# Improved Algorithms for Learning Equilibria in Simulation-Based Games

Enrique Areyan Viqueira
Brown University
Providence, Rhode Island
eareyan@brown.edu

Cyrus Cousins
Brown University
Providence, Rhode Island
cyrus_cousins@brown.edu

Amy Greenwald
Brown University
Providence, Rhode Island
amy_greenwald@brown.edu

## ABSTRACT

We tackle a fundamental problem in empirical game-theoretic analysis (EGTA), that of learning equilibria of simulation-based games. Such games cannot be described in analytical form; instead, a black-box simulator can be queried to obtain noisy samples of utilities. Our first theorem establishes that uniform approximations of simulation-based games are equilibrium preserving. We then design algorithms that uniformly approximate simulation-based games with finite-sample guarantees. Our first algorithm, global sampling (GS), extends previous work that constructs confidence intervals assuming bounded utilities with confidence intervals that are sensitive to variance. The second, progressive sample with pruning (PSP), samples progressively, ceasing the sampling process (i.e., pruning strategies) as soon as it determines that the corresponding utilities have been sufficiently well estimated for equilibrium computation. We experiment with our algorithms using both GAMUT, a state-of-the-art game generator, and Gambit, a state-of-the-art game solver. For a broad swath of games, we show that GS using our variance-sensitive bounds outperforms previous work, and that PSP can significantly outperform GS. Here "outperform" means achieving the same guarantees with far fewer samples.

## KEYWORDS

Empirical Game-Theoretical Analysis; PAC Learning

**ACM Reference Format:**
Enrique Areyan Viqueira, Cyrus Cousins, and Amy Greenwald. 2020. Improved Algorithms for Learning Equilibria in Simulation-Based Games. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 10 pages.

## 1 INTRODUCTION

Game theory is the *de facto* standard conceptual framework used to analyze the strategic interactions among rational agents in multi-agent systems. At the heart of this framework is the theoretical notion of a game. In a game, each player (also referred to as an agent) chooses a strategy and earns a utility that in general depends on the profile (i.e., vector) of strategies chosen by all the agents.

The most basic representation of a game is the *normal form*, which can be visualized as a matrix containing all agents' utilities at all strategy profiles. Analyzing a game means predicting its outcome, i.e., the strategy profiles that one can reasonably expect the agents to play. Perhaps the most common such prediction is that of *Nash equilibrium*, where each agent plays a *best-response* to the strategies of the others: i.e., a strategy that maximizes their utility [20]. More generally, at an $\varepsilon$-Nash equilibrium agents best respond to other agents' strategies up to an additive error of $\varepsilon$.

This paper is concerned with analyzing games for which a complete and accurate description is not available. While knowledge of the number of agents and their respective strategy sets is available, we do not assume *a priori* access to the game's utility function. Instead, we assume access to a simulator from which we can sample noisy utilities associated with any strategy profile. Such games have been called *simulation-based games* [33] and *black-box games* [22], and their analysis is called *empirical game theoretic analysis* (EGTA) [13, 34]. EGTA methodology has been applied in a variety of practical settings for which simulators are readily available, including trading agent analyses in supply chains [13, 32], ad auctions [1, 15], and energy markets [16]; designing network routing protocols [35]; strategy selection in real-time games [27]; and the dynamics of RL algorithms, like AlphaGo [28].

The aim of this work is to design learning algorithms that can accurately estimate all the equilibria of simulation-based games. We tackle this problem using the *probably approximately correct* (PAC) learning framework [29]. Our algorithms learn so-called *empirical games* [34], which are estimates of simulation-based games constructed via sampling. We argue that empirical games so constructed yield *uniform approximations* of simulation-based games, meaning all utilities in the empirical game tend toward their expected counterparts, *simultaneously.*

This notion of uniform approximation is central to our work: we prove that, when one game $\Gamma'$ is a uniform approximation of another $\Gamma''$, all equilibria in $\Gamma'$ are approximate equilibria in $\Gamma''$. In particular, when an estimated game $\hat{\Gamma}$ uniformly approximates a (true) game $\Gamma$, all equilibria in $\Gamma$ are approximate equilibria in $\hat{\Gamma}$, and all equilibria in $\hat{\Gamma}$ are approximate equilibria in $\Gamma$. As a result, a uniform approximation implies perfect recall[1] (*all* true positives—equilibria in $\Gamma$—are at least approximate equilibria in $\hat{\Gamma}$) and *approximately* perfect precision (all false positives—equilibria in $\hat{\Gamma}$ but not necessarily in $\Gamma$—are at least approximate equilibria in $\Gamma$). Our learning algorithms, which learn empirical games that are uniform approximations of simulation-based games, thus well estimate the equilibria of simulation-based games.

Estimating *all* the utilities in an empirical game is non-trivial, in part because of the *multiple comparisons problem* (MCP), which arises when estimating many parameters simultaneously, since accurate inferences for each individual parameter do not necessarily imply a similar degree of accuracy for the parameters in aggregate.

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

[1] We use the term *recall* in the information retrieval sense. The juxtaposition of the two words "perfect" and "recall" is not a reference to extensive-form games.

Controlling the *family-wise error rate* (FWER)—the probability that one or more of the approximation guarantees is violated—is one way to control for multiple comparisons. We control the FWER in empirical games by first using a concentration inequality to establish confidence intervals about each parameter *individually*, and then applying a statistical correction (e.g., Bonferroni; in other words, a union bound) to bound the probability that all approximation guarantees hold *simultaneously*.

The first concentration inequality we apply is Hoeffding's [1963]. We note that Hoeffding's inequality implies sub-Gaussian tail bounds for averages of $m$ $c$-bounded random variables, yielding $1 - \delta$ probability confidence intervals of width $\Theta\sqrt{c^2 \ln(1/\delta)/m}$. Hoefdding's inequality, however, is sensitive only to the range $c$ of the random variables, not their variance $\sigma^2$. Bennett's inequality [1962] relaxes the dependency on $c$, replacing it with a dependency on $\sigma^2 < c^2$, yielding width $\Theta\left(\frac{c \ln(1/\delta)}{m} + \sqrt{\frac{\sigma^2 \ln(1/\delta)}{m}}\right)$; i.e., a sub-gamma tail bound. Tighter still would be to use Gaussian tail bounds, which yield $\Theta\sqrt{\sigma^2 \ln(1/\delta)/m}$. However, the central limit theorem does not provide finite sample guarantees, and any approximation errors are amplified in a MCP setting; when applied to games, a single mistake can lead to a cascade of mistakes in equilibrium estimation.

Thus, we resort to sub-gamma tail bounds, which still drastically improve the situation because of the presence of a fast-decaying, hyperbolic $O(1/m)$ term in addition to the usual root-hyperbolic $O(1/\sqrt{m})$ term. The fast-decaying range term is initially the dominant factor, but it is quickly overcome by the variance term. When the variance is substantially less than $c^2$, these variance-sensitive bounds are substantially tighter than Hoeffding's. They are also of particular interest when only a loose bound on $c$ is available, or in cases where extreme outliers are unlikely albeit possible, thus avoiding the pitfalls of the central limit theorem. In both cases, $c$ rigorously controls for the possibility of a rare event.

As the variance is not generally known, we present a 2-step strategy wherein the variance is upper-bounded in terms of the empirical variance,and the expectation is then upper-bounded by Bennett's inequality in terms of this upper bound, appropriately corrected for the MCP. Analogous to the use of Hoeffding's bound in [28], our new bound (on the expectation) immediately gives rise to a novel algorithm by which to learn an empirical game from a static sample of utilities. We refer to this algorithm generically as *global sampling* (GS), and instantiate with both bounds.

We also describe a second learning algorithm, *progressive sampling with pruning* (PSP), which samples dynamically. Prior art [9, 24, 25] uses progressive sampling to obtain a desired accuracy given a failure probability by guessing an initial sample size, computing Rademacher bounds, and repeating with larger sample sizes until said accuracy is attained. Our work complements this approach with *pruning*: at each iteration, parameters that have already been sufficiently well estimated for the task at hand (here, equilibrium estimation) are pruned as subsequent iterations of PSP do not refine their bounds. Pruning represents both a statistical and computational improvement over earlier progressive sampling techniques.

Finally, we report on extensive experiments designed to evaluate the performance of our algorithms. Producing a robust empirical evaluation is a daunting task, as the space of possible games is vast. Moreover, computing equilibria is computationally intractable [7]. Fortunately, researchers have devised tools to address these issues. The first, GAMUT, is a state-of-the-art suite of game generators capable of producing a myriad of games with rich strategic structure [21]. The second, Gambit, is a state-of-the-art solver to compute Nash equilibria [19]. We use both GAMUT and Gambit to evaluate the performance of our algorithms. Concretely, for a wide variety of game types, we draw multiple random game instances from GAMUT, and for each such instance, we solve for its equilibria using Gambit. Using our variance-sensitive bounds within GS outperforms Hoeffding's; and PSP, with either bound, can significantly outperform the corresponding GS algorithm. Here "outperform" means achieving the same guarantees with far fewer samples—exponentially fewer for PSP in sufficiently large games (in our experiments, two players with ten or more strategies each).

This paper purports to contribute to the literature on EGTA, a methodology for the analysis of multi-agent systems. One important application of our methodology is the estimation of equilibria in meta-games [27]. *Meta-games* are simplified versions of intractably large games, where, instead of modeling every possible strategy an agent might implement, one analyzes a game with a substantially reduced set of strategies, each of which may be given by a complicated algorithmic procedure (i.e., a heuristic). That is, instead of analyzing a game in terms of its (low-level) formal game-theoretic strategies—a computationally intractable task in a game such as Go—one might analyze a reduced version of the game where agents play according to higher-level strategies given by reinforcement learning algorithms: e.g., variants of AlphaGo [26]. Simulation is in order; and since each run of the game can result in either agent winning (depending on various stochastic elements, including perhaps the agents' strategies), utilities are noisy in general. Our techniques are directly applicable to the learning of empirical meta-games, and provide finite-sample guarantees on the quality of the equilibria of the corresponding simulation-based meta-games.

*Related Work.* One distinctive feature of our work *vis à vis* the literature is that we aim to estimate all equilibria of a simulation-based game, rather than just one (e.g., [14]). Notable exceptions include [28], [31], [36], and [2, 30], the latter being our own prior work. The first of these papers argues that all equilibria of a simulation-based game are also approximate equilibria in an empirical game: i.e., they establish perfect recall with finite-sample guarantees. The second paper derives asymptotic results about the quality of the equilibria of empirical games: i.e., they establish perfect precision in the limit, as the number of samples tends to infinity. Our first theorem unifies these two results, obtaining finite sample guarantees for both perfect recall and approximate precision. The third paper is perhaps closest in spirit to ours; the goals are to characterize the quality of the equilibria in empirical games, and to exploit statistical information to save on sampling. Their methods employ bootstrapping [8], so do not immediately afford any theoretical guarantees.

## 2 APPROXIMATION FRAMEWORK

We begin by presenting standard game-theoretic notions. We then introduce the notion of uniform approximation. Given an approximation of one game by another, there is not necessarily a connection between their properties. For example, there may be equilibria in one game with no corresponding equilibria in the other, as small changes to the utility functions can add or remove equilibria. Nonetheless, we show that finding the approximate equilibria of a uniform approximation of a game is sufficient for finding all the (exact) equilibria of the game itself.

A *normal-form game* $\Gamma \doteq \langle P, \{S_p \mid p \in P\}, \boldsymbol{u}(\cdot)\rangle$ consists of a set of agents $P$, with *pure strategy set* $S_p$ available to agent $p \in P$. We define $\boldsymbol{S} \doteq S_1 \times \cdots \times S_{|P|}$ to be the pure strategy profile space of $\Gamma$, and then $\boldsymbol{u} : \boldsymbol{S} \to \mathbb{R}^{|P|}$ is a vector-valued utility function.

Given a normal-form game $\Gamma$, we let $S_p^\diamond$ denote the set of distributions over $S_p$; this set contains $p$'s *mixed strategies.* We define $\boldsymbol{S}^\diamond = S_1^\diamond \times \cdots \times S_{|P|}^\diamond$, and then, overloading notation, we write $\boldsymbol{u}(\boldsymbol{s})$ to denote the expected utility of a mixed strategy profile $\boldsymbol{s} \in \boldsymbol{S}^\diamond$.

A solution to a normal-form game is a prediction of how strategic agents will play the game. One solution concept that has received a great deal of attention in the literature is Nash equilibrium [20], a (pure or mixed) strategy profile at which each agent selects a utility-maximizing strategy, fixing all other agents' strategies. In this paper, we are concerned with $\varepsilon$-Nash equilibrium, an approximation of Nash equilibrium that is amenable to statistical estimation.

Given a game $\Gamma$, fix an agent $p$ and a mixed strategy profile $\boldsymbol{s} \in \boldsymbol{S}^\diamond$. We define $\boldsymbol{T}_{p,\boldsymbol{s}}^\diamond \doteq \{\boldsymbol{t} \in \boldsymbol{S}^\diamond \mid \boldsymbol{t}_q = \boldsymbol{s}_q, \forall q \neq p\}$: i.e., the set of all mixed strategy profiles in which the strategies of all agents $q \neq p$ are fixed at $\boldsymbol{s}_q$. Agent $p$'s *regret* at $\boldsymbol{s}$ is defined as $\operatorname{Reg}_p^\diamond(\Gamma, \boldsymbol{s}) \doteq \sup_{\boldsymbol{s}' \in \boldsymbol{T}_{p,\boldsymbol{s}}^\diamond} \boldsymbol{u}_p(\boldsymbol{s}') - \boldsymbol{u}_p(\boldsymbol{s})$. By restricting $\boldsymbol{s}$ and $\boldsymbol{T}_{p,\boldsymbol{s}}^\diamond$ to pure strategy profiles, agent $p$'s *pure regret* $\operatorname{Reg}_p(\Gamma, \boldsymbol{s})$ can be defined similarly.

Note that $\operatorname{Reg}_p^\diamond(\Gamma, \boldsymbol{s}), \operatorname{Reg}_p(\Gamma, \boldsymbol{s}) \geq 0$, since agent $p$ can deviate to any strategy $s' \in S_p$, including $\boldsymbol{s}_p$ itself. A strategy profile $\boldsymbol{s}$ that has regret at most $\varepsilon \geq 0$, for all $p \in P$, is an $\varepsilon$-Nash equilibrium:

Given $\varepsilon \geq 0$, a mixed strategy profile $\boldsymbol{s} \in \boldsymbol{S}^\diamond$ in a game $\Gamma$ is an *$\varepsilon$-Nash equilibrium* if, for all $p \in P$, $\operatorname{Reg}_p^\diamond(\Gamma, \boldsymbol{s}) \leq \varepsilon$. At a pure strategy $\varepsilon$-Nash equilibrium $\boldsymbol{s} \in \boldsymbol{S}$, for all $p \in P$, $\operatorname{Reg}_p(\Gamma, \boldsymbol{s}) \leq \varepsilon$. Let $\mathrm{E}_\varepsilon^\diamond(\Gamma)$ ($\mathrm{E}_\varepsilon(\Gamma)$) be the set of mixed (pure) $\varepsilon$-Nash equilibria. $\mathrm{E}_\varepsilon(\Gamma) \subseteq \mathrm{E}_\varepsilon^\diamond(\Gamma)$.

We now show that equilibria can be approximated with bounded error, given a uniform approximation. Our main theorem establishes perfect recall: the approximate game contains all true positives: i.e., all (exact) equilibria of the original game. It also establishes approximately perfect precision: all false positives in the approximate game are approximate equilibria in the original game.

We define the *$\ell_\infty$-norm* between two compatible games, with the same agents sets $P$ and strategy profile spaces $\boldsymbol{S}$, and with utility functions $\boldsymbol{u}, \boldsymbol{u}'$, respectively, as follows: $\|\Gamma - \Gamma'\|_\infty \doteq \|\boldsymbol{u}(\cdot) - \boldsymbol{u}'(\cdot)\|_\infty \doteq \sup_{p \in P, \boldsymbol{s} \in \boldsymbol{S}} |\boldsymbol{u}_p(\boldsymbol{s}) - \boldsymbol{u}'_p(\boldsymbol{s})|$. While the $\ell_\infty$-norm as defined applies only to pure normal-form games, it is in fact sufficient to use this metric even to show that the utilities of mixed strategy profiles approximate one another. We formalize this claim presently.[2]

Lemma 2.1. *If $\Gamma, \Gamma'$ differ only in their utilities, then* $\sup_{p \in P, \boldsymbol{s} \in \boldsymbol{S}^\diamond} |\boldsymbol{u}_p(\boldsymbol{s}) - \boldsymbol{u}'_p(\boldsymbol{s})| = \|\Gamma - \Gamma'\|_\infty$.

$\Gamma'$ is said to be a *uniform $\varepsilon$-approximation* of $\Gamma$ when $\|\Gamma - \Gamma'\|_\infty \leq \varepsilon$. They are so-called because the bound between utility deviations in $\Gamma$ and $\Gamma'$ holds *uniformly* over *all* players and strategy profiles.

Theorem 2.2 (Approximate Equilibria). *If two normal-form games, $\Gamma$ and $\Gamma'$, are uniform approximations of one another, then: (1)* $\mathrm{E}(\Gamma) \subseteq \mathrm{E}_{2\varepsilon}(\Gamma') \subseteq \mathrm{E}_{4\varepsilon}(\Gamma)$, *and (2)* $\mathrm{E}^\diamond(\Gamma) \subseteq \mathrm{E}_{2\varepsilon}^\diamond(\Gamma') \subseteq \mathrm{E}_{4\varepsilon}^\diamond(\Gamma)$.

## 3 LEARNING FRAMEWORK

In this section, we move on from approximating equilibria in games to learning them. We present algorithms that learn so-called empirical games, which comprise estimates of the expected utilities of simulation-based games. We further derive uniform convergence bounds, proving that our algorithms output empirical games that uniformly approximate their expected counterparts, with high probability. By Theorem 2.2, the equilibria of these empirical games thus approximate those of the corresponding simulation-based games, with high probability.

A *conditional normal-form game* $\Gamma_\mathcal{X} \doteq \langle \mathcal{X}, P, \{S_p \mid p \in P\}, \boldsymbol{u}(\cdot)\rangle$ consists of a set of conditions $\mathcal{X}$, a set of agents $P$, with pure strategy set $S_p$ available to agent $p$, and a vector-valued conditional utility function $\boldsymbol{u} : \boldsymbol{S} \times \mathcal{X} \to \mathbb{R}^{|P|}$. It is convenient to imagine a condition $x \in \mathcal{X}$ as pertaining to the set $P \times \boldsymbol{S}$. Given such a condition, $\boldsymbol{u}(\cdot; x)$ yields a standard utility function of the form $\boldsymbol{S} \to \mathbb{R}^{|P|}$. Given a conditional normal-form game $\Gamma_\mathcal{X}$ together with distribution $\mathscr{D}$, we also define the *expected utility function* $\boldsymbol{u}(\boldsymbol{s}; \mathscr{D}) = \mathbb{E}_{x \sim \mathscr{D}}[\boldsymbol{u}(\boldsymbol{s}; x)]$, and the *expected normal-form game* as $\Gamma_\mathscr{D} \doteq \langle P, \{S_p \mid p \in P\}, \boldsymbol{u}(\cdot; \mathscr{D})\rangle$.

Expected normal-form games serve as our mathematical model of simulation-based games. They are sufficient not only to model arbitrary black-box games, but additionally games where the *rules* are known but *environmental conditions* are random (e.g., auctions where bidder valuations are random, or *deterministic* war games where initial armies and terrain are random), as well as games with *randomness*, where $\mathcal{X}$ is taken to be a *PRNG seed* or *entropy source.*

Given a conditional normal-form game $\Gamma_\mathcal{X}$ together with a distribution $\mathscr{D}$ and sample conditions $\boldsymbol{X} = (x_1, \ldots, x_m) \sim \mathscr{D}^m$, we define the *empirical utility function* $\hat{\boldsymbol{u}}(\boldsymbol{s}; \boldsymbol{X}) \doteq \frac{1}{m} \sum_{j=1}^m \boldsymbol{u}(\boldsymbol{s}; x_j)$. The corresponding *empirical normal-form game* is then $\hat{\Gamma}_\boldsymbol{X} \doteq \langle P, \{S_p \mid p \in P\}, \hat{\boldsymbol{u}}(\cdot; \boldsymbol{X})\rangle$.

Our present goal, then, is to "uniformly learn" empirical games (i.e., obtain uniform convergence guarantees) from finitely many samples. This learning problem is non-trivial because it involves multiple comparisons. Tuyls et al. [28] use Hoeffding's inequality to estimate a single utility value, and then apply a Šidák correction to estimate all utility values simultaneously, assuming independence among agents' utilities. Similarly, one can apply a Bonferroni correction (i.e., a union bound) to Hoeffding's inequality, which does not require independence, but yields a slightly looser bound.

Theorem 3.1 (Finite-Sample Bounds via Hoeffding's Inequality). *Consider finite, conditional normal-form game $\Gamma_\mathcal{X}$ together with distribution $\mathscr{D}$ and index set $\boldsymbol{I} \subseteq P \times \boldsymbol{S}$ such that for all $x \in \mathcal{X}$ and $(p, \boldsymbol{s}) \in \boldsymbol{I}$, it holds that $\boldsymbol{u}_p(\boldsymbol{s}; x) \in [-c/2, c/2]$, where $c \in \mathbb{R}$. Then, with probability at least $1 - \delta$ over $\boldsymbol{X} \sim \mathscr{D}^m$, it holds that*

$$\sup_{(p,\boldsymbol{s}) \in \boldsymbol{I}} \left|\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})\right| \leq c\sqrt{\frac{\ln(\frac{2|\boldsymbol{I}|}{\delta})}{2m}} \ .$$

**Remark** Given a game, we state all theorems and algorithms for an arbitrary index set $\boldsymbol{I}$. Taking $\boldsymbol{I} = P \times \boldsymbol{S}$, we bound $\|\boldsymbol{u}(\cdot; \mathscr{D}) - \hat{\boldsymbol{u}}(\cdot; \boldsymbol{X})\|_\infty$.

Consider $X_{1:m}$ i.i.d. random variables, and their mean $\bar{X}$. (In our case, $X_j = \boldsymbol{u}(\boldsymbol{s}; x_j)$.) Hoeffding's inequality for bounded random variables can be used to obtain tail bounds on the probability that an empirical mean differs from its expectation. One way to characterize such bounds is to compare them to well-studied cases of common

[2] All proofs appear in the supplemental material, http://github.com/eareyan/pysegta.

random variables. We focus on the case of upper tails, as we apply only symmetric two-tail and upper tail bounds in this work.

Random variables that obey the Gaussian Chernoff bound can be characterized as $\sigma_{\mathcal{N}}^2$*-sub-Gaussian*: i.e.,

$$\mathbb{P}\left(\bar{X} \geq \mathbb{E}[\bar{X}] + \varepsilon\right) \leq \exp\left(\frac{-m\varepsilon^2}{2\sigma_{\mathcal{N}}^2}\right) \ ; \text{ equivalently,}$$

$$\mathbb{P}\left(\bar{X} \geq \mathbb{E}[\bar{X}] + \sqrt{\frac{2\sigma_{\mathcal{N}}^2 \ln(\frac{1}{\delta})}{m}}\right) \leq \delta \ .$$

Here, $\sigma_{\mathcal{N}}^2$ is deemed a *variance proxy*. Using this characterization, Hoeffding's inequality reads "If $X_i$ has range $c$, then $X_i$ is $\frac{c^2}{4}$-sub-Gaussian," because, by Popoviciu's inequality [1935], $\mathbb{V}[X_i] \leq \frac{c^2}{4}$. Thus, Hoeffding's inequality yields sub-Gaussian tail bounds.

This result is not entirely satisfying, however, as it is stated in terms of the *worst-case* variance; when $\mathbb{V}[X_i] \ll \frac{c^2}{4}$, tighter bounds should be possible. We might hope that knowledge of the variance $\sigma^2$ would imply $\sigma^2$-sub-Gaussian, but it does not; taking the range $c$ to $\infty$ allows $X_i$ to exhibit arbitrary tail behaviors. A $(\sigma_\Gamma^2, c)$*-sub-gamma* [6] random variable obeys

$$\mathbb{P}\left(\bar{X} \geq \mathbb{E}[\bar{X}] + \frac{c\ln(\frac{1}{\delta})}{3m} + \sqrt{\frac{2\sigma_\Gamma^2 \ln(\frac{1}{\delta})}{m}}\right) \leq \delta \ .$$

While the form of such bounds is more complicated than in the sub-Gaussian case, there is an intuitive interpretation of the tail behavior as sample size increases. The key is to observe that the additive error consists of a *hyperbolic* $\frac{c\ln(\frac{1}{\delta})}{3m}$ term and a *root-hyperbolic* $\sqrt{2\sigma_\Gamma^2 \ln(\frac{1}{\delta})/m}$ term, which in learning theory are often called *fast* and *slow* terms, respectively. Sub-gamma random variables then yield *mixed convergence rates*, which initially decay quickly while the $c$-term dominates, before slowing to the root-hyperbolic rate when the $\sigma_\Gamma^2$ term comes to dominate.

Bennett's inequality [1962], while usually stated as a sub-Poisson bound, immediately implies that if $X_i$ has range $c$ and variance $\sigma^2$, then $X_i$ is $(\sigma^2, c)$-sub-gamma. Bennett's inequality, however, assumes the range and variance are *known*. Various *empirical* Bennett bounds have been shown [3, 4, 18], which all essentially operate by bounding the variance of a random variable in terms of its empirical variance and range, and then applying Bennett's inequality. Simply put, Bennett's inequality gives us Gaussian-like tail bounds, where the scale-dependent term acts as an *asymptotically negligible* correction for working with non-Gaussian distributions and *empirical estimates* of variance. Asymptotic central-limit-theorem bounds behave similarly, but lack corresponding finite-sample guarantees.

Our work differs from previous applications in that we require confidence intervals of *uniform width*, and thus our bounds are limited by the *maximum* variance over all parameters being estimated. The maximum variance over a set of random variables is known as the *wimpy variance* [6]. We apply a sub-gamma bound to the wimpy variance in terms of an empirical estimate, and then apply Bennett's inequality to the upper and lower tails of each utility function using the wimpy variance bound, as this, by definition, upper bounds the variance of each utility. This strategy yields uniform-width confidence intervals over all utilities, and requires a union bound over $m$ upper tails, $m$ lower tails, and the wimpy variance, whereas bounding each variance separately would require a union bound over $m$ upper, $m$ lower, and $m$ variance bounds. Thus, our method outperforms such strategies, yielding tighter confidence intervals.

Theorem 3.2 (Bennett-Type Variance-Sensitive Finite-Sample Bounds). *Suppose as in Thm. 3.1. Take*

$$\hat{v} \doteq \sup_{(p,\boldsymbol{s})\in\boldsymbol{I}} \frac{1}{m-1}\sum_{j=1}^{m} \Big(\boldsymbol{u}_p(\boldsymbol{s}; x_j) - \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})\Big)^2 \ ;$$

$$\varepsilon_v \doteq \frac{c\ln(\frac{3}{\delta})}{m-1} + \sqrt{\left(\frac{c\ln(\frac{3}{\delta})}{m-1}\right)^2 + \frac{2\hat{v}\ln(\frac{3}{\delta})}{m-1}} \ ; \ \&$$

$$\varepsilon_\mu \doteq \min\left(\underbrace{c\sqrt{\frac{\ln(\frac{3|\boldsymbol{I}|}{\delta})}{2m}}}_{\textsc{Hoeffding}}, \underbrace{\frac{c\ln(\frac{3|\boldsymbol{I}|}{\delta})}{3m} + \sqrt{\frac{2(\hat{v}+\varepsilon_v)\ln(\frac{3|\boldsymbol{I}|}{\delta})}{m}}}_{\textsc{Empirical Bennett}}\right) \ .$$

*Then, with probability at least* $1-\delta$ *over* $\boldsymbol{X} \sim \mathscr{D}^m$*, it holds that*

$$\sup_{(p,\boldsymbol{s})\in\boldsymbol{I}} \Big|\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})\Big| \leq \varepsilon_\mu \ .$$

From the definition of $\varepsilon_\mu$, we see that when $\hat{v}$ is $\approx \frac{c^2}{4}$ (near-maximal), the Hoeffding term applies, so this bound matches Theorem 3.1 to within constant factors (in particular, $\ln(\frac{3|\boldsymbol{I}|}{\delta})$ instead of $\ln(\frac{2|\boldsymbol{I}|}{\delta})$). On the other hand, when $\hat{v}$ is small, Theorem 3.2 is much sharper than Theorem 3.1. A few simplifying inequalities yield

$$\varepsilon_\mu \leq \frac{7c\ln(\frac{3|\boldsymbol{I}|}{\delta})}{3(m-1)} + \sqrt{\frac{2\hat{v}\ln(\frac{3|\boldsymbol{I}|}{\delta})}{m}} \ ,$$

which matches the standard sub-gamma Bennett's inequality up to constant factors, with dependence on $\hat{v}$ instead of $v$. In the extreme, when $\hat{v} \approx 0$ (i.e., the game is *near-deterministic*), then Theorem 3.2 improves asymptotically over Theorem 3.1 by a $\Theta\sqrt{\ln(\frac{|\boldsymbol{I}|}{\delta})/m}$ factor.

## 4 LEARNING ALGORITHMS

We are now ready to present our algorithms. Specifically, we discuss two Monte-Carlo sampling-based algorithms that can be used to uniformly learn empirical games, and hence ensure that the equilibria of the games they are learning are accurately approximated with high probability. Note that our algorithms apply only to finite games, as they require an enumeration of the index set $\boldsymbol{I}$.

A conditional normal form game $\Gamma_{\mathcal{X}}$, together with a black box from which we can sample distribution $\mathscr{D}$, serves as our mathematical model of a black-box simulator from which the utilities of a simulation-based game can be sampled. Given strategy profile $\boldsymbol{s}$, we assume the simulator outputs a sample $\boldsymbol{u}_p(\boldsymbol{s}, x)$, for *all* agents $p \in P$, after drawing a *single* condition value $x \sim \mathscr{D}$.

Our first algorithm, *global sampling* (GS), is a straightforward application of Thms. 3.1 and 3.2. The second, *progressive sampling with pruning* (PSP), iteratively prunes strategies, and thereby has the potential to expedite learning by obtaining tighter bounds than GS, given the same number of samples. We explore PSP's potential savings in our experiments (Sec. 5).

Our first algorithm, GS (Alg. 1), samples all utilities of interest, given a sample size $m$ and a failure probability $\delta$, and returns the ensuing empirical game together with an $\hat{\varepsilon}$ determined by either Thm. 3.1 or 3.2 that guarantees an $\hat{\varepsilon}$-uniform approximation.

More specifically, GS takes in a conditional game $\Gamma_{\mathcal{X}}$, a black box from which we can sample distribution $\mathscr{D}$, an index set $\boldsymbol{I} \subseteq P \times \boldsymbol{S}$, a sample size $m$, a utility range $c$ such that utilities are required to lie in $[-c/2, c/2]$, and a bound type Bd, and then draws $m$ samples to produce an empirical game $\hat{\Gamma}_{\boldsymbol{X}}$, represented by $\tilde{\boldsymbol{u}}(\cdot)$, as well as an additive error $\hat{\varepsilon}$, with the following guarantee:

**Algorithm 1** Global Sampling

```
procedure GS($\Gamma_{\mathcal{X}}, \mathscr{D}, \boldsymbol{I}, m, \delta, c, \textsc{Bd}$) → $(\tilde{\boldsymbol{u}}, \hat{\varepsilon})$
    input: Conditional game $\Gamma_{\mathcal{X}}$, black box from which we can sample distribution $\mathscr{D}$, index set $\boldsymbol{I}$, sample size $m$, failure probability $\delta$, utility range $c$, and bound type Bd.
    output: Empirical utilities $\tilde{\boldsymbol{u}}, \forall(p, \boldsymbol{s}) \in \boldsymbol{I}$; additive error $\hat{\varepsilon}$.
    $\boldsymbol{X} \sim \mathscr{D}^m$                    ▷ Draw $m$ samples from distribution $\mathscr{D}$
    $\forall(p, \boldsymbol{s}) \in \boldsymbol{I} :\ \tilde{\boldsymbol{u}}_p(\boldsymbol{s}) \leftarrow \hat{\boldsymbol{u}}_p(\boldsymbol{s}; \boldsymbol{X})$
    if Bd = H then                                 ▷ See Thm. 3.1 (Hoeffding)
        $\hat{\varepsilon} \leftarrow c\sqrt{\frac{\ln(\frac{2|\boldsymbol{I}|}{\delta})}{2m}}$
    else if Bd = B then            ▷ See Thm. 3.2 (Empirical Bennett)
        $\hat{v} \leftarrow \sup_{(p,\boldsymbol{s}) \in \boldsymbol{I}} \frac{1}{m-1}\sum_{j=1}^{m}\left(\boldsymbol{u}_p(\boldsymbol{s}; x_j) - \tilde{\boldsymbol{u}}_p(\boldsymbol{s})\right)^2$
        $\varepsilon_v \leftarrow \frac{c\ln(\frac{3}{\delta})}{m-1} + \sqrt{\left(\frac{c\ln(\frac{3}{\delta})}{m-1}\right)^2 + \frac{2\hat{v}\ln(\frac{3}{\delta})}{m-1}}$
        $\hat{\varepsilon} \leftarrow \min\left(c\sqrt{\frac{\ln(\frac{3|\boldsymbol{I}|}{\delta})}{2m}},\ \frac{c\ln(\frac{3|\boldsymbol{I}|}{\delta})}{3m} + \sqrt{\frac{2(\hat{v}+\varepsilon_v)\ln(\frac{3|\boldsymbol{I}|}{\delta})}{m}}\right)$
    end if
    return $(\tilde{\boldsymbol{u}}, \hat{\varepsilon})$
end procedure
```

Theorem 4.1 (Approximation Guarantees of Global Sampling). *Consider conditional game $\Gamma_{\mathcal{X}}$ together with distribution $\mathscr{D}$ and take index set $\boldsymbol{I} \subseteq P \times \boldsymbol{S}$ such that for all $x \in \mathcal{X}$ and $(p, \boldsymbol{s}) \in \boldsymbol{I}$, $\boldsymbol{u}_p(\boldsymbol{s}; x) \in [-c/2, c/2]$, for some $c \in \mathbb{R}$. If $\mathrm{GS}(\Gamma_{\mathcal{X}}, \mathscr{D}, \boldsymbol{I}, m, \delta, c, Bd)$ outputs the pair $(\tilde{\boldsymbol{u}}, \hat{\varepsilon})$, then with probability at least $1 - \delta$, it holds that*

$$\sup_{(p,\boldsymbol{s}) \in \boldsymbol{I}} \left|\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \tilde{\boldsymbol{u}}_p(\boldsymbol{s})\right| \leq \hat{\varepsilon} \ .$$

Next, we present PSP (Alg. 2), which, using GS as a subroutine, draws progressively larger samples, refining the empirical game at each iteration, and stopping when the equilibria are approximated to the desired accuracy, or when the sampling budget is exhausted. Although performance ultimately depends on a game's structure, PSP can learn equilibria using vastly fewer resources than GS.

As the name suggests, PSP is a pruning algorithm. The key idea is to prune (i.e., cease estimating the utilities of) strategy profiles that (w.h.p.) are provably not equilibria. Recall that $\boldsymbol{s} \in \mathrm{E}_{\varepsilon}(\Gamma)$ iff $\mathrm{Reg}_p(\Gamma, \boldsymbol{s}) \leq \varepsilon$, for all $p \in P$. Thus, if there exists $p \in P$ s.t. $\mathrm{Reg}_p(\Gamma, \boldsymbol{s}) > \varepsilon$, then $\boldsymbol{s} \notin \mathrm{E}_{\varepsilon}(\Gamma)$. In the search for pure equilibria, such strategy profiles can be pruned.

A strategy $s \in S_p$ is said to *$\varepsilon$-dominate* another strategy $s' \in S_p$ if for all $\boldsymbol{s} \in \boldsymbol{S}$, taking $\boldsymbol{s}' = (s_1, \ldots, s_{p-1}, s'_p, s_{p+1}, \ldots, s_{|P|})$, it holds that $\boldsymbol{u}_p(\boldsymbol{s}) - \varepsilon \geq \boldsymbol{u}_p(\boldsymbol{s}')$. The *$\varepsilon$-rationalizable* strategies $\mathrm{Rat}_{\varepsilon}(\Gamma)$ are those that remain after iteratively removing all $\varepsilon$-dominated strategies. Only strategies in $\mathrm{Rat}_{\varepsilon}(\Gamma)$ can have nonzero weight in a mixed $\varepsilon$-Nash equilibrium [10]; thus eliminating strategies not in $\mathrm{Rat}_{\varepsilon}(\Gamma)$ is a natural pruning criterion for mixed equilibria.

If a strategy $s \in S_p$ is $\varepsilon$-dominated by another strategy $s' \in S_p$, then $p$ always regrets playing strategy $s$, regardless of other agents' strategies. Consequently, the mixed pruning criterion is more conservative than the pure, which means more pruning occurs when learning pure equilibria.

Like GS, PSP takes in a conditional game $\Gamma_{\mathcal{X}}$, a black box from which we can sample distribution $\mathscr{D}$, a utility range $c$, and a bound type Bd. Instead of a single sample size, however, it takes in a *sampling schedule* $\boldsymbol{M}$ in the form of a (possibly infinite) strictly increasing sequence of integers; and instead of a single failure probability, it takes in a *failure probability schedule* $\boldsymbol{\delta}$, with each $\boldsymbol{\delta}_t$ in this sequence and their sum in $(0, 1)$. These two schedules dictate the number of samples to draw and the failure probability to use at each iteration. PSP also takes in a boolean Pure that indicates whether the equilibria of interest are pure or mixed, and an *error threshold* $\varepsilon$, which enables early termination as soon as equilibria are estimated to within the additive factor $\varepsilon$.

**Algorithm 2** Progressive Sampling with Pruning

```
procedure PSP($\Gamma_{\mathcal{X}}, \mathscr{D}, \boldsymbol{M}, \boldsymbol{\delta}, c, \textsc{Bd}, \textsc{Pure}, \varepsilon$) → $((\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{\varepsilon}}), (\hat{E}, \hat{\varepsilon}), \hat{\delta})$
    input: Conditional game $\Gamma_{\mathcal{X}}$, black box from which we can sample distribution $\mathscr{D}$, sampling schedule $\boldsymbol{M}$, failure probability schedule $\boldsymbol{\delta}$, utility range $c$, bound type Bd, equilibrium type Pure, error threshold $\varepsilon$.
    output: Empirical utilities $\tilde{\boldsymbol{u}}$, $\forall(p, \boldsymbol{s}) \in P \times \boldsymbol{S}$, utility error $\tilde{\boldsymbol{\varepsilon}}$, empirical equilibria $\hat{E}$, equilibria error $\hat{\varepsilon}$, failure probability $\hat{\delta}$.
    $\boldsymbol{I} \leftarrow P \times \boldsymbol{S}$                              ▷ Initialize index set
    $\forall(p, \boldsymbol{s}) \in \boldsymbol{I} :\ (\tilde{\boldsymbol{u}}_p(\boldsymbol{s}), \tilde{\boldsymbol{\varepsilon}}_p(\boldsymbol{s})) \leftarrow (0, c/2)$,    ▷ Initialize outputs
    for $t \in 1, \ldots, |\boldsymbol{M}|$ do
        $(\tilde{\boldsymbol{u}}, \hat{\varepsilon}) \leftarrow \mathrm{GS}(\Gamma_{\mathcal{X}}, \mathscr{D}, \boldsymbol{I}, \boldsymbol{M}_t, \boldsymbol{\delta}_t, c, \textsc{Bd})$  ▷ Improve estimates
        $\forall(p, \boldsymbol{s}) \in \boldsymbol{I} :\ \tilde{\boldsymbol{\varepsilon}}_p(\boldsymbol{s}) \leftarrow \hat{\varepsilon}$       ▷ Update confidence intervals
        if $\hat{\varepsilon} \leq \varepsilon$ or $t = |\boldsymbol{M}|$ then          ▷ Termination condition
            $\hat{E} \leftarrow \begin{cases} \textsc{Pure} & : \mathrm{E}_{2\hat{\varepsilon}}(\tilde{\boldsymbol{u}}) \\ \neg\textsc{Pure} & : \mathrm{E}^{\diamond}_{2\hat{\varepsilon}}(\tilde{\boldsymbol{u}}) \end{cases}$
            return $((\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{\varepsilon}}), (\hat{E}, \hat{\varepsilon}), \sum_{i=1}^{t} \boldsymbol{\delta}_i)$
        end if
        $\boldsymbol{I} \leftarrow \begin{cases} \textsc{Pure} : \{(p, \boldsymbol{s}) \in \boldsymbol{I} \mid \mathrm{Reg}_p(\tilde{\boldsymbol{u}}, \boldsymbol{s}) \leq 2\hat{\varepsilon}\} \\ \neg\textsc{Pure} : \left\{(p, \boldsymbol{s}) \in \boldsymbol{I} \,\middle|\, \forall q \in P : \boldsymbol{s}_q \in \mathrm{Rat}_{2\hat{\varepsilon}}(\tilde{\boldsymbol{u}})\right\} \end{cases}$
    end for
end procedure
```

In the PSP pseudocode, and in the following theorem, we overload the Reg and E operators (both pure and mixed) to depend on a utility function, rather than a game.

Theorem 4.2. *Suppose conditional game $\Gamma_{\mathcal{X}}$ and distribution $\mathscr{D}$ such that for all $x \in \mathcal{X}$ and $(p, \boldsymbol{s}) \in P \times \boldsymbol{S}$, $\boldsymbol{u}_p(\boldsymbol{s}; x) \in [-c/2, c/2]$ for some $c \in \mathbb{R}$. If $\mathrm{PSP}(\Gamma_{\mathcal{X}}, \mathscr{D}, \boldsymbol{M}, \boldsymbol{\delta}, c, Bd, Pure, \varepsilon)$ outputs $\left((\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{\varepsilon}}), (\hat{E}, \hat{\varepsilon}), \hat{\delta}\right)$, it holds that:*
*(1) $\hat{\delta} \leq \sum_{\boldsymbol{\delta}_t \in \boldsymbol{\delta}} \boldsymbol{\delta}_t$, $\hat{\delta} \in (0, 1)$;*
*(2) If $\lim_{t \to \infty} \ln(1/\boldsymbol{\delta}_t)/\boldsymbol{M}_t = 0$, then $\hat{\varepsilon} \leq \varepsilon$.*
*Furthermore, if* PSP *terminates, then with probability at least $1 - \hat{\delta}$, the following hold simultaneously:*
*(3) $\left|\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) - \tilde{\boldsymbol{u}}_p(\boldsymbol{s})\right| \leq \tilde{\boldsymbol{\varepsilon}}_p(\boldsymbol{s})$, for all $(p, \boldsymbol{s}) \in P \times \boldsymbol{S}$;*
*(4) If Pure, then $\mathrm{E}(\boldsymbol{u}) \subseteq \mathrm{E}_{2\hat{\varepsilon}}(\tilde{\boldsymbol{u}}) \subseteq \mathrm{E}_{4\hat{\varepsilon}}(\boldsymbol{u})$;*
*(5) If $\neg$Pure, then $\mathrm{E}^{\diamond}(\boldsymbol{u}) \subseteq \mathrm{E}^{\diamond}_{2\hat{\varepsilon}}(\tilde{\boldsymbol{u}}) \subseteq \mathrm{E}^{\diamond}_{4\hat{\varepsilon}}(\boldsymbol{u})$.*

Finally, we propose two possible sampling and failure probability schedules for PSP, $\boldsymbol{M}$ and $\boldsymbol{\delta}$, depending on whether the sampling budget is finite or infinite. Given a finite sampling budget $m < \infty$, a neutral choice is to take $\boldsymbol{M}$ to be a doubling sequence such that $\sum_{M_i \in \boldsymbol{M}} M_i \leq m$, with $\boldsymbol{M}_1$ sufficiently large so as to possibly permit pruning after the first iteration (iterations that neither prune nor achieve $\varepsilon$-accuracy are effectively wasted), and to take $\boldsymbol{\delta}_t = \delta/|\boldsymbol{M}|$, where $\delta$ is some maximum tolerable failure probability. This strategy may fail to produce the desired $\varepsilon$-approximation, as it may exhaust the sampling budget first. To guarantee a particular $\varepsilon$-$\delta$-approximation, then we can take $\boldsymbol{M}$ to be an infinite doubling sequence, and $\boldsymbol{\delta}$ to be a geometrically decreasing sequence such that $\sum_{t=1}^{\infty} \boldsymbol{\delta}_t = \delta$, for which the conditions of Thm. 4.2 (2) hold.

At first glance, it may seem that the sample complexity of estimating a game with our empirical Bennett-type bound should depend on the number, the worst-case variance over, and the range of utility values. However, if the variance of the utilities at all equilibria is small, and if the variance of the utilities at non-equilibrium strategy profiles is no more than half the square of their regret, then PSP runs only until the high-variance high-regret strategy profiles are pruned. Once this critical point is reached, the game is essentially learned, as each subsequent iteration prunes more strategy profiles, further reducing the wimpy variance, yielding tighter bounds for the remaining profiles. Under these circumstances, PSP can provably require fewer samples than GS.

## 5 EXPERIMENTS

We now set out to evaluate the strength of our methodology to learn black-box games and their equilibria from samples. The empirical performance of an algorithm can vary dramatically under different distributions of inputs; in particular, the success of game-theoretic solvers can vary dramatically even within the same class of games [17, 21]. Consequently, we employ GAMUT [21], a state-of-the-art suite of game generators that is capable of producing a wide variety of interesting game inputs of varying scales with rich strategic structure, thereby affording us an opportunity to conduct a robust evaluation of our methodology. Furthermore, we employ Gambit [19], a state-of-the-art equilibrium solver. We bundled both of these packages together with our statistical learning algorithms in a python library for empirical game-theoretic analysis, pySEGTA, to make it easier for other EGTA researchers to benchmark their algorithms against ours.

*Black-Box Game Design.* In all our experiments, we use GAMUT to generate what we call *ground-truth games*. Ground-truth games are ordinarily inaccessible; however, we rely on them here to measure the loss experienced by our algorithms: i.e., the regrets in a learned game as compared those in the corresponding ground-truth game. To simulate a black-box game, we simply add noise drawn from a zero-centered distribution to the utilities of a ground-truth game. We detail this construction presently.

Let $\Gamma$ be a realization of a ground-truth game drawn from GAMUT, and let $\boldsymbol{u}_p(\boldsymbol{s})$ be the utility of player $p$ at profile $\boldsymbol{s}$ in $\Gamma$. Fix a condition set $\mathcal{X} = [a, b]$, where $a < b$. In the conditional game $\Gamma_{\mathcal{X}}$, $\boldsymbol{u}_p(\boldsymbol{s}; x_{p,\boldsymbol{s}}) = \boldsymbol{u}_p(\boldsymbol{s}) + x_{p,\boldsymbol{s}}$, for $x_{p,\boldsymbol{s}} \in \mathcal{X}$. Conditional game $\Gamma_{\mathcal{X}}$ together with distribution $\mathscr{D}$ on $\mathcal{X}$ is then our model for a black-box game. For simplicity, all noise $x_{p,\boldsymbol{s}} \sim \mathscr{D}$ is drawn i.i.d..

We only consider noise distributions where $\mathscr{D}$ is zero-centered. Consequently, the expected-normal form game $\Gamma_{\mathscr{D}}$, which is the object of our algorithms' estimation, exactly coincides with $\Gamma$: i.e., it holds that for every $p$ and $\boldsymbol{s}$, $\boldsymbol{u}_p(\boldsymbol{s}; \mathscr{D}) = \mathbb{E}_{x_{p,\boldsymbol{s}} \sim \mathscr{D}} \left[ \boldsymbol{u}_p(\boldsymbol{s}; x_{p,\boldsymbol{s}}) \right] = \mathbb{E}_{x_{p,\boldsymbol{s}} \sim \mathscr{D}} \left[ \boldsymbol{u}_p(\boldsymbol{s}) + x_{p,\boldsymbol{s}} \right] = \boldsymbol{u}_p(\boldsymbol{s}) + \mathbb{E}_{x_{p,\boldsymbol{s}} \sim \mathscr{D}} \left[ x_{p,\boldsymbol{s}} \right] = \boldsymbol{u}_p(\boldsymbol{s})$, because $\boldsymbol{u}_p(\boldsymbol{s})$ is constant and $\mathscr{D}$ is zero-centered.

*Experimental Setup.* We normalize the utilities generated by GAMUT to lie in the range [-10, 10]. We experiment with three different noise regimes, high, medium, and low variance. Letting $U[a, b]$ be a uniform distribution over $[a, b]$, we model high, medium, and low variance noise by distributions $U[-2.5, 2.5]$, $U[-.5, .5]$, and $U[-.1, .1]$, respectively.

We test both GS, Alg. 1, and PSP, Alg. 2. These algorithms take as input a flag Bd $\in$ {H, B}, indicating which bound, Hoeffding's (Thm. 3.1) or our empirical Bennett-type (Thm. 3.2), to use. Henceforth, to refer to an algorithm that uses bound Bd, we write GS(Bd) and PSP(Bd). Throughout our experiments, we fix $\delta = 0.05$.

*Sample Efficiency of* GS. In this experiment, we investigate the sampling efficiency of our algorithms; that is, the quality of the games learned, as measured by $\varepsilon$, as a function of the number of samples needed to achieve said guarantee. We tested ten different classes of GAMUT games, all of them two-player, with varying numbers of strategies, either indicated in parentheses next to the game's name, or two by default. For each class of games, we draw 60 random ground-truth games from GAMUT, and for each such draw, we run GS 20 times for each of sample sizes $m \in \{10, 20, 40, 80, 160, 320, 640, 1280, 2560, 5120\}$, measuring $\varepsilon$ for all possible combinations of these parameters. We then average the measured values of $\varepsilon$, fixing the number of samples. Fig. 1 plots, in a log-log scale, these averages, comparing the performance of GS given Hoeffding's bound and our empirical Bennett-type bound, for the cases of high and low variance. Fig. 1 depicts only four classes of games, but trends over all ten classes are similar (see the supplemental material). In all cases, we found that our empirical Bennett-type bound produces better estimates for the same number of samples, as measured by $\varepsilon$. Note the initial $1/m$ decay rate, later slowing to $1/\sqrt{m}$, in the Bennett bounds, reflecting the fast $c$ term and slow $\sigma^2$ terms of the sub-gamma bounds.

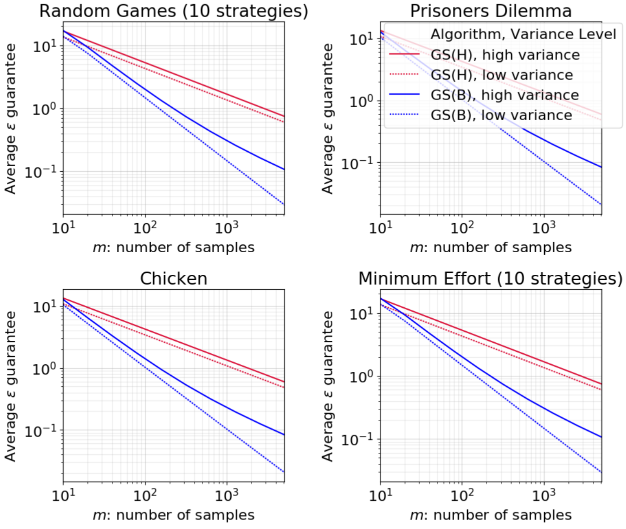

**Figure 1: Quality of Learned Games**

*Empirical Regret of* GS. In this experiment, we investigate the quality of the equilibria learned by our algorithms. To compute the equilibria of a game, we use Gambit [19], a state-of-the-art equilibrium solver. The goal of these experiments is to provide empirical evidence for Thm. 2.2, namely that our algorithms are capable of learning games that approximately preserve the Nash equilibria of black-box games. The goal is not to test the quality of different equilibrium solvers—we refer the reader to [21] for an evaluation along those lines. Hence, we fix one such solver throughout, namely, Gambit's GNM solver, a global Newton method that computes Nash equilibria [11].

To measure the quality of learned equilibria, given a game $\Gamma$ and a subset of the strategy profile space $\boldsymbol{S}' \subseteq \boldsymbol{S}^{\diamond}$, we define the metric $\textsc{Max-Regret}(\Gamma, \boldsymbol{S}') = \sup_{\boldsymbol{s} \in \boldsymbol{S}'} \max_{p \in P} \operatorname{Reg}_p^{\diamond}(\Gamma, \boldsymbol{s})$: i.e, the maximum regret of any player $p$ at any profile $\boldsymbol{s}$ in $\boldsymbol{S}'$. Note that, given two compatible games $\Gamma$ and $\Gamma'$, and $\boldsymbol{S}'$, we can measure $\textsc{Max-Regret}$ of *either* game, since the strategy profile space is shared by compatible games. This is useful because, given a ground-truth game $\Gamma$ and a corresponding empirical estimate $\hat{\Gamma}_{\boldsymbol{X}}$, we can measure the maximum regret of a set of Nash equilibrium profiles in $\Gamma$, say $\boldsymbol{S}^*$, in its empirical estimate $\hat{\Gamma}_{\boldsymbol{X}}$. Thm. 2.2 implies that, given an $\varepsilon$-uniform approximation $\hat{\Gamma}_{\boldsymbol{X}}$ of $\Gamma$, we should observe $\textsc{Max-Regret}(\hat{\Gamma}_{\boldsymbol{X}}, \boldsymbol{S}^*) \leq 2\varepsilon$. Thm. 4.1 then implies that if said $\varepsilon$-uniform approximation holds with probability $1 - \delta$, then we should likewise observe $\textsc{Max-Regret}(\hat{\Gamma}_{\boldsymbol{X}}, \boldsymbol{S}^*) \leq 2\varepsilon$ with probability $1 - \delta$.

We empirically measure $\textsc{Max-Regret}$, where equilibria are computed using Gambit, for the same ten different classes of games as in the previous experiment, again over 60 draws for each class, where for each such draw, we run GS 10 times for each of sample sizes $m \in \{10, 20, 40, 80, 160\}$, measuring $\textsc{Max-Regret}$ for all possible combinations of these parameters. We then average the measured values of $\textsc{Max-Regret}$, fixing the number of samples. Fig. 2 plots, in a log-log scale, both these averages (the markers) and the theoretical guarantees (the lines). This plot complements our theory, establishing experimentally that our algorithms are capable of preserving equilibria in learned black-box games. This learning is robust to various classes of games, for example, for dominance-solvable games (such as Prisoners' Dilemma), games with guaranteed pure-strategy equilibria (such as congestion games), as well as other games with no guarantee on their equilibria other than existence (such as random and zero-sum games).[3] This learning is also robust to different levels of noise, with our algorithms consistently achieving higher accuracy in practice than in theory, all across the board.

*Sample Efficiency of* PSP. In this experiment, we investigate the sample efficiency of PSP as compared to GS. We say that algorithm $A$ has better *sample efficiency* than algorithm $B$ if $A$ requires fewer samples than $B$ to achieve a desired accuracy $\varepsilon$.

Our experimental design is as follows. Fixing a game, and the following values of $\varepsilon \in \{0.125, 0.25, 0.5, 1.0\}$, we compute the number of samples $m(\varepsilon)$ that would be required for GS($H$) to achieve accuracy $\varepsilon$. We then run both GS($H$) and GS($B$) with $m(\varepsilon)$ samples.

For PSP, we use the following doubling strategy as a sampling schedule $\boldsymbol{M}(m(\varepsilon)) = [{}^{m(\varepsilon)}/_4, {}^{m(\varepsilon)}/_2, m(\varepsilon), 2m(\varepsilon)]$, rounding to the nearest integer as necessary. For $\boldsymbol{\delta}$, we use a uniform schedule such that $\sum_{\boldsymbol{\delta}_t \in \boldsymbol{\delta}} \boldsymbol{\delta}_t = \delta$: i.e., $\boldsymbol{\delta} = [0.0125, 0.0125, 0.0125, 0.0125]$. Using these schedules, we run both PSP($H$) and PSP($B$) until completion by setting the desired accuracy to zero. We prune using the mixed-strategy criterion, namely the set of rationalizable strategies.

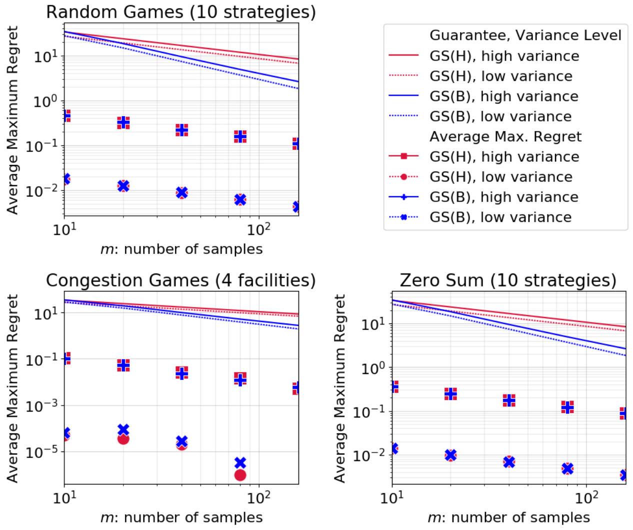

**Figure 2: Average Maximum Regret**

We ran this experiment on three different classes of games: congestion games (2 players, and 2, 3, 4, and 5 facilities; game sizes 18, 98, 450, and 1,922 respectively), random games (2 players, and 5, 10, 20, and 30 strategies each; game sizes 50, 200, 800, and 1,800 respectively), and zero-sum games (2 players, and 5, 10, 20, and 30 strategies; game sizes 50, 200, 800, and 1,800 respectively). As with our other experiments, we draw multiple games for each class of games (in this case 30) and multiple runs (in this case 10 for each draw of each game). We consider medium variance only.

For all algorithms, we measure the total number of samples across all players and strategy profiles. If and when it prunes, PSP requires progressively fewer and fewer samples, with the number decreasing each iteration to the size of the unpruned game.

Table 1 summarizes the results of these experiments for select games. In all cases, we simply report the total number of samples, averaged across all experiments. The theory tells us that given this number of samples, GS must achieve at least the desired $\varepsilon \in \{0.125, 0.25, 0.5, 1.0\}$. Although there is no such guarantee for PSP, in these experiments PSP always achieved a strictly greater accuracy than GS (these accuracies are also reported in Table 1, under the columns labeled $\varepsilon_{\text{PSP}}$). Moreover, PSP tends to exhibit significantly better sample efficiency than GS; notable exceptions include cases where either the games are small or the $\varepsilon$ guarantee is loose (e.g., $\varepsilon \leq 1.0$). These results demonstrate the promise of PSP as an algorithm for learning black-box games, as its sample efficiency generally exceeds that of GS.

*Limitations of* PSP. While our experiments demonstrate that PSP can yield substantial savings when learning games in many different classes, in some GAMUT games, our simple doubling schedule yielded no such gains. We found Grab The Dollar to be a particularly difficult game. In Grab The Dollar, there is a prize

[3] Similar to before, we report on only three game classes here, and refer the reader to the supplemental material for results on all 10 classes.

| | $\varepsilon \leq 0.125$ | | $\varepsilon \leq 0.25$ | | $\varepsilon \leq 0.5$ | | $\varepsilon \leq 1.0$ | |
|---|---|---|---|---|---|---|---|---|
| Bound | Hoeffding | Emp. Bennett | Hoeffding | Emp. Bennett | Hoeffding | Emp. Bennett | Hoeffding | Emp. Bennett |
| Game/Algorithm | GS; PSP; $\varepsilon_{\text{PSP}}$ | GS; PSP; $\varepsilon_{\text{PSP}}$ | GS; PSP; $\varepsilon_{\text{PSP}}$ | GS; PSP; $\varepsilon_{\text{PSP}}$ | GS; PSP; $\varepsilon_{\text{PSP}}$ | GS; PSP; $\varepsilon_{\text{PSP}}$ | GS; PSP; $\varepsilon_{\text{PSP}}$ | GS; PSP; $\varepsilon_{\text{PSP}}$ |
| Congestion Games (5 facilities) | 3,051; **1,654**; 0.08 | 3,051; **1,449**; 0.00 | 762; **464**; 0.17 | 762; **364**; 0.01 | 190; **146**; 0.34 | 190; **93**; 0.01 | **47**; 58; 0.70 | 47; **25**; 0.04 |
| Zero-Sum Games (30 strategies) | 2,841; **1,691**; 0.08 | 2,841; **1,383**; 0.00 | 710; **502**; 0.17 | 710; **349**; 0.01 | 177; **166**; 0.35 | 177; **90**; 0.01 | **44**; 62; 0.71 | 44; **25**; 0.04 |
| Random Games (30 strategies) | 2,841; **1,666**; 0.08 | 2,841; **1,375**; 0.00 | 710; **491**; 0.17 | 710; **347**; 0.01 | 177; **159**; 0.35 | 177; **90**; 0.01 | **44**; 58; 0.71 | 44; **25**; 0.04 |
| Congestion Games (4 facilities) | 622; **492**; 0.09 | 622; **438**; 0.00 | 156; **138**; 0.17 | 156; **110**; 0.01 | **39**; 41; 0.35 | 39; **28**; 0.01 | **10**; 15; 0.71 | 10; **8**; 0.04 |
| Zero-Sum Games (20 strategies) | 1,171; **829**; 0.09 | 1,171; **708**; 0.00 | 293; **240**; 0.17 | 293; **179**; 0.01 | **73**; 77; 0.35 | 73; **46**; 0.01 | **18**; 28; 0.71 | 18; **13**; 0.04 |
| Random Games (20 strategies) | 1,171; **809**; 0.09 | 1,171; **698**; 0.00 | 293; **232**; 0.17 | 293; **176**; 0.01 | **73**; **73**; 0.35 | 73; **45**; 0.01 | **18**; 25; 0.71 | 18; **12**; 0.04 |
| Congestion Games (3 facilities) | **114**; 145; 0.09 | **114**; 135; 0.00 | **29**; 40; 0.18 | **29**; 34; 0.01 | **7**; 12; 0.36 | **7**; 9; 0.02 | **2**; 4; 0.73 | **2**; **2**; 0.05 |
| Zero-Sum Games (10 strategies) | **254**; 268; 0.09 | 254; **242**; 0.00 | **63**; 73; 0.18 | 63; **61**; 0.01 | **16**; 22;0.36 | 16; **15**;0.02 | **4**; 7; 0.73 | **4**; **4**; 0.05 |
| Random Games (10 strategies) | **254**; **254**; 0.09 | 254; **233**; 0.00 | **63**; 69; 0.18 | 63; **59**; 0.01 | **16**; 21;0.36 | 16; **15**; 0.02 | **4**; 7; 0.72 | **4**; **4**; 0.05 |
| Congestion Games (2 facilities) | **17**; 37; 0.09 | **17**; 37; 0.00 | **4**; 10; 0.19 | **4**; 9; 0.01 | **1**; 3; 0.38 | **1**; 2; 0.02 | **1**; **1**; 0.76 | **1**; **1**; 0.05 |
| Zero-Sum Games (5 strategies) | **54**; 94; 0.09 | **54**; 89; 0.00 | **13**; 25; 0.18 | **13**; 22; 0.01 | **3**; 7; 0.37 | **3**; 6; 0.02 | **1**; 2; 0.75 | **1**; **1**; 0.05 |
| Random Games (5 strategies) | **54**; 83; 0.09 | **54**; 90; 0.00 | **13**; 22; 0.18 | **13**; 20; 0.01 | **3**; 6; 0.37 | **3**; 5; 0.02 | **1**; 2; 0.74 | **1**; **1**; 0.05 |

**Table 1:** PSP**'s sample efficiency. Numbers of samples are reported in tens of thousands. The values in bold are smaller than their counterparts; as $\varepsilon$ is fixed, they indicate the more sample efficient algorithms.**

(or "dollar") that two players are free to grab at any time, and there are two utility values, one high and one low. If both players grab for the dollar at the same time, it will rip; so the players earn the low utility. If one grabs the dollar before the other, then that player wins the dollar (and thus high utility), while the other player earns utility somewhere between the high and the low values.

The utility structure of this game is such that the player's utilities are the same across many different strategy profiles—in particular, whenever one player "Grabs The Dollar" before their opponent. As a result, there are few $\varepsilon$-dominated strategies, which in turn makes pruning ineffective. PSP is most effective in cases where utilities between neighboring strategy profiles (i.e., where only one player's strategy differs) are distinct enough that pruning is possible. Arguably, this kind of structure is common in practice, where, fixing all other players' strategies, one player's strategy (like defect in the Prisoners' Dilemma) can yield very different utilities than neighboring strategies (like cooperate). Finally, our PSP algorithm does not compare favorably to the baseline in games where there are few strategies, and hence few opportunities to prune.

## 6 SUMMARY AND CONCLUSION

In this paper, we present and evaluate a methodology for learning games that cannot be expressed analytically. On the contrary, it is assumed that a black-box simulator is available that can be queried to obtain noisy samples of utilities. In many simulation-based games of interest, queries to the simulator are exceedingly expensive, so that the time and effort required to obtain sufficiently accurate utility estimates dwarves that of any other relevant computations, including equilibrium computations. This condition holds in meta-games like Starcraft [28], for example, where agents choices comprise a few high-level heuristic strategies, not intractably many low-level game-theoretic strategies. Thus, our primary concern in this paper is to limit the need for sampling, while still guaranteeing that we are estimating a game well.

Our main contributions come in two flavors. First, we derive a novel bound that can adapt to any naturally occurring variance in a game's black-box simulator, so that fewer samples are required to obtain robust guarantees on the estimated game. Whereas Tuyls et al. [28] control the FWER by applying a Šidák correction to Hoeffding's bound, we improve upon their approach using an empirical variant of Bennett's inequality that does not require *a priori* knowledge of the variance.

Second, we develop an algorithm that progressively samples a game, all the while pruning strategy profiles: i.e., ceasing to estimate those strategy profiles that provably (with high probability) do not comprise any (approximate) equilibria. In extensive experimentation over a broad swath of games, we show that this algorithm, equipped with our variance-sensitive bound, makes frugal use of samples, often requiring far fewer to learn to the same—or even a better—degree of accuracy than a variant of the sampling algorithm in Tuyls et al. [28]'s, which serves as a baseline.

## A pySEGTA: A PYTHON LIBRARY

We carried out our experiments in a Python library we developed and named pySEGTA, for *statistical* EGTA.[4] pySEGTA interfaces with both GAMUT and Gambit, exposing simple interfaces by which users can generate games (GAMUT), learn them (via our learning algorithms, for example), and solve them (Gambit). As the logic concerning game implementation is entirely separate from game learning and/or solving, pySEGTA can be used to analyze arbitrarily complex simulation-based games with arbitrarily complex strategies. pySEGTA already affords access to most GAMUT games, and is designed to be easily extensible to interface with other game generators. To do so only requires describing a game's structure (number of players, and per-player numbers of strategies), and implementing one query method, which takes as input a strategy profile and returns sample utilities for all players at the given strategy profile. pySEGTA also includes parameterizable implementations of both GS and PSP, and was designed with extensibility in mind, so that other users can incorporate their learning algorithms as they are developed. Our intent is that pySEGTA ease the work of benchmarking empirical game-theoretic learning algorithms.

## B ACKNOWLEDGEMENTS

This work was supported in part by NSF Award CMMI-1761546, NSF grant RI-1813444, and DARPA/AFRL grant FA8750-19-2-1006.

[4] pySEGTA is publicly accessible at http://github.com/eareyan/pysegta.

## REFERENCES

[1] Enrique Areyan Viqueira, Cyrus Cousins, Yasser Mohammad, and Amy Greenwald. 2019. Empirical Mechanism Design: Designing Mechanisms from Data. In *UAI*. AUAI Press, 406.

[2] Enrique Areyan Viqueira, Amy Greenwald, Cyrus Cousins, and Eli Upfal. 2019. Learning Simulation-Based Games from Data. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1778–1780.

[3] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. 2007. Tuning bandit algorithms in stochastic environments. In *International conference on algorithmic learning theory*. Springer, 150–165.

[4] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. 2007. Variance estimates and exploration function in multi-armed bandit. In *CERTIS Research Report 07–31*. Citeseer.

[5] George Bennett. 1962. Probability inequalities for the sum of independent random variables. *J. Amer. Statist. Assoc.* 57, 297 (1962), 33–45.

[6] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. 2013. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press.

[7] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. 2009. The complexity of computing a Nash equilibrium. *SIAM J. Comput.* 39, 1 (2009), 195–259.

[8] Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, Boca Raton, Florida, USA.

[9] Tapio Elomaa and Matti Kääriäinen. 2002. Progressive Rademacher sampling. In *AAAI/IAAI*. 140–145.

[10] Robert Gibbons. 1992. *Game theory for applied economists*. Princeton University Press.

[11] Srihari Govindan and Robert Wilson. 2003. A global Newton method to compute Nash equilibria. *Journal of Economic Theory* 110, 1 (2003), 65–86.

[12] Wassily Hoeffding. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association* 58, 301 (1963), 13–30.

[13] Patrick R Jordan, Christopher Kiekintveld, and Michael P Wellman. 2007. Empirical game-theoretic analysis of the TAC supply chain game. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM, 193.

[14] Patrick R Jordan, Yevgeniy Vorobeychik, and Michael P Wellman. 2008. Searching for approximate equilibria in empirical games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 1063–1070.

[15] Patrick R Jordan and Michael P Wellman. 2010. Designing an ad auctions game for the trading agent competition. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*. Springer.

[16] Wolfgang Ketter, Markus Peters, and John Collins. 2013. Autonomous Agents in Future Energy Markets: The 2012 Power Trading Agent Competition.. In *AAAI*.

[17] Kevin Leyton-Brown, Eugene Nudelman, and Yoav Shoham. 2002. Learning the empirical hardness of optimization problems: The case of combinatorial auctions. In *International Conference on Principles and Practice of Constraint Programming*. Springer, 556–572.

[18] Andreas Maurer and Massimiliano Pontil. 2009. Empirical Bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740* (2009).

[19] McLennan Andrew M. McKelvey, Richard D. and Theodore L. Turocy. 2019. Gambit: Software Tools for Game Theory, Version 16.0.1. (2019). Retrieved November 7, 2019 from http://www.gambit-project.org.

[20] John F Nash. 1950. Equilibrium points in n-person games. *Proceedings of the national academy of sciences* (1950).

[21] Eugene Nudelman, Jennifer Wortman, Yoav Shoham, and Kevin Leyton-Brown. 2004. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*. IEEE Computer Society, 880–887.

[22] Victor Picheny, Mickael Binois, and Abderrahmane Habbal. 2016. A Bayesian optimization approach to find Nash equilibria. *arXiv preprint arXiv:1611.02440* (2016).

[23] Tiberiu Popoviciu. 1935. Sur les équations algébriques ayant toutes leurs racines réelles. *Mathematica* 9 (1935), 129–145.

[24] Matteo Riondato and Eli Upfal. 2015. Mining frequent itemsets through progressive sampling with Rademacher averages. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1005–1014.

[25] Matteo Riondato and Eli Upfal. 2018. ABRA: Approximating betweenness centrality in static and dynamic graphs with Rademacher averages. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12, 5 (2018), 61.

[26] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484.

[27] Anderson Tavares, Hector Azpurua, Amanda Santos, and Luiz Chaimowicz. 2016. Rock, paper, starcraft: Strategy selection in real-time strategy games. In *The Twelfth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-16)*.

[28] Karl Tuyls, Julien Perolat, Marc Lanctot, Joel Z Leibo, and Thore Graepel. 2018. A Generalised Method for Empirical Game Theoretic Analysis. *arXiv preprint arXiv:1803.06376* (2018).

[29] Leslie G Valiant. 1984. A theory of the learnable. *Commun. ACM* 27, 11 (1984), 1134–1142.

[30] Enrique Areyan Viqueira, Cyrus Cousins, Eli Upfal, and Amy Greenwald. 2019. Learning Equilibria of Simulation-Based Games. *arXiv preprint arXiv:1905.13379* (2019).

[31] Yevgeniy Vorobeychik. 2010. Probabilistic analysis of simulation-based games. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 20, 3 (2010), 16.

[32] Yevgeniy Vorobeychik, Christopher Kiekintveld, and Michael P Wellman. 2006. Empirical mechanism design: methods, with application to a supply-chain scenario. In *Proceedings of the 7th ACM conference on Electronic commerce*. ACM.

[33] Yevgeniy Vorobeychik and Michael P Wellman. 2008. Stochastic search methods for Nash equilibrium approximation in simulation-based games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 1055–1062.

[34] Michael P Wellman. 2006. Methods for empirical game-theoretic analysis. In *AAAI*. 1552–1556.

[35] Michael P Wellman, Tae Hyung Kim, and Quang Duong. 2013. Analyzing incentives for protocol compliance in complex domains: A case study of introduction-based routing. *arXiv preprint arXiv:1306.0388* (2013).

[36] Bryce Wiedenbeck. 2014. Approximate Game Theoretic Analysis for Large Simulation-based Games. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems (AAMAS '14)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1745–1746. http://dl.acm.org/citation.cfm?id=2615731.2616156