Example 1).

Figure 2.1a presents regret curves for both MSSs using NE-based regret, as well as the uniform-based regret curve for FP. If we violate the consistency criterion and compare uniform-based regret of FP with the NE-based regret of DO (i.e., green versus blue curves in Figure 2.1a), we would conclude FP converges faster than DO in the

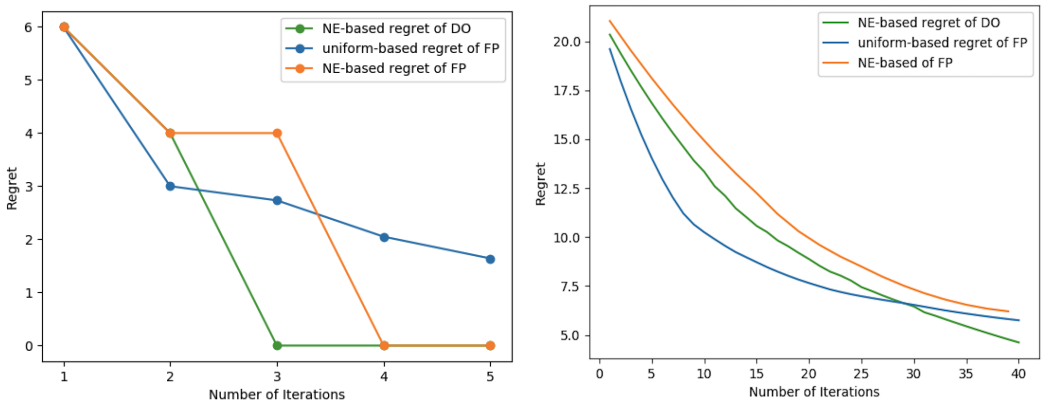
by examination of regret curves. For example, the NE-based regret curve of fictitious play oscillates dramatically while its uniform-based regret curve is much more smooth. So it is easy to identify which MSS was used for evaluation.

Iter#	Strategy Sets	DO proposed strategy
1	$(a_1^1),\ (a_2^1)$	$(1), (1)$
2	$(a_1^1, a_1^3),\ (a_2^1, a_2^3)$	$(0, 1), (0, 1)$
3	$(a_1^1, a_1^2, a_1^3),\ (a_2^1, a_2^2, a_2^3)$	$(0, 1, 0), (0, 1, 0)$

Iter#	Strategy Sets	FP proposed strategy
1	$(a_1^1),(a_2^1)$	$(1), (1)$
2	$(a_1^1, a_1^3),(a_2^1, a_2^3)$	$(\frac{1}{2}, \frac{1}{2}), (\frac{1}{2}, \frac{1}{2})$
3	$(a_1^1, a_1^3),(a_2^1, a_2^3)$	$(\frac{1}{3}, \frac{2}{3}), (\frac{1}{3}, \frac{2}{3})$
4	$(a_1^1, a_1^2, a_1^3),(a_2^1, a_2^2, a_2^3)$	$(\frac{1}{4}, \frac{1}{4}, \frac{1}{2}), (\frac{1}{4}, \frac{1}{4}, \frac{1}{2})$
5	$(a_1^1, a_1^2, a_1^3),(a_2^1, a_2^2, a_2^3)$	$(\frac{1}{5}, \frac{2}{5}, \frac{2}{5}), (\frac{1}{5}, \frac{2}{5}, \frac{2}{5})$

Table 2.5: PSRO process for DO and Fictitious Play.

first two iterations. However, FP cannot actually be better at strategy exploration, as the strategies introduced, a^1 and a^3, are identical under two MSSs. Moreover, at the third iteration, FP fails to add any new strategy, and so the improvement shown is not attributable to the exploration process.



(a) 3×3 game (Example 1). (b) 100×100 game (Example 2).

Figure 2.1: Regret curves evaluating NE and uniform as MSSs strategy exploration, under different solvers.

Comparing the two MSSs under NE-based regret (i.e., green versus orange regret curves), we see that where FP and DO generate identical empirical games their evaluations coincide. Thus, following the rule of consistency avoids reaching a misleading

conclusion about exploration. Note that we would reach the same conclusion if the two MSSs are evaluated under uniform-based regret (i.e., red versus blue curves). However, we observe that not all MSSs are equally effective for evaluation. In this example, although uniform-based regret consistently evaluates equivalent empirical games, its low weight on newly added strategies fails to adequately reflect exploration achievements. For example, the uniform-based regret curve remains well above zero even after the full-game NE has been covered in the empirical game. In Section 2.3.7, we provide a detailed discussion of this phenomenon and propose a scheme for evaluation solver selection.

Of course, if the goal is just to evaluate DO and FP as online algorithms, then the green versus blue comparison is appropriate. A key virtue of the PSRO framework, however, is that it highlights exploration as a distinct issue and provides the MSS abstraction for addressing it. Within an iterative EGTA approach, the choice of solver to employ for decision making at any stage is completely orthogonal to the method used to extend the game model, and so focusing attention on algorithms that couple these in particular ways (e.g., using the same MSS for solving and exploration) is unnecessarily limiting.

Example 2. *I further verified these observations in a synthetic zero-sum game with 100 strategies per player. Resulting regret curves averaged over 10 random starts are shown in Figure 2.1b.*

As for the previous example, comparing uniform-based regret of FP against NE-based regret of DO—breaking my consistency criterion—would lead us astray. First, we see that FP performs best initially, but is ultimately overtaken by DO. More importantly, as demonstrated in Section 2.3.8 below, even the assessment that FP's strategy exploration is more effective than DO's over the first thirty iterations is invalid. Indeed, the blue-versus-green comparison up to iteration 30 shows that the uniform strategy profile in the empirical game of FP is more stable (has lower regret)

than NE in the empirical game of DO. But as in the prior example, this is an artifact of selecting the uniform rather than the NE profile for evaluation. Moreover, as illustrated below in Figure 2.3, we should generally expect there to exist non-NE profiles in the empirical game of DO with significantly lower regret in the true game.

This example demonstrates mixed use of evaluation metrics may result in improper comparison among the performance of MSSs. Indeed, I found that this phenomenon is quite common in prior work, leading in particular to misleading evaluations of FP as a strategy exploration approach. In formulating the general consistency criterion, I emphasized that improper comparisons could be made with any two MSSs; the issue is not limited to FP or any specific MSSs employed in these examples.

2.3.7 Consistency in Poker Games and Evaluation Solver Selection

I further examine the consistency criterion in simplified poker games, specifically two-player Kuhn poker and Leduc poker. These poker games have been commonly employed in prior work within the PSRO framework, facilitating comparison of experimental results. Specifically, I evaluate FP, PRD, and NE as MSSs. Moreover, to select an effective solver to implement the consistency criterion, I propose a new evaluation solver selection scheme, designed to reveal the authentic performance of MSSs for strategy exploration.

2.3.7.1 Solver Consistency with FP

For Leduc poker, Figure 2.2a indicates DO performs better than FP under NE-based regret. However, the uniform-based regret is quite misleading as a measure of exploration performance of DO. It actually increases over much of the range, which would seem to suggest that adding strategies makes the game model worse, which intuitively makes little sense.

In Kuhn poker (Figure 2.2b), DO again outperforms FP under NE-based regret.

Uniform-based regret of DO is misleading for Kuhn as it is for Leduc poker.[3] FP shows much faster convergence under NE-based rather than uniform-based regret after twenty iterations or so. Indeed, the uniform-based regret is far from zero even at a hundred iterations. As we see in the examples above, uniform-based evaluation may misleadingly show smooth improvement where there is none. Here we see again that it can also leave the impression of slow progress even when the empirical game actually contains the key strategies needed for accurate solution.

2.3.7.2 Solver Consistency with PRD

I show experimental results of PSRO with PRD in Leduc poker in Figure 2.2c. We first note that following the rule of consistency, there is little performance gap between PRD and DO (i.e., the blue and orange curves). If we violate consistency and compare PRD-based regret of PRD against NE-based regret of DO (green versus blue curves), however, we would be prone to conclude that PRD clearly and significantly outperforms DO. For Kuhn poker (Figure 2.2d), we would conclude there is little difference, but looking closely and ignoring consistency might lead us to conclude that PRD is slightly worse in the limit. In both cases, we see that the choice of evaluation solvers can drive assessments about exploration performance.

The above examples have shown that not all MSSs are equally suited for evaluation, even if used in compliance with the consistency criterion. Consistency is important for achieving meaningful comparisons, but not sufficient. Conclusions about exploration performance are also sensitive to the selection among MSSs as evaluation solvers.

[3] My conjecture is that the new poker strategies introduced by DO after a point are very good at exploiting vulnerabilities in the current equilibrium, but quite poor as poker players overall. These strategies are quite important to include in the empirical game, to prevent exploitable solutions, even though they should not be part of the solutions themselves. This is a common game-reasoning phenomenon, providing another explanation for why uniform is a poor choice of solver for evaluation.

2.3.7.3 An Evaluation Solver Selection Scheme

Recall that MRCP is the MSS minimizing solver-based regret and thus the regret of the MRCP of an empirical game measures how well the empirical game covers the strategically relevant space. If we could feasibly compute the MRCP or an approximation, that would be a natural choice for solver-based regret. Though this is infeasible in general, we can capture the spirit of MRCP by attempting to minimize solver-based regret. Toward this end, I propose a heuristic evaluation solver selection scheme that chooses the solver with lowest-regret curve among running solvers. I demonstrate the significance of my scheme for evaluating different MSSs by checking the previous PRD example.

In the example, if we merely adhere to solver consistency with NE-based regret (i.e., comparing blue versus orange regret curves in Figure 2.2c), we would not distinguish the performance difference between PRD and DO. In this case, NE in the empirical game exhibits relatively high regret with respect to the true game. we know it is far from MRCP, as the green curve in this plot demonstrates the existence of lower-regret profiles in the same empirical games. Although we cannot tell exactly where the MRCP lies, the PRD solver in this example clearly provides a better approximation than does the NE solver. Considering PRD as the solver for evaluation and following solver consistency, we can likewise evaluate DO using PRD-based regret. The result is shown in the purple curve of Figure 2.2e (other regret curves are as in Figure 2.2c). PRD-based regret of DO is indeed lower than NE-based regret of DO (purple versus blue curves), and thus PRD as an evaluation solver successfully identifies the profiles with lower regret in the empirical games across DO iterations. This achieves the purpose of identifying profiles closer to MRCP as the basis for evaluation.

By comparing the PRD-based regret curves of DO and PRD, I observed that they exhibit similar improvement rates through early iterations, but eventually PRD shows

a small consistent advantage. This I regard as the best available evidence from these experiments on the authentic relationship between PRD and DO. Had we ignored solver consistency and compared the green and blue curves, we would have correctly concluded PRD's superiority but grossly overestimated the performance gap.

To state my work more explicitly: I argue for selecting the solver that minimizes regret in the given context. Specifically, fix a set of MSSs $\mathcal{M}$, typically the same set of MSSs being evaluated for strategy exploration. Let $\mathcal{R}$ be a set of PSRO runs employed to select the evaluation solver. At each iteration t of each run $r \in \mathcal{R}$, we have an empirical game over strategy set X_t^r. For each X_t^r and solver $M \in \mathcal{M}$, we evaluate regret in the full game of the empirical game solution under M. We then designate as evaluation solver M^* the MSS that performs the best over these runs:

$$M^* = \operatorname*{argmin}_{M \in \mathcal{M}} \sum_{r \in \mathcal{R}} \sum_t \rho^{\mathcal{G}}(M(\mathcal{G}_{S \downarrow X_t^r})).$$

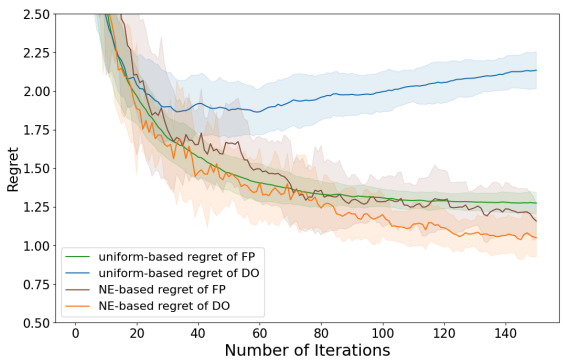
Alternatively, we can accommodate the possibility that which solver minimizes true-game regret may vary over the course of the strategy exploration process. I propose a *pointwise* selection scheme, which designates an evaluation solver M_t^* for each iteration t:

$$M_t^* = \operatorname*{argmin}_{M \in \mathcal{M}} \sum_{r \in \mathcal{R}} \rho^{\mathcal{G}}(M(\mathcal{G}_{S \downarrow X_t^r})).$$

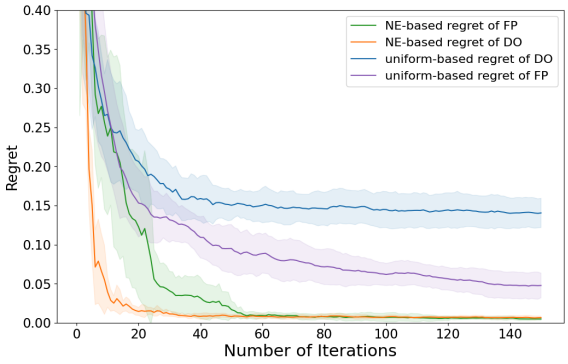
Note that the pointwise scheme, like that for selecting a single solver, accords with the consistency criterion. Variations that combine regrets across runs and time in some way other than summation are also admissible.

2.3.8 Evaluation Performance of MRCP

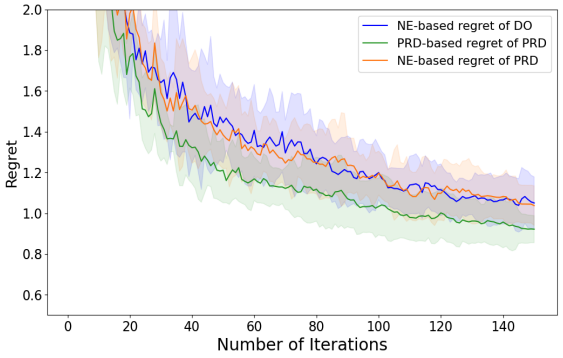
Though computation of MRCP in large games is generally infeasible, for experimental purposes we can evaluate it in a feasible context. Here I present such an evaluation on matrix games of fixed and modest size. Figure 2.3 displays averaged



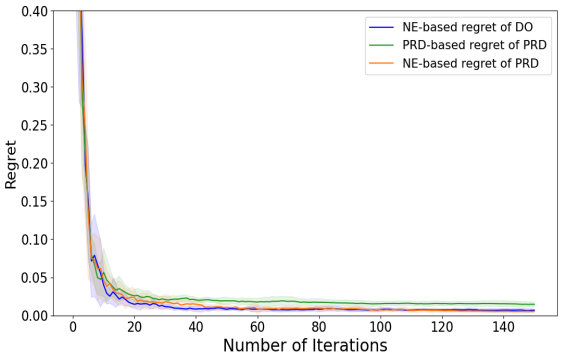
(a) Fictitious Play in 2-player Leduc



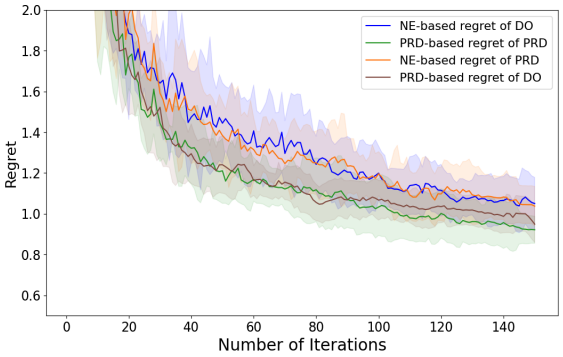
(b) Fictitious Play in 2-player Kuhn



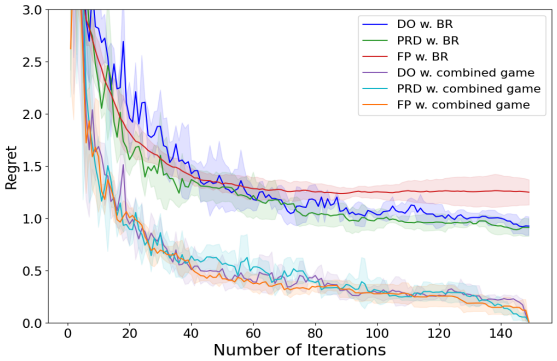
(c) PRD in 2-player Leduc



(d) PRD in 2-player Kuhn



(e) PRD strategies in DO run



(f) Regret curves by Combined Games

Figure 2.2: Experimental regret curves for poker games.

regret curves of PSRO runs on the same synthetic matrix game of Example 2, with FP and DO evaluated by MRCP-based regret. I observed that the MRCP-based regret by definition is lower than its NE-based regret counterpart. In this instance, the comparison using MRCP-based regret validates the qualitative comparison using NE-based regret. Notice that the gap between NE-based regret and MRCP-based regret diminishes as DO and FP gradually converge to a true game NE (i.e., all regrets approach zero). I also observed that the MRCP-based regret curves are much smoother than the NE-based regret curves. MRCP is monotone by definition, the steady performance improvement reflects more accurately the progress in quality of empirical game model achieved by strategy exploration.

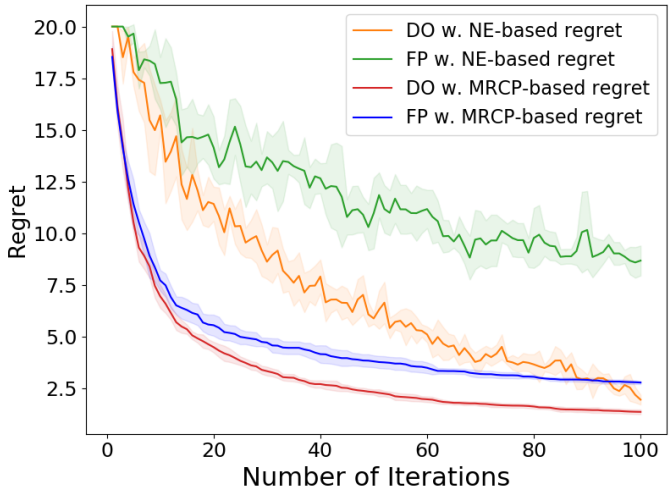


Figure 2.3: MRCP-based Regret vs NE-based regret.

2.3.9 Evaluation without Exact Best Responses

As noted above, calculating profile regret for purposes of evaluating MSSs generally requires identifying a best-response strategy. However, computing the exact best response may not be feasible in complex games. A particular approach is to collect the strategies generated across a set of PSRO runs, and evaluate regret with respect to that set. We refer to the game with all generated strategies as the *combined game*.

In general, regret with respect to the combined game is a lower bound on regret with respect to the true full game. Since the combined game has been used in practice as a heuristic approach to evaluate strategy exploration, it is important to examine its effectiveness.

To test the effectiveness of this approach, I compare results for evaluation with respect to a combined game with that of exact best response (i.e., the ground truth in our context), for some games where calculating exact best responses is feasible. Results are shown in Figure 2.2f. I observed that high-regret profiles in the true game may exhibit quite low regret in the combined game. Most concerning is that the slack in the regret bound may vary across MSSs being evaluated, thus producing misleading comparisons. Specifically in Figure 2.2f, despite the apparent higher regret of FP profiles in the true game, FP profiles exhibit lower regret in the combined game. My explanation for the phenomenon is that when one MSS can explore certain strategy to which strategies generated by other MSSs can deviate largely but not vice versa, the combined game fails to identify the correct ordering of MSSs.

2.4 Conclusion

The primary contributions of this study are methodological considerations for evaluating strategy exploration in EGTA, within the PSRO framework. My observations address nuances that have not been observed before, and may have led to misleading conclusions about the effectiveness of proposed methods. In particular, I proposed an evaluation scheme with a consistency condition, dictating that progress in strategy exploration under different MSSs be evaluated with respect to the same solver. This condition, while seemingly obvious, has not always been followed, perhaps because it is natural in online learning settings to evaluate a method at any point based on its own solution criterion. In the context of strategy exploration, in contrast, what is important is not what the latest strategy is, but how it affects the solution of the

model it is being added to.

CHAPTER III

Strategy Exploration by Setting MSSs

3.1 Introduction

As discussed in the introduction chapter, PSRO generalizes DO by introducing the concept of MSSs for controlling strategy exploration. In this chapter, I investigate how specifying an MSS can affect strategy exploration.

An obvious choice for MSS is the solution concept employed as the objective in game analysis, typically NE. Incrementally adding strategies that are best-responses to NE of the current strategy set is DO, and PSRO with NE as MSS is essentially DO with RL for computing (approximate) best response. Though DO is often effective, there is ample evidence that best-response to NE is not always the best approach to strategy exploration. Schvartzman and Wellman (2009a) observed cases where it would approach a true equilibrium extremely slowly, such that even adding random strategies could provide substantial speedups. More generally, Lanctot, Zambaldi, et al. (2017) argued that best-responding to Nash overfits to the current equilibrium strategies, and thus tends to produce results that do not generalize to the overall space. This was indeed their major motivation for defining a generalized MSS concept for strategy exploration. As an alternative MSS they proposed *projected replicator dynamics* (PRD), which employs an RD search for equilibrium, truncating the replicator updates to ensure a lower bound on probability of playing each pure

strategy. Placing constraints on probabilities for strategies on board enables best responses to strategies outside the equilibrium support, and hence can be viewed as a form of regularization to an exact NE. We credit this regularization for the improved performance of strategy exploration given by PRD.

I adopt an explicit regularization perspective to the specification and analysis of MSSs. I propose a novel MSS called *Regularized Replicator Dynamics* or RRD, which truncates the NE search process in intermediate game models based on a regret criterion. Specifically, at each iteration of PSRO, the best-response target profile is updated by running RD, stopping if the regret of the current profile with respect to the empirical game meets a specified regret threshold. The regret threshold is a hyperparameter, which may be adjusted to suit a particular game class, or annealed to control the degree of regularization across iterations. I assess the performance of RRD in various games and show that RRD outperforms several existing MSSs in terms of convergence rate and quality of intermediate empirical game models.

As the size of a payoff matrix is exponential in the number of players, the cost of maintaining completely specified models over the iterations of PSRO can be prohibitive beyond two players. To mitigate this issue, I employ a PSRO-compatible profile search method, called *backward profile search* (BPS), which finds solution concepts without simulating the whole payoff matrix. I combine RRD with BPS, and demonstrate the effectiveness of this combination in games with more than two players.

Finally, my experiments shed light on the source of the benefit of regularization for strategy exploration. Across a variety of settings, I found that the approximate empirical-game NE produced by RRD tend to have *lower regret in the full game*, compared to exact NE of the empirical game. This not only provides an explanation for the benefits of regulation, it may also suggest a way to evaluate the potential of novel MSS designs in PSRO-related approaches.

3.2 Literature Review

In the first instance of automated strategy generation in EGTA, Phelps et al. (2006) employed genetic search over a parametric strategy space, optimizing the basin size of attraction under replicator dynamics. Schvartzman and Wellman (2009b) combined RL with EGTA in an analogous manner. Questioning whether best response to equilibrium is an ideal way to add strategies, these same authors framed and investigated the general problem of *strategy exploration* in EGTA (Schvartzman and Wellman 2009a). They identified situations where adding a best response to equilibrium would perform poorly, and proposed some alternative approaches. Jordan, Schvartzman, and Wellman (2010) extended this line of work by proposing exploration of strategies that maximize the gain to deviating from a rational closure of the empirical game.

Investigation of strategy exploration was furthered significantly by introduction of the PSRO framework (Lanctot, Zambaldi, et al. 2017). PSRO entails adding strategies that are best responses to *some* designated other-agent profile, where that profile is determined by MSSs applied to the current empirical game. The prior EGTA approaches cited above effectively employed NE as MSS as in the DO algorithm (McMahan, Gordon, and Blum 2003). Lanctot, Zambaldi, et al. (2017) argued that with NE as an MSS the new strategy may overfit to the current equilibrium, and accordingly proposed and evaluated several alternative MSSs, demonstrating their advantages in particular games. For example, their PRD employs an RD search for equilibrium (Smith and Price 1973; Taylor and Jonker 1978), but truncates the replicator updates to ensure a lower bound on probability of playing each pure strategy. Any solution concept for games could in principle be employed as MSS, as for example the adoption by Muller, Omidshafiei, et al. (2020) of an evolutionary-based concept, α-rank (Omidshafiei, Papadimitriou, et al. 2019), within the PSRO framework.

The MSS abstraction also connects strategy exploration to iterative game-solving

methods in general, whether or not based on EGTA. Using a uniform distribution over current strategies as MSS essentially reproduces the classic FP algorithm, and as noted above, an MSS that just selects the most recent strategy equates to self-play. Note that these two MSS instances do not really make substantive use of the empirical game, as they derive from the strategy sets alone.

Wang, Shi, et al. (2019) illustrated the possibility of combining MSSs, employing a mixture of NE and uniform which essentially averages DO and FP. Motivated by the same aversion to overfitting the current equilibrium, Wright, Wang, and Wellman (2019) proposed an approach that starts with DO, but then fine-tunes the generated response by further training against a mix of previously encountered strategies. Balduzzi, Garnelo, et al. (2019) introduced a new MSS, called *rectified Nash*, designed to increase diversity of empirical strategy space. Dinh et al. (2022) proposed to interleave online learning with best responses for computing NE or correlated equilibria. Beyond selecting NE as a solution concept, Marris et al. (2021) proposed maximum welfare coarse correlated equilibrium (MWCCE), and maximum Gini coarse correlated equilibrium (MGCCE) for computing correlated equilibria, within the PSRO framework. McAleer, Wang, et al. (2022) proposed to use MRCP as an MSS for PSRO for two-player zero-sum games, referring as anytime PSRO the property of monotonic decrease in regret as empirical game extends given by MRCP.

The works discussed above focus on improving the efficiency of PSRO through setting MSSs. There are also some prior works improving PSRO through novel implementations. McAleer, Lanier, Fox, et al. (2020) proposed Pipeline PSRO (P2SRO). The key idea of P2SRO is that it initializes a bunch of strategies and assigns each strategy with a level. Then P2SRO warm-starts training each strategy in parallel against the NE of the empirical game involving strategies with lower levels. This pre-training scheme accelerates the overall training of PSRO. Zhou et al. (2022) developed an efficient PSRO (EPSRO) implementation for reducing the computational

cost of PSRO in two-player zero-sum games. The key insight is that the simulation for the empirical game is only used for computing best response target profiles. So as long as best response target profiles can be computed in other ways (e.g., a uniform MSS does not need the evaluation of a game model), there is no need to maintain the empirical game model and thus saving simulations. EPSRO achieves this by solving an unrestricted-restricted game (URR), in which one player is playing according to a restricted set of strategies, at each iteration of PSRO by online learning method and reinforcement learning. Zhou et al. (2022) showed that this procedure requires less simulations compared to the vanilla PSRO. Smith, Anthony, and Wellman (2021) improved the efficiency of best response computation in PSRO through knowledge transferring. The knowledge of best responses to each pure opponent's strategy is transferred to approximating the best response to any mixed opponent's strategy by Q-Mixing (Smith, Anthony, and Wellman 2023), a method for aggregating strategies.

The surveyed works up to this point are based on the normal-form representation of a game, which is also the representation that this thesis focuses on. However, there are also a considerable literature on PSRO leveraging the extensive-form representation to gain benefits in games naturally with tree structures. McAleer, Lanier, Wang, et al. (2021) proposed Extensive-Form Double Oracle (XDO), a double-oracle (DO) algorithm designed for two-player zero-sum extensive-form games. DO is based on the normal-form representation of the empirical games while XDO switches to the extensive-form representation of the empirical games. The extensive-form empirical game tree is constructed by the restricted set of strategies of players. Similar to DO, NE is deployed as an MSS but computed through CFR, which is a reasonable NE solver for two-player zero-sum extensive-form games. Then each player computes a best response against other players' equilibrium strategy profile. The best response operation will add some new actions at some non-terminal infostates until an NE is confirmed. Note that when a new action is added to an information state, multiple

strategies will be added to the corresponding normal-form representation essentially. So one iteration in XDO implicitly needs more simulation for profile evaluation than one iteration in DO.

3.3 Regularized Replicator Dynamics

Given the experiments in prior work, I observed that PRD is essentially a form of regularization and attributed the success of PRD to the regularization. Based on this observation, I adopt an explicit regularization perspective on strategy exploration. Specifically, I propose a method to derive approximate NE by truncating an RD-based search. My new MSS, called *regularized RD* or RRD, simply runs RD on the empirical game, stopping when the regret of the current profile (w.r.t the empirical game) meets a specified regret threshold λ, or a maximum number of iterations is reached. In the RRD procedure (Algorithm 4), each player's strategy is initialized with a uniform distribution over strategies in the empirical game. Then the replicator equation is iteratively applied until the regret of the current profile (w.r.t the empirical game) becomes smaller than the regret threshold λ. Since RD does not generally converge to an exact equilibrium, there is no guarantee a finite regret threshold λ will ever be reached. I therefore set a maximum number of iterations M, and if the limit is reached return the profile with the lowest regret found to that point.

Note that RRD supports direct control of the degree of regularization through an explicit parameter: the regret threshold. This parameter is meaningful across games with different strategy sets, as long as the utility scales on which regret is measured are comparable.

Algorithm 4 RRD

Require: an empirical game $\hat{\mathcal{G}}_{S\downarrow X}$, regret threshold λ, RD step size α, maximal number of iteration M

Initialize RD with $\sigma_i \leftarrow \text{Uniform}(X_i)$

$m \leftarrow 0$

while $\rho^{\hat{\mathcal{G}}_{S\downarrow X}}(\sigma) > \lambda$ and $m < M$ **do**

  for player $i \in N$ **do**

    $\sigma_i \leftarrow P(\sigma_i + \alpha \frac{d\sigma_i}{dt})$

  end for

  $m \leftarrow m + 1$

end while

Return σ

3.4 Convergence of RRD

I provide a theoretical bound for the regret of solution given by RRD.

Theorem 1. *Given the access to an exact best response oracle, PSRO with RRD associated with a reachable regret threshold λ converges to an empirical game containing at least one λ-NE.*

Proof. To prove Theorem 1, I first define the concept *ϵ-closeness*, which can be viewed as a stopping condition of PSRO.

Definition (ϵ-closeness)**.** *An empirical game with strategy space $X \subseteq S$ is ϵ-closed with respect to certain ϵ-NE $\sigma \in \Delta(X)$ and operator o if and only if $o(\sigma) \in X$.*

For example, if o is a best-response operator and $\epsilon = 0$, this definition means there is no beneficial deviation from the NE σ of the empirical game, and thus σ is an NE of the full game. When $\epsilon \neq 0$, ϵ-closeness indicates that the deviation strategy of the ϵ-NE σ of the empirical game already exists in the empirical game. Note that there

could exist an infinite number of ϵ-NE in an empirical game given a specific ϵ, so the definition of ϵ-closeness is associated with a specific ϵ-NE.

Lemma 1. *If an empirical game with strategy space $X \subseteq S$ is ϵ-closed with respect to certain ϵ-NE $\sigma \in \Delta(X)$ and best-response operator o, then σ is an ϵ-NE of the full game $\mathcal{G}$.*

Since σ is an ϵ-NE in the empirical game, there is no deviation strategy within the empirical game that results in regret large than ϵ. Mathematically, we have $\forall i \in N$, $\max_{s_i' \in X_i} u_i(s_i', \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}) \leq \epsilon$. Since the best-response operator finds the best deviation w.r.t the true game and the best deviation falls into the empirical game, we have $\forall i \in N$, $\max_{s_i' \in S_i} u_i(s_i', \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i}) \leq \epsilon$. Then σ is an ϵ-NE of the full game $\mathcal{G}$.

Given a finite strategy space S, by setting ϵ to be a reachable regret threshold λ, ϵ-closeness with respect to certain σ is always reachable by training against an ϵ-NE at each iteration, though all strategies in S should be added in the worst case. Once the ϵ-closeness is reached, the corresponding σ is an ϵ-NE of the full game according to Lemma 1. □

3.5 Selective Profile Evaluation using BPS

One obstacle to scaling PSRO is that the size of the empirical game grows exponentially in the number of players. Even in games with only a few players (e.g., three or four), exhaustive simulation of the payoff matrix may become infeasible as the strategy space grows. To mitigate this issue, I develop a simple profile search method for PSRO, which we call *backward profile search* (BPS, Algorithm 5). BPS resembles that of Brinkman and Wellman (2016), but takes into account the sequence in which the strategies were generated. At each iteration, BPS starts search from the strategies most recently added to the empirical game by PSRO, then searches poten-

tial deviations backward across previous PSRO iterations. The motivation is that the newest strategies are most likely to participate in equilibria. Once BPS confirms a solution of the empirical game, we can apply RRD to the subgame over the support of this solution. By construction, this subgame is completely evaluated (as required by RD), whereas the entire empirical game payoff matrix is only partially evaluated. In my experiments, I show that BPS can successfully find best-response targets in a three-player game, short of exhaustive evaluation of the empirical game.

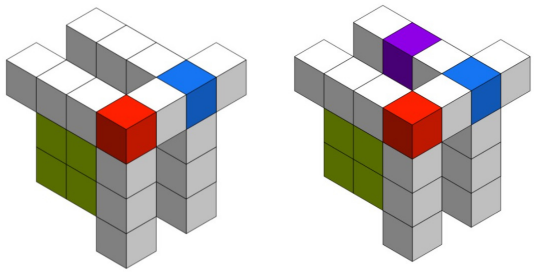


Figure 3.1: An illustration of a partial payoff matrix of a three-player empirical game and the workflow of BPS. Green: evaluated profiles from previous PSRO iterations; White: deviation profiles from NE of current subgame; Red: the profile with the most recently added strategies; Blue and purple: profiles with the largest deviation payoffs.

Figure 3.1 illustrates the mechanism of BPS at the third iteration of PSRO, in which each player has four strategies. The $4 \times 4 \times 4$ cube in Figure 3.1 represents the payoff matrix of the current empirical game. The payoffs of evaluated profiles from previous iterations are represented by green cells, while potential deviations from the equilibrium of the current subgame are represented by white cells. The missing cells indicate profiles that have not been evaluated. It is important to note that the current payoff matrix is incomplete. To determine the NE of the empirical game, the BPS algorithm initiates a search by evaluating the subgame formed by the latest strategy added by each player, which is represented by the red cell. Since the red cell is a pure-strategy profile, it is also the NE of the current subgame. Next, BPS assesses the payoffs of all possible deviations (white cells) from the red cell. If the blue cell

Algorithm 5 Backward Profile Search

Require: an empirical game $\hat{\mathcal{G}}_{S\downarrow X}$ with partial payoff matrix.

Initialize subgame with strategy sets $Z = (Z_i)$, $i \in N$, where $Z_i = \{s^i_\tau\}$, with s^i_τ the player i strategy added in the most recent PSRO iteration τ.
while True **do**
  $\sigma \leftarrow NE(\hat{\mathcal{G}}_{S\downarrow Z})$
  deviation_exists $\leftarrow$ False
  for player $i \in N$ **do**
    $s_i \leftarrow \text{argmax}_{s' \in X_i} u_i(s', \sigma_{-i}) - u_i(\sigma_i, \sigma_{-i})$
    if $s_i \notin Z_i$ **then**
      $Z_i \leftarrow Z_i \cup \{s_i\}$
      deviation_exists $\leftarrow$ True
    end if
  end for
  if $\neg$ deviation_exist **then**
    return σ
  end if
  Evaluate missing profiles of Z through simulation.
end while

contains the profile with the highest deviation payoff for a particular player i, then player i adds the corresponding deviation strategy to her strategy set of the current subgame. This action expands the profile space of the current subgame to include the red and blue profiles. BPS then repeats the process of evaluating all profiles in the current subgame, computing the NE of the subgame, and assessing potential deviations from the NE. Again, if the purple cell contains a deviation profile to the NE of the current subgame, then the corresponding deviating strategy will be added to the strategy set of the subgame. This iterative process is repeated until the NE of the subgame is confirmed, which means that no beneficial deviation could be found in the empirical game.

3.6 Experiments

3.6.1 Two-Player Leduc Poker

In Figure 3.2, I test PSRO with RRD in two-player Leduc poker and plot the regret curves (w.r.t the full game) given by FP, DO, PSRO with PRD, and PSRO with RRD (under two stopping criteria). I first observed that RRD yields a rapid convergence to a low-regret value compared to other MSSs. It is quite striking that RRD outperforms PRD (prior best known for this game) by such a large margin.

To show the benefits of using a regret threshold as a stopping criterion compared to a fixed number of RD updates, I plot the best regret curve of RRD using a fixed number of RD updates. I observed that RRD performs better using a regret threshold. This is because the number of RD updates that produces the right level of regularization varies across empirical games.

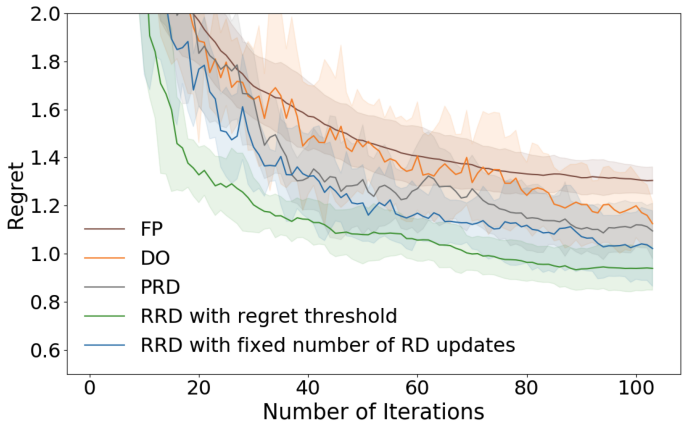


Figure 3.2: RRD performance in two-player Leduc Poker.

3.6.2 Multi-Player Games

In Table 3.1, I list the average of number of profiles evaluated at different PSRO iterations with and without BPS in three-player, four-player, and five-player poker games. Each profile is evaluated by averaging of 1000 payoff samples through simulation. I observed that employing BPS in three-player Leduc poker saved approximately

11% simulation effort, compared to exhaustive estimation, while the percentage of savings significantly increases for four-player and five-player games. Moreover, I noticed that evaluation savings grow as the number of players increases for each poker game. It is worth mentioning that the application of BPS does not incur any additional expense beyond the negligible cost of executing RD in subgames. Therefore, the resulting savings are virtually free.

Game($\|N\|$)	**Iter#**	$\|X\|$ **w. BPS**	$\|X\|$ **w/o BPS**	**Saving Pct.**
Leduc(3)	5	111	125	11.2%
	10	880	1000	12.0%
	15	2953	3375	12.6%
	20	7100	8000	11.3%
Leduc(4)	5	368	625	41.2%
	10	6696	10000	33.0%
	15	29572	50625	41.6%
	20	87953	160000	45.1%
Leduc(5)	5	1025	3125	67.2%
	10	54284	100000	45.7%
	15	400950	759375	47.2%
	20	1.58×10^6	3.2×10^6	51.6%
Kuhn(4)	5	430	625	31.2%
	10	7002	10000	30.0%
	15	35067	50625	31.0%
	20	109636	160000	31.5%
Kuhn(5)	5	1580	3125	49.4%
	10	50320	100000	49.7%
	15	377409	759375	50.3%
	20	1.57×10^6	3.2×10^6	50.9%

Table 3.1: The performance of BPS in poker games.

RRD can be easily combined with BPS for strategy exploration by first employing BPS to find a subgame of the empirical game that contains an empirical-game NE and then applying RRD to the subgame. Fig. 3.3 shows the performance of RRD with BPS in three-player Leduc pokers. I found that although RRD is applied only to the subgame of the empirical game, strategy exploration still benefits from regularization.

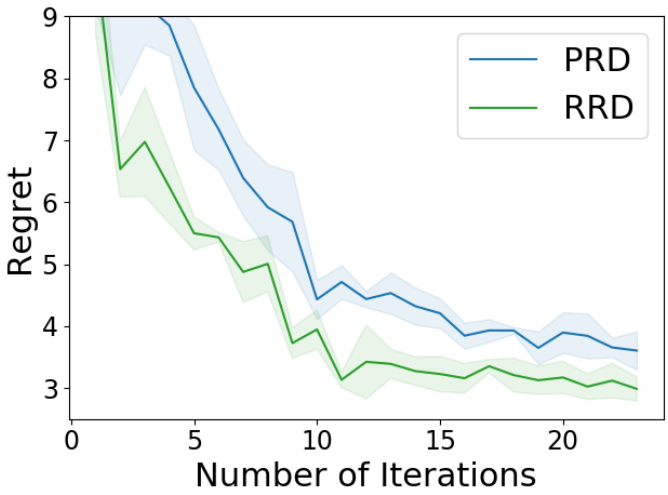


Figure 3.3: RRD performance in three-player Leduc poker with BPS.

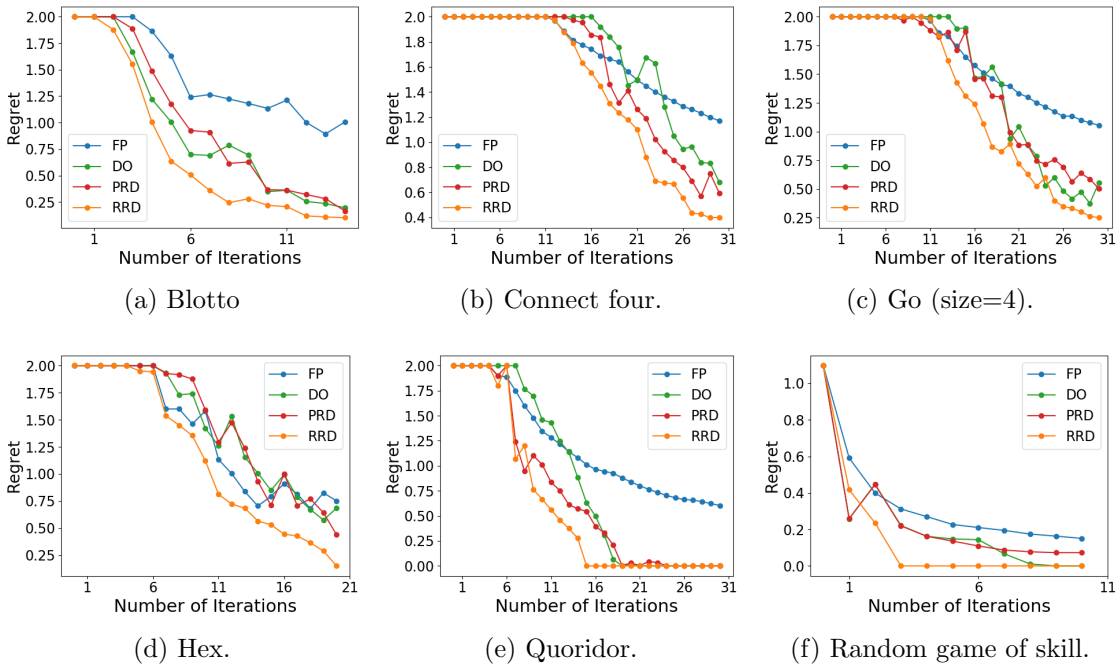
3.6.2.1 Real-World Games

I further evaluate our algorithms in six of the "real-world games" studied by Czarnecki et al. (2020): Blotto, Connect four, Go, Hex, Quoridor, and Random game of skill. I observed that RRD exhibits faster convergence than FP, PRD, and DO in all six games.

3.6.3 Attack-Graph Games

An *attack-graph game* is a two-player general-sum game defined on graph modeling paths of actions that can compromise a cyber-system (Miehling, Rasouli, and Teneketzis 2015).

In Figure 3.5, I show the performance of RRD on a large attack-graph game instance with 100 nodes and hence 2^{100} possible combinatorial actions. From Figure 3.5, I observed that even though the game of interest is large and beyond two-player zero-sum games, RRD still promotes faster convergence and less variance than DO, PRD, and FP.



(a) Blotto (b) Connect four. (c) Go (size=4).

(d) Hex. (e) Quoridor. (f) Random game of skill.

Figure 3.4: RRD performance in six real-world games studied by Czarnecki et al. (2020).

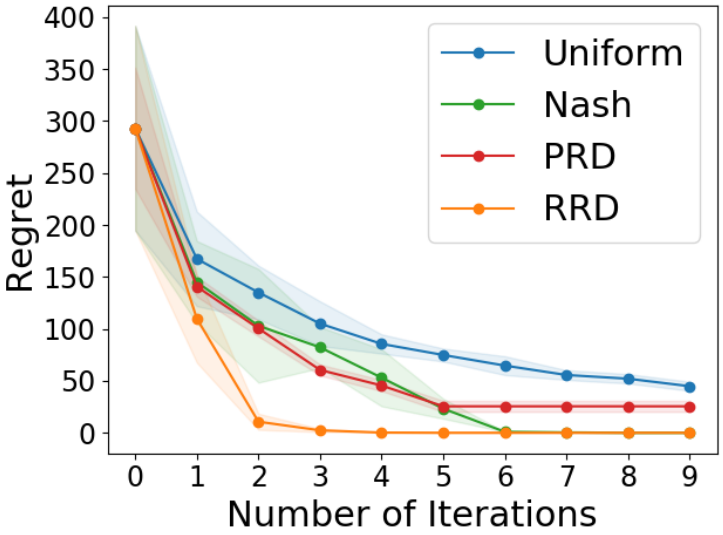


Figure 3.5: RRD outperforms FP, PRD, and DO in the attack-graph game.

3.6.4 Sequential Bargaining Games

I consider another non-zero-sum game with incomplete information, in the domain of *sequential bargaining* (Fudenberg and Tirole 1983; Rubinstein 1982; Rubinstein and Wolinsky 1985). In this game, two players alternately offer deals over multiple types of items, within a given time horizon. As bargaining games contain multiple equilibria, in this experiment, I am particularly interested in the quality of solution, such as social welfare (SW). I found that RRD tends to generate empirical games with higher SW solutions, compared to DO and FP (Figure 3.6). My hypothesis is that in avoiding overfitting a particular NE, regularization enables identification of parts of the solution space that achieve good results for both players.

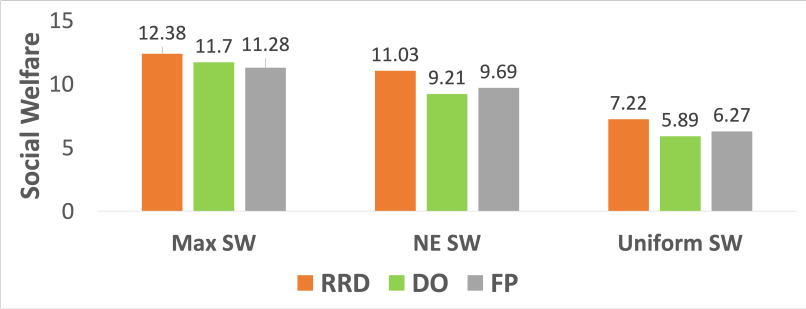
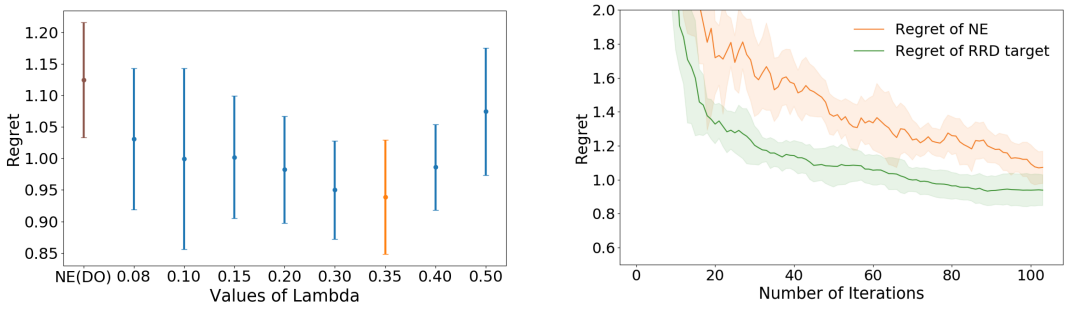


Figure 3.6: RRD performance in bargaining games. Each color represents an MSS and each bundle of colors shows the SW of a given solution concept in the corresponding empirical games. Max SW is the maximum SW among pure strategy profiles.

3.6.5 Stability with Varying Regret Threshold

To investigate the stability of learning performance w.r.t the regret threshold λ, I select a wide range of λs for RRD and compare the regrets at the last iteration of PSRO under these λs with the regret of DO in two-player Leduc poker. I plot the regrets in Figure 3.7a. From Figure 3.7a, I observed that all λs in the range yield a better learning performance than DO, which demonstrates the stability of the performance of RRD w.r.t the regret threshold λ. In addition, I observed that as the value of regret threshold λ increases, the learning performance first improves and

then becomes worse. This means that either excessive or inadequate regularization would damage the overall learning performance.



(a) Range of regret thresholds.

(b) Decreased regret after regularization.

Figure 3.7: Properties of learning with RRD in two-player Leduc Poker.

3.7 A Novel Explanation for Regularization

My key observation is that the performance of strategy exploration is strongly related to the regret of best-response targets *w.r.t the full game*. To observe this phenomenon, in Figure 3.7b, I run PSRO in two-player Leduc poker and use the NE-based and the RRD-based regret for evaluation. The two curves show the NE-based and the RRD-based regrets, respectively, as computed at each PSRO iteration. Note that throughout the run, the regret of the RRD solution is much smaller than that of the empirical NE. In other words, whereas RRD has higher regret than NE in the empirical game (λ versus zero), it reliably has lower regret in the full game. The same observation holds for using either NE or RRD as the MSS for strategy exploration. Since our ultimate objective is a full-game low-regret solution, this helps to explain why the regularization imposed by RRD apparently provides robustly improved performance for strategy exploration.

Note that this observation only goes so far; it is not the case that minimizing full-game regret always provides the optimal best-response target for strategy exploration. This is because the lowest full-game regret profile may not change much from one

PSRO iteration to the next, and so selecting targets on that basis may compromise the diversity of constructed empirical games. I test this explicitly by using MRCP as an MSS. My results confirm that the extreme choice of target is indeed suboptimal for strategy exploration.

3.8 Strategy Exploration with MRCP

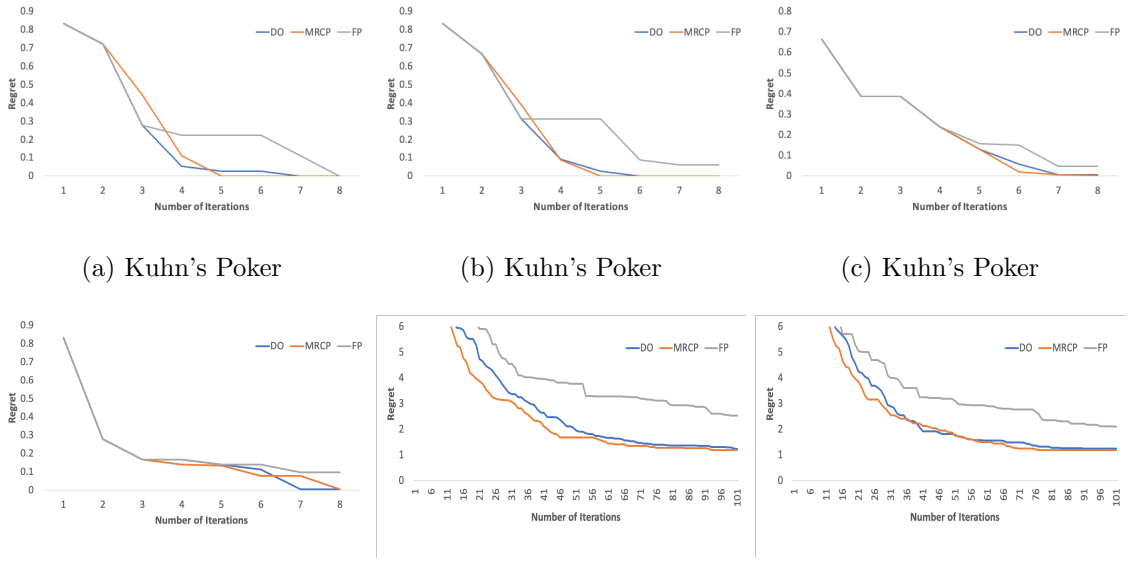
3.8.1 MRCP as an MSS

We have observed the existence of strategy profiles with lower global regret than NE in the empirical game and the experimental results of regularization shows that training against them results in improved learning performance than DO. One natural question to ask is whether training against the most stable profile can benefit strategy exploration the most (i.e., using MRCP as MSS).

To answer this question, I compare the performance of MRCP as MSS against DO and FP in normal-form two-player Kuhn's poker and a synthetic two-player zero-sum game. For Kuhn's poker, I randomly select 4 starting points and implement PSRO. Figure 3.8a-3.8d show that with 3 out of 4 starting points, MRCP converges slight faster than DO. For the synthetic matrix game, Figure 3.8e and 3.8f show the benefits of applying MRCP but the performance varies across different starting points.

In Figure 3.8, I observed that the MRCP has *some* ability for heuristic strategy generation. However, the advantage of using MRCP is not satisfactory in terms of convergence rate and computational complexity. I also found that using MRCP may converge slower in other games like Blotto, compared to DO and PRD.

The experiments show that training against the lowest-regret profile in the empirical game does not necessarily lead to a better overall learning performance. This is because the lowest-regret profile in the empirical game may not be changed much after adding a new strategy to the empirical game, yielding similar strategies contin-



(a) Kuhn's Poker (b) Kuhn's Poker (c) Kuhn's Poker

(d) Kuhn's Poker (e) Zero-Sum Game (f) Zero-Sum Game

Figure 3.8: Performance of MRCP being an MSS in Kuhn's poker and a synthetic matrix game.

ued to be added over PSRO iterations. Continuing adding similar strategies would slow down the overall learning in PSRO.

Now I illustrate why pursuing best response targets with extremely low regret may result in a slow learning using a matrix game shown in table 3.2. The matrix game contains 1000 strategies for each player. All missing entries of the payoff matrix are $(0, 0)$. Let's start PSRO with the first strategy (s_1, s_1). This matrix game is designed to have a long equilibrium search path for DO (as in many real-world games). Specifically, by best-responding to (s_1, s_1), each player adds s_2 to the empirical game, yielding a new NE (s_2, s_2). Similarly, if we repeat best responding to NE, we would first get a new NE (s_3, s_3) and then a long equilibrium path through the diagonal until we reach the NE of the full game (s_{1000}, s_{1000}).

Without loss of generality, suppose we are at iteration 2 (i.e., the empirical game includes (s_1, s_2) and (s_2, s_2) is an empirical NE). The MRCP of this empirical (symmetric) game is approximately $(1s_1, 0s_2)$ with regret $0.0112 \times 2 \approx 0.022$ (sum over players) (the regret of accurate MRCP is even lower). The regret of empirical NE

is $0.1 \times 2 = 0.2$ by deviating to s_3 from (s_2, s_2). When best responding to MRCP, we add s_{500} (only considering deviation strategies outside the empirical) and then the approximated MRCP remains the same (i.e., $(1s_1, 0s_2)$) for the empirical game. Therefore, further best responding to the approximated MRCP may again add some strategies similar to s_{500} and may not improve the learning performance dramatically.

Suppose RRD gives probability $(0.5, 0.5)$ on (s_1, s_2), then best responding to $(0.5s_1, 0.5s_2)$ leads to an equilibrium strategy s_{1000} directly, jumping out of the long equilibrium path of DO. The regret of $(0.5s_1, 0.5s_2)$ is $(0.005 \times 0.5 + 0.199 \times 0.5 - 0.011 \times 0.25 - 0.1 \times 0.25) \times 2 = (0.102 - 0.02775) \times 2 = 0.074252 = 0.1485$. So we can see that the regret of RRD is relatively low but not as low as the regret of MRCP since 0.02 (regret of MRCP) $<$ 0.1485 (regret of RRD) $<$ 0.2 (regret of NE). By best responding to the relatively low full-game regret profile, RRD avoids falling into the long diagonal path as DO. Meanwhile, its regret is not as low as the regret of MRCP so that the best response target at each PSRO iteration would keep being updated significantly rather than staying similarly.

	s_2^1	s_2^2	s_2^3	...	s_2^{500}	...	s_2^{1000}
s_1^1	(0, 0)	(0, 0.011)	(0, 0)	...	(0. 0.01)	...	(0,0.005)
s_1^2	(0.011, 0)	(0.1, 0.1)	(0.1, 0.2)	...	...	...	(0,0.199)
s_1^3	(0, 0)	(0.2, 0.1)	(0.2, 0.2)	...	...	...	(0, 0)
...	...	...	...	...	...	...	...
s_1^{500}	(0.01, 0)	...	...	...	...	...	(0, 0)
...	...	...	...	...	...	...	...
s_1^{1000}	(0.005, 0)	(0.199, 0)	(0, 0)	...	(0, 0)	...	(100, 100)

Table 3.2: A matrix game for demonstrating the slow update of MRCP.

3.8.2 Properties of Learning with MRCP

Theoretically, multiple MRCPs could exist in an empirical game. In addition, purely using MRCP as a MSS does not guarantee convergence to NE since the best-responding strategy to MRCP could already be included in the empirical game. I

define this property of MRCP as follows.

Definition. *An empirical game with strategy space $X \subseteq S$ is closed with respect to MRCP $\bar{\sigma}$ if*

$$\forall i \in N, s_i = \underset{s_i' \in S_i}{\operatorname{argmax}}\, u_i(s_i', \bar{\sigma}_{-i}) \in X_i.$$

To illustrate this concept, consider the symmetric two-player zero-sum matrix game in Table 3.3. Starting from the first strategy of each player and implementing PSRO with MRCP, we have the empirical game including a^1 and a^2. Since the profile $(\frac{10}{11}a_1^1, \frac{1}{11}a_2^1)$ is an MRCP and best responding to the profile could give a^2 again (note that a^3 is also a best response with the same payoff $\frac{10}{11}$), the empirical game is *closed* and never extends to the true game wherein the true NE is (a_1^3, a_2^3). In my experiments, I deal with this issue by only introducing new strategies with the highest deviation payoff outside the empirical game, in which case convergence is guaranteed.

	a_2^1	a_2^2	a_2^3
a_1^1	(0, 0) [2]	(-1, 1) [6]	(-0.5, 0.5)
a_1^2	(1, -1) [6]	(0, 0) [10]	(-5, 5)
a_1^3	(0.5, -0.5)	(5, -5)	(0, 0)

Table 3.3: Symmetric zero-sum game for explaining the closeness of MRCP Regret of profiles is shown in the square parenthesis.

3.9 Strategy Exploration with Quantal Response Equilibrium

One common assumption in game-theoretic analysis is the perfect rationality of players (i.e., players act according to NE). Since RRD prevents players from playing NE to some extent within the empirical game, it can be viewed as a way of restricting the rationality of players, which naturally relates RRD to Quantal Response Equilibrium (QRE) (McKelvey and Palfrey 1995, 1998), an equilibrium notion with bounded rationality.

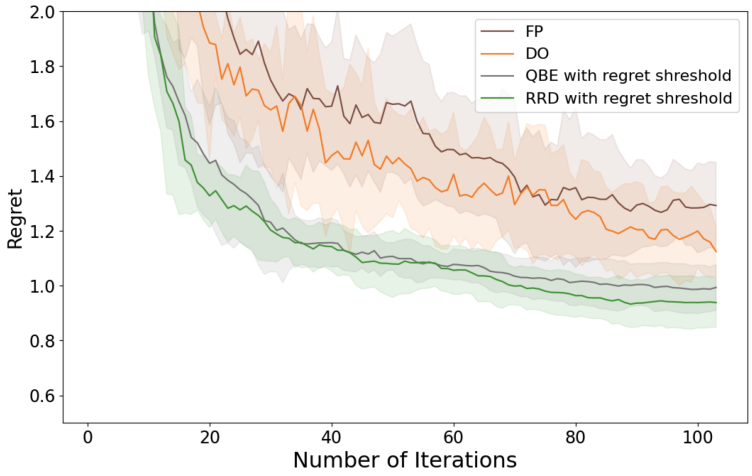


Figure 3.9: Learning performance with QRE.

Figure 3.9 shows the learning performance of QRE in two-player Leduc poker. I compute the QRE of the empirical game at every PSRO iteration using Gambit (McKelvey, McLennan, and Turocy 2006). Gambit will output a sequence of QRE with different rationality parameters and the QRE that reaches a specified regret w.r.t the empirical game is selected as the best response target (similar as computing RRD target). For comparison, I plot the learning curve of RRD with the same regret threshold of QRE as well as DO and FP. From Figure 3.9, I observed that although QRE shows a slight divergence in the end compared to RRD, it outperforms other MSSs, which demonstrates the potential of using QRE as a MSS in PSRO.

3.10 Exact Best Responses and Approximate Best Responses

Best response operation plays a significant role in iterative EGTA. Generally, best response oracles can be classified into two categories: exact best responses and approximate best responses. A wide range of tools can be used for best response oracles (e.g., search, black-box optimization and reinforcement learning) and which tool to choose depends on the particular game setting. My general observation is that the quality of best response oracles will exert a huge influence on the overall

learning performance. Higher-performing best response oracle yields faster learning performance. Therefore, although the main focus of strategy exploration in current research is to design new MSS, I believe choosing effective best response oracles is equally crucial.

3.11 Conclusion and Discussion on Computational Efficiency

I proposed RRD as a novel MSS for PSRO, explicitly based on regularization. By controlling the regret threshold, the degree of regularization can be adjusted to suit a particular strategy exploration context. In my experiments, I showed that RRD outperforms several existing MSSs in various games and investigate many properties of learning with RRD. To help scale beyond two-player games, I proposed BPS, a PSRO-compatible profile search method that avoids exhaustive simulation of the game matrix. I showed the benefit of regularization when combining BPS with RRD in three-player Leduc poker. Finally, I demonstrated that the performance of strategy exploration is strongly related to the regret of best-response targets and regularization could significantly decrease the regret of best-response targets, thus contributing to an improved learning.

Despite the ample evidence that demonstrates the effectiveness of iterative EGTA, several aspects of strategy exploration in EGTA can be further researched. There are two components in iterative EGTA that are computationally demanding: best response computation and empirical game simulation. To improve the efficiency of EGTA, both aspects should be further studied and improved.

One possible solution is to parallelize best response computation and empirical game simulations. Note that best response computation for each player is independent of that of others. At the meantime, the evaluation of each strategy profile is independent of that of other profiles in the empirical game. Based on these properties, parallelism could dramatically accelerate iterative EGTA.

Another possible solution is to leverage the power of learning. For example, one can transfer knowledge through one best response strategy to another to accelerate the best response computation. Or one can learn a high-performing regressor of the utility function and then evaluation can be achieved through querying the regressor rather than running a simulator. This can especially speed up the learning especially in many-player games.

CHAPTER IV

Strategy Exploration by Setting Response Oracles

4.1 Problem Statement

In prior work, PSRO usually proposes a single solution (typically an NE) based on the analysis of the final empirical game. Identifying a single solution is sometimes sufficient for the goals of game analysis, particularly in situations like two-player zero-sum games, where NE are interchangeable. In other cases, we might be interested in characterizing multiple equilibria, or identifying solutions with particular features (e.g., profiles with low regret and high social welfare are preferred in the traveler's dilemma (Basu 1994; Conitzer and Oesterheld 2022)).

As a result, I raise the question of how to steer strategy exploration toward NE with preferred characteristics, or more generally, a preferred game model. I approach this question by setting response objectives, which are objectives (approximately) solved through RL at each iteration of PSRO. Setting response objectives can be viewed as an alternative way to control strategy exploration, as opposed to setting MSSs discussed in Chapter III. I investigate the impact of ROs for strategy exploration and find that the choice of ROs can substantially change equilibrium outcomes.

4.2 Literature Review

4.2.1 Objectives in Classic Learning Dynamics

The best-response operation is a well-established technique used in classic game-learning dynamics, such as FP, weakened FP (Van der Genugten 2000), GWFP (Leslie and Collins 2006), and iterated best response. These dynamics often involve certain modifications in the RO. For example, the smooth FP method (Fudenberg and Levine 1995) perturbs best responses by a smooth and positive definite function (e.g., the Gibbs Entropy). This perturbation is not intended to steer strategy exploration toward a particular equilibrium but aims to achieve convergence through a concave function. Phelps et al. (2006) employed genetic search over a parametric strategy space, optimizing the basin size of attraction under RD. The basin size of attraction can be viewed as a different RO.

4.2.2 Variant Objectives in PSRO

In standard PSRO, the learning player optimizes its own payoff against other players' strategies (i.e., the standard RO). A few prior works have considered some variants of the standard RO. One relevant instance is a method called *diverse PSRO* (Perez-Nieves et al. 2021), which includes a diversity measure defined through a determinantal point process in the response objective. Liu et al. (2022) proposed the unified diversity measure (UDM), as a way to capture a variety of diversity metrics including effective diversity (Balduzzi, Garnelo, et al. 2019), expected cardinality (Perez-Nieves et al. 2021), and population diversity (Parker-Holder et al. 2020). As in diverse PSRO, UDM is combined with FP and PSRO, showing the effectiveness of promoting diversity of agents. Muller, Omidshafiei, et al. (2020) developed a preference-based response objective to enable PSRO to align with the properties of α-rank. Li, Lanctot, et al. (2023) deployed Monte Carlo tree search (MCTS) as the

best response oracle using different values (e.g., social welfare) to update values of nodes along the sample path in the back-propagation step of MCTS. The employment of different back-propagation values can also be viewed as modifications in ROs. They further proposed the joint Nash bargaining solution (NBS_joint) as an MSS and combined NBS_joint with search for computing NE in bargaining games. In these works, varying ROs to include diversity and search were shown to accelerate equilibrium computation in several settings.

4.3 PSRO with Generalized Response Objectives

Algorithm 6 PSRO, parametrized by solver MSS

Require: initial strategy sets X
1: Estimate $\hat{\mathcal{G}}_{S\downarrow X}$ by simulating $\sigma \in X$
2: Initialize target $\sigma \leftarrow \text{MSS}(\hat{\mathcal{G}}_{S\downarrow X})$
3: **for** PSRO iteration $\tau = 1, 2, \ldots, \mathcal{T}$ **do**
4: **for** player $i \in N$ **do**
5: **for** many RL training episodes **do**
6: Sample a profile $s_{-i} \in \sigma_{-i}$
7: **Standard PSRO:** Train best response oracle s'_i against s_{-i}
8: **PSRO with Generalized ROs:** Train a RL agent s'_i against s_{-i} to optimize $RO_i(s'_i, s_{-i})$
9: **end for**
10: $X_i \leftarrow X_i \cup \{s'_i\}$
11: **end for**
12: Update $\hat{\mathcal{G}}_{S\downarrow X}$ by simulating missing profiles over X
13: Compute best-response target $\sigma \leftarrow \text{MSS}(\hat{\mathcal{G}}_{S\downarrow X})$
14: **end for**
15: **Return** $\hat{\mathcal{G}}_{S\downarrow X}$

To understand the impact of ROs on strategy exploration, I introduce PSRO with generalized ROs in Algorithm 2 (with line 8), which generalizes standard PSRO by allowing ROs to be customized for each player. The customized ROs will be (approximately) solved through RL at each iteration of PSRO, typically achieved by setting proper rewards for RL. One natural hypothesis for generalized ROs is that ROs will substantially steer strategy exploration toward preferred equilibria, or more

broadly, empirical game models. To demonstrate this, I propose four RO instances for PSRO representing various strategy exploration preferences and access their impacts in sequential bargaining games and attack-graph games, comparing solutions found according to various criteria.

RO Name	Formula
Original RO	$u_i(s'_i, \sigma_{-i})$
Nash Product RO	$\alpha u_i(s'_i, \sigma_{-i}) + (1-\alpha) u_i(s'_i, \sigma_{-i}) u_{-i}(s'_i, \sigma_{-i})$
Social Welfare RO	$\alpha u_i(s'_i, \sigma_{-i}) + (1-\alpha) u_{-i}(s'_i, \sigma_{-i})$
Social Equity RO	$\alpha u_i(s'_i, \sigma_{-i}) - (1-\alpha) \lvert u_i(s'_i, \sigma_{-i}) - u_{-i}(s'_i, \sigma_{-i}) \rvert$
Minimizing Opponent RO	$\alpha u_i(s'_i, \sigma_{-i}) - (1-\alpha) u_{-i}(s'_i, \sigma_{-i})$

Table 4.1: Five response objective forms. $\alpha \in [0, 1]$ is a weighting parameter.

In Table 4.1, I describe four ROs considered in this work.[1] In each, the learning player i maximizes the RO over $s'_i \in S_i$, responding to the fixed other-player strategy σ_{-i}. First is the *Original RO*—standard in PSRO—which maximizes i's own utility against σ_{-i} (i.e., the *deviation payoff*). My first variant RO is named the *Nash Product Response Objective* (NPRO), which trades off the deviation payoff for the Nash product (i.e., the product of players' utilities). It has been proved by Nash that maximizing the Nash product corresponds to the Nash bargaining solution. By replacing the Nash product with other players' utilities, I obtain the second variant RO, called *Social Welfare Response Objective* (SWRO). When $\alpha = 0.5$, SWRO reproduces social welfare (i.e., the sum of players' utilities). My next RO, the *Social Equity Response Objective* (SERO), aims to balance utilities among players. SERO penalizes the deviation payoff by the difference in utilities among players. The final RO, *Minimizing Opponent Response Objective* (MORO), seeks to explicitly minimize other-player utility, while also maximizing deviation payoff.

[1]The ROs in Table 4.1 are defined for two-player games, so $u_{-i}(s'_i, \sigma_{-i})$ is a scalar. It would be straightforward to generalize these definitions for $|N|$ players.

4.4 Case Study: Sequential Bargaining Games

Sequential bargaining games represent a broad class of situations where two parties attempt to reach a deal through a series of proposals and counter-proposals (Fudenberg and Tirole 1983; Rubinstein and Wolinsky 1985). Variations of this model have been applied extensively, to scenarios including negotiations between nations in trade agreements, and private individuals bargaining over salaries. Sequential bargaining is a salient domain for EGTA due to its strategic complexity, and ubiquity in practice. These games also commonly exhibit multiple equilibria of varying preference, thus making them an especially interesting environment for studying how strategy exploration can affect which equilibria are captured by alternative paths of empirical game models.

4.4.1 Game Setup

I consider a non-zero-sum incomplete-information bargaining game, in which two players alternatively make offers to reach a deal over K types of items within time horizon T. The item of type k has M_k units available. For each bargaining instance, M_k is drawn from a uniform distribution, and revealed to both players. Each player has a private per-unit valuation for each item type, drawn independently from a specified distribution. Also for each instance the players are assigned independently drawn *disagreement values*, from player-specific distributions.

During each time step $t \leq T$, one player makes an offer and the other player decides whether to accept or reject it. Offers are made in vector form, representing the quantities of each item requested by the player (e.g., (3, 1, 1) requests 3 units of the first item and 1 unit each of the second and third items). If a deal is reached, the players receive a sum of their private values for the items in the offer, discounted by a factor of γ^t. If no deal is reached, they receive their disagreement values.

4.4.2 Experimental Results

In sequential bargaining games, a key consideration for the bargaining outcome is social welfare. Under the PSRO framework, it means that an empirical game model that contains higher welfare solutions will be more desirable than others. In my first experiment, I show that PSRO with either NPRO or SWRO tends to support higher welfare solutions compared to other PSRO variants. My experimental results support that ROs can substantially impact strategy exploration and equilibrium outcomes.

Specifically, I run PSRO with a combination of five MSSs (i.e., RRD, Nash equilibrium, uniform, MWCCE, and MGCCE) and three ROs (i.e., the original RO, the NPRO and the SWRO), producing fifteen MSS-RO combinations in total. PSRO with each MSS-RO combination will generate one empirical game. To evaluate the quality of solutions in the fifteen resulting empirical games, I adopt the consistency criterion discussed in Chapter II. The criterion states that whereas empirical games can be generated by different MSS-RO combinations, they should be evaluated based on measures of interest (e.g., regret, social welfare) applied to the same solution concept. For our purposes, I choose to compare the social welfare of the same solution concept across the generated empirical games. I select five solution concepts for evaluation (shown in Table 4.2), reflecting the quality of solutions in the empirical games from different angles. For example, NE, MWCCE, and MGCCE represent the common solution concepts whilst uniform reflects the average performance of strategies in the empirical game.

Solution Concept	Description
Max SW	The maximum SW across pure strategy profiles.
NE SW	SW of Nash equilibrium.
Uniform SW	SW of a uniform distribution over strategies.
MWCCE SW	SW of maximum social welfare coarse correlated equilibrium.
MGCCE SW	SW of maximum Gini coarse correlated equilibrium.

Table 4.2: Five solution concepts used for evaluation.

In Figure 4.1, I first show the impact of NPRO and SWRO on strategy exploration by replacing the original RO in Nash with NPRO and SWRO, respectively. Specifically, each color in Figure 4.1 represents an MSS-RO combination (if the RO is the original RO, it is omitted for simplicity), and seven empirical games were generated in total, one for each combination. Then the social welfare of the same concept[2] across the seven empirical games were bundled. For example, in the Max SW group (i.e., the left-most bundle), I plot the maximal social welfare in pure strategy profiles for each of the seven empirical games. In the NE SW group, the social welfare of NE of each empirical game (approximated by RD) is listed for comparison.

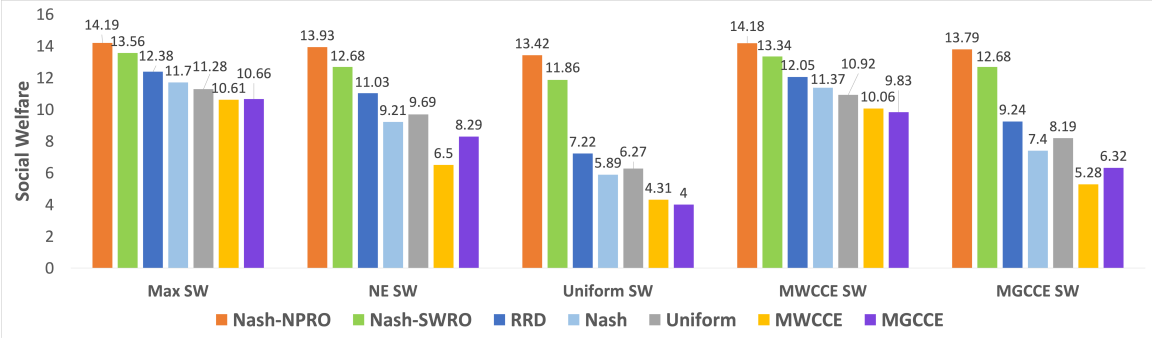


Figure 4.1: Social welfare of PSRO with various MSS-RO combinations. Each color represents an MSS and each bundle of colors shows the SW of a given solution concept in the corresponding empirical games. Max SW is the maximum SW among pure strategy profiles.

From Figure 4.1, I observed that the Nash-NPRO combination generates the greatest social welfare across all five solution concepts and Nash-SWRO earns the second highest social welfare. By comparing Nash with either Nash-NPRO or Nash-SWRO, we can see a significant increase in social welfare across solution concepts, associated with replacing the original RO with NPRO or SWRO. This observation confirms our concern for DO (i.e., PSRO with Nash) that it can stop at an NE with arbitrary features, and shows that NPRO and SWRO can steer strategy exploration toward the specified objective. As discussed below, this observation remains valid, regardless

[2]The social welfare is averaged over 15 random seeds.

of the MSSs employed. It is worth mentioning that Nash-SWRO achieve the highest social welfare with a weighting parameter $\alpha = 0.8$, as opposed to the setting $\alpha = 0.5$ that exactly captures social welfare. In other words, there is a benefit to considering the other-agent value in constructing a response strategy, but not to the same degree as one's own value.

Since there might exist multiple equilibria in an empirical game, which equilibrium to select for evaluation is pivotal. I demonstrate that this issue is relieved given the results in our particular situation. In particular, I assume the solution concept of interest is NE and use Nash-SWRO as an example. From Figure 4.1, we can see that the social welfare of NE found by Nash-SWRO (i.e., 12.68) is higher than that of any other combinations in Max SW. Since the social welfare of any mixed strategy profile is upper bounded by the maximal social welfare over pure strategy profiles, the social welfare of NE found by Nash-SWRO is determined to be higher than the social welfare of any profiles (including NE) found by other MSS-RO combinations. Therefore, which NE is picked from the empirical games for evaluation is not a concern given our results. Another way to reason about this argument is that since the set of NE is a subset of CCE, the social welfare of NE is upper bounded by the social welfare of MWCCE in the corresponding empirical game, which is further bounded by Max SW. As the social welfare of NE found by Nash-SWRO is higher than that of MWCCE given by other MSS-RO combinations, Nash-SWRO must result in NE with higher social welfare than others. The same observation holds for Nash-NPRO.

In Figure 4.2, I combine NPRO with each MSS and plot the social welfare of the same five evaluation concepts. I observed that the social welfare of solutions given by Nash-NPRO remains highest across all combinations. One interesting observation is that Nash-NPRO outperforms RRD-NPRO even though Nash performs worse than RRD (green vs dark blue in Figure 4.1). This observation indicated that MSSs and ROs have a coupled influence on strategy exploration. My hypothesis for the

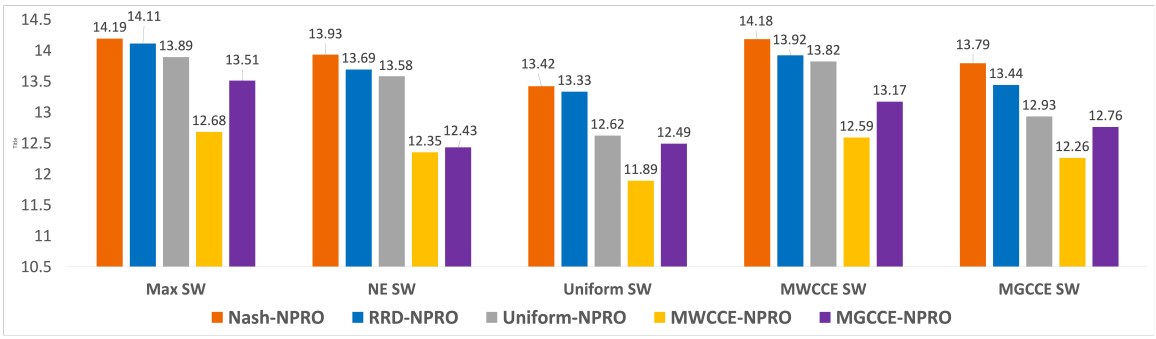
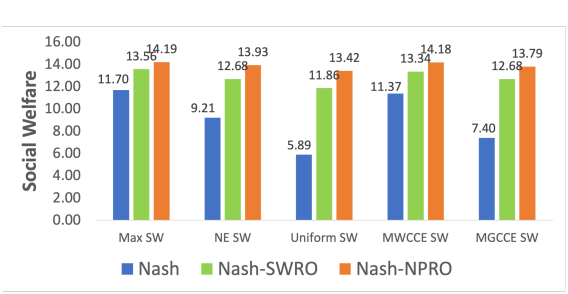


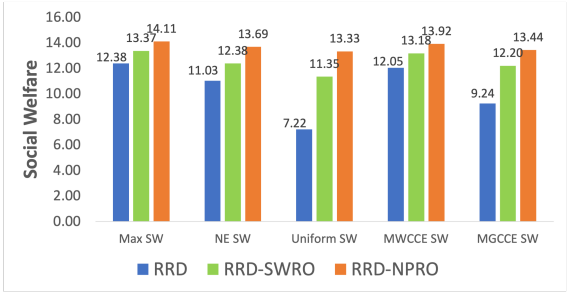
Figure 4.2: Social welfare of PSRO with MSSs and NPRO evaluated under the same set of solution concepts.

reduced performance of RRD with NPRO is that the regularization imposed by RRD is superfluous given that it varies from BR. I observed the same phenomenon for SWRO.

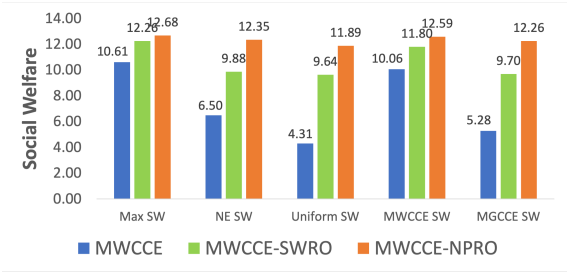
In Figure 4.3, I plot the social welfare before and after integrating NPRO and SWRO with each individual MSS, respectively. I observed that the social welfare of all evaluation concepts increases after applying either NPRO or SWRO, regardless of the MSS employed. This showed that either NPRO or SWRO can direct strategy



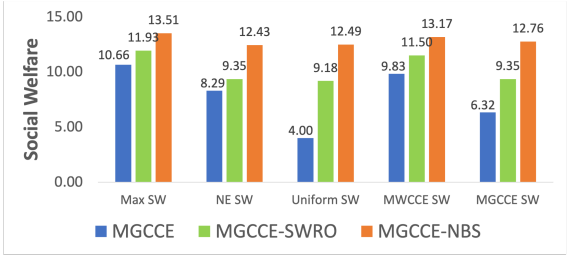
(a) Nash as an MSS.



(b) RRD as an MSS.



(c) MWCCE as an MSS.



(d) MGCCE as an MSS.

Figure 4.3: Social welfare of MSSs with and without SWRO and NPRO.

exploration and identify strategy spaces that cover solutions with higher social welfare, though a proper choice of MSSs will further raise the social welfare (i.e., Nash-NPRO yields the highest SW).

Furthermore, I found that the social welfare given by Nash-NPRO or Nash-SWRO is strikingly high compared to global maximum of the full game. I measure the social welfare distance between the Nash bargaining solution and the NE given by Nash-NPRO and Nash-SWRO through *Nash bargaining ratio*. NBS describes a bargaining solution with properties including invariant to affine transformations or invariant to equivalent utility representations, Pareto optimality, independence of irrelevant alternatives, and symmetry. A Nash bargaining solution can be found by maximizing the Nash product $u_1(\sigma_1)u_2(\sigma_2)$. I defined the Nash bargaining ratio as

$$\text{NBS\_Ratio} = \frac{SW(\text{NE})}{SW(\text{NBS})},$$

which is the social welfare of any NE divided by the social welfare of NBS. The NBS can be computed by assuming a **complete information** of the bargaining game and then searching deals that maximizes the Nash product. Therefore, the computation of NBS includes more information of the game than my experiments with PSRO and it is not surprising that NBS will result in a higher social welfare. In my bargaining game instance, the maximum social welfare achievable with complete information is 15.08 whereas the social welfare of NBS is 15.01, which is almost the maximum. Therefore, the NBS ratios given the NE of Nash-NPRO and Nash-SWRO are $13.93/15.01 = 0.928$ and $12.68/15.01 = 0.845$, respectively. The ratio is strikingly high especially when the setting for Nash-NPRO and Nash-SWRO is incomplete information.

Figure 4.4 plots individual player utilities in the final NEs produced across 11 PSRO runs with different MSS-RO combinations. The convex hull of these points represent the empirical Pareto frontier of equilibria of the bargaining game. Points

with the same color are obtained by running PSRO with different random seeds. From the plot, I observed that the equilibria given by both Nash-NPRO and Nash-SWRO are on the frontier and appear dense while the equilibria found by other combinations spread out in the utility space. This means that both Nash-NPRO and Nash-SWRO can steer strategy exploration toward preferred game models, in a relatively stable manner. Moreover, I found that player 1 earns a higher utility than player 2 in all equilibria found by Nash-NPRO and Nash-SWRO, which reveals the advantage of moving first in these equilibria with higher social welfare.

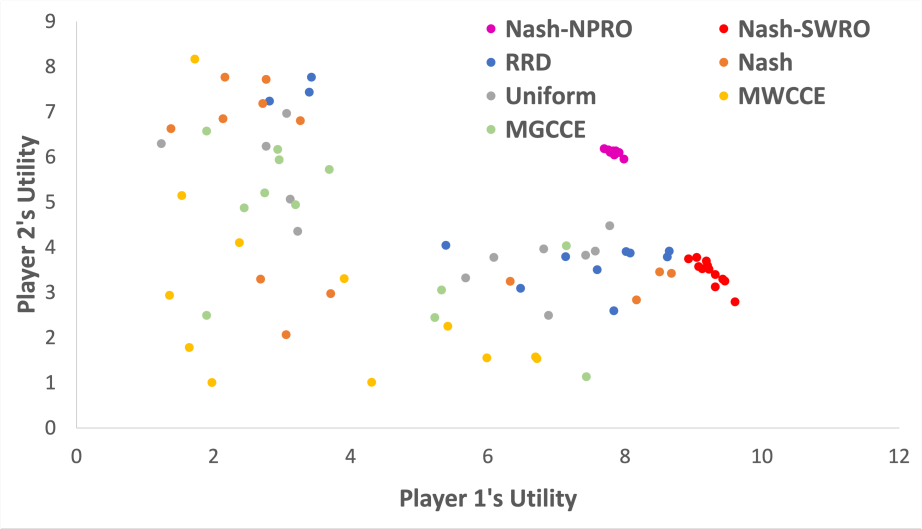


Figure 4.4: NE scatters in the utility space. Each color represents an MSS-RO combination. Points with the same color are obtained by running PSRO with different random seeds.

To further demonstrate the impact of ROs on strategy exploration, I combine Nash with SERO and list the averaged utilities in equilibria given by some selected MSS-RO combinations in Table 4.3. I observed that Nash-SERO can efficaciously reduce the utility difference between two players, compared to other combinations. Moreover, I noticed that Nash-SERO causes an increase in social welfare from Nash. This rise can be attributed to the transformation of SERO into a formula that accounts for the utility of both players when $u_i(s'_i, \sigma_{-i}) - u_{-i}(s'_i, \sigma_{-i}) \geq 0$ and $\alpha > 0.5$.

MSS-RO	$u_1(\sigma^*)$	$u_2(\sigma^*)$	$\lvert u_1(\sigma^*) - u_2(\sigma^*)\rvert$	Social Welfare
Nash-NPRO	7.82	6.11	1.71	13.93
Nash-SWRO	9.24	3.44	5.80	12.68
Nash-SERO	5.56	5.83	**0.27**	11.39
Nash	4.26	4.95	0.69	9.21

Table 4.3: A shrinkage in the utility gap caused by the SERO.

4.4.3 Disagreement Offers and Discount Factor

In sequential bargaining games, disagreement offers and discount factor play an important role in making a complex and meaningful bargaining situation. Without disagreement offers and a discount factor, the bargaining process will reduce to an ultimatum game. In an ultimatum game, a sum of money is given to a single player, known as the proposer, who must divide it with another player, known as the responder. The responder is aware of the total sum. After the proposer makes its decision, the responder has the option to either accept or decline the proposal. If the responder accepts, the money is distributed as per the proposal. However, if the responder declines, both players receive nothing. Both players are aware of the outcomes of the responder's decision to accept or reject the offer beforehand.

To understand the importance of disagreement offers and discount factor visually, I analyze the equilibria of sequential bargaining games with and without disagreement offers and discount factor. In Figure 4.5, I plot the bargaining procedures given by equilibrium strategies. In these two bargaining instance, two bargaining players, denoted as P1 and P2, exchange offers over a basket of 3 types of fruits. The private values of players for items are listed as P1 values and P2 values. At each round, I list which player is making an offer and what the offer is. The table on the left shows the bargaining procedure without disagreement offers and discount factor, in comparison to the case on the right where a disagreement offer [1.2, 1.5] and a discount factor $\gamma = 0.9$ are applied.

	Basket	1 🍎	4 🍐	1 🍊
	P1 Values	1	2	1
Round	P2 Values	0	1	6
1	P1	1	4	1
2	P2	1	2	1
3	P1	1	4	1
4	P2	1	1	1
5	P1	1	4	1
6	P2	1	4	1
7	P1	1	4	1
8	P2	1	4	1
9	P1	1	3	1
10	P2		Agree	

	Basket	1 🍎	5 🍐	1 🍊
	P1 Values	0	1	5
Round	P2 Values	2	1	3
1	P1	1	5	1
2	P2	1	4	1
3	P1	1	5	1
4	P2	1	2	1
5	P1	1	4	1
6	P2	1	4	1
7	P1	1	4	1
8	P2	1	4	1
9	P1	1	4	1
10	P2	1	4	1

Figure 4.5: Playthroughs of sequential bargaining. **Left:** No disagreement offers and discount factor. **Right:** Having both disagreement offers and discount factor.

In the left instance, since there is no disagreement offer and discount factor, player 1 keeps requesting all items until the second to the last round, in which it can give another player a small portion of items to avoid gaining nothing at the end. Note that the optimal action of the player 1 should be giving one unit of items that is worth the least for itself and has positive value for another player. However, since player 2's value is not observable, player 1 chooses to give one unit of pear with value 2 to player 2. Note that the player's behavior in this instance needs not to be exactly optimal due to the application of approximate best responses and the fact that the player is optimizing its expected utility over the instance distribution. In the right instance, since we have both a disagreement offer and a discount factor, the players would like to receive the disagreement offer when the value of the offer is less than the disagreement offer after discounting. So player 2 prefers to receive the disagreement offer rather than accepting the offer made by player 1 at the final round. This example shows that disagreement offers and discount factor play a crucial role in making a complex and meaningful bargaining situation, and reveals the rationality of players in equilibria learned through PSRO.

4.5 Case Study: Attack-Graph Games

The attack-graph games often have several equilibria exhibiting differing offensive and defensive interactions, thus providing a particularly intriguing setting for investigating the impact of ROs for strategy exploration.

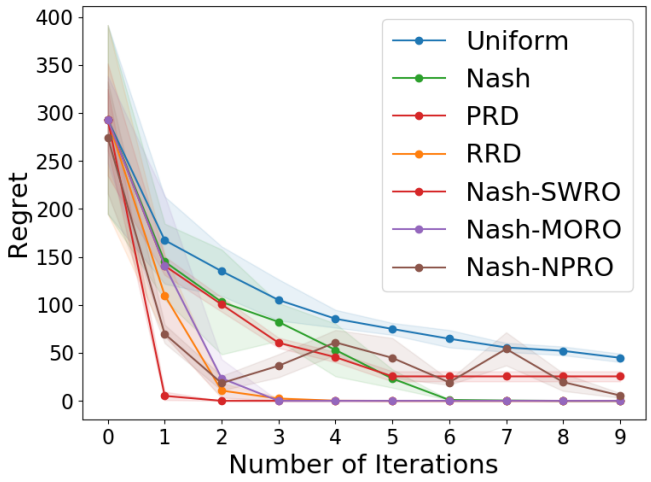


Figure 4.6: PSRO with different MSS-RO combinations in attack-graph games.

In Figure 4.6, I plot the regret curves of different MSS-RO combinations in the attack graph game. I observed that strategy exploration with generalized ROs can affect the convergence speed to an NE. In particular, Nash-SWRO converges to an NE faster than others in this instance. Then I compute the averaged utilities in equilibrium strategies for both players, where the defender (D) and the attacker (A) earn utilities (D: -56.67, A: 27.50) for Nash, (D: -13.43, A: 53.43) for Nash-SWRO, and (D: -84.91, A: 84.98) for Nash-MORO. An interesting observation is that Nash-SWRO can improve both players' utilities in the equilibrium, though the attack-graph games appear to be purely adversarial. This observation exhibits the strategic complexity in the attack-graph games and reveals a possibility for both players to cooperate in these games. Additionally, I observed that Nash-MORO can enlarge the utility difference between two players and the attacker can cause more damage to the defender.

4.6 Case Study: Computing Berge Equilibrium

In previous sections, we have seen that generalized ROs can substantially affect some features (e.g., social welfare) of common solutions. Now I demonstrate that PSRO with generalized ROs can also be employed for computing specific solution concepts. In particular, I focus on computing Berge equilibrium (Berge 1957), an equilibrium concept commonly used and studied in social games. In a BE, each player ensures that all other players will receive the highest payoff. I demonstrate how a BE in two-player games can be found by PSRO by employing a simple utility swapping trick.

4.6.1 Berge Equilibrium

I follow the definition of BE from an individual perspective given by Zhukovskii (1985), though BE was first defined in terms of coalitions by Berge (1957).

A strategy profile $\sigma^B \in \Delta(S)$ is a Berge equilibrium if for $i \in \{1, 2\}$ and $s_{-i} \in S_{-i}$,

$$u_i(\sigma^B) \geq u_i(\sigma_i^B, s_{-i}). \tag{4.1}$$

This definition means that for any particular player i, its utility would not increase if it sticks to its own BE strategy while other players can change their strategies. This can be viewed as the altruism in the game playing since in a BE each player ensures the highest payoff for all other players who are also employing their BE strategy. Note that this is different from the spirit of NE, where players are assumed to be selfish and only maximize their own payoff.

4.6.2 Computing Berge Equilibria with PSRO

To adapt PSRO for computing BE, the following two questions should be answered:

1. How can we compute a BE in the current empirical game if BE is employed as an MSS?

2. How can we design a corresponding response objective such that a full-game BE is reached when PSRO stops?

To answer the first question, I first prove that a BE of the empirical game can be obtained by computing NE of the corresponding utility-swapping game, assuming a BE of the empirical game exists.

Proposition 1. *Given a two-player finite game* $\mathcal{G} = (\{1,2\},(S_i),(u_i))$, σ^B *is a BE of* $\mathcal{G}$ *if and only if it is an NE of the utility function swapping game* $\mathcal{G}' = (\{1,2\},(S_i),(z_i))$, *where* $z_i = u_{-i}$ *for* $i \in \{1,2\}$.

Proof. According to the definition of BE, a BE σ for the game $\mathcal{G}$ satisfies

$$u_1(\sigma^B) \geq u_1(\sigma_1^B, s_2), \forall s_2 \in S_2$$

and

$$u_2(\sigma^B) \geq u_2(s_1, \sigma_2^B), \forall s_1 \in S_1.$$

By setting $z_1 = u_2$ and $z_2 = u_1$, we have

$$z_2(\sigma^B) \geq z_2(\sigma_1^B, s_2), \forall s_2 \in S_2$$

and

$$z_1(\sigma^B) \geq z_1(s_1, \sigma_2^B), \forall s_1 \in S_1.$$

We can see that each player changes to maximizing their own payoff after swapping the utility functions. Therefore, by the definition of NE, σ^B is an NE of the utility-swapping game $\mathcal{G}'$. By the same reasoning procedure, we can prove that an NE of the utility-swapping game is a BE of the original game. □

Now I examine our assumption in Proposition 1 that a BE exists in an empirical game.

Corollary 1. *For two-player finite games, a BE exists in both the full game and the empirical game.*

Proof. Since both the full game and the empirical game are finite two-player games, the corresponding utility-swapping games are well-defined and are also finite two-player games. By the classic theorem of the existence of NE (Nash Jr. 1950a), at least one NE exists in the utility-swapping games. By the same argument as in the proof of Proposition 1, the NE in the utility-swapping game is a BE of the original game, so a BE in the original game exists. □

To answer the second question, I propose the *Berge Equilibrium Response Objective* (BERO) based on the definition of BE, which simply takes the form of $u_{-i}(s_i, \sigma_{-i})$. With BE as an MSS and BERO, I show that the Berge PSRO algorithm for computing BE in Algorithm 7 and Algorithm 8.

Algorithm 7 Berge PSRO

Require: initial strategy sets X
Initialize target $\sigma \leftarrow \text{BE}(\hat{\mathcal{G}}_{S\downarrow X})$
deviation $\leftarrow$ True
while deviation **do**
  deviation $\leftarrow$ False
  for player $i \in \{1, 2\}$ **do**
    for many RL training episodes **do**
      Sample a profile $s_{-i} \in \sigma_{-i}$
      BERO: Train a RL agent s_i' against s_{-i} to maximize $u_{-i}(s_i', s_{-i})$
    end for
    if $s_i' \notin X_i$ **then**
      $X_i \leftarrow X_i \cup \{s_i'\}$
      deviation $\leftarrow$ True
    end if
  end for
  Compute response target $\sigma \leftarrow \text{BE}(\hat{\mathcal{G}}_{S\downarrow X})$
end while
Return σ

Algorithm 8 BE as an MSS

Require: an empirical game $\hat{\mathcal{G}}_{S\downarrow X}$
  if $\exists$ An utility-swapping game $\hat{\mathcal{G}}'_{S\downarrow X'}$ from previous iterations **then**
    $X' \leftarrow X$
  else
    Construct a new utility-swapping game $\hat{\mathcal{G}}'_{S\downarrow X'}$ with $X' = X$
  end if
  Update $\hat{\mathcal{G}}'_{S\downarrow X'}$ by simulating missing profiles over X'
  $\sigma \leftarrow NE(\hat{\mathcal{G}}'_{S\downarrow X})$
  Return σ

Proposition 2. *With exact best response oracles, PSRO with BE as an MSS and BERO stops when the full-game BE exists in the empirical game.*

Proof. Suppose σ^B is a BE of the current empirical game. According to Algorithm 7, when the algorithm stops, for every player i, there is no other player's strategy $s_{-i} \in S_i/X_i$ such that

$$u_i(\sigma^B) < u_i(\sigma_i^B, s_{-i}).$$

Therefore, according to the definition of BE, the BE of the empirical game is a BE of the full game.

□

4.7 Conclusion

I studied the effectiveness of setting customized ROs for guiding strategy exploration toward desired empirical games under the PSRO framework. Through experiments in sequential bargaining games and attack-graph games, I showed that ROs can steer strategy exploration toward games with solutions aligned with specified objectives.

One future research direction could be designing ROs to find certain equilibrium refinements. This may require an extensive-form representation of PSRO if the refinement (e.g., subgame perfect equilibrium) is defined only for extensive-form games.

CHAPTER V

Game Model Learning for Mean Field Games

5.1 Introduction

In previous chapters, we investigate strategy exploration and its evaluation in finite games. I extend the prior results of strategy exploration to MFGs (Huang, Malhamé, Caines, et al. 2006; Lasry and Lions 2007), a model for analyzing games with a large number of players. Specifically, MFGs model strategic interactions among a conceptually infinite number of agents and consider their aggregate behavior. Aggregate agent behavior is summarized by a distribution over states of the population. Then the analysis can be reduced to the characterization of the optimal behavior of a single representative agent in its interactions with the full population, as represented by the mean field. MFG model can support game-theoretic analysis that would be intractable for a standard corresponding model of a game among a large but finite number of players. I first formally define a game model for MFGs and then present the EGTA framework for MFGs in the next chapter.

In EGTA, a game model serves as a fundamental element for various types of analysis and hence crucial for a complete EGTA framework for MFGs. Due to the non-linearity of the utility function of MFGs in the mean field and mean fields are continuous, it is infeasible to represent the utility function with a finite number of values as in finite games. Therefore, a game model cannot be defined in terms of

usual components. I fill in this gap and propose a game model learning approach, which is essentially a form of regression that learns a utility function over a restricted set of strategies and distributions derived by these strategies. I study a general setup of MFGs where strategies and distributions are both time-dependent (i.e., non-stationary), and hence encoding them explicitly as inputs to a learner entails impractically high dimensionality. To handle the time-dependency, I propose a *coding scheme* method and learn a game model (i.e., the utility function) that takes as inputs sufficiently-statistical representations of strategies and distributions, and outputs a utility value. With the method, time-dependent strategies and distributions are no longer explicitly encoded as inputs as for the true utility function and hence our method circumvents the representation complexity induced by the time-dependency.

To learn an effective game model for MFGs, it is important to endow the model with the ability of generalization across the space of strategies that induce mean fields. To reach this goal, the training data set is required to include uniformly-sampled mixed strategies in the restricted strategy space. For a large MFG, a game model typically involves dozens of strategically significant strategies, which creates a high-dimensional strategy space. To obtain samples in such high-dimensional spaces, I propose a combination of two sampling schemes: grid sampling and sampling from Dirichlet distributions with varying concentration parameters. By combining coarse coding with the data sampling methods, I demonstrate that my approach can successfully achieve effective generalization and accurate predictions on utilities. I also show that the learned game model can support game-theoretic analysis, that is, both FP and RD empirically converge to NE with the model.

5.2 Literature Review on Game Model Learning in Finite Games

Vorobeychik, Wellman, and Singh (2007) first introduced the concept of learning normal-form game models as utility function regression from simulation data sampled over continuous strategy spaces. They demonstrated the approach using single-parameter strategy representation. Ficici, Parkes, and Pfeffer (2008) clustered a large number of players into two roles based on data consisting of strategy profiles and utilities. Then regression of the utility function was applied for each role.

Wiedenbeck, Yang, and Wellman (2018) deployed Gaussian process regression to learn the utility function of large symmetric games. The regressor takes as input a pure strategy profile and outputs a utility vector. In symmetric games, since the utility function depends on how many players choose each strategy (and not which players), the input vector to the regressor can be represented by a non-negative tally vector with one dimension per strategy. Once the utility function for pure strategy profiles is learned, the extension to mixed strategy profiles can be achieved by taking the expectation. The authors further investigated the use of neural networks for the regression.

Sokota, Ho, and Wiedenbeck (2019) extended game model learning to role-symmetric games by regressing the deviation payoff function rather than the payoff function. Deviation payoffs can be directly used by algorithms such as replicator dynamics, saving the effort to derive deviation payoffs first from the payoff function. This approach has also been deployed by Li and Wellman (2021) to learn NE in Bayesian games.

There has also been some work on learning game models on some succinct descriptions of games. For example, Duong et al. (2009) and Fearnley et al. (2015) learned graphical game models (Kearns 2007) from utility data. Li and Wellman (2020) combined structure learning and payoff regression to induce tractable game models with

many players.

For MFGs, there has been a considerable literature on learning equilibra (Cardaliaguet and Hadikhanloo 2017; Guo et al. 2019; Laurière et al. 2022; Mishra, Vasal, and Vishwanath 2020; Muller, Rowland, et al. 2021; Perrin, Pérolat, et al. 2020; Wang and Wellman 2023b). Despite we show that the learned game model can facilitate equilibrium search in our experiments, our approach **differs from** these works in nature since our object is to learn the game model of MFGs rather than presenting an equilibrium learning algorithm. Therefore, they are not fair baselines for our approach.

More broadly, a related research field that studies learning utility function or reward function is Multiagent Inverse Reinforcement Learning (MIRL) (Chen, Liu, and Khoussainov 2021; Lin, Adams, and Beling 2019; Natarajan et al. 2010; Yu, Song, and Ermon 2019). However, my setting is distinct from that of MIRL in the following ways. First, the goal of inverse RL is to observe agents' actions and determine the reward function they are optimizing. In my setting, players are not necessarily optimizing any reward function but playing according to their strategies. The returns of their plays (i.e., utilities) are observable by the learning algorithm. With these utilities, my objective is to learn a mapping from the set of strategies to the utilities. Second, my goal is to induce a normal-form representation of the game, based on utility observations obtained through a black-box simulator. Although the underlying MFG may contain temporal structure, we have no access to the transitions (i.e., current states of players, actions that are taken, immediate rewards, next states) of the game, which are essential for MIRL. Third, I specifically study MFGs featuring time-dependent strategies and distributions. Time dependency dramatically increases the complexity of learning and that is why I propose the coarse encoding. Existing works in MIRL for MFGs such as Chen, Liu, and Khoussainov (2021) bypass this complexity induced by time-dependency by focusing on stationary strategies.

5.3 Methods

5.3.1 Time-dependent Strategies and Distributions

In this chapter, I focus on game model learning in single-population MFGs, so the index for population i is omitted. The primary goal of game model learning is to learn the utility function $u(\sigma, \mu)$ over a restricted set of strategies. Since $u(\sigma, \mu)$ is an expectation of $u(s, \mu)$ over $s \in \Lambda$ (this follows Equation 1.3), it is sufficient to learn the utility function $u(s, \mu)$ on pure strategies. In MFGs, a strategy s and distributions μ are generally both time-dependent (i.e., $s = (s_t)_{t\in[0,T-1]}$ and $\mu = (\mu_t)_{t\in[0,T]}$). Explicitly encoding them as inputs to a learner would result in a high-dimensional feature vector and entail an impractically complex regression setup. To handle this issue, I propose a coarse coding scheme.

5.3.2 Coarse Coding

In finite games, a utility function can be treated as a black box (illustrated in Figure 5.1) that abstracts away the details of strategies as well as the mechanism of utility computation. A black-box utility function takes a simple representation of a strategy profile $I(\mathbf{s})$ (i.e., a vector of strategy indices one per player) as input and outputs utility samples.

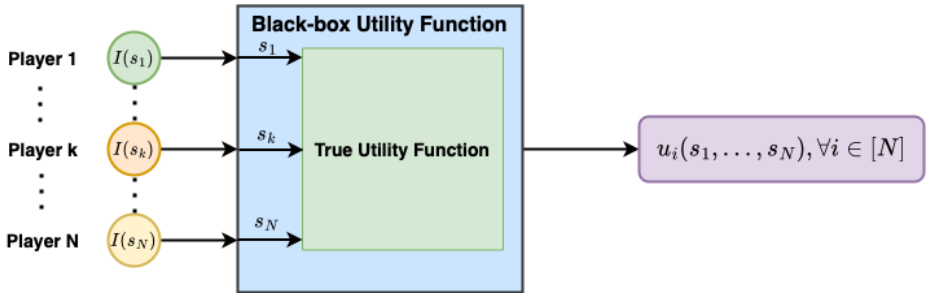


Figure 5.1: An illustration of a black-box utility function.

I was inspired by this abstraction and handle the time-dependency by learning a black-box version of the true utility function u. Mathematically, consider a restricted

strategy set $\Lambda \subseteq S$. Let $I : \Lambda \to \mathbb{Z}_+$ be a function that index each strategy $s \in \Lambda$ with a positive integer. Let σ be the mixed strategy that induces the distributions μ^σ. Since the forward equation (Eq. 1.4) is deterministic, it is sufficient for σ to determine μ^σ given a fixed initial distribution $\mu_0 \in \Delta(Z)$. Instead of learning $u(s, \mu^\sigma)$ with time-dependent inputs, we learn a black-box utility function $\hat{u} : I(\Lambda) \times \Delta(\Lambda) \to \mathbb{R}$ as a game model using sufficient representations $I(s)$ and σ of s and μ^σ. I refer to this representation as *coarse coding*.

The object is to predict the true utility $u(s, \mu^\sigma)$ by $\hat{u}(I(s), \sigma)$ and thus minimizing the mean square loss $E[(u(s, \mu^\sigma) - \hat{u}(I(s), \sigma))^2]$. The regression is based on neural networks and the structure of the neural networks is depicted in Figure 5.2. In practice, $I(s)$ can be any representation of categorical inputs (e.g., one-hot encoding) and σ is a vector of strategy probabilities.

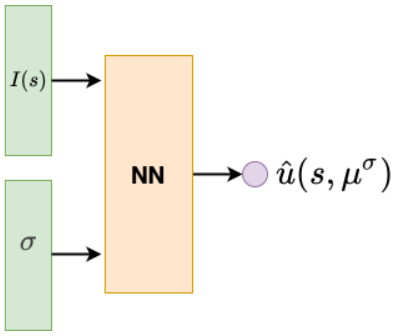


Figure 5.2: A neural network structure for coarse coding.

There are two main benefits of coarse coding. First, the coarse coding scheme only requires a simple network structure with a one-hot encoded strategy and a vector representation of a mixed strategy. So the implementation of coarse coding is straightforward.

Second, the coarse coding scheme can dramatically simplify the execution of equilibrium search algorithms for MFGs. Consider FP as an example. FP for MFGs can be summarized as iterating three steps: (1) computing a best response strategy against given distributions μ, (2) updating the averaged strategy after adding a new

best response strategy, and (3) computing the distributions μ induced by the averaged strategy through the forward equation (Eq. 1.4). Averaging strategies into one merged strategy and distribution induction (i.e., the second step and the third step) are usually considered to be computationally expensive (Laurière et al. 2022). With coarse coding, the computation in step (2) and step (3) can be largely reduced. In step (2), the averaged strategy is now represented by a probability vector on Λ rather than a strategy entity and the averaged strategy update becomes a update of the probability vector. Hence, there is no need to merge strategies across states, actions, and time horizon. Meanwhile, for step (3), distribution induction is no longer needed since the probability vector (i.e., a mixed strategy σ) can sufficiently represent the distributions μ^σ (because the initial distribution μ_0 is fixed and the forward equation is deterministic). In essence, learning with coarse coding can be viewed as an end-to-end learning approach, in which distribution induction is implicitly learned from data. Based on these benefits, FP can be significantly accelerated given a coarse-coded utility model.

5.3.3 Data Sampling

For regression of the utility function, a data point constitutes an index of a pure strategy $I(s)$, a mixed strategy σ, and a true utility $u(s, \mu^\sigma)$. To collect these data points, the basic requirement is that the sampled mixed strategies σ's should uniformly distribute in the restricted strategy space so as to endow the learner with the ability of generalization across the space of induced distributions. For a large MFG, a game model typically contains dozens of strategies, which makes the sample space high-dimensional. It is well-known that uniformly sampling in a high-dimension space suffers the curse of dimension and a finite number of samples will mainly concentrate at the center of the space while less samples appear at its corners.[1] To handle this

[1]Uniformly sampling can be implemented using Dirichlet distributions.

issue, I combine two sampling schemes.

My first sampling scheme is grid sampling. In grid sampling, a grid of points on the surface of a strategy simplex are sampled. The grid sampling can be achieved by combinatorial algorithms (Nijenhuis 1975). Specifically, denote the parameter K as the sum of each unnormalized sample and $|S|$ as the number of strategies in a restricted set. The total number of the samples is $\frac{(K+|S|-1)!}{K!(|S|-1)!}$ (i.e., $K + |S| - 1$ choose $|S| - 1$). For example, for $K = 4$ and $|S| = 2$, samples have the vector form $[(0, 4), (1, 3), (2, 2), (3, 1), (4, 0)]$. After being normalized by K for each sample, the samples generate a grid on the $|S|$-simplex (i.e., $[(0, 1), (\frac{1}{4}, \frac{3}{4}), (\frac{2}{4}, \frac{2}{4}), (\frac{3}{4}, \frac{1}{4}), (1, 0)]$). For my experiments, I select $K = 4$ and $|S|$ varies based on specific MFGs.

The next sampling scheme relies on symmetric Dirichlet distributions with different concentration parameters α. Mathematically, the density function of a symmetric Dirichlet distribution is represented in terms of Gamma function, as follows

$$f(x_1, \ldots, x_{|S|}; \alpha) = \frac{\Gamma(\alpha|S|)}{\Gamma(\alpha)^{|S|}} \prod_{i=1}^{|S|} x_i^{\alpha-1}.$$

A concentration parameter α controls the density of samples. With $\alpha > 1$, samples are dense near the centroid while samples are sparsely distributed with $\alpha < 1$ and close to corners of the simplex. I set a range of values for α and sample the corresponding Dirichlet distribution, aiming at generating a sufficient number of samples covering the strategy simplex.

Combining samples from these two sampling schemes, we can obtain a set of mixed strategy samples in the strategy space. Then we can induce μ^σ for each sampled σ by the forward equation (i.e., μ_t will be induced by σ_t given μ_{t-1} throughout the time horizion T) and evaluate $u(s, \mu^\sigma)$ for each pure strategy $s \in \Lambda$ and μ^σ. Therefore, we can obtain data points that contain indices of the pure strategy $I(s)$, mixed strategies σ's, and the corresponding utilities $u(s, \mu^\sigma)$.

In the discussion above, I assume a restricted set of strategies in a game model and our object is to learn a utility function over this set of strategies. To obtain such a set of strategies, I apply a query-based iterative EGTA approach (Muller, Rowland, et al. (2021) and Chapter VI), where the term "query-based" means the utility data is simulated whenever it is needed. This is a common approach for assembling a strategy portfolio for a game model. The generated strategies are both diverse and exhibit interesting strategic interactions (e.g., containing an NE of the true MFG).

5.3.4 Approximating Nash Equilibrium

Whether a learned game model $\hat{u}$ is sufficiently accurate to support NE computation is crucial for analyzing MFGs. To conduct this evaluation, I consider two learning dynamics FP and RD, and implement them using the learned game model. For each learning dynamic, I measure the regret of intermediate strategies generated by the dynamic with the true utility function and the model, respectively. If the regrets given by the true utility function and the model are close to each other and both converge to 0 (i.e., approach an NE), we can claim that the learned game model provides good prediction and can support NE computation.

In Algorithm 9 and Algorithm 10, I show the implementations of FP and RD with a game model. Compared to implementing FP and RD with the true utility function (Algorithm 12 and 13 introduced in the next chapter), the implementation with a game model becomes much simpler due to coarse coding, which we have discussed in Section 5.3.2.

In Algorithm 9, for each iteration of FP, we predict the utility value $\hat{u}(I(s), \bar{\sigma})$ for all strategies s in the restricted strategy set Λ using the model $\hat{u}$. Then a best response against the current averaged strategy $\bar{\sigma}$ is selected and its count of being a best response is increased by 1. Finally, we update the averaged strategy $\bar{\sigma}$ by normalizing the the count of being a best response across strategies. The output of

FP is the averaged strategy $\bar{\sigma}$ after J iterations.

Algorithm 9 Fictitious Play with a Game Model

Require: a game model $\hat{u}$. Define the initial strategy $\bar{\sigma}$ as the average of strategies in the restricted set $\Lambda = (s_1, \ldots, s_\tau)$.

1: **for** Iteration $j \in \{1, \ldots, J\}$ **do**
2: Evaluate $\hat{u}(I(s), \bar{\sigma})$ for all $s \in \Lambda$
3: Select a best response $s \leftarrow \text{argmax}_{s' \in \Lambda} \hat{u}(I(s'), \bar{\sigma})$
4: Update $\bar{\sigma}$: $\bar{\sigma}(s) \leftarrow \frac{1}{j} n_s$ for all $s \in \Lambda$, where n_s is the count of strategy s being a best response
5: **end for**
6: **Return** $\bar{\sigma}$

In Algorithm 10, for each iteration of RD, we again predict the utility value $\hat{u}(I(s), \bar{\sigma})$ for all strategies $s \in \Lambda$ using the model $\hat{u}$. Then we compute the fitness, which is the expected utility given the current averaged strategy $\bar{\sigma}$. Finally, the probability of each strategy $s \in \Lambda$ in the averaged strategy $\bar{\sigma}$ is updated proportional to the deviation payoff from the averaged strategy $\bar{\sigma}$. The output of RD is the averaged strategy $\bar{\sigma}$ after J iterations, which serves as an approximate NE of the restricted game.

Algorithm 10 Replicator Dynamics with a Game Model

Require: a game model $\hat{u}$. Define the initial strategy $\bar{\sigma}$ as the average of strategies in the restricted set $\Lambda = (s_1, \ldots, s_\tau)$. A learning rate dt.

1: **for** Iteration $j \in \{1, \ldots, J\}$ **do**
2: Evaluate $\hat{u}(I(s), \bar{\sigma})$ for all $s \in \Lambda$ and compute the average fitness $F \leftarrow \sum_{s \in \Lambda} \bar{\sigma}(s) \hat{u}(I(s), \bar{\sigma})$
3: **for** $s \in \Lambda$ **do**
4: Update $\bar{\sigma}(s)' \leftarrow \bar{\sigma}(s) + dt * \bar{\sigma}(s)[\hat{u}(I(s), \bar{\sigma}) - F]$
5: **end for**
6: $\bar{\sigma} \leftarrow \bar{\sigma}'$
7: **end for**
8: **Return** $\bar{\sigma}$

5.4 Experimental Results

5.4.1 Experimental Environments

5.4.1.1 Linear Quadratic MFGs

A linear quadratic MFG is defined on a discretized state space $Z = \{-L, \dots, L\}$. With an action space $\{-M, \dots, M\}$, a representative player can move up to M states to the left or to the right or stay still. The transition function is given by

$$z_{t+1} = z_t + (K(m_t - z_t) + a_t)\delta_t + c\epsilon_t\delta_t$$

and a reward function

$$r(z_t, a_t, \mu_t) = [-\frac{1}{2}|a_t|^2 + qa_t(m_t - z_t) - \frac{\kappa}{2}(m_t - z_t)^2]\delta_t$$

with terminal reward

$$r(z_T, a_T, \mu_T) = -\frac{C}{2}(m_T - z_T)^2$$

where K, q, κ, C are given non-negative constant and ϵ_t represents the randomness of the environment regularized by a constant c. δ_t measures the time lapse between two time steps and $m_t = \sum_{z \in Z} z\mu(z)$ is the expectation of states. Agents with this reward function are encouraged to follow the average state of the population.

5.4.1.2 Beach Bar Problems

I consider a simplified version of Santa Fe bar problem (Arthur 1994; Farago, Greenwald, and Hall 2002) and adopt the model by Perrin, Pérolat, et al. (2020). Specifically, a 1-D beach bar problem for a single-population MFG is a Markov Decision Process with $|Z|$ states on a one-dimensional torus ($Z = 0, \dots, |Z| - 1$). Without loss of generality, we designate a bar to the state 0. Positions of players are initial-

ized according to a uniform distribution. Players can keep still ($a_t = 0$) or move left ($a_t = -1$) or right ($a_t = 1$) at time step t on the torus to get as close as possible to the bar, while avoiding the crowded areas. The transition function is given by:

$$z_{t+1} = z_t + a_t + \epsilon_t$$

where a_t is an action of the representative player at time t and ϵ_t represents the randomness of the environment. The immediate reward function is given by:

$$r(z_t, a_t, \mu_t) = \tilde{r}(z_t) - \frac{|a_t|}{|Z|} - \log(\mu_t(z_t))$$

where $\tilde{r}(z_t)$ measures the closeness to the bar from state z_t, $-\frac{|a_t|}{|Z|}$ is the running cost and $-\log(\mu_t(z_t))$ represents the aversion of players to the crowded areas. Compared to the 1-D beach bar problem, the 2-D beach bar problem includes extra actions (i.e., up, down, left, right, stay) yielding a more complex environment.

5.4.2 Generalization of a Learned Model

For coarse coding, the training data are a combination of grid samples with a precision $K = 4$ and Dirichlet samples with $\alpha \in (0, 1]$ and a step of $\alpha = 0.05$. For each α, I sampled 150 data points from the corresponding Dirichlet distribution. The testing data are sampled from Dirichlet distributions with $\alpha \in (0.01, 2.01]$ and a step of $\alpha = 0.1$. For each α, I again sampled 150 test data points.

Table 5.1 reports the R^2 score of the model on testing data in the aforementioned three MFGs. R^2 score measures the total variation explained by the model. Mathematically,

$$R^2 = 1 - \frac{\sum_k (\hat{y}_k - y_k)^2}{\sum_k (y_k - \bar{y})^2},$$

where k is the index of samples, $\hat{y}$ is the estimate given by the model, and $\bar{y}$ is the

mean of targets y. R^2 score is between 0 and 1. if R^2 score is close to 1, the variation of the data is well-explained by the model. From Table 5.1, I observed a high R^2 score of the model in all of the three MFGs.

MFGs	R^2 **Score**
Linear Quadratic	0.99998 ± 0.00001
1-d Crowd Modeling	0.9724 ± 0.0024
2-d Crowd Modeling	0.9465 ± 0.0034

Table 5.1: Test results.

5.4.3 Approximating NE with a Game Model

5.4.3.1 FP with a Game Model

In Figure 5.3, I plot the regret curves of FP with the true utility function and the game model respectively in three MFGs. Since these MFGs support exact strategy evaluation (e.g., through dynamic programming), the regret curves can be exactly computed and hence no error bar is reported in the plots. In all cases, I observed that the regret curve generated with our game model can quickly coincide with the one using the true utility function, and both successfully converge to 0 (i.e., reaching an NE). This means the learned game model has good prediction accuracy on the equilibrium search path of FP and is able to support game-theoretic analysis of MFGs.

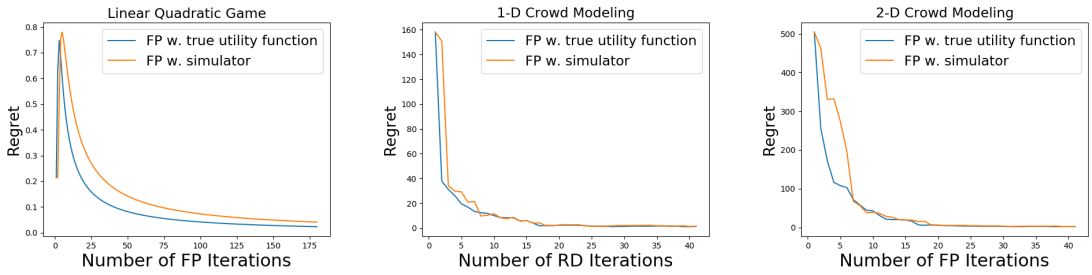


Figure 5.3: Regret curves with FP.

5.4.3.2 RD with a Game Model

In Figure 5.4, I plot the regret curves of RD in three MFGs. In the linear-quadratic game, I again observed that two regret curves quickly coincide and RD successfully converge to 0. However, in 1-D and 2-D crowd modeling games, I observed that the regret curves with the learned game model first quickly decrease as usual and then slightly diverge later.[2] This divergence is caused by the prediction error of the game model near the equilibrium. To improve the accuracy of the game model around the equilibrium, I re-sampled utilities $u(s, \mu^\sigma)$ in the neighborhood of the σ at the divergence point and fine-tune the game model. Then I continued RD with the fine-tuned game model. In the plots, I re-sampled at iteration 19 (indicated by the red vertical line) and observed that the divergence quickly disappears and RD converges to NE. This again shows that the learned model can support game-theoretic analysis of MFGs.

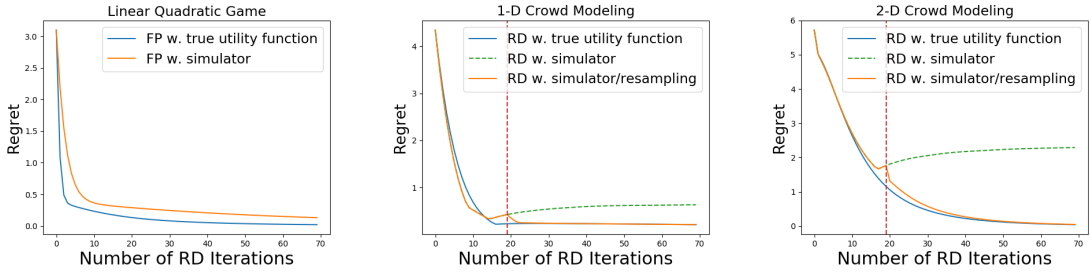


Figure 5.4: Regret curves with RD.

5.4.4 Mean Field Estimation

To verify that the learned game model can estimate the mean field (i.e., distributions) accurately, I plot the time-dependent distributions induced by the equilibrium strategies computed with the true utility function and the game model, respectively. Figure 5.5 and Figure 5.6 show the equilibrium distributions at $t \in [11, 16, 21, 26, 30]$

[2]Slightly divergence means that the error caused by the divergence is still much smaller than the scale of the utility, which may not affect further game-theoretical analysis dramatically.

in 1-D and 2-D crowd modeling games with the utility function and the game model. For visualization, states in the 1-D game are reshaped into 2 dimensions. By comparing plots on the top and at the bottom in Figure 5.5 and Figure 5.6 respectively, I observed that the distributions generated with the game model are almost indistinguishable by inspection from those generated with the true utility function. This accuracy can be quantified by Wasserstein distance, which as I report in Table 5.2 are all quite tiny (< 0.0005) though with a tendency to increase over time.

MFGs	$t = 11$	$t = 16$	$t = 21$	$t = 26$	$t = 30$
1-D Crowd	1.8	2.1	2.3	3.8	4.1
2-D Crowd	3.9	3.9	4.9	4.0	4.6

Table 5.2: Wasserstein distances ($\times 10^{-4}$) in the 1-D and 2-D crowd modeling games.

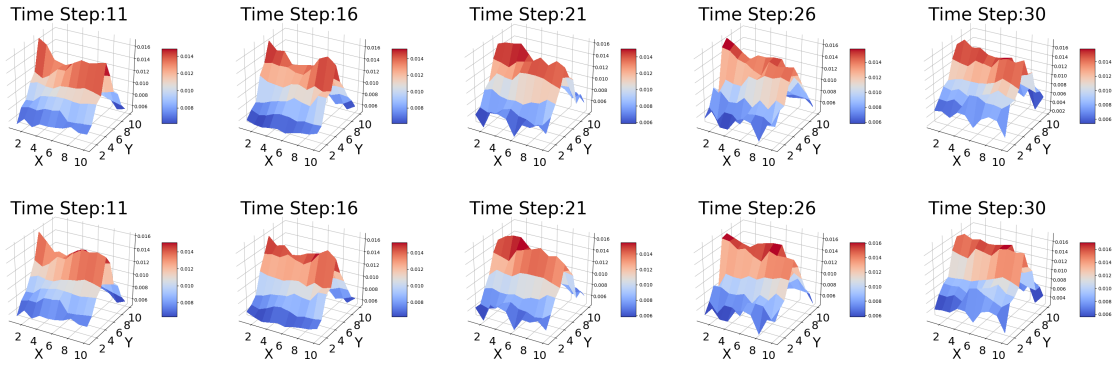


Figure 5.5: Distribution estimation in 1-D crowd modeling: (top) true utility function; (bottom) game model.

5.5 Conclusion and Discussion

I developed a game model learning approach for MFGs. I introduced a coarse coding scheme to handle the high-dimensional inputs in the utility function of MFGs and a data sample scheme for MFGs with dozens of strategies. I showed that the learning curves of FP and RD almost coincide with the true utility function and the learned utility simulation, respectively. This demonstrates that the learned game

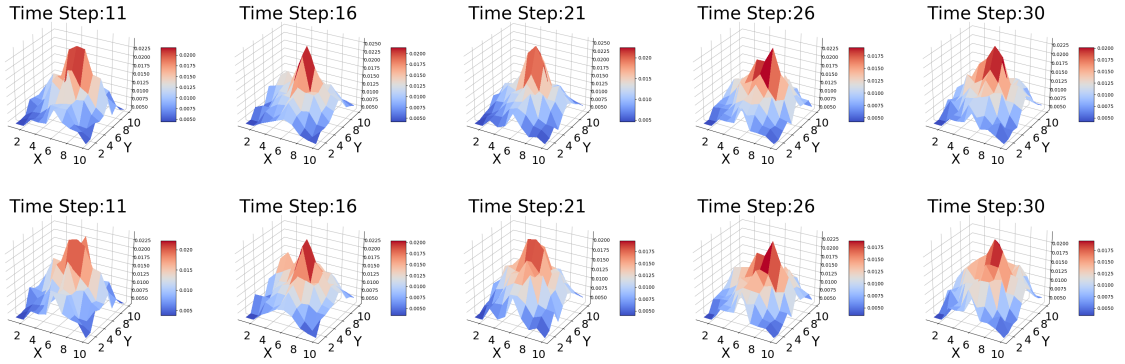


Figure 5.6: Distribution estimation in 2-D crowd modeling: (top) true utility function; (bottom) game model.

model can support game-theoretic analysis in MFGs. With an accurate game model, we can apply the EGTA framework for solving MFGs.

One limitation of the game model learning approach is that the learned model cannot be generalized to strategies outside the restricted strategy set. This is because each strategy is represented by an index. Although an index coding is beneficial to handle the time-dependent strategies, it does not provide any information about the similarity and difference between two strategies. Therefore, introducing a new strategy means that both extra sampling and model fine-tuning are needed.

CHAPTER VI

MFGs: An EGTA Framework

6.1 Introduction

Given a well-defined game model through game model learning (GML), the primary purpose of this chapter is to introduce an iterative framework for numerically solving MFGs with EGTA. I first introduce a primary method to demonstrate the basic implementation of the framework, and then I apply GML and RRD to the primary method, aiming at improving the sample efficiency of the primary method without sacrificing the overall learning performance. The primary method extends DO to MFGs, iteratively adding strategies based on best response to the equilibrium of the empirical MFG among strategies considered so far. I propose FP and RD as two subroutines for approximating an NE of the empirical MFG and demonstrate that both subroutines are effective for the empirical game analysis. Each subroutine is implemented with a query-based method, in which we query the utilities of different strategies through simulations as needed rather than maintaining an explicit payoff matrix as in typical EGTA methods. This tweak is caused by the non-linearity of the MFG utility function in the population distribution, which I highlight for MFGs. I test the primary iterative EGTA framework in MFGs with various configurations and demonstrate the improved learning performance of EGTA over directly applying FP (Perrin, Pérolat, et al. 2020) to MFGs.

Despite its effectiveness for solving MFGs, the primary method requires a large number of utility simulations due to the query-based method. I refer to this as low sample efficiency in EGTA. To improve the sample efficiency, I introduce a GML approach and apply RRD to our iterative EGTA framework. The GML approach is a form of regression that learns the utility function progressively over EGTA iterations. With a learned utility function, utility information can be predicted, thus reducing the number of queries to the simulator. RRD improves sample efficiency through reducing the number of iterations of subroutines in each EGTA iteration. In Chapter III, we have shown that properly regularizing the best response target (i.e., not best-responding to an exact equilibrium) will lead to an improved learning performance for EGTA and the regularization can be achieved by early stopping a subroutine within each EGTA iteration. For our purposes, early stopping a subroutine means less utility queries for each iteration of EGTA and improved sample efficiency if the overall learning performance would not decline. By introducing GML and RRD, I demonstrate a significant improvement on the sample efficiency (i.e., EGTA with GML and RRD only requires 1/6 of simulations needed by the primary method) over a variety of MFG configurations.

The theoretical results of this work are twofold. First, I prove the existence of NE in an empirical MFG with a restricted strategy space under a mild assumption, assuming the MFG is fully symmetric. Second, I prove that the iterative EGTA converges to NE of the full game if the best response target is NE across iterations and an exact best-response oracle is available.

6.2 Literature Review

There is a large prior literature on learning solutions of MFGs. Here I list few that are closely related to my approach. Elie et al. (2020) first studied the convergence of approximate discrete-time FP in MFGs. Perrin, Pérolat, et al. (2020) further

proved the convergence rate of continuous-time FP in MFGs and extended the study to MFGs with common noise. In their work, a FP algorithm for MFGs with finite time horizons is demonstrated effective in various environments. Perolat et al. (2022) criticized that FP is not scalable to MFGs with large state spaces due to the best response calculation and thus proposing Online Mirror Descent (OMD) as a solution. They empirically showed that OMD converges significantly faster than FP in MFGs with a large number of states. In the aforementioned algorithms, a fixed initial distribution of the population is required. Perrin, Laurière, Pérolat, Élie, et al. (2022) argued that a fixed initial distribution restricts the practical applications of MFGs since a real initial distribution could be different from the one used for training. They proposed a learning algorithm to learn a *Master Policy*, which takes the distribution of population as input and thus taking the initial distribution into consideration. They demonstrated the ability of generalization of the learned master policy.

In a recent work that is simultaneous with and independent of the primary method, Muller, Rowland, et al. (2021) adapted PSRO to MFGs and analyzed convergence properties with various solution concepts. Both theirs and the primary method are based on an iterative EGTA/PSRO framework for MFGs, highlighting the issue that the utility function for MFGs is not generally linear in the distribution. Theoretically, both works prove the existence of NE in the empirical game with restricted strategy set and the convergence of iterative EGTA/PSRO to NE. What is unique to their work is that they investigate the modifications for PSRO to converge to (coarse) correlated equilibria in MFGs as well as the theoretical counterparts. The contribution of the primary method focuses on practical techniques for the empirical game analysis and goes beyond their work in including full details of how online learning algorithms (i.e., FP and RD as subroutines) realize the framework of EGTA for MFGs. I also include more experimental results on performance of these methods for deriving approximate equilibria for MFGs. Since both theirs and the primary method rely on either black-

box optimization or online approaches for computing NE of intermediate empirical games, the issue of low sample efficiency exists. As a major contribution of this work, the GML and regularization successfully addresses this issue.

6.3 Iterative EGTA for MFGs

6.3.1 Framework

In iterative EGTA, the restricted strategy space is expanded incrementally based on analysis of intermediate game models. In finite games, a common approach of iterative strategy generation is presented by the well-known DO algorithm, which adds strategies that best respond to a current equilibrium. I extend DO to MFGs and provide a learning framework for MP-MFGs in Algorithm 11.

In Algorithm 11, for each population i, we initialize the representative player with policy $s_{i,0}$ and an initial distribution $\mu_{i,0}$. At each iteration τ, a best response target σ^e and its induced distribution μ^e are computed through the analysis of the empirical game. Here the tools for the empirical game analysis are determined by the solution concepts and convergence properties one pursues. Then the representative player of each population i finds an exact/approximate best response strategy $s_{i,\tau}$ to the distribution μ^e and adds it to the empirical game. This process repeats for a fixed number iterations, typically set to be large enough so that no beneficial deviation strategy could be found.

6.3.2 Analyzing an Empirical MFG

In EGTA, analyzing an intermediate empirical game is crucial for generating effective strategies.[1] Although an MFG can be viewed as an interaction between two parties (i.e., a representative player and the population), the empirical game analysis

[1]For discussion simplicity, I assume that players are fully symmetric (i.e., $N_p = 1$). So the population index i is dropped when the context is clear.

Algorithm 11 Iterative EGTA for MP-MPG

Input: for each population i, an initial policy $s_{i,0}$ and an initial distribution $\mu_{i,0}$

1: **for** $\tau \in \{1, \ldots, \mathcal{T}\}$ **do**
2: Compute (σ^e, μ^e) by empirical game analysis
3: **for** $i \in \{1, \ldots, N_p\}$ **do**
4: Find a best response policy $s_{i,\tau}$ to empirical equilibrium distribution μ^e
5: Add $s_{i,\tau}$ to the strategy set of population i
6: **end for**
7: **end for**
8: **Return** (σ^e, μ^e)

for MFGs is more than constructing an explicit matrix and then applying a game-solver for two parties. Proposition 3 shows that MFGs can only be solved with an explicit payoff matrix under certain restrictive conditions.

Assume an empirical MFG with an explicit payoff matrix representation, shown in Table 6.1. In the empirical game, there are 4 strategies $\Lambda = \{s_0, s_1, s_2, s_3\}$ in the restricted strategy set. Since the game is fully symmetric, the population would act following distributions $\mu = \{\mu_0, \mu_1, \mu_2, \mu_3\}$ induced by corresponding strategies in Λ. In the payoff matrix, the value in entry $(j, k), j, k \in \{0, 1, 2, 3\}$ is $u(s_j, \mu_k)$, where μ_k is the distribution induced by s_k.

	s_0	s_1	s_2	s_3
s_0	$u(s_0, \mu_0)$	$u(s_0, \mu_1)$	$u(s_0, \mu_2)$	$u(s_0, \mu_3)$
s_1	$u(s_1, \mu_0)$	$u(s_1, \mu_1)$	$u(s_1, \mu_2)$	$u(s_1, \mu_3)$
s_2	$u(s_2, \mu_0)$	$u(s_2, \mu_1)$	$u(s_2, \mu_2)$	$u(s_2, \mu_3)$
s_3	$u(s_3, \mu_0)$	$u(s_3, \mu_1)$	$u(s_3, \mu_2)$	$u(s_3, \mu_3)$

Table 6.1: Single-population MFG payoff matrix.

Proposition 3. *The NE of the aforementioned payoff matrix will not generally be an NE of the mean field empirical game unless the utility function is linear in μ.*

Proof. Assume σ is an NE computed from the payoff matrix. According to the

definition of NE, we have

$$\sum_{j\in[|\Lambda|]}\sum_{k\in[|\Lambda|]}\sigma(s_j)\sigma(s_k)u(s_j,\mu_k)\geq\sum_{k\in[|\Lambda|]}\sigma(s_k)u(s',\mu_k),\forall s'\in\Lambda. \tag{6.1}$$

According to the definition of NE σ^* in MFGs, we have

$$\sum_{j\in[|\Lambda|]}\sigma^*(s_j)u(s_j,\mu^{\sigma^*})\geq u(s',\mu^{\sigma^*}),\forall s'\in\Lambda \tag{6.2}$$

where μ^{σ^*} is induced by σ^*.

By comparing inequalities 6.1 and 6.2, to make the NE σ an MFG NE σ^*, the following condition should hold

$$\sum_{k\in[|\Lambda|]}\sigma(s_k)u(s_j,\mu_k)=u(s_j,\mu^{\sigma}),\forall s_j\in\Lambda$$

indicating the requirement of linearity in μ at least at the equilibrium point.

□

Since the utility function in MFGs will not generally be linear in μ, analysis of an empirical game based on an explicit payoff matrix is impractical. Instead, we can rely on query-based methods. I propose FP and RD as two subroutines for solving empirical games and query utility information through simulation as needed.

6.3.2.1 FP as a Subroutine

FP for MFGs has been studied by Elie et al. (2020) and Perrin, Pérolat, et al. (2020). I adapt it to analyzing an empirical game with a restricted set of strategies. In Algorithm 12, I demonstrate how to apply FP to empirical games. Specifically, starting from the uniform strategy $\bar{\sigma}$ over the strategies in the restricted strategy set and its induced distribution $\bar{\mu}$, at each iteration $j\in[1,J]$, the representative player

of a population i finds a best response strategy $s_{i,j} \in \Lambda_i$ against the populations $\bar{\mu}$. The probability in the mixed strategy of playing $s_{i,j}$ is updated by the frequency of $s_{i,j}$ appearing as a best response, that is, the number of being a best response $n_{s_{i,j}}$ divided by the total count up to the j^{th} iteration. Mathematically, for all $i \in [N_p]$ and $k \in [|\Lambda_i|]$, the number of $s_{i,k}$ being a best response $n_{s_{i,k}}$ is incremented by 1 if it is the best response at the current iteration (i.e., $s_{i,k} = s_{i,j}$) and remains the same otherwise.

$$n_{s_{i,k}} = \begin{cases} n_{s_{i,k}} + 1 & s_{i,k} = s_{i,j} \\ n_{s_{i,k}} & s_{i,k} \neq s_{i,j} \end{cases}$$

Then we update the corresponding probability in the mixed strategy by

$$\bar{\sigma}_i(s_{i,k}) = \frac{n_{s_{i,k}}}{|\Lambda_i| + \sum_{k \in [|\Lambda_i|]} n_{s_{i,k}}}$$

Algorithm 12 Fictitious Play for Empirical MFGs

Input: An empirical game. Define initial policy $\bar{\sigma}_i$ as the average of strategies in the restricted set $\Lambda_i = (s_{i,1}, \ldots, s_{i,\tau})$ of population i and $\bar{\mu}$ is induced by $\bar{\sigma}$

1: **for** $j \in \{1, \ldots, J\}$ **do**
2: **for** $i \in \{1, \ldots, N_p\}$ **do**
3: Find a best response strategy $s_{i,j} \in \Lambda_i$ to $\bar{\mu}$
4: **end for**
5: Update $\bar{\sigma}_i$ and induce $\bar{\mu}_i$, for all $i \in [N_p]$
6: **end for**
7: **Return** $(\bar{\sigma}, \bar{\mu})$

To induce a corresponding distribution $\bar{\mu}_i$, I first build a weighted average strategy that is equivalent to the mixed strategy $\bar{\sigma}_i$. Specifically, consider a mixed strategy $\sigma \in \Delta(\Lambda)$ defined on the empirical game with a restricted strategy set Λ. An equivalent

strategy $\bar{s}$ is defined as, for each population i,

$$\bar{s}_{i,t}(a \mid x) = \frac{\sum_{k=1}^{|\Lambda_i|} \sigma_i(s_{i,k,t}) \mu_t^{s_{i,k,t}}(x) s_{i,k,t}(a \mid x)}{\sum_{k=1}^{|\Lambda_i|} \sigma_i(s_{i,k,t}) \mu_t^{s_{i,k,t}}(x)}, \forall t \in [0, T-1]$$

where $s_{i,k,t}$ is the k^{th} strategy of population i at time step t. Then the induced distribution is computed through Equation 1.4 or estimated through various approaches (e.g., empirical density estimation (Perrin, Pérolat, et al. 2020) or generative models (Perrin, Laurière, Pérolat, Geist, et al. 2021)).

The empirical game analysis terminates until certain stopping criterion is satisfied (e.g., reaching a fixed number of iterations). Note that FP will not generally converge to an NE (or even CE) in MFGs with multiple populations and here I use FP practically for illustration purpose.

6.3.2.2 RD as a Subroutine

RD describes an evolving trajectory of mixed profiles and is commonly employed as a heuristic equilibrium search algorithm in finite games. In Algorithm 13, I adapt RD to our MFG model and propose it as a practical subroutine for empirical game analysis. Similar to RD in finite games, at each iteration the update of a strategy's probability in population i is in proportion to the deviation payoff of that strategy from the average fitness, weighted by its probability from the previous iteration and a learning rate. Theoretically, RD has not been proved for convergence as FP. However, I show that RD exhibits empirical convergence with an even more stable learning manner than FP in mys experiments.

Algorithm 13 Replicator Dynamics for Empirical MFGs

Input: an empirical game. Define initial policy $\bar{\sigma}_i$ as the average of strategies in the restricted set $\Lambda_i = (s_{i,1}, \ldots, s_{i,\tau})$ of population i and $\bar{\mu}$ is induced by $\bar{\sigma}$ A learning rate dt.

for Iteration $j \in \{1, \ldots, J\}$ **do**
  for $i \in \{1, \ldots, N_p\}$ **do**
    Compute the average fitness $F_i = u_i(\bar{\sigma}_i, \bar{\mu}^{\bar{\sigma}})$
    for $s_i \in \Lambda_i$ **do**
      Evaluate $u_i(s_i, \bar{\mu}^{\bar{\sigma}})$
      Update $\bar{\sigma}_i(s)' = \bar{\sigma}_i(s) + dt * \bar{\sigma}_i(s)[u_i(s_i, \bar{\mu}^{\bar{\sigma}}) - F_i]$
    end for
  end for
  $\bar{\sigma} = \bar{\sigma}'$
  Induce new distribution $\bar{\mu}$ based on updated $\bar{\sigma}$.
end for
Return $(\bar{\sigma}, \bar{\mu})$

6.3.3 Best Response Oracles

In Algorithm 11, one key step is to find a best-response strategy to a distribution. For games with a moderate size of the state and action spaces, best response can be computed through tabular RL or backward dynamic programming. For example, a strategy can be represented by a Q-value table (i.e., a tabular strategy) and then Q-learning for MFGs can be applied.[2]

However, the scalability of tabular approach is problematic for two reasons. One is that the optimal strategy is not necessarily time-homogeneous in MFGs, meaning that for each time step an optimal Q-value table should be stored. This causes a memory burden especially when the time horizon T is large. Another issue is that tabular approaches are not scalable with the size of the state and action spaces.

For games with large state and action spaces, deep RL has been employed in prior work to find an approximate best-response strategy. To handle the time-heterogeneity of optimal strategies in MFGs, one possible approach is to encode the time information as part of the input to the function approximator in deep RL, aiming to learn the

[2]For detailed algorithms such as Q-learning and backward induction, please refer to the Appendix by Perrin, Pérolat, et al. (2020).

temporal structure of the optimal strategy, without keeping an optimal strategy for each time step.

6.4 Convergence to NE

I analyze the theoretical properties of iterative EGTA for MFGs from two aspects: the existence of NE in mean field empirical games and the convergence of iterative ETGA to NE of the full game. Theoretical results for the existence of NE in general MFGs have been widely studies in prior work. For discrete-time MFGs with finite state and action spaces, Gomes, Mohr, and Souza (2010) proved the existence of NE in general case and Doncel, Gast, and Gaujal (2019) further extended the existence results to scenarios where transition functions also depend on the population. For an empirical game model, since it restricts the set of strategies available to players, it is necessary to re-examine the existence result for NE. Here I assume that players are fully symmetric and prove that the NE exists in an empirical game under one mild assumption.

Assumption 1. *The utility function $u(s, \mu)$ is continuous in the distribution μ.*

Theorem 2. *Under Assumption 1, for games with finite state and action spaces, there exists a Nash equilibrium in the empirical game.*

Proof. To prove the existence of an NE in the empirical game with strategy sets Λ using Kakutani's fixed point theorem (Kakutani 1941), we need to show

1. The empirical strategy space $\Delta(\Lambda)$ is non-empty, closed and bounded (compactness by Heine-Borel Theorem (Borel 1895)) and a convex subset of certain Euclidean space.

2. The best response correspondence br is a set-valued function such that br has a closed graph and $br(\cdot)$ is non-empty and convex.

Then according to Kakutani's fixed point theorem, an NE exists in an empirical game.

For the first condition, since Λ is non-empty, then $\Delta(\Lambda)$ is just the simplex of Λ so it is non-empty. For the compactness, we need to prove $\Delta(\Lambda)$ is closed and bounded. First note that $|\Lambda|$ is finite since there are finite number of strategies in the empirical game. Since $\Delta(\Lambda)$ is the intersection of the closed sets $\mathcal{R}_+^{|\Lambda|}$ and $\{\lambda \in \mathcal{R}^{|\Lambda|} : \sum_{j\in[|\Lambda|]} \lambda_j = 1\}$, $\Delta(\Lambda)$ is closed. Since $\Delta(\Lambda)$ is a subset of $[0,1]^{|\Lambda|}$, it is bounded. Then $\Delta(\Lambda)$ is compact. For the convexity, consider any strategies s and s', and coefficient $\lambda \in (0,1)$, according to the definition of a simplex, $\lambda s + (1-\lambda) \in \Delta(\Lambda)$. So $\Delta(\Lambda)$ is a convex set. We now complete the verification of the first condition.

For the second condition, given a strategy $\sigma \in \Delta(\Lambda)$, define a best-response correspondence to σ as

$$\begin{aligned} br(\sigma) &= \operatorname*{argmax}_{s'\in\Lambda} u(s', \mu^\sigma) \\ &= \operatorname*{argmax}_{\sigma'\in\Delta(\Lambda)} u(\sigma', \mu^\sigma) \\ &= \operatorname*{argmax}_{\sigma'\in\Delta(\Lambda)} \sum_{s\in\Lambda} \sigma'(s) u(s, \mu^\sigma) \end{aligned}$$

Due to the compactness of $\Delta(\Lambda)$ and the continuity assumption of u, the best-response correspondence $br(\sigma)$ is non-empty. To show it is convex, consider two strategies $s_1, s_2 \in br(\sigma))$ associated with coefficients c_1 and c_2 such that $c_1, c_2 \geq 0$ and $c_1 + c_2 = 1$. Since all optima share the same utility

$$u(s_1, \mu^\sigma) = u(s_2, \mu^\sigma)$$

and the definition of the expected utility

$$u(c_1 s_1 + c_2 s_2, \mu^\sigma) = c_1 u(s_1, \mu^\sigma) + c_2 u(s_2, \mu^\sigma)$$

We have $c_1 s_1 + c_2 s_2 \in br(\sigma)$ and then $br(\sigma)$ is convex. Note that $br(\sigma)$ is a set-valued function since there could be multiple strategies maximizing the value function, which constitutes a power set of Λ.

Next, I claim that $Gr(br) := \{(\sigma, br(\sigma)) : \sigma \in \Delta(\Lambda), br(\sigma) \in \Delta(\Lambda)\}$ is a closed graph. By the Berge's maximum theorem (Berge 1997) and the continuity assumption, the set-valued function br is upper-hemicontinous. Since $br(\sigma)$ is closed for all $\sigma \in \Delta(\Lambda)$ and $\Delta(\Lambda)$ is a metrizable space, $Gr(br)$ is a closed graph. This completes the proof of the second condition. With condition 1 and 2, according to Kakutani's fixed point theorem, an NE exists in an empirical game.

□

According to Theorem 2, an empirical NE exists and hence it can be obtained through some theoretically proven equilibrium search subroutines (e.g., FP). The remaining problem is whether iterative EGTA converges to the NE of the full game.

Theorem 3. *For games with finite state and action spaces, suppose the empirical NE is the best response target at each iteration of iterative EGTA and an exact best response oracle is available, then the empirical NE converges to the NE of the full game.*

Proof. Suppose σ^* is an empirical NE associated with a population distribution μ^* such that

$$\max_{s \in \Lambda} u(s, \mu^*) - u(\sigma^*, \mu^*) = 0$$

Suppose there is no beneficial deviation can be found at certain iteration, indicating

$$\max_{s \in S} u(s, \mu^*) - u(\sigma^*, \mu^*) = 0$$

Then σ^* is an NE of the full game. Since the strategy space is finite, σ^* is always reachable with the worst case where all strategies are included in the empirical game.

□

6.5 Game Model Learning and Regularization for Improved Sample Efficiency

One crucial factor that affects the practicality of the primary iterative EGTA method is sample efficiency. Sample efficiency here refers to the number of simulations needed for estimating the utilities in the equilibrium search at each iteration of EGTA. In finite games, the utilities of any mixed strategy profile can be computed by taking an expectation of utilities across pure strategy profiles in the support, so utility information of pure strategy profiles can be re-used.

Unlike finite games, Proposition 3 shows that the utility $u(s, \mu^\sigma)$ of playing strategy s against population distributions μ^σ needs to be evaluated for different μ^σ and generally cannot be computed by first computing $u(s, \mu^{s'})$ for each pure strategy s' in the support of σ and then taking an expectation as in finite games. Moreover, since it is very unlikely to encounter a same distribution μ^σ across EGTA iterations, it is also not useful to store the corresponding utility information. Due to these characteristics, instead of storing utility information and re-using them, the iterative EGTA approaches up to this point (including the primary method and the version by Muller, Rowland, et al. (2021)) need to compute or simulate $u(s, \mu^\sigma)$ whenever μ^σ changes in the equilibrium computation (i.e., the query-based implementation), which results in a low sample efficiency. To improve the sample efficiency, I introduce GML and apply RRD to the primary method. For discussion purposes in this section, I assume that MFGs are single-population (i.e., $N = 1$) and my approach can be readily extended to MFGs with multiple populations.

6.5.1 Game Model Learning

My GML approach is a form of regression that learns the utility function based on utility information collected over previous EGTA iterations. This approach is based on GML discussed in Chapter V and extended to the iterative setting. With a game model (i.e., a learned utility function), sample efficiency can be improved by querying the game model rather than running simulations as long as the game model is able to provide high-fidelity predictions on these queries. In the following discussion, I first discuss how a game model fits in the EGTA framework and then elaborate the method for learning a game model given utility samples.

6.5.1.1 Applying GML to EGTA

To implement GML in iterative EGTA, one key step is to periodically update the game model based on collected utilities and apply the game model to running a subroutine for equilibrium computation of the current empirical game. I select RD as a subroutine for illustration purposes.

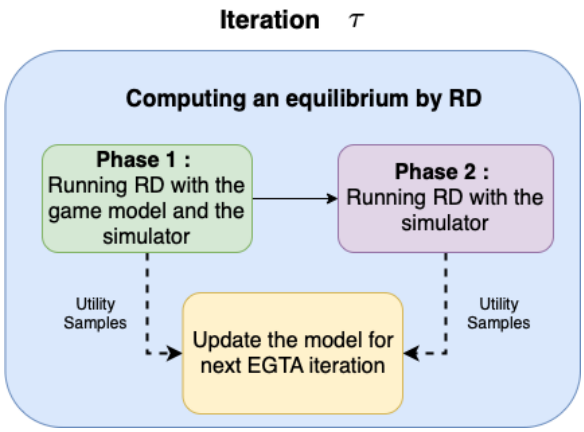


Figure 6.1: RD with a game model.

Denote the current EGTA iteration as iteration τ. Our object is to approximate the NE of the current empirical game with the strategy set Λ_τ, using RD and the game model $\hat{u}_{\tau-1}$ learned based on $\Lambda_{\tau-1}$ (i.e., the game model learned from previous

iterations). Since the model $\hat{u}_{\tau-1}(s, \mu^\sigma)$ only contains utility information of $s \in \Lambda_{\tau-1}$ and $\sigma \in \Delta\Lambda_{\tau-1}$ from previous iterations, while RD requires utilities $u(s, \mu^\sigma), \forall s \in \Lambda_\tau, \forall \sigma \in \Delta\Lambda_\tau$ for the current iteration, we cannot directly apply the model $\hat{u}_{\tau-1}$ due to the lack of information of s_τ. To handle this issue, I interleave utility approximation with simulations for different strategies.

In particular, for $s \in \Lambda_{\tau-1}$ and $\sigma \in \Delta\Lambda_{\tau-1}$, I directly apply the model to predict $u(s, \mu^\sigma)$. For $s \in \Lambda_{\tau-1}$ and $\sigma \in \Delta\Lambda_\tau$, we first project σ onto $\Delta\Lambda_{\tau-1}$ by a projection operator $P_{\tau-1}(\sigma) = \text{argmin}_{\sigma' \in \Delta\Lambda_{\tau-1}} ||\sigma' - \sigma||_2$ and approximate $u(s, \mu^\sigma)$ by $\hat{u}_{\tau-1}(s, \mu^{P(\sigma)})$. The assumption here is that if $P(\sigma)$ is close to σ, then $u_{\tau-1}(s, \mu^\sigma)$ is close to $u_{\tau-1}(s, \mu^{P(\sigma)})$ and the model is valid for estimating $u(s, \mu^\sigma)$. The scenario in the assumption often holds at the start of running RD when RD is initialized with the equilibrium strategy from last iteration. Since the update of strategy in RD is controlled by a small step size, $P(\sigma)$ will be close to σ within first few RD iterations. For $s = s_\tau$, I query the simulator (e.g., a noiseless simulator) to obtain the exact utility $u(s, \mu^\sigma)$ because the game model does not contain its utility information. I refer to this procedure as the first phase.

The number of RD iterations for the first phase is determined by the quality of utility estimations given by the projection. To measure this quality, I set a threshold for the L-2 distance between σ and its projection $P(\sigma)$. If the distance goes beyond the threshold, it means that the game model with projection becomes less accurate on the utility predictions. So we should stop using the model and switch to the simulator (i.e., the second phase).

In the second phase, I directly run RD with the simulator for all $s \in \Lambda_\tau$ and $\sigma \in \Delta\Lambda_\tau$ for a fixed number of iterations. At the end the second phase, utilities collected from the simulator at both phases are used to fine-tune the current game model. Note that the sampling of these utilities is guided by RD to avoid sampling the whole strategy space, where the latter will hurt our motivation of improving sample

efficiency. The overall framework is depicted in Figure 6.1.

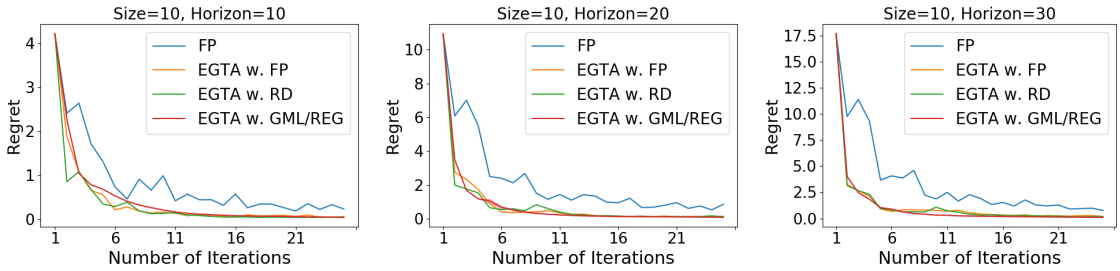
6.5.1.2 Learning Utility Functions

To learn a game model, I apply the *coarse coding* scheme described in Chapter V. Based on coarse coding, a utility data point is constructed to include an index of a pure strategy $I(s)$, a mixed strategy σ, and a utility target $u(s, \mu^\sigma)$. Since the object is to predict the true utility $u(s, \mu^\sigma)$ by $\hat{u}(I(s), \sigma)$, I fine-tune $\hat{u}_{\tau-1}$ by minimizing the mean square error $E[u(s, \mu^\sigma) - \hat{u}(I(s), \sigma))^2]$.

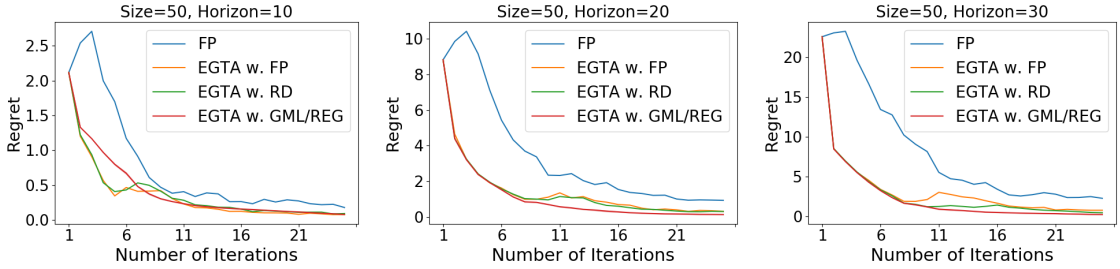
6.5.2 Regularization by RRD

The regularization method improves sample efficiency through reducing the number of iterations of RD in each EGTA iteration. In the iterative EGTA approach up to this point, the number of iterations for the subroutines (e.g., RD and FP) is set to be large enough so that they can approximately converge to the current equilibrium of the empirical game. This mimics DO, in which a best response to the current equilibrium is computed at each iteration.

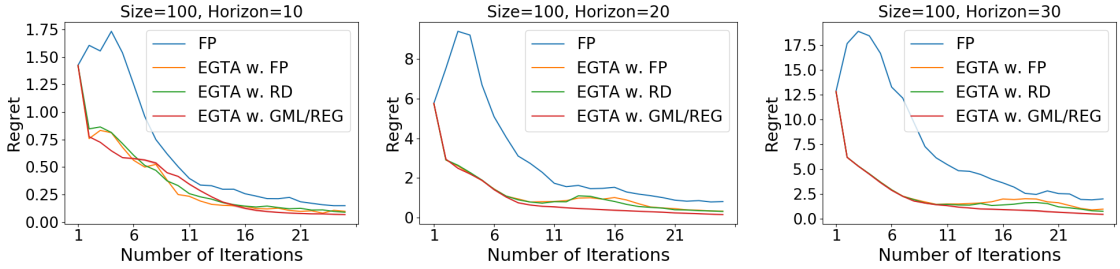
In Chapter III, we have shown that properly regularizing the best response target (i.e., not best-responding to an exact equilibrium) will lead to improved learning performance for EGTA. Regularization can be achieved by early stopping RD when the regret of the current profile with respect to the empirical game exceeds a regret threshold. For our purposes, early stopping of RD means less utility queries from the simulator and improved sample efficiency if the overall learning performance would not decline. I apply this approach to the EGTA framework by replacing the regret threshold with a fixed number of RD iterations since the prediction error given by the game model could affect the regret estimation for reaching a threshold.



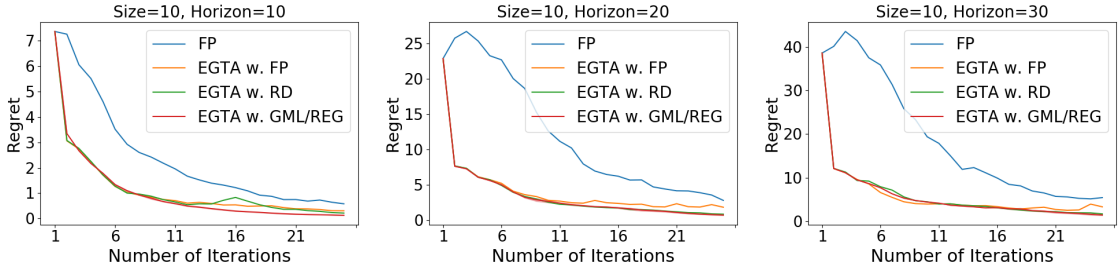
(a) 1-D(10, 10) (b) 1-D(10, 20) (c) 1-D(10, 30)



(d) 1-D(50, 10) (e) 1-D(50, 20) (f) 1-D(50, 30)



(g) 1-D(100, 10) (h) 1-D(100, 20) (i) 1-D(100, 30)



(j) 2-D(100, 10) (k) 2-D(100, 20) (l) 2-D(100, 30)

Figure 6.2: Experimental results of 1-D and 2-D beach bar problems.

6.5.3 Algorithms

I show the full EGTA framework with GML and regularization for single-population MFGs in Algorithm 14 and Algorithm 15. Compared to the primary method (Algorithm 11), the main differences are the introduction of a game model $\hat{u}$ in Algorithm 14 and how the model $\hat{u}$ is updated and then applied to RD in Algorithm 15 (discussed in Section 6.5.1). Note that, in Algorithm 15, RD is initialized with the equilibrium σ^e from last EGTA iteration, perturbed by a function δ to guarantee a full support. This makes sure that every strategy can be played with non-zero probability and hence can be updated by RD. Regularization is achieved by controlling the maximal number of RD iteration J.

Algorithm 14 Iterative EGTA with GML and RRD

Require: an initial strategy $\Lambda_0 = \{s_0\}$ and an initial distribution μ_0. A neural network $\hat{u}$.

$\sigma^e \leftarrow s_0$
Initialize μ^e by Eq. 1.4 using s_0
for EGTA iteration $\tau \in \{1, \dots, \mathcal{T}\}$ **do**
    Compute a best response strategy s_τ to the empirical equilibrium distribution μ^e
    Add s_τ to the strategy set of population i: $\Lambda_\tau \leftarrow \Lambda_{\tau-1} \bigcup s_\tau$
    Compute $\sigma^e, \mu^e, \hat{u} \leftarrow$ a subroutine $\Psi(\mathcal{G}_{S\downarrow\Lambda_\tau}, \hat{u}, \sigma^e, \mu^e)$
end for
Return (σ^e, μ^e)

6.6 Experimental Results

6.6.1 The 1-D Beach Bar

I test the performance of iterative EGTA in the 1-D beach bar problem (see description in Chapter V) with various configurations, which are determined by the Cartesian product of $|Z| \in \{10, 50, 100\}$ and $T \in \{10, 20, 30\}$, denoted by 1-D($|Z|, T$). In Figure 6.2a-6.2i, I plot the regret curves of iterative EGTA (with FP and RD as subroutines respectively) against directly applying FP to MFGs (Perrin, Pérolat, et

Algorithm 15 RRD as a Subroutine Ψ

Require: an empirical game $\mathcal{G}_{S\downarrow\Lambda_\tau}$. A learned utility simulator $\hat{u}$. Equilibrium strategy σ^e, and distribution μ^e.
Parameters: A distance threshold γ. A maximal number of iterations for applying the model M. A learning rate dt.

Initialize a strategy $\bar{\sigma} \leftarrow \delta(\sigma^e)$
for RD iteration $j \in \{1, \ldots, J\}$ **do**
  $\bar{\sigma}^p \leftarrow P_{\tau-1}(\bar{\sigma})$
  if $||\bar{\sigma}, \bar{\sigma}^p||_2 < \gamma$ and $j < M$ **then**
    Approximate $u(s, \mu^{\bar{\sigma}})$ by $\hat{u}(I(s), \bar{\sigma}^p), \forall s \in \Lambda_{\tau-1}$
    Simulate $u(s_\tau, \mu^{\bar{\sigma}})$
  else
    Simulate $u(s, \mu^{\bar{\sigma}}), \forall s \in \Lambda_\tau$
  end if
  Save new data points $I(s)$, $\bar{\sigma}$, and $u(s, \mu^{\bar{\sigma}})$
  Compute fitness $F = \sum_{s\in\Lambda} \bar{\sigma}(s) u(s, \mu^{\bar{\sigma}})$
  for $s \in \Lambda_\tau$ **do**
    $\bar{\sigma}(s) \leftarrow \bar{\sigma}(s) + dt * \bar{\sigma}(s)[u(s, \mu^{\bar{\sigma}}) - F]$
  end for
end for
Fine-tune $\hat{u}$ with all new data points
Compute the induced distributions $\mu^{\bar{\sigma}}$ by Eq. 1.4 using $\bar{\sigma}$
Return $\bar{\sigma}$, $\mu^{\bar{\sigma}}$, and $\hat{u}$

al. 2020), where x-axis being the EGTA iterations. Since the game size supports exact best response calculation and exact strategy evaluation given a fixed initialization and parameters of the randomness of the environment (i.e., the simulator is noiseless), the regret curves can be exactly computed and hence no error bar is reported in the plots.

From Figure 6.2a-6.2i, I observed that the performance of our primary EGTA method (orange and green curves) dominates FP in all instances. Moreover, as the instance becomes complex (i.e., with more states and longer horizon), the performance gap between our primary method and FP becomes more apparent. For the subroutine selection, I observed that RD in some cases exhibits a more stable learning manner (e.g., Fig 6.2f) than FP while in most cases their performances are almost indistinguishable. For EGTA with GML and regularization (abbr. REG in the plots) (red curve), I observed that its performance also almost coincides with our primary

method, but becomes even most stable as time horizon increases due to regularization. Thanks to GML and regularization, we obtain this performance with only 1/6 of the total utility queries compared to our primary method, which demonstrates a significant improvement on the sample efficiency. I show the number of simulations needed across EGTA iterations in Figure 6.3.

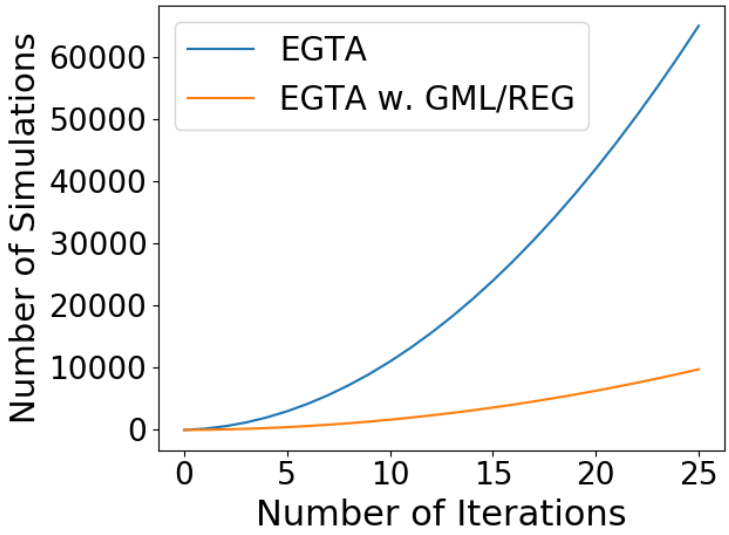


Figure 6.3: The number of utility samples across EGTA iterations.

6.6.2 The 2-D Beach Bar

I test the performance of our primary EGTA method in the 2-D beach bar problem with fixed $|Z| = 100$ and different time horizon $T \in \{10, 20, 30\}$, denoted by 2-D($|Z|$, T). Compared to the 1-D beach bar problem, the 2-D beach bar problem includes extra actions (i.e., up, down, left, right, stay) yielding a more complex environment. From Figure 6.2j-6.2l, I observed the same phenomenon as in the 1-D problem, that is, iterative EGTA dominates FP and as the game instance becomes larger, the advantage of iterative EGTA becomes apparent. EGTA with GML and regularization again exhibits similar performance compared to the primary method but only requires 1/6 of the utility queries.

6.6.3 Multi-Population Chasing

For MP-MFGs, I test the performance of iterative EGTA in a three-population chasing problem (Perolat et al. 2022), which closely relates to the game Hens-Foxes-Snakes, where hens, snakes and foxes are chasing cyclically. The reward structure of this game is shown in table 6.6.3, denoted as R.

	Hens	Snakes	Foxes
Hens	(0, 0)	(-1, 1)	(1, -1)
Snakes	(1, -1)	(0, 0)	(-1, 1)
Foxes	(-1, 1)	(1, -1)	(0, 0)

The immediate reward function of population $i \in [N_p]$ is defined as

$$r_i(z, a, \mu) = -\log(\mu_i(z)) + \sum_{j \neq i} \mu_j(z) R(i, j)$$

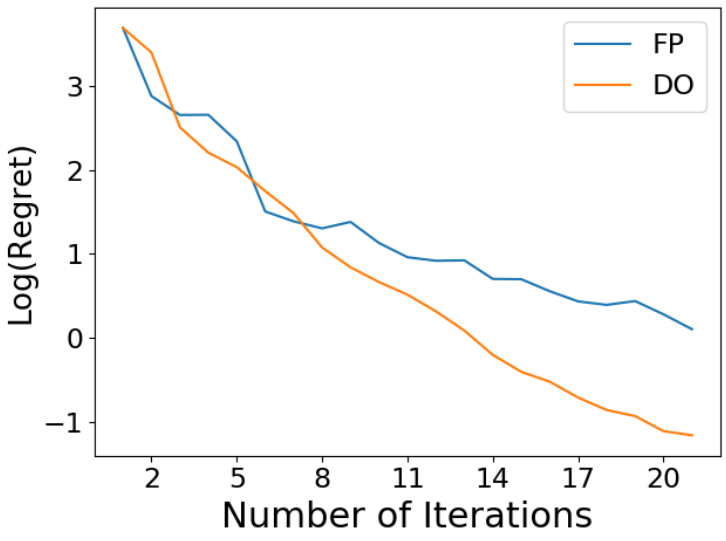


Figure 6.4: Regret curves of FP in multi-population chasing.

In Figure 6.4, I observed that at the early stage of learning, both FP and iterative EGTA can learn the game quickly and improve the stability of strategies. Iterative EGTA keeps its momentum as learning proceeds while the learning curve of FP

becomes flatten over time. I conjecture that the reason for the two learning curves of EGTA and FP close to each other is the best response target given by FP is not as effective as the one in single-population setting since FP will not generally converge to NE in MP-MFGs.

6.7 Conclusion and Discussion

I proposed an iterative EGTA framework for computing NE in MFGs and a sample-efficient version by combining GML and regularization. I demonstrated the efficacy of our approaches in various MFGs. Theoretically, I proved the existence of NE in empirical MFGs and the convergence of the iterative EGTA framework.

6.7.1 Complexity of the Empirical Game Analysis

In my experiments, I measured the regret with respect to EGTA iterations and show superior performance of iterative EGTA against FP. However, in terms of running time, I noticed that directly applying FP or OMD to the MFGs in OpenSpiel results in faster convergence since in EGTA the analysis of the empirical game (i.e., estimate the payoffs of mixed strategies, update the mixed strategies and compute the corresponding distributions) turns out to be computationally expensive. This is because without an explicit payoff matrix, the same strategy or similar ones could be evaluated repeatedly. Besides, computing the induced distribution for every FP update also could be costly. In a word, I believe that the acceleration of the empirical game analysis is a crucial step for the application of iterative EGTA for MFGs, which is a potential research direction in the future.

In the plots, the reason for measuring the regret with respect to EGTA iterations is based on one assumption in EGTA, that is, it is common that the cost on best response calculation becomes dominant than the empirical game analysis in complex games especially when deep reinforcement learning is deployed. Based on this assumption,

we prefer to build an effective game model with a minimal number of iterations and thus evaluating the performance with respect to the EGTA iterations. For the same reason, I did not plot the regret curves of OMD (Perolat et al. 2022) with respect to PSRO iterations since I found that the definition of one iteration in OMD is different from that in PSRO, which could lead to improper comparison.

For MP-MFGs, as I observed in the multi-population chasing experiments, since FP will not generally converge to an NE in the multi-population setting, using FP for the empirical game analysis in EGTA may affect the learning performance. Therefore, selecting an effective best response target is also one future research direction.

6.7.2 Re-Evaluating Strategies in Finite Games

A key motivation for using MFGs is that the MFG model dramatically simplifies game learning compared to directly solving the corresponding finite game. Meanwhile, the solution of an MFG approximates the NE of the finite game. One future research direction is how to reduce the approximation error while applying the MFG solution to the corresponding finite game. It is apparent that the error depends on many factors (e.g., the number of players and the game size of a finite game).

A key distinction between EGTA and other learning dynamics is that the constructed empirical game model incorporates a set of strategies, which are considered strategically important to understand the game. The construction of the set of strategies is called the *strategy exploration* problem in EGTA, which aims to construct effective models with minimal iteration. The iterative EGTA can be viewed as an approach for strategy exploration in MFGs.

Based on this feature of EGTA, to reduce the approximation error of the MFG solution in the corresponding finite game, one potential approach is to conduct strategy exploration in the MFG while re-evaluating the generated strategies in the finite game. This takes the virtue of the MFG model for fast strategy exploration (i.e.,

quickly obtain strategically important strategies of a game) as well as improving the accuracy of the empirical game model for the finite game. This approach is also compatible with the factors that affect the approximation errors. For example, an MFG solution could behave much worse in a ten-player symmetric game than in a one-hundred-player symmetric game. In this case, re-evaluating the empirical game model in the ten-player game could improve the accuracy of the empirical game model and provide stable solutions.

CHAPTER VII

Conclusion

The goal of this thesis is to provide algorithms for game solving and evaluation. I specifically focused on games with large state and action spaces, which makes a direct exhaustive game analysis infeasible. To approach these games, I leveraged the EGTA framework, which describes a broad set of methods that are building and reasoning about game models based on simulation data. I concentrated on understanding and solving the strategy exploration problem in EGTA. Strategy exploration is a key component of iterative EGTA since it directly relates to the quality of game models, on which the game analysis is based. I designed novel algorithms for controlling and evaluating strategy exploration and demonstrated the effectiveness of my algorithms in various games.

For strategy exploration evaluation, I explained what makes evaluating for strategy exploration distinct from evaluating other game learning algorithms. I highlighted that in strategy exploration the generated empirical games create a space of strategies and evaluation should reflect how well the space of strategies covers the strategically relevant space of the full game. To characterize this fact in evaluation, I introduced a systematic evaluation procedure for strategy exploration. Specifically, I proposed to use the regret of MRCP as a measure for a game model since MRCP represents the profile in a game model that is the most close to NE in terms of regret. Then I

presented a consistency criterion that states whereas empirical games can be generated by different MSS-RO combinations, they should be evaluated based on measures of interest (e.g., regret, social welfare) applied to the same solution concept. With informative examples as well as examples in large games, I demonstrated the importance of the consistency criterion and the evaluation issues if the criterion is violated. The intriguing part of this research for me is to discover an evaluation issue in the method that people took for granted in prior works, and provide a refined evaluation procedure for strategy exploration.

For controlling strategy exploration, I first studied how to efficiently build a game model by setting MSSs. After observing a simple regularization deployed in PRD can accelerate the overall equilibrium computation, I applied an explicit regularization approach to strategy exploration and introduced a novel MSS RRD. I showed the effectiveness of RRD on identifying strategically important strategies in few-player games with large strategy spaces and offered an explanation on the enhanced performance of RRD, supported by empirical observations. Apart from RRD, I underscored the effectiveness of alternative approaches such as MRCP and QRE for exploring strategies.

While studying the impact of MSSs in strategy exploration, I realized that some MSSs could lead to an arbitrary NE, which might not possess desired properties such as high social welfare. In other words, we might be interested in not only the solution but its characteristics. To steer strategy exploration toward NE with specified characteristics, I introduced the concept of generalized ROs within the PSRO framework. Unlike the standard PSRO approach, generalized ROs go beyond optimizing utility against opponents' strategies and can incorporate specified preferences. I presented three instances of ROs for PSRO, each reflecting different strategy exploration preferences. To evaluate the effectiveness of these generalized ROs, I conducted experiments using sequential bargaining games and attack-graph games. By comparing

the solutions obtained using different criteria, I found that the choice of ROs can substantially affect equilibria outcomes and steer strategy exploration toward equilibria with preferred features.

For games with a large number of players, I extended the iterative EGTA (PSRO) results from finite games to MFGs. This extension suffered from several issues such as the non-linearity of the utility function in MFGs (in which case a game model cannot be define in terms of the usual components) and the existence issue of NE in an MFG model. To handle these issues, I first provided a game model learning approach to learn the utility function of MFGs. I developed a coarse coding representation for the high-dimensional inputs (i.e., time-dependent strategies and distributions) of MFG utility functions based on the features of EGTA. I also developed a data sampling scheme that effectively samples data in large strategy spaces. I showed that the learned game model exhibits the ability of generalization and can successfully support game-theoretic analysis. After defining game models through learning, we can employ the EGTA framework. I first proved the existence of NE in MFG models and then adapted EGTA to MFGs. My experimental results showed that the EGTA framework can successfully construct a game model incorporating the NE of MFGs in various configurations. Additionally, I illustrated that employing a learned game model significantly reduces the computational cost of analyzing intermediate game models, compared to simulating utility queries for all scenarios. This reduction in cost is attributed to the low expense associated with querying a learned model.

BIBLIOGRAPHY

BIBLIOGRAPHY

Arthur, W Brian (1994). "Inductive reasoning and bounded rationality". In: *The American Economic Review* 84.2, pp. 406–411.

Aumann, Robert J (1964). "Markets with a continuum of traders". In: *Econometrica*, pp. 39–50.

Balduzzi, David, Marta Garnelo, et al. (2019). "Open-ended learning in symmetric zero-sum games". In: *36th International Conference on Machine Learning*.

Balduzzi, David, Karl Tuyls, et al. (2018). "Re-evaluating evaluation". In: *32nd Annual Conference on Neural Information Processing Systems*, pp. 3272–3283.

Basu, Kaushik (1994). "The traveler's dilemma: Paradoxes of rationality in game theory". In: *The American Economic Review* 84.2, pp. 391–395.

Bensoussan, Alain, Jens Frehse, Phillip Yam, et al. (2013). *Mean field games and mean field type control theory*. Vol. 101. Springer.

Berge, Claude (1957). *Théorie générale des jeux à n personnes*. Vol. 138. Gauthier-Villars Paris.

— (1997). *Topological Spaces: including a treatment of multi-valued functions, vector spaces, and convexity*. Courier Corporation.

Borel, Émile (1895). "Sur quelques points de la théorie des fonctions". In: *Annales Scientifiques de L'École Normale Supérieure*. Vol. 12, pp. 9–55.

Brinkman, Erik and Michael P. Wellman (2016). "Shading and efficiency in limit-order markets". In: *IJCAI-16 Workshop on Algorithmic Game Theory*.

Brown, George W (1951). "Iterative solution of games by fictitious play". In: *Activity Analysis of Production and Allocation* 13.1, pp. 374–376.

Cardaliaguet, Pierre and Saeed Hadikhanloo (2017). "Learning in mean field games: the fictitious play". In: *ESAIM: Control, Optimisation and Calculus of Variations* 23.2, pp. 569–591.

Carmona, René and François Delarue (2018). *Probabilistic Theory of Mean Field Games with Applications I-II*. Springer.

Chen, Yang, Jiamou Liu, and Bakhadyr Khoussainov (2021). "Agent-level maximum entropy inverse reinforcement learning for mean field games". In: *arXiv preprint arXiv:2104.14654*.

Conitzer, Vincent and Caspar Oesterheld (2022). "Foundations of cooperative AI". In: *37th AAAI Conference on Artificial Intelligence*.

Czarnecki, Wojciech Marian et al. (2020). "Real World Games Look Like Spinning Tops". In: *31st Annual Conference on Neural Information Processing Systems*.

Dinh, Le Cong et al. (Oct. 2022). "Online Double Oracle". In: *Transactions on Machine Learning Research*.

Doncel, Josu, Nicolas Gast, and Bruno Gaujal (2019). "Discrete mean field games: Existence of equilibria and convergence". In: *arXiv preprint arXiv:1909.01209*.

Duong, Quang et al. (2009). "Learning graphical game models". In: *20th International Joint Conference on Artificial Intelligence*. Pasadena, pp. 116–121.

Elie, Romuald et al. (2020). "On the convergence of model free learning in mean field games". In: *34th AAAI Conference on Artificial Intelligence*, pp. 7143–7150.

Farago, Julie, Amy Greenwald, and Keith Hall (2002). "Fair and efficient solutions to the Santa Fe bar problem". In: *Grace Hopper Celebration of Women in Computing*. Citeseer.

Fearnley, John et al. (2015). "Learning equilibria of games via payoff queries." In: *Journal of Machine Learning Research* 16, pp. 1305–1344.

Ficici, Sevan G, David C Parkes, and Avi Pfeffer (2008). "Learning and solving many-player games through a cluster-based representation". In: *In Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, pp. 187–195.

Fudenberg, Drew and David K Levine (1995). "Consistency and cautious fictitious play". In: *Journal of Economic Dynamics and Control* 19.5-7, pp. 1065–1089.

Fudenberg, Drew and Jean Tirole (1983). "Sequential bargaining with incomplete information". In: *The Review of Economic Studies* 50.2, pp. 221–247.

Gomes, Diogo A, Joana Mohr, and Rafael Rigao Souza (2010). "Discrete time, finite state space mean field games". In: *Journal De Mathématiques Pures et Appliquées* 93.3, pp. 308–328.

Guo, Xin et al. (2019). "Learning mean-field games". In: *33rd Annual Conference on Neural Information Processing Systems* 32.

Huang, Minyi, Roland P Malhamé, Peter E Caines, et al. (2006). "Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash cer-

tainty equivalence principle". In: *Communications in Information & Systems* 6.3, pp. 221–252.

Jordan, Patrick R., L. Julian Schvartzman, and Michael P. Wellman (2010). "Strategy Exploration in Empirical Games". In: *AAMAS*. Toronto, pp. 1131–1138.

Kakutani, Shizuo (1941). "A generalization of Brouwer's fixed point theorem". In: *Duke Mathematical Journal* 8.3, pp. 457–459.

Kearns, Michael (2007). "Graphical games". In: *Algorithmic Game Theory* 3, pp. 159–180.

Lanctot, Marc, Edward Lockhart, et al. (2019). "OpenSpiel: A framework for reinforcement learning in games". In: *arXiv preprint arXiv:1908.09453*.

Lanctot, Marc, Vinicius Zambaldi, et al. (2017). "A unified game-theoretic approach to multiagent reinforcement learning". In: *31st Annual Conference on Neural Information Processing Systems*. Long Beach, CA, pp. 4190–4203.

Lasry, Jean-Michel and Pierre-Louis Lions (2007). "Mean field games". In: *Japanese journal of mathematics* 2.1, pp. 229–260.

Laurière, Mathieu et al. (2022). "Scalable Deep Reinforcement Learning Algorithms for Mean Field Games". In: *39th International Conference on Machine Learning*.

Leslie, David S and Edmund J Collins (2006). "Generalised weakened fictitious play". In: *Games and Economic Behavior* 56.2, pp. 285–298.

Li, Zun, Marc Lanctot, et al. (2023). "Combining Tree-Search, Generative Models, and Nash Bargaining Concepts in Game-Theoretic Reinforcement Learning". In: *arXiv preprint arXiv:2302.00797*.

Li, Zun and Michael P Wellman (2021). "Evolution Strategies for Approximate Solution of Bayesian Games". In: *35th AAAI Conference on Artificial Intelligence*. Vol. 35. 6, pp. 5531–5540.

— (2020). "Structure learning for approximate solution of many-player games". In: *34th AAAI Conference on Artificial Intelligence*. Vol. 34. 02, pp. 2119–2127.

Lin, Xiaomin, Stephen C Adams, and Peter A Beling (2019). "Multi-agent inverse reinforcement learning for certain general-sum stochastic games". In: *Journal of Artificial Intelligence Research* 66, pp. 473–502.

Liu, Zongkai et al. (2022). "A Unified Diversity Measure for Multiagent Reinforcement Learning". In: *36th Annual Conference on Neural Information Processing Systems*.

Marris, Luke et al. (2021). "Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers". In: *38th International Conference on Machine Learning*, pp. 7480–7491.

McAleer, Stephen, John Lanier, Roy Fox, et al. (2020). "Pipeline PSRO: A Scalable Approach for Finding Approximate Nash Equilibria in Large Games". In: *34th Annual Conference on Neural Information Processing Systems*.

McAleer, Stephen, John Lanier, Kevin A. Wang, et al. (2021). "XDO: A Double Oracle Algorithm for Extensive-Form Games". In: *34th Annual Conference on Neural Information Processing Systems*.

McAleer, Stephen, Kevin Wang, et al. (2022). "Anytime PSRO for two-player zero-sum games". In: *AAAI-22 Workshop on Reinforcement Learning in Games*.

McKelvey, Richard D., Andrew M. McLennan, and Theodore L. Turocy (2006). *Gambit: Software Tools for Game Theory*.

McKelvey, Richard D. and Thomas R. Palfrey (1995). "Quantal response equilibria for normal form games". In: *Games and Economic Behavior* 10.1, pp. 6–38.

— (1998). "Quantal response equilibria for extensive form games". In: *Experimental Economics* 1.1, pp. 9–41.

McMahan, H. Brendan, Geoffrey J. Gordon, and Avrim Blum (2003). "Planning in the presence of cost functions controlled by an adversary". In: *20th International Conference on Machine Learning*, pp. 536–543.

Miehling, Erik, Mohammad Rasouli, and Demosthenis Teneketzis (2015). "Optimal defense policies for partially observable spreading processes on Bayesian attack graphs". In: *2nd ACM Workshop on Moving Target Defense*, pp. 67–76.

Mishra, Rajesh K, Deepanshu Vasal, and Sriram Vishwanath (2020). "Model-free reinforcement learning for non-stationary mean field games". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 1032–1037.

Muller, Paul, Shayegan Omidshafiei, et al. (2020). "A generalized training approach for multiagent learning". In: *8th International Conference on Learning Representations*. virtual.

Muller, Paul, Mark Rowland, et al. (2021). "Learning Equilibria in Mean-Field Games: Introducing Mean-Field PSRO". In: *31st International Conference on Autonomous Agents and Multi-Agent Systems*.

Nash Jr., John F. (1950a). "Equilibrium points in n-person games". In: *Proceedings of the National Academy of Sciences* 36.1, pp. 48–49.

— (1950b). "The bargaining problem". In: *Econometrica*, pp. 155–162.

Natarajan, Sriraam et al. (2010). "Multi-agent inverse reinforcement learning". In: *9th International Conference on Machine Learning and Applications*. IEEE, pp. 395–400.

Nelder, John A. and Roger Mead (1965). "A simplex method for function minimization". In: *Computer Journal* 7.4, pp. 308–313.

Nijenhuis, Albert (1975). *HS Wilf Combinatorial Algorithms*.

Omidshafiei, Shayegan, Christos Papadimitriou, et al. (2019). "α-rank: Multi-agent evaluation by evolution". In: *Scientific Reports* 9.1, pp. 1–29.

Omidshafiei, Shayegan, Karl Tuyls, et al. (2020). "Navigating the Landscape of Games". In: *Nat Commun 11, 5603*.

Parker-Holder, Jack et al. (2020). "Effective diversity in population based reinforcement learning". In: *34th Annual Conference on Neural Information Processing Systems*.

Perez-Nieves, Nicolas et al. (2021). "Modelling behavioural diversity for learning in open-ended games". In: *International Conference on Machine Learning*. PMLR, pp. 8514–8524.

Perolat, Julien et al. (2022). "Scaling up Mean Field Games with Online Mirror Descent". In: *31st International Conference on Autonomous Agents and Multi-Agent Systems*.

Perrin, Sarah, Mathieu Laurière, Julien Pérolat, Romuald Élie, et al. (2022). "Generalization in Mean Field Games by Learning Master Policies". In: *36th AAAI Conference on Artificial Intelligence*.

Perrin, Sarah, Mathieu Laurière, Julien Pérolat, Matthieu Geist, et al. (2021). "Mean Field Games Flock! The Reinforcement Learning Way". In: *30th International Joint Conference on Artificial Intelligence*.

Perrin, Sarah, Julien Pérolat, et al. (2020). "Fictitious play for mean field games: Continuous time analysis and applications". In: *34th Annual Conference on Neural Information Processing Systems*.

Phelps, S. et al. (2006). "A novel method for automatic strategy acquisition in N-player non-zero-sum games". In: *5th International Joint Conference on Autonomous Agents and Multi-Agent Systems*. Hakodate, pp. 705–712.

Rubinstein, Ariel (1982). "Perfect equilibrium in a bargaining model". In: *Econometrica: Journal of the Econometric Society*, pp. 97–109.

Rubinstein, Ariel and Asher Wolinsky (1985). "Equilibrium in a market with sequential bargaining". In: *Econometrica*, pp. 1133–1150.

Schmeidler, David (1973). "Equilibrium points of nonatomic games". In: *Journal of statistical Physics* 7, pp. 295–300.

Schvartzman, L. Julian and Michael P. Wellman (2009a). "Exploring Large Strategy Spaces in Empirical Game Modeling". In: *AAMAS-09 Workshop on Agent-Mediated Electronic Commerce*. Budapest.

— (2009b). "Stronger CDA strategies through empirical game-theoretic analysis and reinforcement learning". In: *18th International Conference on Autonomous Agents and Multi-Agent Systems*. Budapest, pp. 249–256.

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

Smith, J Maynard and George R Price (1973). "The logic of animal conflict". In: *Nature* 246.5427, pp. 15–18.

Smith, Max Olan, Thomas Anthony, and Michael P Wellman (2021). "Iterative empirical game solving via single policy best response". In.

— (2023). "Learning to play against any mixture of opponents". In: *Frontiers in Artificial Intelligence*.

Sokota, Samuel, Caleb Ho, and Bryce Wiedenbeck (2019). "Learning deviation payoffs in simulation-based games". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01, pp. 2173–2180.

Taylor, Peter D and Leo B Jonker (1978). "Evolutionary stable strategies and game dynamics". In: *Mathematical biosciences* 40.1-2, pp. 145–156.

Tuyls, Karl et al. (2020). "Bounds and dynamics for empirical game-theoretic analysis". In: *29th International Conference on Autonomous Agents and Multi-Agent Systems* 34.7.

Van der Genugten, Ben (2000). "A weakened form of fictitious play in two-person zero-sum games". In: *International Game Theory Review* 2.04, pp. 307–328.

Vorobeychik, Yevgeniy (2010). "Probabilistic Analysis of Simulation-Based Games". In: *ACM Transactions on Modeling and Computer Simulation* 20.3, 16:1–25.

Vorobeychik, Yevgeniy, Michael P Wellman, and Satinder Singh (2007). "Learning payoff functions in infinite games". In: *Machine Learning* 67.1, pp. 145–168.

Wang, Yongzhao, Qiurui Ma, and Michael P. Wellman (2022). "Evaluating Strategy Exploration in Empirical Game-Theoretic Analysis". In: *31st International Conference on Autonomous Agents and Multi-Agent Systems*.

Wang, Yongzhao and Michael P Wellman (2023a). "Regularization for Strategy Exploration in Empirical Game-Theoretic Analysis". In: *arXiv preprint arXiv:2302.04928*.

— (2023b). "Empirical Game-Theoretic Analysis for Mean Field Games". In: *32nd International Conference on Autonomous Agents and Multi-Agent Systems*.

Wang, Yongzhao and Michael P. Wellman (2023c). "Game Model Learning for Mean Field Games". In: *32nd International Conference on Autonomous Agents and Multi-Agent Systems*.

Wang, Yufei, Zheyuan Ryan Shi, et al. (2019). "Deep reinforcement learning for green security games with real-time information". In: *33rd AAAI Conference on Artificial Intelligence*.

Wellman, Michael P. (2006). "Methods for Empirical Game-Theoretic Analysis (Extended Abstract)". In: *Twenty-First National Conference on Artificial Intelligence*. Boston, pp. 1552–1555.

Wellman, Michael P. et al. (2005). "Approximate strategic reasoning through hierarchical reduction of large symmetric games". In: *Twentieth National Conference on Artificial Intelligence*. Pittsburgh, pp. 502–508.

Wiedenbeck, Bryce, Ben-Alexander Cassell, and Michael P. Wellman (2014). "Bootstrap techniques for empirical games". In: *23rd International Conference on Autonomous Agents and Multi-Agent Systems*. Paris, pp. 597–604.

Wiedenbeck, Bryce and Michael P. Wellman (2012). "Scaling simulation-based game analysis through deviation-preserving reduction". In: *21st International Conference on Autonomous Agents and Multi-Agent Systems*. Valencia, pp. 931–938.

Wiedenbeck, Bryce, Fengjun Yang, and Michael P. Wellman (2018). "A regression approach for modeling games with many symmetric players". In: *32nd AAAI Conference on Artificial Intelligence*. New Orleans, pp. 1266–1273.

Wright, Mason, Yongzhao Wang, and Michael P. Wellman (2019). "Iterated Deep Reinforcement Learning in Games: History-Aware Training for Improved Stability". In: *20th ACM Conference on Economics and Computation*. Phoenix, pp. 617–636.

Yu, Lantao, Jiaming Song, and Stefano Ermon (2019). "Multi-agent adversarial inverse reinforcement learning". In: *International Conference on Machine Learning*. PMLR, pp. 7194–7201.

Zhou, Ming et al. (2022). "Efficient Policy Space Response Oracles". In: *arXiv preprint arXiv:2202.00633*.

Zhukovskii, Vladislav I (1985). "Some problems of non-antagonistic differential games". In: *Matematiceskie metody v issledovanii operacij*, pp. 103–195.