Exploiting Extensive-Form Structure in Empirical Game-Theoretic Analysis

Christine Konicki(✉), Mithun Chakraborty, and Michael P. Wellman

University of Michigan, Ann Arbor, MI 48109, USA
{ckonicki,dcsmc,wellman}@umich.edu

Abstract. Empirical game-theoretic analysis (EGTA) is a general framework for reasoning about complex games using agent-based simulation. Data from simulating select strategy profiles is employed to estimate a cogent and tractable game model approximating the underlying game. To date, EGTA methodology has focused on game models in normal form; though the simulations play out in sequential observations and decisions over time, the game model abstracts away this temporal structure. Richer models of *extensive-form games* (EFGs) provide a means to capture temporal patterns in action and information, using tree representations. We propose *tree-exploiting EGTA* (TE-EGTA), an approach to incorporate EFG models into EGTA. TE-EGTA constructs game models that express observations and temporal organization of activity, albeit at a coarser grain than the underlying agent-based simulation model. The idea is to exploit key structure while maintaining tractability. We establish theoretically and experimentally that exploiting even a little temporal structure can vastly reduce estimation error in strategy-profile payoffs compared to the normal-form model. Further, we explore the implications of EFG models for iterative approaches to EGTA, where strategy spaces are extended incrementally. Our experiments on several game instances demonstrate that TE-EGTA can also improve performance in the iterative setting, as measured by the quality of equilibrium approximation as the strategy spaces are expanded.

1 Introduction

Empirical game-theoretic analysis (EGTA) (Wellman 2016) employs agent-based simulation to induce a game model over a restricted set of strategies. The methodology is salient for games that are too complex for analytic description and reasoning. Complexity in dynamics and information can be expressed in a simulator, but abstracted from the game model. In typical EGTA practice, simulation data is used to estimate a *normal-form game* (NFG) model, associating a payoff vector with each combination of strategies available to the agents. But game theory offers richer model forms that capture sequentiality in agent play and conditional information. Specifically, *extensive-form game* (EFG) models represent the game as a tree, where nodes or sets of nodes represent states,

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-22832-2_8.

K. A. Hansen et al. (Eds.): WINE 2022, LNCS 13778, pp. 132–149, 2022.
https://doi.org/10.1007/978-3-031-22832-2_8

and edges represent player moves and chance events. Whereas NFGs treat agent strategies as atomic objects, EFGs afford a finer-grained expression of the observations and actions that define these strategies, capturing structure that may be shared among many strategies. The goal of this work is to take advantage of extensive-form structure, at flexible granularity, for complex game environments described by agent-based simulation. Our approach, *Tree-Exploiting EGTA* (TE-EGTA), follows the basic framework of EGTA, but employs a parameterized EFG model to leverage part of the game's tree structure.

Taking advantage of extensive form necessitates two key modifications to the EGTA process. First, we require methods to estimate the more complex model form: an abstracted game tree parameterized by player utilities at terminal nodes and probability distributions over successors for stochastic events represented by *chance nodes* in the tree. These stochastic events, together with *information-set* structure, model the *imperfect information* available to the players. We introduce straightforward techniques to estimate these game-tree parameters, and describe how the structure intuitively affords more effective use of available simulation data. Second, we require methods for extending extensive-form models as the strategy space is expanded, across iterations of the EGTA process. We introduce techniques for iterative augmentation of empirical game-tree models with new (best-response) strategies, within a standard approach that incorporates deep RL within EGTA (Lanctot et al. 2017).

To establish the benefits of tree-exploitation for EGTA, we show that an extensive-form empirical game model provides (with high probability) a more accurate approximation of the true game than a normal-form model constructed from the same simulation data. As it is generally intractable to construct a game tree expressing the full fidelity of the game simulated, our approach is designed to operate on highly abstracted models capturing only selected tree structure. To ground the meaning of such abstractions, we provide an algorithm that produces a coarsened model given the full game and a description of what to abstract away. We demonstrate the efficacy of TE-EGTA through experiments on three stylized games, and over varying levels of abstraction. We compare TE-EGTA to normal-form EGTA on two key performance measures. The first is the average error incurred from estimating the true player payoffs for all strategy combinations in the empirical game. The second is the *regret* of empirical-game solutions with respect to the full multiagent scenario, computed over successive empirical game models in an iterative EGTA process.

Outline. Section 2 provides technical preliminaries, including a formal exposition of the EFG representation and precise elaboration of the EGTA framework and process. Sect. 3 delineates our algorithmic contribution, TE-EGTA, starting with the structure of an extensive-form empirical game model and how to estimate its parameters from simulation data (Sect. 3.1). We then give a theoretical procedure for generating a (usually) coarsened extensive-form model from the underlying game (Sect. 3.2), and explain how to iteratively refine the model via simulation-aided strategy exploration (Sect. 3.3). In Sect. 4, we present theoretical results on the advantage of TE-EGTA over normal-form EGTA in approximating true payoffs given a set of strategy profiles. All proofs are available in the full version. In Sect. 5, we report experiments that demonstrate the improvement in strategy-profile payoff estimation (Sect. 5.1) and in model refinement using the PSRO approach (Lanctot et al. 2017) (Sect. 5.2) produced via tree exploitation. Sect. 6 concludes.

2 Preliminaries

2.1 Extensive-Form Games (EFGs)

An *extensive-form game* (EFG) is a standard model for strategic multi-agent scenarios where agents act *sequentially* with potentially varying degrees of *imperfect information* about the history of game play. Early algorithmic work on EFGs showed how to generalize the Lemke-Howson method for computing Nash equilibria (NE) for two-player games with perfect recall (Koller et al. 1996). Well-known game-theoretic methods such as replicator dynamics (Gatti et al. 2013) and fictitious self-play (Heinrich et al. 2015) have also been adapted for EFGs. The task of successful abstraction with exploitability guarantees has also been investigated: Kroer and Sandholm (2018) gave a framework for analyzing abstractions of large-scale EFGs, and Zhang and Sandholm (2020) introduced the notion of small certificates carrying proofs of approximate NE. Other works have developed algorithms that search for optimal strategies or approximate equilibria that minimize exploitability (Johanson et al. 2012; Lockhart et al. 2019). In this paper, we will only consider games with perfect recall, so no player can forget what it observed or knew earlier.

Tree Structure. Formally, a finite, imperfect-information EFG is a tuple $G := \langle N, H, V, \{\mathcal{I}_j\}_{j=0}^{n}, \{\Pi_j\}_{j=1}^{n}, X, P, u\rangle$. The components of G are defined as follows (see Fig. 1 for an illustrative example):

- $N = \{0, \ldots, n\}$ is the set of *players*. Player 0 represents *Nature*, a non-strategic agent responsible for stochastic events that impact the course of play; the remaining players are strategic rational agents.
- H is the finite game tree, rooted at a node h_0, that captures the dynamic nature of interactions. Each node $h \in H$ represents a *state* of the game, also identified with a history of actions (see below) beginning at the *initial state* h_0 which corresponds to the null history $\emptyset$. The leaves or *terminal nodes* $T \subset H$ represent possible end-states of the game. We refer to the non-terminal nodes of H as *decision nodes*, represented by the set $D = H \setminus T$.
- $V : D \rightarrow N$ assigns a player to each decision node h.
- For each player $j \in N$, $\mathcal{I}_j$ is a partition of $V^{-1}(j)$ where each $I \in \mathcal{I}_j$ is an *information set (infoset)* of j. All nodes $h \in I$ are indistinguishable from the viewpoint of player j.
- At each information set $I \in \mathcal{I}_j$, player j has a set of available actions $\Pi_j(I)$.
- A node h where $V(h) = 0$ is called a *chance node*. $X(h)$ is the set of actions available to Nature (i.e., possible outcomes of the stochastic event) at h, and $P(\cdot \mid h)$ is the probability distribution over $X(h)$.
- The *utility function* $u : T \rightarrow \mathbb{R}^n$ maps each terminal node to a real-valued vector of players' utilities $\{u_j(t)\}_{j=1}^{n}$.

The directed edge connecting any $h \in I$ to its child $\mathsf{child}[h]$ represents a state transition resulting from $V(h)$'s move, and is labeled with an action $\pi \in \Pi_{V(h)}(I)$ if $V(h) \neq 0$, or an outcome $x \in X(h)$ otherwise. We denote by $\varphi(h, j)$ the history of actions belonging to player j up to node h.

Strategies and Payoffs. A *pure strategy* for player $j \in N \setminus \{0\}$ specifies the action $\pi_j \in \Pi_j(I)$ that j selects at information set $I \in \mathcal{I}_j$. More generally, a *mixed strategy* or simply *strategy* $\sigma_j(\cdot \mid I)$ defines a probability distribution over $\Pi_j(I)$ at each information set of agent j; that is, action π_j is selected with probability $\sigma_j(\pi_j \mid I)$. The vector $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)$ is called a *strategy profile*, and $\boldsymbol{\sigma}_{-j}$ represents the combination of strategies for players other than j. We denote the set of all strategies available to player j by Σ_j and the space of joint strategy profiles by $\Sigma = \times_{j=1}^{n} \Sigma_j$. Let $r_j(t, \sigma_j)$ denote the probability that node t is reached if player j adopts strategy σ_j and all other players (including Nature) always choose actions that lead to h when possible; the probability that t is reached under strategy profile $\boldsymbol{\sigma}$ is given by its *reach probability*, $r(t, \boldsymbol{\sigma}) = \prod_{j \in N} r_j(t, \sigma_j)$. Likewise, the contribution of Nature to the reach probability of t is $r_0(t) = \prod_{h \in H,\, e \in X(h) \cap \varphi(t,0)} P\left(e \mid h\right)$. We define the *payoff* from joint strategy profile $\boldsymbol{\sigma}$ to player j as its expected utility over all end-states: $U_j(\boldsymbol{\sigma}) := \sum_{t \in T} u_j(t) r(t, \boldsymbol{\sigma})$.

Best Response Formulation and Regret. A *best response (BR)* of player $j \in N \setminus \{0\}$ to $\boldsymbol{\sigma}_{-j}$ is a strategy $\sigma_j \in \arg\max_{\sigma'_j \in \Sigma_j} U_j(\sigma'_j, \boldsymbol{\sigma}_{-j})$ that maximizes the payoff for j given $\boldsymbol{\sigma}_{-j}$. The *regret* of player j from playing $\boldsymbol{\sigma}$ is given by $\mathsf{Reg}_j(\boldsymbol{\sigma}) = \max_{\sigma_j \in \Sigma_j} U_j(\sigma_j, \boldsymbol{\sigma}_{-j}) - U_j(\boldsymbol{\sigma})$. The total regret of the strategy profile $\boldsymbol{\sigma}$ is the sum: $\mathsf{Reg}(\boldsymbol{\sigma}) = \sum_{j=1}^{n} \mathsf{Reg}_j(\boldsymbol{\sigma})$. For $\varepsilon > 0$, an ε-*Nash equilibrium* is a strategy profile $\boldsymbol{\sigma}$ such that $\mathsf{Reg}_j(\boldsymbol{\sigma}) \leq \varepsilon$ for every player $j \in N \setminus \{0\}$; a strategy profile $\boldsymbol{\sigma}$ with $\mathsf{Reg}(\boldsymbol{\sigma}) = 0$ is a *Nash equilibrium.*

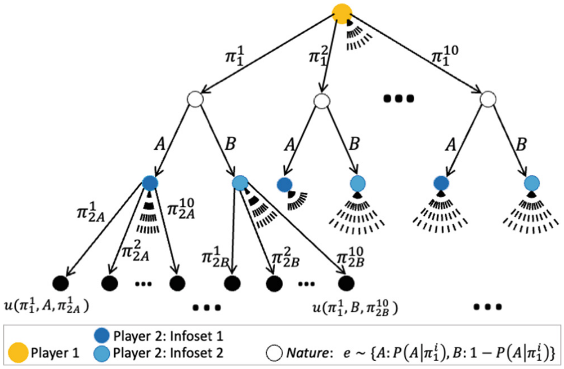


Fig. 1. EFG representation of Game$_1$, our running example also used in our experiments. Dashed lines indicate outgoing edges to nodes omitted from this illustration.

Running Example. Consider the two-agent strategic scenario depicted in Fig. 1, which we call Game$_1$. First, Player 1 chooses an action from $\Pi_1 =$

$\{\pi_1^i\}_{i=1}^{10}$; then, a single stochastic event $X(\pi_1^i) \in \{A, B\}$ occurs, outcome A having probability $P(A \mid \pi_1^i)$ dependent on Player 1's choice π_1^i. Player 2 observes the outcome $e \in \{A, B\}$ but not Player 1's chosen action, which induces two information sets for Player 2. Player 2 also has ten actions to choose from in each information set, $\Pi_{2A} = \{\pi_{2A}^i\}_{i=1}^{10}$ and $\Pi_{2B} = \{\pi_{2B}^i\}_{i=1}^{10}$. Each leaf with history $(\pi_1^i, e, \pi_{2e}^{i'})$ is labeled with the 2-dimensional vector of Player 1 and 2's realized utilities. Neither the conditional probabilities $P(A \mid \pi_1^i)$ nor the leaf utilities $u(\pi_1^i, e, \pi_{2e}^{i'})$ are known *a priori* to the game analyst.

2.2 Empirical Game-Theoretic Analysis (EGTA)

The framework of EGTA was developed for the application of game-theoretic reasoning to scenarios too complex for analytic description, accessible only in the form of a procedural simulation (Wellman 2016). Over the years, EGTA has been applied to multifarious problem domains including recreational strategy games (Tuyls et al. 2020), security games (Wang et al. 2019), social dilemmas (Leibo et al. 2017), and auctions (Wellman 2020). There is also substantial work on methodological questions such as how to decide which strategy profiles to simulate (Fearnley et al. 2015; Jordan et al. 2008), and how to reason statistically about estimated game models (Areyan Viqueira et al. 2020; Tuyls et al. 2020; Vorobeychik 2010). Recently, EGTA has received newfound attention, as the simulation-based approach meshes well with powerful new strategy generation methods from deep reinforcement learning (RL) (Lanctot et al. 2017).

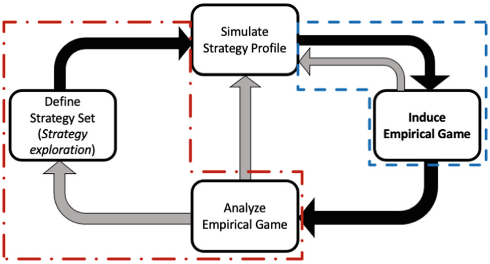

Simulate Strategy Profile
Define Strategy Set (*Strategy exploration*)
Induce Empirical Game
Analyze Empirical Game

Fig. 2. Schematic illustration of EGTA. TE-EGTA modifies two subprocesses to incorporate the tree structure of EFGs: accumulation of simulation data into the game model (enclosed in blue, described in Sect. 3.1); and the procedure for augmenting $\hat{\Sigma}$ with new strategies (enclosed in red, described in Sect. 3.3). Black (resp. grey) arrows represent the sequence of operations (resp. direction of possible information flow). (Color figure online)

The main feature of EGTA is its construction of an *empirical game model* $\hat{G}$ of a much larger game of interest, called the *true game* G, from simulation data. A typical EGTA process (see Fig. 2) iteratively refines and extends $\hat{G}$ by cumulative simulation over an incrementally growing strategy space. $\hat{G}$ is a *simplification* of the underlying G since: (1) it is defined on restricted subsets $\hat{\Sigma}_j \subset \Sigma_j$ of the players' true-game strategy spaces, and the *restricted strategy profile space*, given by $\hat{\Sigma} = \times_{j=1}^{n} \hat{\Sigma}_j$, is typically a vast reduction of Σ; (2) some information revelation and conditioning structure may be abstracted away. Moreover, we assume that G is accessible only through a high-fidelity but expensive *simulator* that executes a given strategy profile in G and outputs limited observation histories and noisy utility samples. $\hat{G}$ is thus also an *approximation* of G since its parameters must be estimated from this simulation data.

Almost all EGTA literature to date expresses game models in *normal form*, given by a (multi-dimensional) matrix of payoff estimates for combinations of agents' strategies from the restricted set. The multi-agent scenarios themselves are typically dynamic in nature, as represented by an agent-based simulator; agent strategies are generally conditional on partial observations. For example, a normal-form game model for GAME_1 in Sect. 2.1 would treat each pure strategy $\pi_2^{i'}$ of player 2 as atomic, abstracting away the nuanced conditioning on whether A or B happened, and record estimated utility vectors for strategy combinations of the form $(\pi_1^i, \pi_2^{i'})$ from restricted set.

As our objective is to extend EGTA to extensive-form modeling, we will call this normal-form baseline *NF-EGTA*. In NF-EGTA, the sole simulator output of concern is the noisy sample of players' payoffs, from which we compute estimates $\{\hat{U}_j^{NF}(\boldsymbol{\sigma})\}_{j=1}^{n}$ of the true utilities $\{U_j(\boldsymbol{\sigma})\}_{j=1}^{n}$ to obtain the empirical game model $\hat{G}$. We then analyze or solve this tractable, multi-dimensional game matrix by standard techniques to obtain a result for the next iteration. Termination may be decided by a criterion such as the *true-game regret* of a solution (i.e., the maximum payoff increase achievable by any player j by deviating to a strategy in Σ_j rather than $\hat{\Sigma}_j$) falling below a specified threshold. If termination criteria are not met, we expand the restricted strategy sets through a process called *strategy exploration* (Balduzzi et al. 2019; Jordan et al. 2010), and update $\hat{G}$ through further simulation and model induction.

Game Model Estimation. Consider the process of estimating a normal-form model for an underlying extensive-form game implicitly represented by traces from the simulator. Suppose we simulate each strategy profile in $\hat{\Sigma}$ m times. Each simulated play traces a path through the game tree ending at some undisclosed terminal node $t \in T$ and returns a vector of noisy payoffs for all players sampled from a distribution with expectation $u(t)$. Let $\{\bar{u}_j^i\}_{j=1}^{n}$ denote the realized payoff sample at the end of the i^{th} simulation for $i = 1, \ldots, m$; Typically, NF-EGTA's payoff estimate $\hat{U}_j^{NF}(\boldsymbol{\sigma})$ is the simple average of these samples. $\hat{U}_j^{NF}(\boldsymbol{\sigma})$ is an unbiased estimator of the true payoff, as shown in Proposition 1. In practice, the number of samples m that can be acquired is limited by the computational cost of simulation. This begs the question: can incorporating tree structure into $\hat{G}$ improve the accuracy of estimated payoffs, relative to *NF-EGTA*, for a fixed simulation budget m? We address this question in Sect. 3.1.

Proposition 1. *For every player $j \in N \setminus \{0\}$ and strategy profile $\boldsymbol{\sigma} \in \hat{\Sigma}$, $\mathbb{E}\left[\hat{U}_j^{NF}(\boldsymbol{\sigma})\right] = U_j(\boldsymbol{\sigma})$.*

Policy-Space Response Oracles (PSRO). A fully automated implementation of the iterative EGTA framework of Fig. 2 requires the ability to automatically generate new strategies based on analysis of the empirical game model at a given point. Phelps et al. (2006) first introduced automated strategy generation to EGTA via genetic search, and Schvartzman and Wellman (2009) first employed RL for this purpose. The advent of deep RL methods brought significant new power to this approach, which is now the predominant means of accumulating a set of restricted strategies in EGTA algorithms.

Lanctot et al. (2017) developed a general framework for interleaving empirical game modeling with deep RL techniques, which they termed *policy-space response oracles*. A key idea of PSRO is that of a *meta-strategy solver* (MSS), an abstract operation that implements the "Analyze Empirical Game" block of Fig. 2. The output of an MSS is a strategy profile, which provides the other-agent context for a BR calculation performed by deep RL. The policy generated by RL as a BR to the MSS result is then added as a new strategy to expand the current restricted strategy space, leading to another round of simulation and induction for the next EGTA iteration. The MSS concept provides a useful abstraction for expressing a variety of approaches to strategy exploration (Wang et al. 2022). For example, using Nash equilibrium as an MSS yields the double oracle (DO) algorithm (McMahan et al. 2003). If the MSS simply returns the uniform distribution over the restricted strategy sets, the algorithm reduces to fictitious play.

Prior work has extended the DO algorithm to exploit game-tree structure. Bošanský et al. (2014) developed a *sequence-form double-oracle* algorithm for zero-sum EFGs that maintains a restricted game model based on partial action sequences. The XDO algorithm of McAleer et al. (2021) for two-player zero-sum games computes a mixed BR at each information set, as compared to normal-form DO which mixes policies only at the root level. It modifies PSRO for EFGs while still using a normal-form empirical model. The benefits over normal-form demonstrated by these works suggest EGTA can be similarly extended to exploit game-tree structure beyond the strategy exploration block.

3 Tree-Exploiting EGTA

We call our approach for augmenting empirical game models to incorporate extensive-form game elements *tree-exploiting EGTA* (TE-EGTA). In the typical normal-form treatment of EGTA, the underlying game is parameterized by entries in a payoff matrix $\{U_j(\boldsymbol{\sigma})\}_{j\in N\setminus\{0\},\boldsymbol{\sigma}\in\Sigma}$.[1] TE-EGTA instead parameterizes the underlying game to capture the EFG tree structure through a set of *leaf utilities* $\{u(t)\}_{t\in T}$, and *conditional probability distributions* that are dependent on possibly unobserved previous choices made in the game play and estimated from observations of stochastic events.

[1] More general approaches based on regression have been proposed (Sokota et al. 2019; Vorobeychik et al. 2007), which also amount to parameterized representations of a payoff function.

We assume that the structure of decisions and stochastic events in the empirical EFG model is given (typically a high-level abstraction of the game tree implicitly represented by the simulator, as discussed in Sect. 3.2). This ensures that the order of player choices and stochastic events in the empirical game tree matches the order in the true game, from root to leaf. In particular, the true game's information sets must be a refinement of the empirical game's information sets. Given this structure, we treat observations of Nature's actions as conditioned on past game play. The empirical game tree therefore must associate with each chance node a conditional probability distribution over the relevant set of outgoing edges. Leaves of the tree are associated with payoff estimates, which depend on the entire path from the root.

Each simulation of a strategy profile yields sample payoffs, as well as a trace of publicly or privately *observable actions* from both the players and Nature that are made over the course of the game. This is a key point of contrast with the normal-form model, for which only payoffs are relevant. The trace of actions tells us which leaf node in the abstract model is reached and what stochastic event outcomes were realized along the way.

To explain our tree-exploiting estimation approach, we first restate the expression for $U_j(\boldsymbol{\sigma})$ in a way that explicitly factors in probabilities of specific *observations* of stochastic events. We assume that a game theorist working with the black-box simulator's partial observations in order to formulate an empirical model is aware of the game's rules, and so can surmise where in the game the observation has occurred. We also assume that the observation labels used by the simulator allow the game theorist to distinguish the observations from each other and associate them with the appropriate chance nodes. A stochastic observation during gameplay is captured in the tree by an edge $e \in \varphi(t, 0)$ from a chance node h such that $V(h) = 0$ to a node with history he. The reach probability of he from the perspective of Nature is $r_0(he) = P(e \mid h)$, and recall $r_0(t)$ is the joint probability of Nature's choices along the path from the root to t. Hence,

$$U_j(\boldsymbol{\sigma}) = \sum_{t \in T} u_j(t) \prod_{k=1}^{n} r_k(t, \sigma_k) r_0(t). \tag{1}$$

3.1 TE-EGTA Game Model Estimation

The probabilities $r_k(t, \sigma_k)$, for all terminal nodes t, are directly determined by the strategy profile $\boldsymbol{\sigma}$. Hence, to estimate $U_j(\boldsymbol{\sigma})$ based on Eq. (1), we need estimates for $u(t)$ and $\{r_0(t)\}_{t \in T}$. These are, in fact, the game parameters for TE-EGTA (leaf utilities and conditional probabilities respectively) that we introduced above. We denote the respective estimates by $\{\hat{u}_j(t)\}_{j=1}^{n}$ and $\{\hat{r}_0(t)\}_{t \in T}$.

A key feature of TE-EGTA is that, in modeling the payoff of strategy profile $\boldsymbol{\sigma}$, we estimate the parameters using *all* relevant simulation data, not just the data from simulating $\boldsymbol{\sigma}$. Different strategy profiles may lead to overlapping or identical paths being taken through the game tree, with some probability. We compute $\hat{u}_j(t)$ as the sample average of player j's payoffs across simulation runs that terminate at node t. Similarly, we estimate chance node probabilities using all simulations. Suppose a chance node h is reached m_h times across all simulation data, and the node with history he (reflecting Nature's choice e) is reached $m_{he} < m_h$ times. The empirical probability of observing the stochastic outcome represented by e in the game tree is $\frac{m_{he}}{m_h}$. Note that m_h can never be zero because the algorithm for constructing the empirical game model includes only nodes that are reached in simulation. Finally, we give player j's estimated payoff for strategy profile $\boldsymbol{\sigma}$:

$$\hat{U}_j^{TE}(\boldsymbol{\sigma}) = \sum_{t\in T} \hat{u}_j(t) \prod_{k=1}^{n} r_k(t,\sigma_k) \left(\prod_{e\in\varphi(t,0)} \frac{m_{he}}{m_h}\right).$$

Recall that each strategy profile $\boldsymbol{\sigma}$ in $\hat{\Sigma}$ is simulated m times, resulting in m game play sequences for each. Some strategies that end at different terminal nodes t_1 and t_2 may still include the same node h in their respective paths and result in the same observation $e \in \hat{X}(h)$. The observation occurs with the same probability for both strategies since their histories diverge only at node he. This feature is what allows the empirical game model to take into account the role of different decision points in the formulation of player strategies in a way that the normal-form model does not.

To illustrate the difference in model estimation between NF- and TE-EGTA, consider the following example from Game$_1$. Suppose we simulate the strategy profile (π_1^1, π_2^1) 10 times, and obtain the following payoff samples for Player 1: $99, 95, 100, 96, 95, 100, 92, 95, 93, 94$; we also observe outcome A of the stochastic event in the first 6 of these 10 simulations. NF-EGTA would simply average the 10 payoff samples and record $\hat{U}_1^{NF}(\pi_1^1, \pi_2^1) = 95.9$. In contrast, TE-EGTA distinguishes the 6 samples corresponding to the leaf (π_1^1, A, π_{2A}^1) from the 4 samples corresponding to the leaf (π_1^1, B, π_{2B}^1), and separately averages them to get the estimates $\hat{u}_1(\pi_1^1, A, \pi_{2A}^1) = 97.5$ and $\hat{u}_1(\pi_1^1, B, \pi_{2B}^1) = 93.5$. Now, suppose we also have data from 10 simulations of another strategy profile (π_1^1, π_2^2), $\pi_2^2 \neq \pi_2^1$, A being realized in 5 of these simulations. From this experience, our overall estimated probability of A conditioned on π_1^1 is $\frac{6+5}{10+10} = 0.55$. Thus, using all relevant sample data, $\hat{U}_1^{TE}(\pi_1^1, \pi_2^1) = 0.55 \times 97.5 + (1 - 0.55) \times 93.5 = 95.7$.

The following proposition shows that, like NF-EGTA, TE-EGTA produces unbiased estimates of strategy-profile payoffs. However, our theoretical results in Sect. 4 suggest that TE-EGTA offers more accurate payoff estimates with a high probability.

Proposition 2. *For every player $j \in N \setminus \{0\}$ and strategy profile $\boldsymbol{\sigma} \in \hat{\Sigma}$, $\mathbb{E}_{t\sim r(T,\boldsymbol{\sigma})}\left[\hat{U}_j^{TE}(\boldsymbol{\sigma})\right] = U_j(\boldsymbol{\sigma})$.*

3.2 The Game Model as an Abstraction

Abstraction methods have extended the state of the art in solving imperfect-information games over the years (Sandholm 2010), particularly poker. An abstraction algorithm takes as input a complete game description and produces a simpler version of the tree. TE-EGTA incorporates some of the tree structure from the true game into the empirical game model; in order to ground this game model as a coarse abstraction of the underlying game, we describe **Coarsen**, an algorithm that coarsens a game tree by abstracting away chance nodes.

We express *coarseness* as the fraction of chance nodes from the true game that are included in the empirical game model. An empirical game that matches the true game's structure would include all of them; conversely, an empirical game in normal-form would include none of them. We are primarily concerned with games represented by agent-based simulation where the representation of the true game as an EFG is intractable, and thus we would not expect to obtain a coarsened model by actually applying **Coarsen**. Our intent is to contextualize a coarsened game as one that could in principle be produced by abstracting away chance nodes.

Coarsen Algorithm for coarsening an input game G

```
Require: Input game $G$, partition $C' \subseteq C$ and map $\rho : C' \rightarrow X$
  Copy $H' = H$, with each node $h$ represented by its history
  for $c \in C'$, beginning at the chance node furthest from the root do
      Let $I_j(c)$ be the set of infosets induced by each event $e \in X(c)$ for player $j$.
      Compute power set $Z^*$ of intersections $Z = \bigcap_{I \in I_j(c)} \{h \mid he \in I\}$ of all the histories $h$ across $I_j(c)$.
      for $Z \in Z^*$ do
          $\{I'_j, \Pi'_j(I'_j)\}, H' =$ CoarsenInfosets $(I_j(c), Z, \rho, G)$
          Assign $X'(c) = X(c) \setminus \rho(c)$
          $\mathcal{I}'_j, \Pi'_j =$ CondenseBranching $\left(\{I'_j, \Pi'_j(I'_j)\}, \mathcal{I}'_j\right)$
      end for
  end for
  Assign $X'(c) = X(c)$ for all $c \notin C'$.
  Assign all player $j$'s infosets not conditioned on chance events from any $c \in C'$ to $\mathcal{I}'_j$
  For all nodes $h$ that preceded or did not follow any nodes in $C'$, assign $V'(h) = V(h)$

  return $G' = (N, H', V', \{\mathcal{I}'\}_{j=1}^n, \{\Pi'_j\}_{j=1}^n, X')$
```

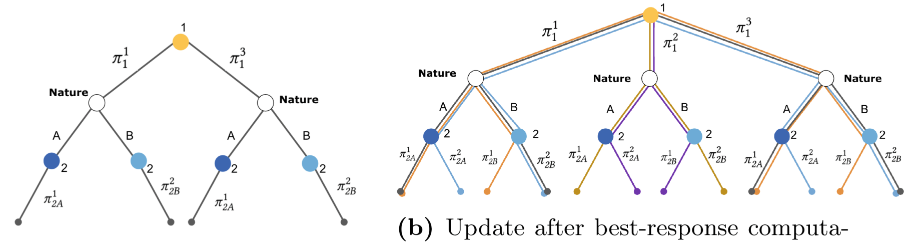
The algorithm is given a partition of both G's chance nodes $C = \{h \in H \mid V(h) = 0\}$ and the set of outcomes $X(h)$ for each chance node, denoting what to exclude from the coarsened tree. One important restriction on G is that the child nodes of a given chance node in C' must all belong to the same player so that they can be collapsed into one node. We denote the abstracted game by $G' = \langle N, H', V', \{\mathcal{I}'_j\}_{j=1}^n, \{\Pi'_j\}_{j=1}^n, X' \rangle$ whose components are defined as in Sect. 2. The nodes identified, information sets, and action spaces will necessarily differ from those of G, depending on what information is coarsened and where. Without loss of generality, **Coarsen** treats both G and G' as binary trees in order to limit the branching factor of G'. **CoarsenInfosets** transforms the intersecting information sets of the children of each $c \in C'$ into a new information set for G'

whose action space is the Cartesian product of the old infosets' action spaces. To keep the branching factor equal to 2, **CondenseBranching** transforms these action spaces (comprised of tuples) into binary (sub-)trees where each edge is part of an action tuple.

3.3 Tree-Exploiting PSRO

Recall the PSRO framework for iterative EGTA with deep RL, introduced in Sect. 2.2. Like EGTA more generally, past work within the PSRO framework has relied on normal-form representations of the empirical game, even though the games of interest are inherently sequential. We call PSRO that uses a normal-form (resp. tree-exploiting) empirical game NF-PSRO (TE-PSRO). In addition to exploiting extensive structure for estimation (Sect. 3.1), TE-PSRO also takes advantage of the tree representation for managing the restricted strategy space. A single pure strategy profile can result in multiple different paths depending on Nature's choices. If a new best response for a given infoset is part of the profile, new paths with their own new utilities and stochastic distributions at Nature's decision points are discovered and added to the empirical game tree. If one of those paths includes moves from other players that are already part of the game tree, then additional samples from this new combination can be included in the (tighter) estimation of the old parameters pertinent to that path.

Consider the empirical game in Fig. 3a with restricted strategy sets $\hat{\Pi}_1, \hat{\Pi}_{2A}$, and $\hat{\Pi}_{2B}$ for each information set as shown; the true game here is GAME_1. Let $BR_1(\sigma_{2A}, \sigma_{2B})$ and $(BR_{2A}(\sigma_1), BR_{2B}(\sigma_1))$ denote the respective best responses from GAME_1 (the true game) to the strategy profile $(\sigma_1, (\sigma_{2A}, \sigma_{2B}))$. Suppose, in an iteration, $BR_1(\sigma_{2A}, \sigma_{2B}) = \pi_1^2$, $BR_{2A}(\sigma_1) = \pi_{2A}^2$, and $BR_{2B}(\sigma_1) = \pi_{2B}^1$.



(a) Starting empirical game tree.

(b) Update after best-response computation.

Fig. 3. Two successive steps of possible TE-PSRO instantiation on GAME_1.

In the next round, the new best-response elements are considered in conjunction with the pre-existing strategy combinations from the restricted set, as well as other players' new best responses. The resulting trajectories are shown in Fig. 3b: (1) $BR_1 \times \Pi_{2A} \times \Pi_{2B}$ highlighted in yellow; (2) $\Pi_1 \times BR_{2A} \times \Pi_{2B}$ highlighted in blue; (3) $\Pi_1 \times \Pi_{2A} \times BR_{2B}$ highlighted in orange; and (4) (BR_1, BR_{2A}, BR_{2B}) highlighted in purple. See the full paper for more detail. This expansion of the empirical game tree captures finer-grained structural information about the true game than simply adding a matrix entry for each new best-response combination.

4 Payoff Estimation Improvement: Theoretical Results

To develop a formal framework for comparing the efficacy of payoff estimation (Sect. 3.1) by TE-EGTA and NF-EGTA, we apply the concept of *uniform approximation of a game* (Areyan Viqueira et al. 2020) to our setting. Consider a true EFG G and an empirical game $\hat{G}$ with the same set of players and with restricted set $\hat{\Sigma}$ constructed from accumulated simulation data upon termination of EGTA. Let $\hat{U}_j(\boldsymbol{\sigma})$ be the estimate in $\hat{G}$ of an arbitrary player j's true payoff under strategy profile $\boldsymbol{\sigma}$.

Definition 1. *The ℓ_∞-norm between games G and $\hat{G}$ is given by*

$$\parallel G - \hat{G} \parallel_\infty = \max_{j \in N \setminus \{0\}, \boldsymbol{\sigma} \in \hat{\Sigma}} |U_j(\boldsymbol{\sigma}) - \hat{U}_j(\boldsymbol{\sigma})|.$$

If $\parallel G - \hat{G} \parallel_\infty \leq \varepsilon$, then $\hat{G}$ is said to be a unifor ε-approximation of G.

Note that in this definition, the maximization is only over the restricted set $\hat{\Sigma} \subseteq \Sigma$. An important consequence of $\hat{G}$ being a uniform approximation of G upon EGTA's termination is that a strategy profile that is an approximate Nash equilibrium in $\hat{G}$ is an approximate Nash equilibrium in G as well:

Proposition 3. *If $\hat{G}$ is a uniform ε-approximation of G and $\boldsymbol{\sigma}$ is a γ-Nash equilibrium of $\hat{G}$ for some $\gamma \geq 0$, then $\mathsf{Reg}_j(\boldsymbol{\sigma}) \leq 2\varepsilon + \gamma$ for each player $j \in N \setminus \{0\}$ upon the termination of EGTA.*

The main result of this section is that for a given EFG, under reasonable assumptions, TE-EGTA induces an empirical game model that is a tighter uniform approximation of the EFG than that induced by NF-EGTA, with a high probability. Given an arbitrary true game G, let $\hat{G}_{NF}$ and $\hat{G}_{TE}$ denote respectively the empirical game models induced by the application of NF-EGTA and TE-EGTA to G over the same restricted set $\hat{\Sigma}$.[2] We further assume an upper and a lower bound for all agent payoff samples returned by the simulator. Let c be the number of strategy profiles from the restricted set that, after each profile is sampled m times, result in a path taken through the tree that includes the first edge of $\varphi(t)$. c can be as small as 1 and as large as $O(|\Sigma_j|)$ for some $j \in N$ depending on the game structure and when the selected EGTA method terminates. Combined with Proposition 3, we have the following result, which also

[2] In the iterative application of EGTA, the NF- and TE- variants may produce different choices of strategies to add; hence, strategy sets covered at a given iteration number tend to diverge. However, for comparing model estimation accuracy, however, it makes sense to start with a common baseline of strategy space. Our experiments (Sect. 5) provide empirical corroboration that the benefits accrue as well when we examine the trajectory of models produced within the iterative PSRO framework.

implies a tighter upper bound for player regret in G under approximate equilibria in the empirical game model computed using payoffs estimated through TE-EGTA.

Theorem 1. *For any $\delta \in (0, 1)$ and the same number m of game simulation repetitions in each iteration of either type of EGTA, there exist positive constants ε_{NF} and ε_{TE} such that $\frac{\varepsilon_{TE}}{\varepsilon_{NF}} = \frac{1}{\sqrt{c}}$, and with probability at least $1 - \delta$ w.r.t. the randomness in the simulator payoff output, $\hat{G}_{NF}$ (respectively, $\hat{G}_{TE}$) is a uniform ε_{NF}-approximation (respectively, ε_{TE}-approximation) of G.*

5 Experiments

We conducted two sets of experiments comparing TE-EGTA with varying levels of tree structure exploitation to NF-EGTA. Each set used three different EFGs, chosen so that the corresponding empirical game models induced by our flexible tree-exploiting framework would vary in size and complexity. We implemented a simulator for each game that produced observations in accordance with the corresponding stochastic events, and end-state payoff samples that were normally distributed about the true utilities at the respective terminal nodes with a noise variance $\epsilon = 0.1$. The first game was Game$_1$ (Sect. 2.1). In our experiments, for each instance of Game$_1$, we randomly assigned $P(A \mid \pi_1^i)$ from $U[0, 1]$ for each $\pi_1^i \in \Pi_1$ and $u(t)$ from $\{0, 0.25, \ldots, 4.75, 5\}$ for each leaf utility. During each game play sample, the simulator returned the realized outcome A or B of the single stochastic event and a noisy payoff vector.

The second game was Game$_2$, an extension of Game$_1$ having a second stochastic event $e_2 \in \{C, D\}$ after Player 2's turn and a second turn for Player 1 afterward. Player 1 only observes its first action and the second event e_2. Thus Player 2 has 2 information sets whereas Player 1 has $1+2\cdot 10 = 21$. For its second turn, Player 1 has ten options depending on which outcome of e_2 it observed: $\Pi_{1C} = \{\pi_{1C}^i\}_{i=1}^{10}$ and $\Pi_{1D} = \{\pi_{1D}^i\}_{i=1}^{10}$. See the full version of this paper for an illustration. For each instance of Game$_2$ and each π_{2A}^i (respectively, π_{2B}^i), we sampled $P(C \mid A, \pi_{2A}^i)$ (respectively, $P(C \mid B, \pi_{2B}^i)$) from $U[0, 1]$. Each leaf utility was chosen uniformly at random from the set $\{0, 0.1, \ldots, 9.9, 10\}$. We experimented with two game model forms: one for when the simulator returned a noisy payoff vector and e_1 only, and one for when it returned the vector and outcomes of both events.

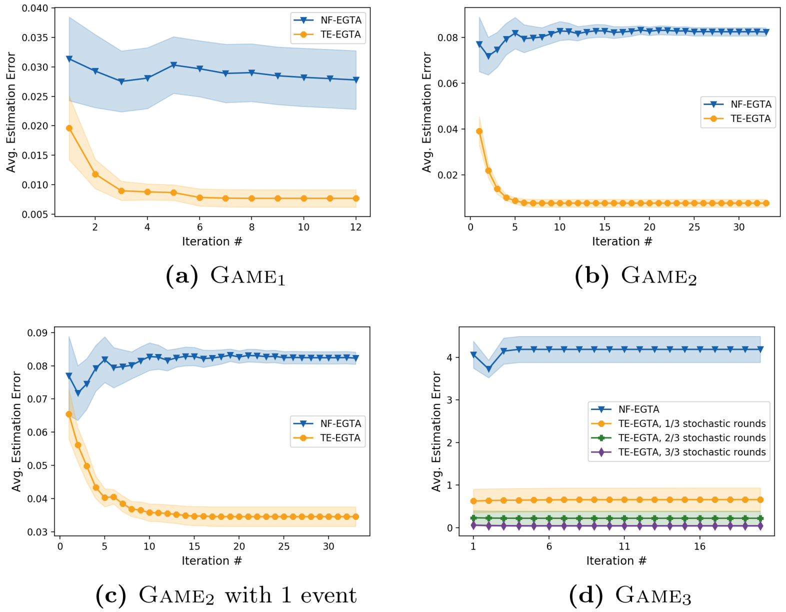
The final game was Game$_3$, which begins with a stochastic event $e_1 \in \{A, B, C, D\}$. Player 1 observes the event and then takes a turn, choosing one of four possible actions. Next, Player 2 observes the event (but not Player 1's action) and also chooses from four possible actions. This 3-round sequence is repeated twice, but in each subsequent sequence, the only outcomes available to Nature and the agents are the remaining ones that have not yet been chosen. For instance, if $e_1 = A$, then Nature can only output $e_2 \in \{B, C, D\}$ during its second turn and $e_3 \in \{B, C, D\} \setminus \{e_2\}$ during its third. Likewise, the players are restricted to the actions that they have not yet played in the previous 3-round sequence(s). Since the players are only unable to observe the other player's actions during the *current* 3-round sequence, each player has $4 + 4^3 \cdot 3 + 4^3 \cdot 3^3 \cdot 2 = 3652$ information sets. To compare the effects of varying degrees of tree exploitation, we examined three different game model forms: (1) simulator reports observation e_1 only; (2) simulator reports e_1 and e_2 only; and (3) simulator reports all three events. We believe that a model that includes only the first stochastic event would generally yield only a negligible difference in accuracy from a model that includes only the second (or third) stochastic event.

Each iteration of EGTA had a fixed budget of 500 total samples available for all strategy combinations to be fed into the simulator for Game_1 and Game_2. Due to the larger size, we allotted 5000 total samples for Game_3. We ran the experiments for Game_1 on a standard laptop (Quad-Core Intel Core i7 Processor, 2.7 GHz, 16 GB RAM). Each repetition of both TE-PSRO and NF-PSRO for Game_1 finished in less than 1 min. We ran the experiments for Game_2 and Game_3 on a single core of the Great Lakes Slurm cluster at the University of Michigan, with 786MB of memory. NF-PSRO on Game_2 consistently finished within 6 min, and took 4–90 min for Game_3. TE-PSRO required between 3 min and 5 h for Game_2 (depending on the MSS used, see Sect. 5.2), and at most 1 h for Game_3. All figures include the metrics' initial values at time-step 0.

5.1 TE-EGTA Payoff Estimation

The aim of the first set of experiments was to assess the improvement in strategy profile payoff estimation produced by incorporating the EFG tree structure into the empirical game model. We ran NF-EGTA and TE-EGTA on each true game with the same number $m = 500$ of simulations for each strategy-profile payoff vector estimation. To update the game model for either variant of EGTA, we implemented the PSRO framework using an oracle that returns the best response to the other player's strategy for Game_1 and Game_2. However, the size of Game_3 made a best response oracle infeasible, so we instead used Q-learning to compute an approximate best response from the true game. For newly selected strategy profiles that were simulated in each iteration, we computed estimated payoffs $\hat{U}_j^{NF}(\boldsymbol{\sigma})$ (resp. $\hat{U}_j^{TE}(\boldsymbol{\sigma})$) for NF-EGTA (resp. TE-EGTA) from accumulated simulation data using the approach described in Sect. 2.2 (resp. Sect. 3.1). We evaluated the *estimation error* for that iteration of either variant as the average absolute difference between true and estimated payoffs for all players over all strategy combinations in the current empirical game. We repeated this operation for 25 initial restricted sets, each consisting of a single randomly chosen policy, and reported the estimation error averaged over all 25 repetitions for each iteration of PSRO in Fig. 4.

As the plots show, TE-EGTA achieves significantly lower payoff estimation error compared to NF-EGTA across all games. It is also clear that while the vast number of infosets in Game_3 led NF-EGTA to perform worse as more strategy combinations were added despite an unchanging sample budget m; such was not the case for TE-EGTA, which converged very quickly. We attribute this to the relatively small number of actions (2, 3, or 4) available at each information set, as



(a) GAME$_1$

(b) GAME$_2$

(c) GAME$_2$ with 1 event

(d) GAME$_3$

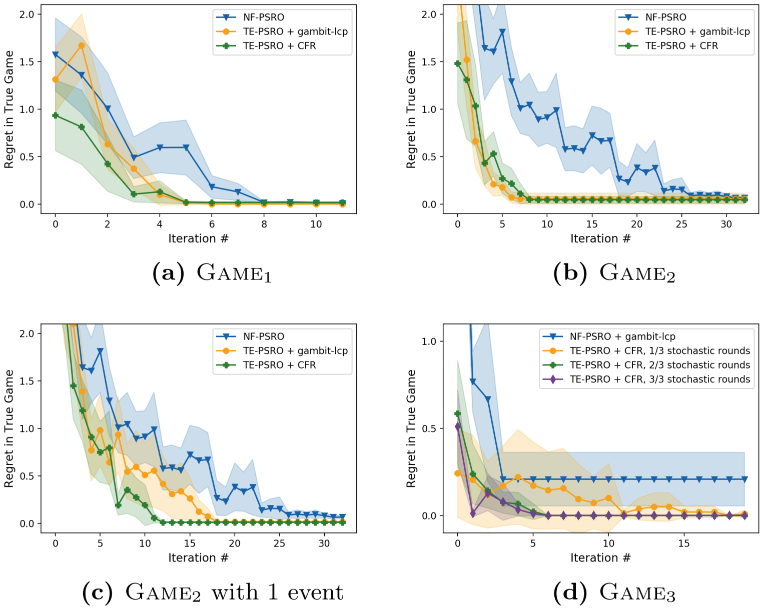
Fig. 4. Average estimation error of strategy payoffs over the course of EGTA's runtime. Shaded areas represent the standard error of the mean. The estimation errors at iteration 0 are identical since the restricted sets for both models contain the same randomly chosen policy; hence, they are omitted.

well as the large number of infosets relative to the total number of game paths. Q-learning returned a best response for every infoset that could be reached, given $\boldsymbol{\sigma}$, so the empirical game ceased growing after only a few iterations. Finally, we note that the more stochastic events included in $\hat{G}$, the more tree structure is exploited by TE-EGTA, and the lower the resulting payoff error. In fact, the inclusion of even a single stochastic event or round in the model dramatically decreased the payoff error in comparison to NF-EGTA.

5.2 Iterative Model Refinement in PSRO

Our second set of experiments compared the power of NF-PSRO and TE-PSRO to iteratively explore the EFG's strategy space and fine-tune their respective empirical game models. PSRO terminates once no new best responses can be added to $\hat{\Sigma}$. To evaluate the efficacy of this iterative fine-tuning, we computed the regret $\mathsf{Reg}(\boldsymbol{\sigma})$ (as defined in Sect. 2.1) in the true game G of the solution $\boldsymbol{\sigma}$ returned by the MSS in every iteration.

For NF-PSRO, we used the Python-Gambit interface to represent the empirical game and used Gambit's `lcp` solver as the MSS. The solver takes as input an NFG or EFG, converts it into a linear complementarity program, and solves for all NE. We also used the `lcp` as the TE-PSRO solver for GAME$_1$ and GAME$_2$. It



(a) GAME$_1$ **(b)** GAME$_2$

(c) GAME$_2$ with 1 event **(d)** GAME$_3$

Fig. 5. Average regret of solution profiles over the course of PSRO's runtime. Shaded areas represent standard error of the mean.

is important to note that Gambit's solvers can become intractable for medium or large game trees. However, when possible, we intentionally chose an MSS that finds exact solutions to the empirical game in order to minimize any error/variability in the solutions resulting from the iterative process of adding strategies and fine-tuning the empirical game models. For medium-to-large game trees like GAME$_3$, we used counterfactual regret minimization (CFR) (Zinkevich et al. 2007) to find an approximate NE and Q-learning to learn an approximate best response from the true game. We used CFR as the MSS for GAME$_2$ as well for comparison to the exact `lcp` solver. As in Sect. 5.1, we repeated PSRO for 25 different restricted sets, each consisting of a single, randomly chosen strategy profile. We report the regret curves, averaged over 25 repetitions, in Fig. 5.

TE-PSRO converged on average to a regret at least as tight as NF-PSRO using the same simulation budget and regardless of which pure $\boldsymbol{\sigma}$ the initial restricted set contained. It also converged in fewer iterations, particularly in GAME$_2$ and GAME$_3$ as more tree structure was included in $\hat{G}$. Additional plots in the full version of this paper demonstrate the same result for different numbers of samples. However, the standard error shadings for GAME$_1$ overlap mainly due to the high volatility in NF-PSRO regret in earlier iterations. Since, in each iteration, we add new, pertinent best responses to $\hat{\Sigma}$, we hypothesize that their absence from the previous strategy space caused the regret to increase. A one-sided two-sample t-test on each of the iterations of GAME$_1$'s regret curves established that TE-PSRO's regret improvement was statistically significant.

These results suggest that including even some tree structure in $\hat{G}$ results in PSRO converging at least as quickly and to a solution that has lower regret in the true game.

6 Conclusions and Future Work

This study represents a first step towards the goal of leveraging extensive-form structure within the EGTA framework. Our work complements prior research that showed benefits of exploiting tree structure in game reasoning and learning, for example studies that demonstrated advantages of extensive form in techniques based on the double oracle algorithm (Bošanský et al. 2014; McAleer et al. 2021). In future work, we hope to draw on further insights from this line of work, combining the best features of techniques from game reasoning, machine learning, and simulation-based game modeling. One particularly fruitful direction may be consideration of strategy exploration methods that explicitly consider extensive structure in the currently defined strategy space.

Acknowledgments. This work was supported in part by a grant from the Effective Altruism Foundation, and by the US National Science Foundation under CRII Award 2153184.

References

Areyan Viqueira, E., Cousins, C., Greenwald, A.: Improved algorithms for learning equilibria in simulation-based games. In: 19th International Conference on Autonomous Agents and Multi-Agent Systems (2020)

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

Bošanský, B., Kiekintveld, C., Lisý, V., Pěchouček, M.: An exact double-oracle algorithm for zero-sum extensive-form games with imperfect information. J. Artif. Intell. Res. **51**, 829–866 (2014)

Fearnley, J., Gairing, M., Goldberg, P., Savani, R.: Learning equilibria of games via payoff queries. J. Mach. Learn. Res. **16**, 1305–1344 (2015)

Gatti, N., Panozzo, F., Restelli, M.: Efficient evolutionary dynamics with extensive-form games. In: 27th AAAI Conference on Artificial Intelligence (2013)

Heinrich, J., Lanctot, M., Silver, D.: Fictitious self-play in extensive-form games. In: 32nd International Conference on Machine Learning (2015)

Johanson, M., Bard, N., Burch, N., Bowling, M.: Finding optimal abstract strategies in extensive-form games. In: 26th AAAI Conference on Artificial Intelligence (2012)

Jordan, P.R., Vorobeychik, Y., Wellman, M.P.: Searching for approximate equilibria in empirical games. In: 7th International Conference on Autonomous Agents and Multi-Agent Systems (2008)

Jordan, P.R., Schvartzman, L.J., Wellman, M.P.: Strategy exploration in empirical games. In: 9th International Conference on Autonomous Agents and Multi-Agent Systems, pp. 1131–1138 (2010)

Koller, D., Megiddo, N., von Stengel, B.: Efficient computation of equilibria for extensive two-person games. Games Econom. Behav. **14**, 247–259 (1996)

Kroer, C., Sandholm, T.: A unified framework for extensive-form game abstraction with bounds. In: 32nd Conference on Neural Information Processing Systems (2018)

Lanctot, M., et al.: A unified game-theoretic approach to multiagent reinforcement learning. In: 31st Annual Conference on Neural Information Processing Systems (2017)

Leibo, J.Z., Zambaldi, V., Lanctot, M., Marecki, J., Graepel, T.: Multi-agent reinforcement learning in sequential social dilemmas. In: 16th International Conference on Autonomous Agents and Multi-Agent Systems (2017)

Lockhart, E., et al.: Computing approximate equilibria in sequential adversarial games by exploitability descent. In: 28th International Joint Conference on Artificial Intelligence (2019)

McAleer, S., Lanier, J., Wang, K.A., Baldi, P., Fox, R.: XDO: a double oracle algorithm for extensive-form games. In: 35th Annual Conference on Neural Information Processing Systems (2021)

McMahan, H.B., Gordon, G.J., Blum, A.: Planning in the presence of cost functions controlled by an adversary. In: 20th International Conference on Machine Learning, pp. 536–543 (2003)

Phelps, S., Marcinkiewicz, M., Parsons, S., McBurney, P.: 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, pp. 705–712 (2006)

Sandholm, T.: The state of solving large incomplete-information games, and application to poker. AI Mag. **31**(4), 13–32 (2010)

Schvartzman, L.J., Wellman, M.P.: Stronger CDA strategies through empirical game-theoretic analysis and reinforcement learning. In: 8th International Conference on Autonomous Agents and Multi-Agent Systems, pp. 249–256 (2009)

Sokota, S., Ho, C., Wiedenbeck, B.: Learning deviation payoffs in simulation-based games. In: 33rd AAAI Conference on Artificial Intelligence (2019)

Tuyls, K.: Bounds and dynamics for empirical game-theoretic analysis. Auton. Agents Multi-Agent Syst. **34**(7) (2020)

Vorobeychik, Y.: Probabilistic analysis of simulation-based games. ACM Trans. Modeling Comput. Simul. **20**(3), 16:1–16:25 (2010)

Vorobeychik, Y., Wellman, M.P., Singh, S.: Learning payoff functions in infinite games. Mach. Learn. **67**, 145–168 (2007)

Wang, Y.: Deep reinforcement learning for green security games with real-time information. In: 33rd AAAI Conference on Artificial Intelligence (2019)

Wang, Y., Ma, Q., Wellman, M.P.: Evaluating strategy exploration in empirical game-theoretic analysis. In: 21st International Conference on Autonomous Agents and Multi-Agent Systems (2022)

Wellman, M.P.: Putting the agent in agent-based modeling. Auton. Agent. Multi-Agent Syst. **30**(6), 1175–1189 (2016). https://doi.org/10.1007/s10458-016-9336-6

Wellman, M.P.: Economic reasoning from simulation-based game models. Œconomia **10**, 257–278 (2020)

Zhang, B.H., Sandholm, T.: Small Nash equilibrium certificates in very large games. In: 34th Annual Conference on Neural Information Processing Systems (2020)

Zinkevich, M., Johanson, M., Bowling, M.H., Piccione, C.: Regret minimization in games with incomplete information. In: 21st Conference on Neural Information Processing Systems (2007)