No-regret Sample-efficient Bayesian Optimization for Finding Nash Equilibria with Unknown Utilities

Sebastian Shenghong Tay[1 2] **Quoc Phong Nguyen**[1] **Chuan Sheng Foo**[2 3] **Bryan Kian Hsiang Low**[1]
[1]Department of Computer Science, National University of Singapore
[2]Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore
[3]Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR), Singapore

Abstract

The *Nash equilibrium* (NE) is a classic solution concept for normal-form games that is stable under potential unilateral deviations by self-interested agents. *Bayesian optimization* (BO) has been used to find NE in continuous general-sum games with unknown costly-to-sample utility functions in a sample-efficient manner. This paper presents the first no-regret BO algorithm that is sample-efficient in finding pure NE by leveraging theory on high probability confidence bounds with Gaussian processes and the maximum information gain of kernel functions. Unlike previous works, our algorithm is theoretically guaranteed to converge to the optimal solution (i.e., NE). We also introduce the novel setting of applying BO to finding mixed NE in unknown discrete general-sum games and show that our theoretical framework is general enough to be extended naturally to this setting by developing a no-regret BO algorithm that is sample-efficient in finding mixed NE. We empirically show that our algorithms are competitive w.r.t. suitable baselines in finding NE.

1 INTRODUCTION

The *Nash equilibrium* (NE) is a classic solution concept for normal-form games with self-interested utility-maximizing agents (Nash, 1951). An NE is a stable solution such that no agent can increase its utility by unilaterally deviating from the NE. So, it provides predictability of obtained utilities compared to unstable solutions which may be arbitrarily worse in deployment. The NE has been used to analyze many real-world scenarios like sustainable use of natural resources (Thorpe et al., 2017), international conflicts (Schelling, 1980), and traffic, power, and wireless networks (Djehiche et al., 2017).

There is extensive literature on the theory and computation of NE for both discrete (Shoham and Leyton-Brown, 2008) and continuous games (Başar, 1987; Debreu, 1952; Reeves and Wellman, 2012) with *known* utility functions. On the other hand, empirical, simulation-based, or black-box games have *unknown* utility functions and hence require using a learning-based approach to find an NE from samples of the utility function (through oracle calls) (Vorobeychik et al., 2007, 2008; Wellman, 2006). However, these works do not assume a cost (e.g., time or money) for the samples and can become impractical when the samples are *costly*. To illustrate the importance of sample efficiency, suppose that the vector $\mathbf{x} := (x_1, x_2) \in \mathbb{R}^2$ of decision variables represents an abstract joint policy over two private hire drivers 1 and 2 with abstract individual policies x_1 and x_2 and unknown utility functions $u_1 : \mathbb{R}^2 \to \mathbb{R}$ and $u_2 : \mathbb{R}^2 \to \mathbb{R}$, respectively. The utility of each driver depends on the individual policies of both drivers: If both drivers try to pick up passengers in the same areas, they will reduce each other's utility. The controller may want to deploy the drivers at an NE so that they do not deviate from their prescribed policies and there is predictability of the obtained utilities. However, the utility functions can be unknown and must be learned from real-world deployments (through samples). A single sample of the utility function may take days, so the controller must find an NE in as few samples as possible.

To additionally account for the *costly* samples (i.e., our problem setting), a novel line of work has investigated the use of *Bayesian optimization* (BO) (Garnett, 2022) to find NE with unknown utility functions in a sample-efficient manner (Al-Dujaili et al., 2018; Picheny et al., 2019). These works have proposed heuristic algorithms that have been shown to perform well empirically, but do not provide theoretical performance guarantees which are desirable in ensuring generalization to untested settings. Also, such

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

works have only considered *pure NE* in which each agent selects only one strategy deterministically. There may be situations in which each agent selects its strategy stochastically and their decisions are better modeled as a probability distribution over strategies instead. In this case, we wish to find *mixed NE*. It is unclear how existing algorithms can be adapted for this purpose.

To tackle the above challenges, this paper presents novel BO algorithms with provable performance guarantees that are sample-efficient in finding pure and mixed NE in general-sum games with unknown costly-to-sample utility functions. To achieve this, we first leverage classic proof techniques from sequential optimization (Srinivas et al., 2010) to develop a BO algorithm with a *no-regret* performance guarantee, i.e., its incurred average cumulative regret tends to 0 by selecting decisions to sample function values arbitrarily close to an optimum (in our setting, an NE) as the number of BO iterations tends to infinity. Interestingly, such a theory informs our algorithm (as opposed to prescribing an algorithm based on heuristics) by explicitly selecting the decisions (to sample the function values) for both exploitation and exploration that are required to guarantee convergence. Then, we develop the novel setting of applying BO to finding mixed NE and show that our theoretical framework is general enough to be extended naturally to this setting by developing a no-regret BO algorithm for finding mixed NE. The specific contributions of our work here are as follows:

- To our best knowledge, we develop the first no-regret BO algorithm that is sample-efficient in finding pure NE in unknown continuous general-sum games (Sec. 4);
- We introduce the novel setting of finding mixed NE with BO in unknown discrete general-sum games, show that our general theoretical framework can be extended naturally to this setting (albeit not a simple application of our aforementioned theory for pure NE due to the adoption of a practical learning setting, as discussed in Sec. 5.1), and consequently develop a no-regret BO algorithm that is sample-efficient in finding mixed NE (Sec. 5);
- We provide experimental results to show that our algorithms are competitive w.r.t. previous work in finding pure NE as well as w.r.t. suitable baselines in finding mixed NE (Sec. 6).

2 RELATED WORK

To use BO for finding NE with unknown utility functions in a sample-efficient manner, Picheny et al. (2019) have developed probability of equilibrium (i.e., analogous to a conventional BO acquisition function called probability of improvement) and an entropy search algorithm, while Al-Dujaili et al. (2018) have proposed an ϵ-greedy algorithm with best-response approximation. However, they do not provide theoretical guarantees on the convergence of their algorithms and only consider pure strategies. Vorobeychik et al. (2008) have tackled the same problem using simulated annealing. However, since they do not assume a prior over the utility functions to exploit a probabilistic model, their method is sample-inefficient (i.e., $\approx 1000\times$ the number of function samples compared to BO algorithms). Another line of work has focused on minimax problems, both with BO (Bogunovic et al., 2018; Marchesi et al., 2020) and without (Liu et al., 2020; Wang et al., 2022). These works are applicable to zero-sum games but not in our more general setting of finding NE for general-sum games. Viqueira et al. (2019, 2020) have focused on a related but different setting in which they have access to a conditional game and sample entire utility functions at a time, as opposed to our setting in which we only sample the utility function values of specific strategy profiles at a time. More generally, other recent works leverage BO within a multi-agent framework for purposes other than finding NE (Dai et al., 2020a; Sessa et al., 2019, 2020, 2021).

3 BAYESIAN OPTIMIZATION (BO) AND GAUSSIAN PROCESSES (GP)

Before we describe the problem setting of finding Nash equilibria, we will present a primer on standard BO (Garnett, 2022). BO is a well-established framework for sample-efficient black-box optimization that has seen numerous successes in real-world applications with unknown costly-to-sample objective functions such as hyperparameter optimization of machine learning models (Chen et al., 2018) and drug and antibody sequence design (Stanton et al., 2022). A learner is required to find the maximizer $\mathbf{x}^* := \operatorname{argmax}_{\mathbf{x}} f(\mathbf{x})$ of an unknown function f w.r.t. decision variables $\mathbf{x}$. The learner obtains information about f through *samples* of f at arbitrary *decisions*. Specifically, in each iteration t, the learner selects a decision $\tilde{\mathbf{x}}_t$ to sample f at and receives a corresponding sampled noisy function value $y_t := f(\tilde{\mathbf{x}}_t) + \xi_t$ to form the sample $(\tilde{\mathbf{x}}_t, y_t)$ where $\xi_t \sim \mathcal{N}(0, \sigma^2)$ with noise variance σ^2. Such samples are assumed to be costly in terms of time, money, or some other resource, hence it is in the learner's interest to find $\mathbf{x}^*$ in as few iterations as possible. To achieve sample efficiency, BO adopts a Bayesian approach by assuming a prior over f and using the samples gathered thus far to derive a posterior over f that is exploited by an *acquisition function* to guide the learner in selecting future decisions so as to reduce the number of samples required to find the maximizer.

Though any Bayesian model can be used, the *Gaussian process* (GP) model (Williams and Rasmussen, 2006) is the standard model of choice as it allows tractable exact posterior inference with small datasets, as is the case with BO. Given a dataset $\mathcal{D}_t := \{(\tilde{\mathbf{x}}_j, y_j)\}_{j=1}^t$ of samples gathered up till iteration t, the GP posterior mean and variance at any

decision $\mathbf{x}$ in the decision space are given by

$$\mu_t(\mathbf{x}) \coloneqq \mathbf{k}_t(\mathbf{x})^\top(\mathbf{K}_t + \lambda\mathbf{I})^{-1}\mathbf{y}_t^\top \ , \quad (1)$$

$$\sigma_t^2(\mathbf{x}) \coloneqq k(\mathbf{x},\mathbf{x}) - \mathbf{k}_t(\mathbf{x})^\top(\mathbf{K}_t + \lambda\mathbf{I})^{-1}\mathbf{k}_t(\mathbf{x}) \quad (2)$$

where $\mathbf{y}_t \coloneqq (y_j)_{j=1}^t \in \mathbb{R}^t$, k is a positive semidefinite kernel (i.e., covariance function), $\mathbf{k}_t(\mathbf{x}) \coloneqq (k(\mathbf{x},\tilde{\mathbf{x}}_j))_{j=1}^t \in \mathbb{R}^t$, $\mathbf{K}_t \coloneqq (k(\tilde{\mathbf{x}}_j,\tilde{\mathbf{x}}_{j'}))_{j,j'=1}^t \in \mathbb{R}^{t\times t}$, and λ is a free parameter for algorithm design (Chowdhury and Gopalan, 2017) (to recover the true posterior in this setting, $\lambda = \sigma^2$). The choice of kernel k encodes our prior distribution over f: Briefly, k determines the *reproducing kernel Hilbert space* (RKHS) in which the GP posterior mean lies (Schölkopf and Smola, 2002). The kernel choice also affects a quantity of interest known as the *maximum information gain* in iteration T (Srinivas et al., 2010): $\gamma_T \coloneqq \max_{(\tilde{\mathbf{x}}_t)_{t=1}^T} 0.5\log\det(\mathbf{I}+\sigma^{-2}\mathbf{K}_T)$ where the maximum is taken over all possible combinations of T decisions. Note that γ_T is considered a measure of sample complexity as it is used to upper bound the regret of an algorithm in various works on sequential optimization (Abbasi-Yadkori, 2012; Chowdhury and Gopalan, 2017; Srinivas et al., 2010) and in this work (Theorems 1, 2, and 3).

4 PURE NASH EQUILIBRIA (NE)

In the pure NE setting, we consider continuous normal-form general-sum games with n self-interested agents: $\mathcal{A} \coloneqq \{1,...,n\}$. A *pure strategy* (or *action*) $\mathbf{x}_i$ of each agent $i \in \mathcal{A}$ lies in a compact set $\mathcal{X}_i \subset \mathbb{R}^{d_i}$ of possible pure strategies (i.e., a.k.a. pure strategy set). A *pure strategy profile* is denoted by a vector $\mathbf{x} \in \mathbb{R}^d$ concatenating all agents' pure strategies $\mathbf{x}_1,\ldots,\mathbf{x}_n$ where $d = \sum_{i=1}^n d_i$. Each agent $i \in \mathcal{A}$ is associated with an unknown utility function $u_i : \mathbb{R}^d \to \mathbb{R}$ that maps each pure strategy profile to its obtained utility when that strategy profile is played. Each u_i is assumed to belong to the RKHS associated with k. The space of all possible pure strategy profiles is denoted by $\mathcal{X} \coloneqq \times_{i=1}^n \mathcal{X}_i \subset \mathbb{R}^d$. Given a strategy profile $\mathbf{x} \in \mathcal{X}$, the *best-response payoff* of agent $i \in \mathcal{A}$ for the pure NE setting is defined as $\max_{\mathbf{x}_i'\in\mathcal{X}_i} u_i(\mathbf{x}_i',\mathbf{x}_{-i}) - u_i(\mathbf{x})$ where $\mathbf{x}_{-i}$ denotes a vector of all agents' strategies in $\mathbf{x}$ except agent i's; we overload the notation $u_i(\mathbf{x}_i',\mathbf{x}_{-i}) = u_i((\mathbf{x}_i',\mathbf{x}_{-i}))$ here s.t. $(\mathbf{x}_i',\mathbf{x}_{-i})$ is a concatenated vector. The larger this payoff is, the less stable (i.e., worse) $\mathbf{x}$ is since agent i has a larger incentive to deviate. To cast our setting as a maximization problem, we consider agent i's *negative best-response payoff* instead:

$$f_i(\mathbf{x}) \coloneqq u_i(\mathbf{x}) - \max_{\mathbf{x}_i'\in\mathcal{X}_i} u_i(\mathbf{x}_i',\mathbf{x}_{-i}) \leqslant 0 \ . \quad (3)$$

A pure NE is a strategy profile $\mathbf{x}_*$ s.t.

$$\mathbf{x}_* \in \mathcal{X}_* \coloneqq \{\mathbf{x}\in\mathcal{X} \mid \forall i\in\mathcal{A}\ \ f_i(\mathbf{x}) = 0\} \ .$$

In general, a pure NE is not guaranteed to exist. We thus rely on a relaxation known as a pure ϵ-*Nash equilibrium* (ϵ-NE) which is denoted by strategy profile $\mathbf{x}_\epsilon$:

$$\mathbf{x}_\epsilon \in \mathcal{X}_\epsilon \coloneqq \{\mathbf{x}\in\mathcal{X} \mid \forall i\in\mathcal{A}\ \ f_i(\mathbf{x}) \geqslant -\epsilon\} \ .$$

A pure ϵ-NE always exists if ϵ is arbitrary as ϵ can simply increase until some strategy profile $\mathbf{x}_\epsilon$ satisfies the condition $\forall i\in\mathcal{A}\ \ f_i(\mathbf{x}_\epsilon) \geqslant -\epsilon$. Fig. 1a illustrates an example of a 2-agent game with 1-D strategy sets and the game's NE.

4.1 Finding Pure NE with No-regret BO Algorithm

Our learning setting involves a single learner who is able to sample at a pure strategy profile $\tilde{\mathbf{x}}_t \in \mathcal{X}$ in each iteration t; so, the learner's decisions are pure strategy profiles. The learner then receives a sampled noisy function value $y_{i,t} = u_i(\tilde{\mathbf{x}}_t)+\xi_{i,t}$ from every agent $i \in \mathcal{A}$ to form the *sample* $(\tilde{\mathbf{x}}_t, y_{i,t})$ for updating a separate GP model (each with kernel k) for each agent i. We assume that the learner has full control over the agents during learning and the game (with potential unilateral deviations from the prescribed strategy profile) only starts during deployment after learning; these are similarly and commonly adopted in multi-agent reinforcement learning utilizing *centralized training for decentralized execution* (Lyu et al., 2021).[1]

In each iteration t, the learner also reports a pure strategy profile that it thinks to be an NE using any available information. This *reported strategy profile* $\mathbf{x}_t$ serves a different purpose from the *sampled strategy profile* $\tilde{\mathbf{x}}_t$ and is used to measure an algorithm's performance through the notion of regret. We define the *cumulative pure Nash regret* as

$$R_T \coloneqq \textstyle\sum_{t=1}^T -\epsilon_* - \min_{i\in\mathcal{A}} f_i(\mathbf{x}_t) \ ,$$
$$\epsilon_* \coloneqq \inf\{\epsilon\in\mathbb{R} \mid \mathcal{X}_\epsilon \neq \varnothing\} \ .$$

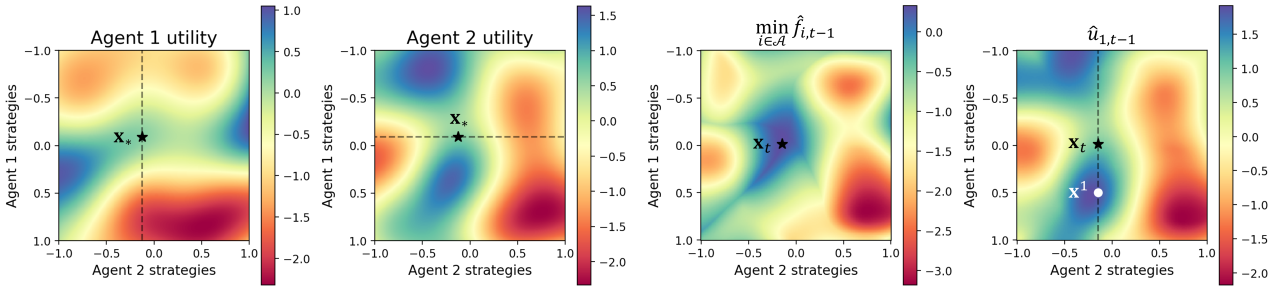
Since a pure NE may not exist, the pure Nash regret compares the performance of each reported pure strategy profile against the best achievable ϵ-NE (i.e., one with the smallest relaxation ϵ possible). The learner is assumed to be *unaware* of the value of ϵ_*.

To develop our algorithm, we first define *upper and lower confidence bounds* of the underlying utility functions:

$$\widehat{u}_{i,t-1}(\mathbf{x}) \coloneqq \mu_{i,t-1}(\mathbf{x}) + \beta_t\sigma_{t-1}(\mathbf{x}) \ ,$$
$$\breve{u}_{i,t-1}(\mathbf{x}) \coloneqq \mu_{i,t-1}(\mathbf{x}) - \beta_t\sigma_{t-1}(\mathbf{x}) \ ,$$
$$\beta_t \coloneqq B + \sigma(2\,(\gamma_{t-1} + 1 + \ln(1/\delta)))^{1/2}$$

where B is an upper bound of the RKHS norms of each u_i. From Lemma 9 (Chowdhury and Gopalan, 2017), with probability of at least $1-\delta$, for any $\mathbf{x}$ and t, the

[1] We emphasize that our setting differs from that of *multi-agent online learning* in which the agents aim to minimize their own losses, have individual policies, and usually know their utility functions, e.g., in (Cesa-Bianchi and Lugosi, 2006, Ch. 7). In contrast, our setting considers a single learner who aims to find a NE, coordinates the actions of all the agents during learning, and does not know their utility functions.



(a) Utility functions and pure NE $\mathbf{x}_*$.

(b) UCB-PNE's reported and exploring strategy profiles.

Figure 1: **Pure NE: (a)** shows the underlying utility functions of a 2-agent continuous general-sum game and the NE $\mathbf{x}_*$. At $\mathbf{x}_*$, neither of the agents can improve their utilities by unilaterally changing their strategies; each agent's space of possible other strategies is denoted by the dotted lines. The left plot in **(b)** shows the minimum (over agents) of the upper confidence bounds of each agent's negative best-response payoff in iteration t. The reported strategy profile $\mathbf{x}_t$ maximizes this function. The right plot assumes the exploring agent $j_t = 1$ and shows the upper confidence bound of agent 1's utility in iteration $t-1$. Agent 1's exploring strategy profile $\mathbf{x}^1$ maximizes this function over all strategy profiles with agent 2's strategy being held constant.

true $u_i(\mathbf{x})$ will lie between these confidence bounds, i.e., $u_i(\mathbf{x}) \in [\check{u}_{i,t-1}(\mathbf{x}), \widehat{u}_{i,t-1}(\mathbf{x})]$ for all $\mathbf{x} \in \mathcal{X}$ and $t \in \mathbb{Z}^+$.

We use these confidence bounds to further define the upper and lower confidence bounds of f_i in iteration t as

$$\begin{aligned}\widehat{f}_{i,t-1}(\mathbf{x}) &:= \widehat{u}_{i,t-1}(\mathbf{x}) - \max_{\mathbf{x}'_i \in \mathcal{X}_i} \check{u}_{i,t-1}(\mathbf{x}'_i, \mathbf{x}_{-i}) \\ \check{f}_{i,t-1}(\mathbf{x}) &:= \check{u}_{i,t-1}(\mathbf{x}) - \max_{\mathbf{x}'_i \in \mathcal{X}_i} \widehat{u}_{i,t-1}(\mathbf{x}'_i, \mathbf{x}_{-i})\end{aligned} \tag{4}$$

and the *exploring strategy profile* of agent i w.r.t. $\mathbf{x}$ in iteration t as

$$\mathbf{x}^i := \left(\operatorname{argmax}_{\mathbf{x}'_i \in \mathcal{X}_i} \widehat{u}_{i,t-1}(\mathbf{x}'_i, \mathbf{x}_{-i}), \mathbf{x}_{-i}\right). \tag{5}$$

Our first lemma proves that these confidence bounds of f_i are valid and the width of the resulting confidence interval at any $\mathbf{x}$ is upper bounded by some function of the GP posterior standard deviations at $\mathbf{x}$ and $\mathbf{x}^i$:

Lemma 1 *With probability of at least $1-\delta$, for all $i \in \mathcal{A}$, $\mathbf{x} \in \mathcal{X}$, and $t \in \mathbb{Z}^+$, the following hold:*

$$\begin{aligned}\check{f}_{i,t-1}(\mathbf{x}) \leqslant f_i(\mathbf{x}) \leqslant \widehat{f}_{i,t-1}(\mathbf{x}) \,, \\ \widehat{f}_{i,t-1}(\mathbf{x}) - \check{f}_{i,t-1}(\mathbf{x}) \leqslant 2\beta_t \left(\sigma_{t-1}(\mathbf{x}) + \sigma_{t-1}(\mathbf{x}^i)\right).\end{aligned}$$

Note that the GP posterior standard deviation σ_{t-1} (Equ. 2) does not have a subscript to index the agent since it is the same for all agents. This is because the GP posterior standard deviation only depends on the past selected decisions $(\tilde{\mathbf{x}}_j)_{j=1}^{t-1}$ (and not on the corresponding noisy function values $(y_{i,j})_{j=1}^{t-1}$) which are available to all agents.

Our acquisition function for finding pure NE called UCB-PNE is described in Algo. 1. UCB-PNE samples at either the reported strategy profile $\mathbf{x}_t$ or the exploring strategy profile $\mathbf{x}_t^{j_t}$ depending on which has the higher GP posterior standard deviation. Fig. 1b illustrates an example of

Algorithm 1 UCB-PNE

Input: n GPs each with kernel k, max. iteration T
for iteration $t = 1$ **to** T **do**
  Report strategy profile
    $\mathbf{x}_t := \operatorname{argmax}_{\mathbf{x}\in\mathcal{X}} \min_{i\in\mathcal{A}} \widehat{f}_{i,t-1}(\mathbf{x})$
  Sample at strategy profile
    $\tilde{\mathbf{x}}_t := \operatorname{argmax}_{\mathbf{x}\in\{\mathbf{x}_t, \mathbf{x}_t^{j_t}\}} \sigma_{t-1}(\mathbf{x})$ where
    $j_t := \operatorname{argmin}_{j\in\mathcal{A}} \check{f}_{j,t-1}(\mathbf{x}_t)$
  for agent $i = 1$ **to** n **do**
    Observe $y_{i,t} := u_i(\tilde{\mathbf{x}}_t) + \xi_i$ where $\xi_i \sim \mathcal{N}(0, \sigma^2)$
    Update agent i's GP posterior with
      $\mathcal{D}_{i,t} := \mathcal{D}_{i,t-1} \bigcup \{(\tilde{\mathbf{x}}_t, y_{i,t})\}$

an iteration of UCB-PNE with the reported and exploring strategy profiles of that iteration. Algo. 1 can be interpreted as a double application of the classic '*optimism in the face of uncertainty*' (OFU) principle (Lai et al., 1985): The reported strategy profile $\mathbf{x}_t := \operatorname{argmax}_{\mathbf{x}\in\mathcal{X}} \min_{i\in\mathcal{A}} \widehat{f}_{i,t-1}(\mathbf{x})$ is the maximizer of the minimum (over agents) of upper confidence bounds of f_i while the *exploring agent* $j_t := \operatorname{argmin}_{j\in\mathcal{A}} \check{f}_{j,t-1}(\mathbf{x}_t)$ is the agent with the minimum of lower confidence bounds of f_j (i.e., the agent who can potentially receive the largest best-response payoff). The exploring strategy profile $\mathbf{x}_t^{j_t}$ (Equ. 5) of j_t then optimistically assumes that the upper confidence bounds of u_{j_t} determine what this largest best-response payoff is. Computing the reported strategy profile $\mathbf{x}_t$ requires solving a bilevel optimization problem; we elaborate on our method of solving such a problem in App. B.1. Using Lemma 1, we guarantee the no-regret performance of UCB-PNE:

Theorem 1 *With probability of at least $1-\delta$, the sequence of reported strategy profiles $(\mathbf{x}_t)_{t=1}^T$ selected by UCB-PNE*

(Algo. 1) incurs a cumulative pure Nash regret bounded by

$$R_T = \mathcal{O}\left(\beta_T\sqrt{T\gamma_T}\right). \tag{6}$$

For the commonly used squared exponential kernel, $\gamma_T = \mathcal{O}((\log T)^{d+1}) \leqslant \mathcal{O}(\sqrt{T})$ (Srinivas et al., 2010). By dropping the polylogarithmic terms, Theorem 1 implies that the average pure Nash regret $R_T/T = \mathcal{O}(1/\sqrt{T}) \rightarrow 0$ as $T \rightarrow \infty$. So, UCB-PNE incurs no regret, which implies that $\mathbf{x}_t$ converges to an ϵ_*-NE. The proof bounds the pure Nash regret in each iteration t in terms of the GP posterior standard deviations of the reported strategy profile and exploring strategy profile of exploring agent j_t via Lemma 1. Then, R_T can be written as a sum of GP posterior standard deviations which can be bounded by the RHS of Equ. 6 via Lemma 10 (Chowdhury and Gopalan, 2017). UCB-PNE's design is determined by Theorem 1 that prescribes the sampled strategy profiles required to incur no regret.

Interestingly, under some further assumptions, the sequence of sampled strategy profiles (along with the reported ones) can also be shown to be no regret:

Theorem 2 *Suppose that the following assumptions hold:* ***(1)*** *For all $i \in \mathcal{A}$, u_i is continuous and bounded;* ***(2)*** *Agent-wise maximizers are unique, i.e., $\forall \mathbf{x} \in \mathcal{X}\ \ \forall i \in \mathcal{A}\ \ \exists! \mathbf{x}_i \in \mathcal{X}_i\ \ u_i(\mathbf{x}_i, \mathbf{x}_{-i}) = \max_{\mathbf{x}'_i \in \mathcal{X}_i} u_i(\mathbf{x}'_i, \mathbf{x}_{-i})$;* ***(3)*** *There exists a unique NE $\mathbf{x}_*$;* ***(4)*** *$\gamma_T < \mathcal{O}(T)$. Then, with probability of at least $1-\delta$, the sequence of sampled strategy profiles $(\tilde{\mathbf{x}}_t)_{t=1}^T$ chosen by UCB-PNE (Algo. 1) is no regret (in terms of cumulative pure Nash regret):*

$$\lim_{T\rightarrow\infty} T^{-1} \textstyle\sum_{t=1}^T -\epsilon_* - \min_{i\in\mathcal{A}} f_i(\tilde{\mathbf{x}}_t) = 0\ .$$

To see the intuition of Theorem 2, the reported strategy profiles $\mathbf{x}_t$ converge to the unique NE $\mathbf{x}_*$ by Theorem 1. Since agent-wise maximizers are unique and the exploring strategy profiles $\mathbf{x}_t^{j_t}$ maximize the upper confidence bounds along an agent's dimensions, the same maximum information gain argument can be leveraged to show that $\mathbf{x}_t^{j_t}$ converges to $\mathbf{x}_*$ as well. The challenge is in proving this fact when $\mathbf{x}_t$ only converges to $\mathbf{x}_*$ instead of being equal.

5 MIXED NASH EQUILIBRIA (NE)

In the mixed NE setting, each agent $i \in \mathcal{A}$ is still associated with a compact set $\mathcal{X}_i \subset \mathbb{R}^{d_i}$ of pure strategies. However, we now consider discrete normal-form general-sum games in which each agent i's decision is a vector-valued *mixed strategy* $\mathbf{x}_i^\triangle \in \mathbb{R}^{m_i}$ (subject to $\mathbf{1}^\top \mathbf{x}_i^\triangle = 1$ and $\mathbf{x}_i^\triangle \succeq \mathbf{0}$) corresponding to a probability distribution over a finite set $\tilde{\mathcal{X}}_i$ of m_i strategies that is a discretization of $\mathcal{X}_i$. A *mixed strategy profile* is denoted by a vector $\mathbf{x}^\triangle \in \mathbb{R}^{\sum_{i=1}^n m_i}$ concatenating all agents' mixed strategies $\mathbf{x}_1^\triangle, \ldots, \mathbf{x}_n^\triangle$. Each agent is again associated with an unknown utility function $u_i : \mathbb{R}^d \rightarrow \mathbb{R}$ in the RKHS of k that maps from each pure strategy profile to its obtained utility when that strategy profile is played. In this discrete game setting with $m := \prod_{i=1}^n m_i$ possible pure strategy profiles, each u_i can be equivalently represented by a vector $\mathbf{u}_i := (u_i(\tilde{\mathbf{x}}^{(j)}))_{j=1}^m \in \mathbb{R}^m$ where $\tilde{\mathbf{x}}^{(j)}$ is the j-th pure strategy profile in $\tilde{\mathcal{X}} := \times_{i=1}^n \tilde{\mathcal{X}}_i \subset \mathbb{R}^d$ s.t. $|\tilde{\mathcal{X}}| = m$. Define the mapping $\mathbf{p}(\mathbf{x}^\triangle) : \mathbb{R}^{\sum_{i=1}^n m_i} \rightarrow \mathbb{R}^m$ s.t. the j-th element of $\mathbf{p}(\mathbf{x}^\triangle)$ is the probability that the j-th pure strategy profile $\tilde{\mathbf{x}}^{(j)} \in \tilde{\mathcal{X}}$ is played when mixed strategy profile $\mathbf{x}^\triangle$ is played. In other words, $\mathbf{p}(\mathbf{x}^\triangle)$ is a probability distribution over $\tilde{\mathcal{X}}$ (subject to $\mathbf{1}^\top \mathbf{p}(\mathbf{x}^\triangle) = 1$ and $\mathbf{p}(\mathbf{x}^\triangle) \geq \mathbf{0}$) and can be constructed by taking each agent's mixed strategy in $\mathbf{x}^\triangle$ to be independent and taking the product of the probabilities of the strategies that correspond to a particular pure strategy profile. Fig. 2a illustrates an example of $\mathbf{x}^\triangle$ with $\mathbf{p}(\mathbf{x}^\triangle)$. We can thus overload the notation u_i to refer to agent i's *expected utility* under $\mathbf{x}^\triangle$:

$$u_i(\mathbf{x}^\triangle) = \mathbf{p}(\mathbf{x}^\triangle)^\top \mathbf{u}_i\ .$$

We define agent i's negative best-response payoff in the mixed NE setting as

$$f_i(\mathbf{x}^\triangle) := u_i(\mathbf{x}^\triangle) - \max_{\mathbf{x}_i \in \mathcal{X}_i} u_i(\mathbf{x}_i, \mathbf{x}_{-i}^\triangle)\ .$$

Note that when taking the maximum over agent i's strategies, we only need to consider pure strategies and not mixed ones as the maximum is attained by assigning all probability mass to the one strategy with the largest expected utility given all other agents' mixed strategies. The space of all possible mixed strategy profiles is denoted by $\mathcal{X}^\triangle$. A mixed NE is a strategy profile $\mathbf{x}_*^\triangle$ s.t.

$$\mathbf{x}_*^\triangle \in \mathcal{X}_*^\triangle := \{\mathbf{x}^\triangle \in \mathcal{X}^\triangle \,|\, \forall i \in \mathcal{A}\ \ f_i(\mathbf{x}_*^\triangle) = 0\}\ .$$

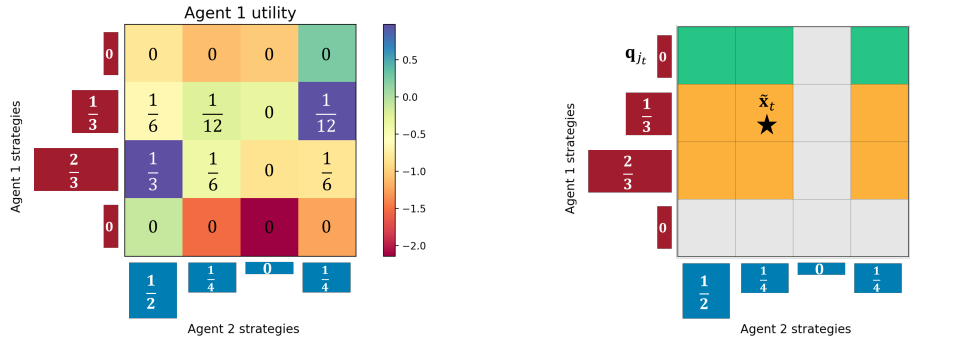
The ϵ-relaxation is unnecessary here as there always exists at least a mixed NE when the number of agents and the number of strategies are both finite (Nash, 1951).

5.1 Finding Mixed NE with No-regret BO Algorithm

In each iteration t, the learner is allowed to sample at any pure strategy profile $\tilde{\mathbf{x}}_t \in \tilde{\mathcal{X}}$ and receive sampled noisy function values in the same manner as that in the pure NE setting (Sec. 4.1). The learner is not allowed to sample at arbitrary mixed strategy profiles as it is not feasible in practice to directly observe the expected utility of a mixed strategy profile, especially since the samples are costly (Sec. 1). In each iteration t, the learner reports a mixed strategy profile $\mathbf{x}_t^\triangle$ that it thinks to be an NE. We define the *cumulative mixed Nash regret* as

$$R_T^\triangle := \textstyle\sum_{t=1}^T - \min_{i\in\mathcal{A}} f_i(\mathbf{x}_t^\triangle)\ . \tag{7}$$

There are two possible model choices for constructing a GP posterior over each agent's utility: Firstly, we can



(a) Agent 1's utility function and an example mixed strategy profile.

(b) UCB-MNE's potentially sampled strategy profiles.

Figure 2: **Mixed NE: (a)** shows agent 1's utility (and omits agent 2's utility to ease exposition) in a 2-player discrete general-sum game with an example mixed strategy profile $\mathbf{x}^\triangle$ (red and blue bars). Each row (or column) corresponds to one of agent 1's (or 2's) pure strategies with $m_1 = m_2 = 4$, and each box corresponds to a pure strategy profile. The colour of each box represents the utility agent 1 receives from that pure strategy profile, and the numbers in the boxes are the joint probability distribution $\mathbf{p}(\mathbf{x}^\triangle)$ induced by $\mathbf{x}^\triangle$. **(b)** assumes that $\mathbf{x}_t^\triangle$ is the shown mixed strategy profile, $j_t = 1$, and $\mathbf{q}_{j_t}$ is the strategy represented by the first row. The set of pure strategy profiles $\text{supp}(\mathbf{x}_t^\triangle)$ is shaded yellow and $\text{supp}(\mathbf{q}_{j_t}, \mathbf{x}_{-j_t,t}^\triangle)$ is shaded green. UCB-MNE samples at the pure strategy profile $\tilde{\mathbf{x}}_t$ with the largest posterior standard deviation among the shaded strategy profiles.

use a *pure-space GP* model to represent $\mathbf{u}_i$ with decision space $\tilde{\mathcal{X}}$ and kernel k. Since $\mathbf{u}_i$ is a finite vector, this simplifies to placing a multivariate Gaussian prior on $\mathbf{u}_i$ with distribution $\mathcal{N}(\mathbf{0}, \tilde{\mathbf{K}} := (k(\tilde{\mathbf{x}}^{(j)}, \tilde{\mathbf{x}}^{(\ell)}))_{j,\ell=1}^{m})$. In this model, if $\mathbf{u}_i$ is distributed according to $\mathcal{N}(\mathbf{m}, \mathbf{S})$ for some mean vector $\mathbf{m}$ and covariance matrix $\mathbf{S}$, then the expected utility $u_i(\mathbf{x}^\triangle)$ will be distributed according to $\mathcal{N}(\mathbf{p}(\mathbf{x}^\triangle)^\top \mathbf{m}, \mathbf{p}(\mathbf{x}^\triangle)^\top \mathbf{S}\ \mathbf{p}(\mathbf{x}^\triangle))$ since it is a linear transformation of a Gaussian. Alternatively, we can use a *mixed-space GP* model to directly represent $u_i(\mathbf{x}^\triangle)$ with decision space $\mathcal{X}^\triangle$. However, since the decision space is no longer a subset of $\mathcal{X}$, kernel k cannot be directly used for the mixed-space GP model. In order for any finite collection of $u_i(\mathbf{x}_1^\triangle), \ldots, u_i(\mathbf{x}_q^\triangle)$ to be distributed in the same way as if we had used a pure-space GP model, we require an appropriate transformation to k, as given by the next result:

Proposition 1 *Let $k^\triangle$ be a positive semidefinite kernel defined as $k^\triangle(\mathbf{x}_j^\triangle, \mathbf{x}_{j'}^\triangle) := \mathbf{p}(\mathbf{x}_j^\triangle)^\top \tilde{\mathbf{K}}\ \mathbf{p}(\mathbf{x}_{j'}^\triangle)$. Then, a mixed-space GP model with prior mean $\mathbf{0}$ and kernel $k^\triangle$ yields the same predictive mean of any $u_i(\mathbf{x}^\triangle)$ and covariance between any $u_i(\mathbf{x}_j^\triangle)$ and $u_i(\mathbf{x}_{j'}^\triangle)$ as that obtained with a linear transformation of the pure-space GP predictive distribution over $\mathbf{u}_i$ with prior mean $\mathbf{0}$ and kernel k.*

It may seem at first glance that we can use Algo. 1 and Theorem 1 with a mixed-space GP model and the kernel $k^\triangle$ to guarantee convergence towards a mixed NE by treating $\mathcal{X}^\triangle$ as a 'pure' strategy profile space. This would be the case if the learner is allowed to sample at an arbitrary $\mathbf{x}^\triangle$ to directly observe a noise-corrupted $u_i(\mathbf{x}^\triangle)$. However, this violates our practical learning setting that the learner can only sample at pure strategy profiles. Algo. 1 and Theorem 1 are thus not directly applicable with a mixed-space GP model. Moreover, using Algo. 1 prevents us from exploiting the structure of the mixed NE problem (without non-trivial modifications), e.g., by choosing to first search for mixed NE at which all agents have supports of equal size (Stengel, 2007, pp. 56). We find that working with pure-space GP models allows the previously developed theory for pure NE to transfer elegantly with some necessary modifications and also enables us to take advantage of existing mixed NE solvers that exploit the problem structure. So, the rest of this section is devoted to finding mixed NE via pure-space GP models.

We first define upper and lower bounds of the negative best-response payoff using the overloaded notations of the upper and lower confidence bounds of the underlying utility functions $\widehat{u}_i(\mathbf{x}^\triangle) := \mathbf{p}(\mathbf{x}^\triangle)^\top \widehat{\mathbf{u}}_i$ and $\breve{u}_i(\mathbf{x}^\triangle) := \mathbf{p}(\mathbf{x}^\triangle)^\top \breve{\mathbf{u}}_i$:

$$\widehat{f}_{i,t-1}(\mathbf{x}^\triangle) := \widehat{u}_{i,t-1}(\mathbf{x}^\triangle) - \max_{\mathbf{x}_i \in \mathcal{X}_i} \breve{u}_{i,t-1}(\mathbf{x}_i, \mathbf{x}_{-i}^\triangle)$$
$$\breve{f}_{i,t-1}(\mathbf{x}^\triangle) := \breve{u}_{i,t-1}(\mathbf{x}^\triangle) - \max_{\mathbf{x}_i \in \mathcal{X}_i} \widehat{u}_{i,t-1}(\mathbf{x}_i, \mathbf{x}_{-i}^\triangle)\ .$$

We also redefine the exploring pure strategy profile w.r.t. a mixed strategy profile $\mathbf{x}^\triangle$ for mixed NE as

$$\mathbf{x}^i := \operatorname{argmax}_{\mathbf{x}' \in \text{supp}(\mathbf{q}_i, \mathbf{x}_{-i}^\triangle)} \sigma_{t-1}(\mathbf{x}') \tag{8}$$
$$\mathbf{q}_i := \operatorname{argmax}_{\mathbf{x}_i' \in \mathcal{X}_i} \widehat{u}_{i,t-1}(\mathbf{x}_i', \mathbf{x}_{-i}^\triangle) \tag{9}$$

where $\text{supp}(\mathbf{x}^\triangle)$ is the set of all pure strategy profiles with non-zero probability under $\mathbf{p}(\mathbf{x}^\triangle)$, and define an additional *exploiting pure strategy profile* w.r.t. a mixed strategy profile $\mathbf{x}^\triangle$ as

$$\bar{\mathbf{x}} := \operatorname{argmax}_{\mathbf{x}' \in \text{supp}(\mathbf{x}^\triangle)} \sigma_{t-1}(\mathbf{x}')\ . \tag{10}$$

The next result is a direct analog of Lemma 1 for the mixed NE setting. The additional insight is that the pure strategy profiles forming the upper bound of the width of the confidence interval of f_i are those in the supports of $\mathbf{x}^\triangle$ and $(\mathbf{q}_i, \mathbf{x}^\triangle_{-i})$ with the largest GP posterior standard deviations.

Lemma 2 *With probability of at least $1-\delta$, for all $i \in \mathcal{A}$, $\mathbf{x}^\triangle \in \mathbf{X}^\triangle$, and $t \in \mathbb{Z}^+$, the following hold:*

$$\breve{f}_{i,t-1}(\mathbf{x}^\triangle) \leqslant f_i(\mathbf{x}^\triangle) \leqslant \widehat{f}_{i,t-1}(\mathbf{x}^\triangle)\ ,$$
$$\widehat{f}_{i,t-1}(\mathbf{x}^\triangle) - \breve{f}_{i,t-1}(\mathbf{x}^\triangle) \leqslant 2\beta_t \left(\sigma_{t-1}(\overline{\mathbf{x}}) + \sigma_{t-1}(\mathbf{x}^i)\right).$$

Our acquisition function for finding mixed NE called UCB-MNE is described in Algo. 2 and illustrated in Fig. 2b. In each iteration t, UCB-MNE samples a utility function $\tilde{\mathbf{u}}_{i,t-1}$ for each agent i s.t. $\breve{\mathbf{u}}_{i,t-1} \preceq \tilde{\mathbf{u}}_{i,t-1} \preceq \widehat{\mathbf{u}}_{i,t-1}$ (using the vector representation of utility functions) and computes the reported mixed NE $\mathbf{x}^\triangle_t$ using each $\tilde{\mathbf{u}}_{i,t-1}$ as ground truth; we elaborate on our method of computing mixed NE in App. B.2. Based on $\mathbf{x}^\triangle_t$, UCB-MNE then samples at either the exploiting strategy profile $\overline{\mathbf{x}}_t$ (Equ. 10) or the exploring pure strategy profile $\mathbf{x}^{j_t}_t$ (Equ. 8) of the exploring agent $j_t := \operatorname{argmin}_{j\in\mathcal{A}} \breve{f}_{j,t-1}(\mathbf{x}^\triangle_t)$ depending on which of the two has the larger uncertainty. Similar to UCB-PNE, this may be interpreted as a double application of the OFU principle: By sampling the utility functions within the confidence bounds, we optimistically assume these functions to be ground truth and compute a potential mixed NE based on those functions. Since j_t is the agent with the potentially largest best-response payoff, UCB-MNE again optimistically assumes that the upper confidence bounds of u_{j_t} determine what the potential best-response payoff is when selecting the exploring pure strategy profile. The theory transfers elegantly: We prove using Lemma 2 that this sequence of reported mixed strategy profiles incurs no regret and so, $\mathbf{x}^\triangle_t$ converges to a mixed NE; proof sketch is similar to that of Theorem 1 but adapted to mixed NE setting:

Theorem 3 *With probability of at least $1-\delta$, the sequence of reported strategy profiles $(\mathbf{x}^\triangle_t)^T_{t=1}$ selected by UCB-MNE (Algo. 2) incurs a cumulative mixed Nash regret bounded by*

$$R^\triangle_T = \mathcal{O}\left(\beta_T\sqrt{T\gamma_T}\right).$$

6 EXPERIMENTS AND DISCUSSION

We empirically evaluate our acquisition functions in various games in both the pure and mixed NE settings. We first describe the games defined by their utility (objective) functions, followed by the acquisition functions tested in both settings; App. C details the full description of the utility functions, acquisition function hyperparameters, and computation time. All experiments are tested over 5 RNG seeds which affect the utilities, the initial samples, and acquisition functions with randomness. The code for the experiments can be found at `https://github.com/sebtsh/nash-bo`.

Algorithm 2 UCB-MNE

Input: n GPs each with kernel k, max. iteration T
for iteration $t = 1$ **to** T **do**
  Sample utility functions $\tilde{\mathbf{u}}_{i,t-1}$ s.t. $\breve{\mathbf{u}}_{i,t-1} \preceq \tilde{\mathbf{u}}_{i,t-1} \preceq \widehat{\mathbf{u}}_{i,t-1}$ for all $i \in \mathcal{A}$
  Report mixed strategy profile $\mathbf{x}^\triangle_t \leftarrow$ mixed NE computed using $\tilde{\mathbf{u}}_{i,t-1}$ for all $i \in \mathcal{A}$ as ground truth
  Sample at strategy profile $\tilde{\mathbf{x}}_t := \operatorname{argmax}_{\mathbf{x}\in\{\overline{\mathbf{x}}_t, \mathbf{x}^{j_t}_t\}} \sigma_{t-1}(\mathbf{x})$ where $j_t := \operatorname{argmin}_{j\in\mathcal{A}} \breve{f}_{j,t-1}(\mathbf{x}^\triangle_t)$
  for agent $i = 1$ **to** n **do**
    Observe $y_{i,t} := u_i(\tilde{\mathbf{x}}_t) + \xi_i$ where $\xi_i \sim \mathcal{N}(0, \sigma^2)$
    Update agent i's GP posterior with $\mathcal{D}_{i,t} := \mathcal{D}_{i,t-1} \bigcup \{(\tilde{\mathbf{x}}_t, y_{i,t})\}$

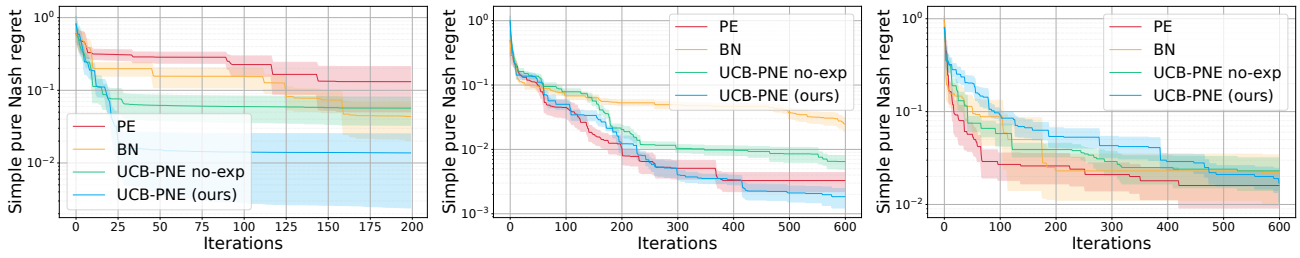
Synthetic Random Functions (RF): We construct a 2-player general-sum game by sampling two random 2-D functions from a GP prior and using them as utility functions. Each agent's strategy is in $[-1, 1]$. To test the acquisition functions' ability to handle games with non-0 ϵ_*, we chose 5 RNG seeds s.t. $\epsilon_* \geqslant 0.05$ in the resulting games.

Generative Adversarial Networks (GAN) on 1-D Data Manifold: GANs are a class of generative models that consist of a generator and a discriminator. The desired generator and discriminator parameters form a Nash equilibrium of a 2-player general-sum game (Salimans et al., 2016). We evaluate our algorithms on a simple GAN setup inspired by the experiments in Fedus et al. (2018). The generator generates data on a 1-D manifold with strategies (parameters) in $[-1, 1]^2$, while the discriminator is a binary classifier with strategies in $[-1, 1]^3$. This setup has $\epsilon_* = 0$.

Binary Classifiers' Additive Adversarial Attack and Defense (BCAD): We evaluate our algorithms on finding NE in the 2-player zero-sum game of additive adversarial attacks and defenses on binary classifiers, as described in Pal and Vidal (2020). We use a simple binary classifier that classifies points in $\mathbb{R}^2$. The attacker selects a vector field of perturbations parameterized by strategies in $[-1, 1]^3$. The defender selects a constant perturbation parameterized by strategies in $[-1, 1]^2$. Pal and Vidal (2020) show that an NE always exists when the utility functions are defined in a specific way and so, $\epsilon_* = 0$ in this setup.

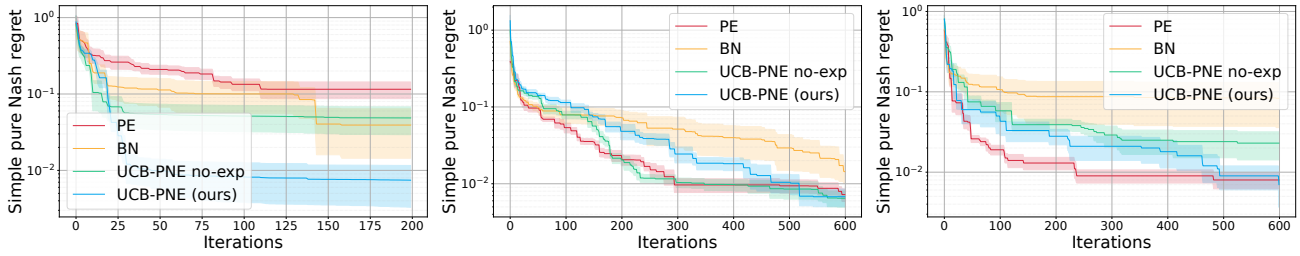
Pure NE. For the pure NE setting, we compare UCB-PNE with Probability of Equilibrium (PE)[2] (Picheny et al., 2019)

[2]PE is not designed for continuous strategy profile sets: To make it work, we discretize each agent's strategy set into a finite



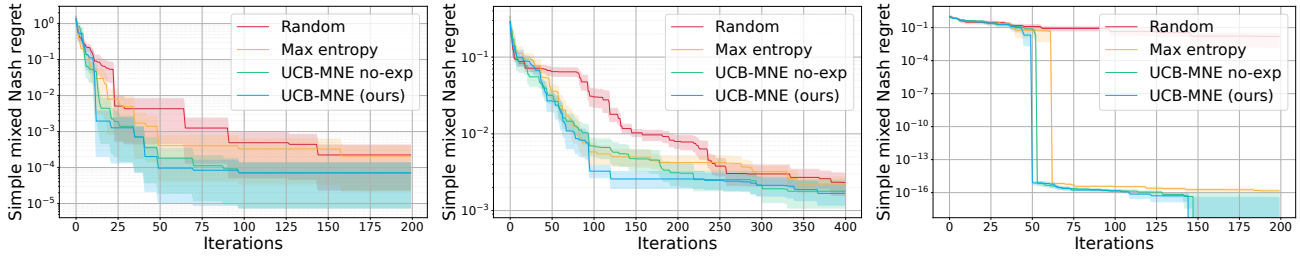
(a) RF (b) GAN (c) BCAD

Figure 3: Mean and standard error of simple pure Nash regrets of **reported** strategy profiles.



(a) RF (b) GAN (c) BCAD

Figure 4: Mean and standard error of simple pure Nash regrets of **sampled** strategy profiles.



(a) RF (b) GAN (c) BCAD

Figure 5: Mean and standard error of simple mixed Nash regrets.

and BN (Al-Dujaili et al., 2018). We also do an ablation study of UCB-PNE by removing its ability to select an exploring strategy profile to sample at and call this UCB-PNE no-exp. For PE and BN, the reported strategy profile is computed in each iteration by taking the GP posterior means as surrogates to each u_i and solving the bilevel optimization problem $\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}} \min_{i\in\mathcal{A}} f_i(\mathbf{x})$ (App. B.1). Fig. 3 shows the simple pure Nash regret $\min_{j\leqslant t} -\epsilon_* - \min_{i\in\mathcal{A}} f_i(\mathbf{x}_j)$ incurred by the **reported** strategy profiles of each acquisition function with each game at each iteration; 0 indicates that the ϵ_*-NE was exactly reported at some iteration. It can be observed that UCB-PNE incurs the least simple pure Nash regret by the end in RF and GAN. While PE performs well in the games with $\epsilon_* = 0$ (GAN and BCAD), it performs poorly in RF when $\epsilon_* > 0$, which is expected as its behavior is unknown when the probability of an equilibrium is 0 everywhere. In GAN and BCAD, UCB-PNE continues to improve its simple regret in the last few iterations while PE stops improving in the last 200 or so iterations. We also plot the simple pure Nash regret incurred by the **sampled** strategy profiles in Fig. 4. It can be observed that UCB-PNE again generally outperforms the rest and only incurs slightly higher regret than UCB-PNE no-exp by the end in GAN.

Mixed NE. For the mixed NE setting, since there is no prior work, we compare UCB-MNE to its ablated form without exploring strategy profiles (and call this UCB-MNE no-exp) as well as to two simple heuristics: one sampling at the pure strategy profile with the largest GP posterior variance (i.e., maximum entropy) and one sampling at a profile selected uniformly at random. We discretize each agent's strategy set into 10 strategies for RF and GAN, and 16 strategies for BCAD. Fig. 5 shows the simple mixed

set with uniform random sampling in each iteration as we found performance to be poor with a fixed strategy set at every iteration. We do not compare to SUR (Picheny et al., 2019) (i.e., also not designed for continuous spaces) as PE was empirically shown to perform on par with or better than SUR in both previous works.

Nash regrets $\min_{j \leqslant t} - \min_{i \in \mathcal{A}} f_i(\mathbf{x}_j^{\triangle})$ for different acquisition functions and utility functions. It can be observed that UCB-MNE and UCB-MNE no-exp consistently incur the least simple mixed Nash regret by the end. While both are able to outperform the simple heuristics by the end, UCB-MNE finds solutions with low regret in slightly less iterations: The ability to sample at exploring strategy profiles allows the learner to find better solutions faster.

7 CONCLUSION

This paper describes the first no-regret BO algorithms that are sample-efficient in finding pure and mixed NE in general-sum games with unknown costly-to-sample utility functions. This line of research has several potential avenues of further exploration. In particular, the algorithms tested in this work are difficult to scale to domains with a large number of dimensions due to the bilevel optimization for UCB-PNE and due to the need of discretization for PE and UCB-MNE. Future work may explore acquisition functions that are able to find NE in high dimensions (Hoang et al., 2018) in a computationally tractable manner. Finally, we will consider generalizing our algorithms to cater to deep neural networks as the surrogate model (instead of a GP model) (Dai et al., 2022b), batch mode (Daxberger and Low, 2017), information-theoretic acquisition functions (Nguyen et al., 2021c,e), preferences (Nguyen et al., 2021d), multi-fidelity function evaluations (Zhang et al., 2017, 2019), nonmyopic planning with lookaheads (Kharkovskii et al., 2020b; Ling et al., 2016), uncontrollable environmental random variables (Nguyen et al., 2021a,b; Tay et al., 2021), fairness (Sim et al., 2021), differential privacy (Kharkovskii et al., 2020a), early stopping (Dai et al., 2019), delayed feedback (Verma et al., 2022), federated learning (Dai et al., 2020b, 2021, 2023) and meta-learning (Dai et al., 2022a) settings, and consider its application to neural architecture search (Shu et al., 2022a,b,c) and inverse reinforcement learning (Balakrishnan et al., 2020). For applications with a huge budget of function evaluations, we like to couple our algorithm with the use of distributed/decentralized (Chen et al., 2012, 2013a,b, 2015; Hoang et al., 2016, 2019; Low et al., 2015; Ouyang and Low, 2018), online/stochastic (Hoang et al., 2015, 2017; Low et al., 2014; Xu et al., 2014; Yu et al., 2019b), or deep (Yu et al., 2019a, 2021) sparse GP models to represent the belief of the unknown utility functions efficiently.

Acknowledgements

This research is part of the programme DesCartes and is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. Sebastian Shenghong Tay is supported by the Agency for Science, Technology and Research (A*STAR), Singapore.

References

Abbasi-Yadkori, Y. (2012). *Online Learning for Linearly Parametrized Control Problems*. Ph.D. Thesis, University of Alberta.

Al-Dujaili, A., Hemberg, E., and O'Reilly, U.-M. (2018). Approximating Nash equilibria for black-box games: A Bayesian optimization approach. arXiv:1804.10586.

Balakrishnan, S., Nguyen, Q. P., Low, B. K. H., and Soh, H. (2020). Efficient exploration of reward functions in inverse reinforcement learning via Bayesian optimization. In *Proc. NeurIPS*, pages 4187–4198.

Başar, T. (1987). Relaxation techniques and asynchronous algorithms for on-line computation of non-cooperative equilibria. *J. Economic Dynamics and Control*, 11(4):531–549.

Bogunovic, I., Scarlett, J., Jegelka, S., and Cevher, V. (2018). Adversarially robust optimization with Gaussian processes. In *Proc. NeurIPS*.

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

Chen, J., Cao, N., Low, K. H., Ouyang, R., Tan, C. K.-Y., and Jaillet, P. (2013a). Parallel Gaussian process regression with low-rank covariance matrix approximations. In *Proc. UAI*, pages 152–161.

Chen, J., Low, K. H., Jaillet, P., and Yao, Y. (2015). Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems. *IEEE Trans. Autom. Sci. Eng.*, 12:901–921.

Chen, J., Low, K. H., and Tan, C. K.-Y. (2013b). Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system. In *Proc. RSS*.

Chen, J., Low, K. H., Tan, C. K.-Y., Oran, A., Jaillet, P., Dolan, J. M., and Sukhatme, G. S. (2012). Decentralized data fusion and active sensing with mobile sensors for modeling and predicting spatiotemporal traffic phenomena. In *Proc. UAI*, pages 163–173.

Chen, Y., Huang, A., Wang, Z., Antonoglou, I., Schrittwieser, J., Silver, D., and de Freitas, N. (2018). Bayesian optimization in AlphaGo. arXiv:1812.06855.

Chowdhury, S. R. and Gopalan, A. (2017). On kernelized multi-armed bandits. In *Proc. ICML*, pages 844–853.

Dai, Z., Chen, Y., Low, B. K. H., Jaillet, P., and Ho, T.-H. (2020a). R2-B2: Recursive reasoning-based Bayesian optimization for no-regret learning in games. In *Proc. ICML*, pages 2291–2301.

Dai, Z., Chen, Y., Yu, H., Low, B. K. H., and Jaillet, P. (2022a). On provably robust meta-Bayesian optimization. In *Proc. UAI*, pages 475–485.

Dai, Z., Low, B. K. H., and Jaillet, P. (2020b). Federated Bayesian optimization via Thompson sampling. In *Proc. NeurIPS*, pages 9687–9699.

Dai, Z., Low, B. K. H., and Jaillet, P. (2021). Differentially private federated Bayesian optimization with distributed exploration. In *Proc. NeurIPS*, pages 9125–9139.

Dai, Z., Shu, Y., Low, B. K. H., and Jaillet, P. (2022b). Sample-then-optimize batch neural Thompson sampling. In *Proc. NeurIPS*.

Dai, Z., Shu, Y., Verma, A., Fan, F. X., Low, B. K. H., and Jaillet, P. (2023). Federated neural bandits. In *Proc. ICLR*.

Dai, Z., Yu, H., Low, B. K. H., and Jaillet, P. (2019). Bayesian optimization meets Bayesian optimal stopping. In *Proc. ICML*, pages 1496–1506.

Daxberger, E. A. and Low, B. K. H. (2017). Distributed batch Gaussian process optimization. In *Proc. ICML*, pages 951–960.

Debreu, G. (1952). A social equilibrium existence theorem. *Proc. National Academy of Sciences of the United States of America*, 38(10):886–893.

Djehiche, B., Tcheukam, A., and Tembine, H. (2017). Mean-field-type games in engineering. *AIMS Electronics and Electrical Engineering*, 1(1):18–73.

Fedus, W., Rosca, M., Lakshminarayanan, B., Dai, A. M., Mohamed, S., and Goodfellow, I. (2018). Many paths to equilibrium: GANs do not need to decrease a divergence at every step. In *Proc. ICLR*.

Garnett, R. (2022). *Bayesian Optimization*. Cambridge Univ. Press. In preparation.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Proc. NeurIPS*, pages 2672–2680.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

Hoang, Q. M., Hoang, T. N., and Low, K. H. (2017). A generalized stochastic variational Bayesian hyperparameter learning framework for sparse spectrum Gaussian process regression. In *Proc. AAAI*, pages 2007–2014.

Hoang, T. N., Hoang, Q. M., and Low, B. K. H. (2018). Decentralized high-dimensional Bayesian optimization with factor graphs. In *Proc. AAAI*, pages 3231–3238.

Hoang, T. N., Hoang, Q. M., and Low, K. H. (2015). A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data. In *Proc. ICML*, pages 569–578.

Hoang, T. N., Hoang, Q. M., and Low, K. H. (2016). A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In *Proc. ICML*, pages 382–391.

Hoang, T. N., Hoang, Q. M., Low, K. H., and How, J. P. (2019). Collective online learning of Gaussian processes in massive multi-agent systems. In *Proc. AAAI*.

Jones, D. R., Perttunen, C. D., and Stuckman, B. E. (1993). Lipschitzian optimization without the Lipschitz constant. *J. Optimization Theory and Applications*, 79(1):157–181.

Kharkovskii, D., Dai, Z., and Low, B. K. H. (2020a). Private outsourced Bayesian optimization. In *Proc. ICML*, pages 5231–5242.

Kharkovskii, D., Ling, C. K., and Low, B. K. H. (2020b). Nonmyopic Gaussian process optimization with macro-actions. In *Proc. AISTATS*, pages 4593–4604.

Lai, T. L., Robbins, H., et al. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22.

Lemke, C. E. and Howson, Jr, J. T. (1964). Equilibrium points of bimatrix games. *Journal of the Society for Industrial and Applied Mathematics*, 12(2):413–423.

Ling, C. K., Low, B. K. H., and Jaillet, P. (2016). Gaussian process planning with Lipschitz continuous reward functions: Towards unifying Bayesian optimization, active learning, and beyond. In *Proc. AAAI*, pages 1860–1866.

Liu, S., Lu, S., Chen, X., Feng, Y., Xu, K., Al-Dujaili, A., Hong, M., and O'Reilly, U.-M. (2020). Min-max optimization without gradients: Convergence and applications to black-box evasion and poisoning attacks. In *Proc. ICML*, pages 6282–6293.

Low, K. H., Xu, N., Chen, J., Lim, K. K., and Özgül, E. B. (2014). Generalized online sparse Gaussian processes with application to persistent mobile robot localization. In *Proc. ECML/PKDD Nectar Track*, pages 499–503.

Low, K. H., Yu, J., Chen, J., and Jaillet, P. (2015). Parallel Gaussian process regression for big data: Low-rank representation meets Markov approximation. In *Proc. AAAI*, pages 2821–2827.

Lyu, X., Xiao, Y., Daley, B., and Amato, C. (2021). Contrasting centralized and decentralized critics in multi-agent reinforcement learning. In *Proc. AAMAS*, pages 844–852.

Marchesi, A., Trovò, F., and Gatti, N. (2020). Learning probably approximately correct maximin strategies in simulation-based games with infinite strategy spaces. In *Proc. AAMAS*, page 834–842.

Matthews, A. G. d. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. (2017). GPflow: A Gaussian process library using TensorFlow. *J. Machine Learning Research*, 18(40):1–6.

Nash, J. (1951). Non-cooperative games. *Annals of Mathematics*, 54(2):286–295.

Nguyen, Q. P., Dai, Z., Low, B. K. H., and Jaillet, P. (2021a). Optimizing conditional value-at-risk of black-box functions. In *Proc. NeurIPS*.

Nguyen, Q. P., Dai, Z., Low, B. K. H., and Jaillet, P. (2021b). Value-at-risk optimization with Gaussian processes. In *Proc. ICML*, pages 8063–8072.

Nguyen, Q. P., Low, B. K. H., and Jaillet, P. (2021c). An information-theoretic framework for unifying active learning problems. In *Proc. AAAI*, pages 9126–9134.

Nguyen, Q. P., Tay, S., Low, B. K. H., and Jaillet, P. (2021d). Top-k ranking Bayesian optimization. In *Proc. AAAI*, pages 9135–9143.

Nguyen, Q. P., Wu, Z., Low, B. K. H., and Jaillet, P. (2021e). Trusted-maximizers entropy search for efficient Bayesian optimization. In *Proc. UAI*, pages 1486–1495.

Ouyang, R. and Low, K. H. (2018). Gaussian process decentralized data fusion meets transfer learning in large-scale distributed cooperative perception. In *Proc. AAAI*, pages 3876–3883.

Pal, A. and Vidal, R. (2020). A game theoretic analysis of additive adversarial attacks and defenses. In *Proc. NeurIPS*, pages 1345–1355.

Picheny, V., Binois, M., and Habbal, A. (2019). A Bayesian optimization approach to find Nash equilibria. *J. Global Optimization*, 73(1):171–192.

Porter, R., Nudelman, E., and Shoham, Y. (2008). Simple search methods for finding a Nash equilibrium. *Games and Economic Behavior*, 63(2):642–662.

Reeves, D. and Wellman, M. P. (2012). Computing best-response strategies in infinite games of incomplete information. arXiv:1207.4171.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training GANs. In *Proc. NeurIPS*, pages 2234–2242.

Schelling, T. C. (1980). *The Strategy of Conflict: With a New Preface by the Author*. Harvard Univ. Press.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.

Sessa, P. G., Bogunovic, I., Kamgarpour, M., and Krause, A. (2019). No-regret learning in unknown games with correlated payoffs. In *Proc. NeurIPS*, pages 13624–13633.

Sessa, P. G., Bogunovic, I., Kamgarpour, M., and Krause, A. (2020). Learning to play sequential games versus unknown opponents. In *Proc. NeurIPS*, pages 8971–8981.

Sessa, P. G., Bogunovic, I., Krause, A., and Kamgarpour, M. (2021). Online submodular resource allocation with applications to rebalancing shared mobility systems. In *Proc. ICML*, pages 9455–9464.

Shoham, Y. and Leyton-Brown, K. (2008). *Multiagent Systems: Algorithmic, Game-theoretic, and Logical Foundations*. Cambridge Univ. Press.

Shu, Y., Cai, S., Dai, Z., Ooi, B. C., and Low, B. K. H. (2022a). NASI: Label- and data-agnostic neural architecture search at initialization. In *Proc. ICLR*.

Shu, Y., Chen, Y., Dai, Z., and Low, B. K. H. (2022b). Neural ensemble search via Bayesian sampling. In *Proc. UAI*, pages 1803–1812.

Shu, Y., Dai, Z., Wu, Z., and Low, B. K. H. (2022c). Unifying and boosting gradient-based training-free neural architecture search. In *Proc. NeurIPS*.

Sim, R. H. L., Zhang, Y., Low, B. K. H., and Jaillet, P. (2021). Collaborative Bayesian optimization with fair regret. In *Proc. ICML*, pages 9691–9701.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proc. ICML*, pages 1015–1022.

Stanton, S., Maddox, W., Gruver, N., Maffettone, P., Delaney, E., Greenside, P., and Wilson, A. G. (2022). Accelerating Bayesian optimization for biological sequence design with denoising autoencoders. In *Proc. ICML*, pages 20459–20478.

Stengel, B. v. (2007). *Equilibrium Computation for Two-Player Games in Strategic and Extensive Form*, page 53–78. Cambridge University Press.

Tay, S. S., Foo, C. S., Daisuke, U., Leong, R., and Low, B. K. H. (2021). Efficient distributionally robust Bayesian optimization with worst-case sensitivity. In *Proc. ICML*, pages 21180–21204.

Thorpe, R. B., Jennings, S., and Dolder, P. J. (2017). Risks and benefits of catching pretty good yield in multispecies mixed fisheries. *ICES Journal of Marine Science*, 74(8):2097–2106.

Verma, A., Dai, Z., and Low, B. K. H. (2022). Bayesian optimization under stochastic delayed feedback. In *Proc. ICML*.

Viqueira, E. A., Cousins, C., and Greenwald, A. (2020). Improved algorithms for learning equilibria in simulation-based games. In *Proc. AAMAS*, pages 79–87.

Viqueira, E. A., Cousins, C., Upfal, E., and Greenwald, A. (2019). Learning equilibria of simulation-based games. arXiv:1905.13379.

Vorobeychik, Y., Wellman, M. P., et al. (2008). Stochastic search methods for Nash equilibrium approximation in simulation-based games. In *Proc. AAMAS*, pages 1055–1062.

Vorobeychik, Y., Wellman, M. P., and Singh, S. (2007). Learning payoff functions in infinite games. *Machine Learning*, 67(1):145–168.

Wang, Z., Balasubramanian, K., Ma, S., and Razaviyayn, M. (2022). Zeroth-order algorithms for nonconvex-strongly-concave minimax problems with improved complexities. *J. Global Optimization*.

Wellman, M. P. (2006). Methods for empirical game-theoretic analysis. In *Proc. AAAI*, pages 1552–1556.

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA.

Xu, N., Low, K. H., Chen, J., Lim, K. K., and Özgül, E. B. (2014). GP-Localize: Persistent mobile robot localization using online sparse Gaussian process observation model. In *Proc. AAAI*, pages 2585–2592.

Yu, H., Chen, Y., Dai, Z., Low, B. K. H., and Jaillet, P. (2019a). Implicit posterior variational inference for deep Gaussian processes. In *Proc. NeurIPS*, pages 14475–14486.

Yu, H., Hoang, T. N., Low, K. H., and Jaillet, P. (2019b). Stochastic variational inference for Bayesian sparse Gaussian process regression. In *Proc. IJCNN*.

Yu, H., Liu, D., Low, K. H., and Jaillet, P. (2021). Convolutional normalizing flows for deep Gaussian processes. In *Proc. IJCNN*.

Zhang, Y., Dai, Z., and Low, B. K. H. (2019). Bayesian optimization with binary auxiliary information. In *Proc. UAI*, pages 1222–1232.

Zhang, Y., Hoang, T. N., Low, B. K. H., and Kankanhalli, M. (2017). Information-based multi-fidelity Bayesian optimization. In *Proc. NIPS Workshop on Bayesian Optimization*.

No-regret Sample-efficient Bayesian Optimization for Finding Nash Equilibria with Unknown Utilities: Supplementary Materials

A PROOFS

A.1 Proof of Lemma 1

Lemma 1 *With probability of at least* $1-\delta$*, for all* $i \in \mathcal{A}$*,* $\mathbf{x} \in \mathbf{X}$*, and* $t \in \mathbb{Z}^+$*,*

$$\check{f}_{i,t-1}(\mathbf{x}) \leqslant f_i(\mathbf{x}) \leqslant \widehat{f}_{i,t-1}(\mathbf{x}) \,,$$
$$\widehat{f}_{i,t-1}(\mathbf{x}) - \check{f}_{i,t-1}(\mathbf{x}) \leqslant 2\beta_t \left(\sigma_{t-1}(\mathbf{x}) + \sigma_{t-1}(\mathbf{x}^i)\right) .$$

Proof From Lemma 9, with probability of at least $1-\delta$, for all $i \in \mathcal{A}$, $\mathbf{x} \in \mathbf{X}$, and $t \in \mathbb{Z}^+$, the following holds:

$$\check{u}_{i,t-1}(\mathbf{x}) \leqslant u_i(\mathbf{x}) \leqslant \widehat{u}_{i,t-1}(\mathbf{x}) \,.$$

For the first statement of the lemma,

$$\begin{aligned}
\check{f}_{i,t-1}(\mathbf{x}) &\coloneqq \check{u}_{i,t-1}(\mathbf{x}) - \max_{\mathbf{x}'_i \in \mathcal{X}_i} (\widehat{u}_{i,t-1}(\mathbf{x}'_i, \mathbf{x}_{-i})) \\
&\leqslant u_{i,t-1}(\mathbf{x}) - \max_{\mathbf{x}'_i \in \mathcal{X}_i} (\widehat{u}_{i,t-1}(\mathbf{x}'_i, \mathbf{x}_{-i})) \\
&\leqslant u_{i,t-1}(\mathbf{x}) - \max_{\mathbf{x}'_i \in \mathcal{X}_i} (u_{i,t-1}(\mathbf{x}'_i, \mathbf{x}_{-i})) = f_{i,t-1}(\mathbf{x}) \\
&\leqslant \widehat{u}_{i,t-1}(\mathbf{x}) - \max_{\mathbf{x}'_i \in \mathcal{X}_i} (u_{i,t-1}(\mathbf{x}'_i, \mathbf{x}_{-i})) \\
&\leqslant \widehat{u}_{i,t-1}(\mathbf{x}) - \max_{\mathbf{x}'_i \in \mathcal{X}_i} (\check{u}_{i,t-1}(\mathbf{x}'_i, \mathbf{x}_{-i})) = \widehat{f}_{i,t-1}(\mathbf{x}).
\end{aligned}$$

For the second statement of the lemma, define

$$\begin{aligned}
\mathbf{q}_i &\coloneqq \operatorname*{argmax}_{\mathbf{x}'_i \in \mathcal{X}_i} (\widehat{u}_{i,t-1}(\mathbf{x}'_i, \mathbf{x}_{-i})) \\
\mathbf{w}_i &\coloneqq \operatorname*{argmax}_{\mathbf{x}'_i \in \mathcal{X}_i} (\check{u}_{i,t-1}(\mathbf{x}'_i, \mathbf{x}_{-i})).
\end{aligned}$$

The quantity $\widehat{f}_{i,t}(\mathbf{x}) - \check{f}_{i,t-1}(\mathbf{x})$ is bounded by

$$\begin{aligned}
\widehat{f}_{i,t}(\mathbf{x}) - \check{f}_{i,t-1}(\mathbf{x}) &= \widehat{u}_{i,t-1}(\mathbf{x}) - \check{u}_{i,t-1}(\mathbf{w}_i, \mathbf{x}_{-i}) - \check{f}_{i,t-1}(\mathbf{x}) \\
&= \widehat{u}_{i,t-1}(\mathbf{x}) - \check{u}_{i,t-1}(\mathbf{x}) + \widehat{u}_{i,t-1}(\mathbf{q}_i, \mathbf{x}_{-i}) - \check{u}_{i,t-1}(\mathbf{w}_i, \mathbf{x}_{-i}) \\
&= 2\beta_t \sigma_{t-1}(\mathbf{x}) + \widehat{u}_{i,t-1}(\mathbf{q}_i, \mathbf{x}_{-i}) - \check{u}_{i,t-1}(\mathbf{w}_i, \mathbf{x}_{-i}) \\
&\overset{(i)}{\leqslant} 2\beta_t \sigma_{t-1}(\mathbf{x}) + \widehat{u}_{i,t-1}(\mathbf{q}_i, \mathbf{x}_{-i}) - \check{u}_{i,t-1}(\mathbf{q}_i, \mathbf{x}_{-i}) \\
&= 2\beta_t \left(\sigma_{t-1}(\mathbf{x}) + \sigma_{t-1}(\mathbf{q}_i, \mathbf{x}_{-i})\right) \\
&= 2\beta_t \left(\sigma_{t-1}(\mathbf{x}) + \sigma_{t-1}(\mathbf{x}^i)\right)
\end{aligned}$$

where (i) follows from the definition of $\mathbf{w}_i$, hence completing the proof. ■

A.2 Proof of Theorem 1

Theorem 1 *With probability of at least $1-\delta$, the sequence of reported strategy profiles $(\mathbf{x}_t)_{t=1}^T$ selected by UCB-PNE (Algo. 1) incurs a pure Nash regret bounded by*

$$R_T \leqslant \mathcal{O}\left(\beta_T\sqrt{T\gamma_T}\right).$$

Proof Recall the pure Nash regret:

$$\begin{aligned} R_T &\coloneqq \sum_{t=1}^{T} -\epsilon_* - \min_{j\in\mathcal{A}} f_j(\mathbf{x}_t) \\ &= \sum_{t=1}^{T} \min_{i\in\mathcal{A}} f_i(\mathbf{x}_{\epsilon_*}) - \min_{j\in\mathcal{A}} f_j(\mathbf{x}_t). \end{aligned}$$

We now derive an upper bound for R_T.

$$\begin{aligned} R_T &= \sum_{t=1}^{T} \min_{i\in\mathcal{A}} f_i(\mathbf{x}_{\epsilon_*}) - \min_{j\in\mathcal{A}} f_j(\mathbf{x}_t) \\ &\overset{(i)}{\leqslant} \sum_{t=1}^{T} \min_{i\in\mathcal{A}} \widehat{f}_{i,t-1}(\mathbf{x}_{\epsilon_*}) - \min_{j\in\mathcal{A}} f_j(\mathbf{x}_t) \\ &\overset{(ii)}{\leqslant} \sum_{t=1}^{T} \min_{i\in\mathcal{A}} \widehat{f}_{i,t-1}(\mathbf{x}_t) - \min_{j\in\mathcal{A}} f_j(\mathbf{x}_t) \\ &\leqslant \sum_{t=1}^{T} \min_{i\in\mathcal{A}} \widehat{f}_{i,t-1}(\mathbf{x}_t) - \min_{j\in\mathcal{A}} \check{f}_{j,t-1}(\mathbf{x}_t) \\ &\leqslant \sum_{t=1}^{T} \widehat{f}_{j_t,t-1}(\mathbf{x}_t) - \check{f}_{j_t,t-1}(\mathbf{x}_t) \\ &\overset{(iii)}{\leqslant} \sum_{t=1}^{T} 2\beta_t \left(\sigma_{t-1}(\mathbf{x}_t) + \sigma_{t-1}(\mathbf{x}_t^{j_t})\right) \\ &\leqslant \sum_{t=1}^{T} 4\beta_t \max\left(\sigma_{t-1}(\mathbf{x}_t), \sigma_{t-1}(\mathbf{x}_t^{j_t})\right) \\ &\leqslant 4\beta_T \left(\sum_{t=1}^{T} \max\left(\sigma_{t-1}(\mathbf{x}_t), \sigma_{t-1}(\mathbf{x}_t^{j_t})\right)\right) \\ &\overset{(iv)}{\leqslant} 4\beta_T\sqrt{4(T+2)\gamma_T} = \mathcal{O}\left(\beta_T\sqrt{T\gamma_T}\right) \end{aligned}$$

where $j_t \coloneqq \operatorname{argmin}_{j\in\mathcal{A}} \check{f}_{j,t-1}(\mathbf{x}_t)$, (i) follows from Lemma 1, (ii) follows from our choice of reported strategy profile $\mathbf{x}_t$, (iii) follows from Lemma 1 again, and (iv) follows from Lemma 10 and the algorithm's choice of pure strategy profiles $\tilde{\mathbf{x}}_t$ sampled at. ■

A.3 Proof of Theorem 2

Theorem 2 *If the following assumptions hold:*

1. *For all $i \in \mathcal{A}$, u_i is continuous and bounded;*
2. *Unique agent-wise maximizers, i.e., $\forall \mathbf{x} \in \mathcal{X}, \forall i \in \mathcal{A}, \exists! \mathbf{x}_i \in \mathcal{X}_i$ such that $u_i(\mathbf{x}_i, \mathbf{x}_{-i}) = \max_{\mathbf{x}_i'\in\mathcal{X}_i} u_i(\mathbf{x}_i', \mathbf{x}_{-i})$;*

3. *There exists a unique NE* $\mathbf{x}_*$*;*

4. $\gamma_T < \mathcal{O}(T)$*,*

then with probability of at least $1-\delta$*, the sequence of sampled strategy profiles* $(\tilde{\mathbf{x}}_t)_{t=1}^T$ *chosen by UCB-PNE (Algo. 1) incurs no regret (in terms of pure Nash regret), i.e.,*

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} -\epsilon_* - \min_{i\in\mathcal{A}} f_i(\tilde{\mathbf{x}}_t) = 0.$$

Proof For ease of exposition, define the following functions:

$$\tau_i(\mathbf{x}) \coloneqq \left(\operatorname*{argmax}_{\mathbf{x}_i'\in\mathcal{X}_i} u_i(\mathbf{x}_i', \mathbf{x}_{-i}), \mathbf{x}_{-i}\right)$$
$$\hat{\tau}_{i,t-1}(\mathbf{x}) \coloneqq \left(\operatorname*{argmax}_{\mathbf{x}_i'\in\mathcal{X}_i} \hat{u}_{i,t-1}(\mathbf{x}_i', \mathbf{x}_{-i}), \mathbf{x}_{-i}\right) = \mathbf{x}^i.$$

Denote $T_{\text{rep}} \subset \mathbb{Z}_0^+$ as the set of all iterations in which the reported strategy profile was sampled at, and $T_{\text{exp}} \subset \mathbb{Z}_0^+$ as the set of all iterations in which the exploring strategy profile was sampled at. The sequence of sampled strategy profiles $(\tilde{\mathbf{x}}_t)_{t=1}^T$ can be decomposed into two disjoint subsequences: the decisions that are chosen as the reported strategy profiles $(\mathbf{x}_t)_{t\in T_{\text{rep}}}$, and the decisions that are chosen as the exploring strategy profiles $(\hat{\tau}_{j_t,t-1}(\mathbf{x}_t))_{t\in T_{\text{exp}}}$. Since all infinite subsequences converge to the same limit if the original sequence converges, by Theorem 1 and Assumption 4, $(\mathbf{x}_t)_{t\in T_{\text{rep}}}$ has the desired no-regret performance guarantee. To prove that the entire sequence incurs no regret, it remains to show that $(\hat{\tau}_{j_t,t-1}(\mathbf{x}_t))_{t\in T_{\text{exp}}}$ incurs no regret (or is sublinear). To do this, it is sufficient to show that (noting that $\epsilon_* = 0$ under our assumptions)

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} -\min_{i\in\mathcal{A}} f_i(\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)) = 0.$$

From Lemma 3 and Assumption 1, for all $i \in \mathcal{A}$, f_i is continuous, and hence so is $\min_{i\in\mathcal{A}} f_i$. Since $\min_{i\in\mathcal{A}} f_i(\mathbf{x}_*) = 0$ where $\mathbf{x}_*$ is the unique NE by Assumption 3, for all $\epsilon_f > 0$ there exists $\delta > 0$ such that

$$\forall \mathbf{x} \in \mathcal{B}(\mathbf{x}_*, \delta),\ \min_{i\in\mathcal{A}} f_i(\mathbf{x}) > -\epsilon_f \tag{11}$$

where $\mathcal{B}(\mathbf{x}_*, \delta)$ is an open ball centered at $\mathbf{x}_*$ with radius δ. To reduce notational clutter, let $\mathcal{B}$ denote $\mathcal{B}(\mathbf{x}_*, \delta)$. Note that $\mathbf{x}_* \in \mathcal{B}$.

Consider the preimage of $\mathcal{B}$ under τ_i:

$$\tau_i^{-1}(\mathcal{B}) \coloneqq \{\mathbf{x} \in \mathcal{X} \mid \tau_i(\mathbf{x}) \in \mathcal{B}\}.$$

Note that $\tau_i(\mathbf{x}_*) = \mathbf{x}_* \in \mathcal{B}$ for all $i \in \mathcal{A}$ by Assumption 2, so $\mathbf{x}_* \in \tau_i^{-1}(\mathcal{B})$. Denote the intersection of these sets as

$$\tau^{-1}(\mathcal{B}) \coloneqq \cap_{i\in\mathcal{A}} \tau_i^{-1}(\mathcal{B})$$

which contains $\mathbf{x}_*$.

From Lemma 4, $\mathcal{X}\backslash\tau^{-1}(\mathcal{B})$ is a compact set. Since $\mathcal{X}\backslash\tau^{-1}(\mathcal{B})$ is compact and $\min_{i\in\mathcal{A}} f_i$ is a continuous function, $\min_{i\in\mathcal{A}} f_i$ achieves a maximum value on $\mathcal{X}\backslash\tau^{-1}(\mathcal{B})$. Since $\mathcal{X}\backslash\tau^{-1}(\mathcal{B})$ does not contain $\mathbf{x}_*$, this maximum value, denoted as ϵ_f', is less than 0. If $\mathcal{X}\backslash\tau^{-1}(\mathcal{B})$ is an empty set, we may simply reduce δ until $\mathcal{X}\backslash\tau^{-1}(\mathcal{B})$ is non-empty, and Equ. 11 will still hold. Concretely,

$$\epsilon_f' \coloneqq \max_{\mathbf{x}\in\mathcal{X}\backslash\tau^{-1}(\mathcal{B})} \min_{i\in\mathcal{A}} f_i(\mathbf{x}) < 0.$$

Hence, we can decompose the average cumulative regret of the exploring strategy profiles:

$$\begin{aligned}
&\lim_{T\to\infty} -\frac{1}{T}\sum_{t=1}^{T} \min_{i\in\mathcal{A}} f_i(\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)) \\
&= -\lim_{T\to\infty} \frac{1}{T}\Big(\sum_{t=1}^{T} \mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})} \min_{i\in\mathcal{A}} f_i(\hat{\tau}_{j_t,t-1}(\mathbf{x})) \\
&\quad + \sum_{i=1}^{T} \mathbb{1}_{\mathbf{x}_t\in(\mathcal{X}\setminus\tau^{-1}(\mathcal{B}))} \min_{i\in\mathcal{A}} f_i(\hat{\tau}_{j_t,t-1}(\mathbf{x}))\Big).
\end{aligned}$$

From Lemma 5 and the fact that f_i is bounded (e.g., by M) since u_i is bounded by Assumption 1,

$$\begin{aligned}
&\lim_{T\to\infty} -\frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{\mathbf{x}_t\in(\mathcal{X}\setminus\tau^{-1}(\mathcal{B}))} \min_{i\in\mathcal{A}} f_i(\hat{\tau}_{j_t,t-1}(\mathbf{x})) \\
&< \lim_{T\to\infty} -\frac{1}{T} M|\mathcal{X}_T \cap \big(\mathcal{X}\setminus\tau^{-1}(\mathcal{B})\big)| = 0
\end{aligned}$$

where $\mathcal{X}_T \coloneqq \{\mathbf{x}_t\}_{t=1}^{T}$ is the set of reported strategy profiles until iteration T. Hence,

$$\begin{aligned}
&\lim_{T\to\infty} -\frac{1}{T}\sum_{t=1}^{T} \min_{i\in\mathcal{A}} f_i(\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)) \\
&= -\lim_{T\to\infty} \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})} \min_{i\in\mathcal{A}} f_i(\hat{\tau}_{j_t,t-1}(\mathbf{x})) \\
&= -\lim_{T\to\infty} \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})}\mathbb{1}_{\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)\in\mathcal{B}} \min_{i\in\mathcal{A}} f_i(\hat{\tau}_{j_t,t-1}(\mathbf{x})) \\
&\quad -\lim_{T\to\infty} \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})}\mathbb{1}_{\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)\in\mathcal{X}\setminus\mathcal{B}} \min_{i\in\mathcal{A}} f_i(\hat{\tau}_{j_t,t-1}(\mathbf{x})).
\end{aligned}$$

Similar to the above argument, from Lemma 6, Assumption 4 and the fact that f_i is bounded,

$$\lim_{T\to\infty} \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})}\mathbb{1}_{\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)\in\mathcal{X}\setminus\mathcal{B}} \min_{i\in\mathcal{A}} f_i(\hat{\tau}_{j_t,t-1}(\mathbf{x})) = 0.$$

Hence,

$$\begin{aligned}
&\lim_{T\to\infty} -\frac{1}{T}\sum_{t=1}^{T} \min_{i\in\mathcal{A}} f_i(\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)) \\
&= -\lim_{T\to\infty} \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})}\mathbb{1}_{\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)\in\mathcal{B}} \min_{i\in\mathcal{A}} f_i(\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)) \\
&\overset{(i)}{\leqslant} \lim_{T\to\infty} \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})}\mathbb{1}_{\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)\in\mathcal{B}}\ \epsilon_f \\
&\leqslant \epsilon_f
\end{aligned}$$

where (i) follows from the definition of $\mathcal{B}$. As the above is true for all $\epsilon_f > 0$, it follows that

$$\lim_{T\to\infty} -\frac{1}{T}\sum_{t=1}^{T} \min_{i\in\mathcal{A}} f_i(\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)) = 0$$

which concludes the proof. ■

Lemma 3 *If u_i is continuous, then τ_i and f_i are continuous.*

Proof We first prove that τ_i is continuous with a proof by contradiction. Assume that τ_i is not continuous. This means that at some decision $\mathbf{x}^\circ$,

$$\exists\, \epsilon_0 > 0 \text{ s.t. } \forall\, \delta > 0, \exists\, \mathbf{x}' \text{ s.t. } (\|\mathbf{x}^\circ - \mathbf{x}'\| < \delta) \wedge (\tau_i(\mathbf{x}') \notin \mathcal{B}_{\epsilon_0}(\tau_i(\mathbf{x}^\circ))). \tag{12}$$

where $\mathcal{B}_{\epsilon_0}(\tau_i(\mathbf{x}^\circ))$ is an open ball centered at $\tau_i(\mathbf{x}^\circ)$ with radius ϵ_0 and $\|\cdot\|$ is an arbitrary norm on $\mathbb{R}^d$.

Let $\bar{\mathcal{X}}_i(\mathbf{x}^\circ)$ denote the following affine set that contains $\mathbf{x}^\circ$:

$$\bar{\mathcal{X}}_i(\mathbf{x}^\circ) \coloneqq \{(\mathbf{x}_i, \mathbf{x}^\circ_{-i}) |\ \mathbf{x}_i \in \mathcal{X}_i\}.$$

Note that $\tau_i(\mathbf{x}^\circ) \in \bar{\mathcal{X}}_i(\mathbf{x}^\circ)$.

Since there is a unique agent-wise maximizer $\tau_i(\mathbf{x}^\circ)$ by Assumption 2, for any ϵ_0 above

$$\omega \coloneqq u_i(\tau_i(\mathbf{x}^\circ)) - \max_{\mathbf{x} \in \bar{\mathcal{X}}_i(\mathbf{x}^\circ) \setminus \mathcal{B}_{\epsilon_0}(\tau_i(\mathbf{x}^\circ))} u_i(\mathbf{x}) > 0. \tag{13}$$

Construct a sequence of distances from $\mathbf{x}^\circ$ $(\delta^{(j)})_{j=1}^\infty$ such that $\lim_{j\to\infty} \delta^{(j)} = 0$. For each $\delta^{(j)}$, by Equ. 12, there exists a $\mathbf{x}^{(j)}$ such that

$$(\|\mathbf{x}^\circ - \mathbf{x}^{(j)}\| < \delta^{(j)}) \wedge (\tau_i(\mathbf{x}^{(j)}) \notin \mathcal{B}_{\epsilon_0}(\tau_i(\mathbf{x}^\circ))).$$

Construct such a sequence $(\mathbf{x}^{(j)})_{j=1}^\infty$. This gives us the associated sequence $(\tau_i(\mathbf{x}^{(j)}))_{j=1}^\infty$. By construction, all the elements of this sequence lie outside the ϵ_0 open ball around $\tau_i(\mathbf{x}^\circ)$. Furthermore, by letting $\rho(\mathbf{x}) \coloneqq (\mathbf{x}_i, \mathbf{x}^\circ_{-i})$ be the projection of $\mathbf{x}$ onto $\bar{\mathcal{X}}_i(\mathbf{x}^\circ)$ and noting that τ_i does not change the strategies of players other than i and that $\mathbf{x}^{(j)} \to \mathbf{x}$, we claim (to be proven later)

$$\lim_{j\to\infty} \|\tau_i(\mathbf{x}^{(j)}) - \rho(\tau_i(\mathbf{x}^{(j)}))\| = 0. \tag{14}$$

Using the triangle inequality,

$$\|\tau_i(\mathbf{x}^\circ) - \rho(\tau_i(\mathbf{x}^{(j)}))\| + \|\tau_i(\mathbf{x}^{(j)}) - \rho(\tau_i(\mathbf{x}^{(j)}))\| \geqslant \|\tau_i(\mathbf{x}^\circ) - \tau_i(\mathbf{x}^{(j)})\| \geqslant \epsilon_0.$$

Hence,

$$\lim_{j\to\infty} \left\|\tau_i(\mathbf{x}^\circ) - \rho(\tau_i(\mathbf{x}^{(j)}))\right\| \geqslant \epsilon_0\ . \tag{15}$$

As u_i is a continuous function on a compact set $\mathcal{X}$, u_i is uniformly continuous on $\mathcal{X}$. Hence, given $\omega/3$, there exists $\epsilon_2 < \epsilon_0$ such that

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}\ \ \|\mathbf{x} - \mathbf{x}'\| < \epsilon_2 \Rightarrow |u_i(\mathbf{x}) - u_i(\mathbf{x}')| < \omega/3\ . \tag{16}$$

From Equ. 14 and Equ. 15, choose a value j' such that

$$\|\tau_i(\mathbf{x}^\circ) - \rho(\tau_i(\mathbf{x}^{(j')}))\| > \epsilon_0 - \epsilon_2 \tag{17}$$

$$\|\tau_i(\mathbf{x}^{(j')}) - \rho(\tau_i(\mathbf{x}^{(j')}))\| < \epsilon_2. \tag{18}$$

This implies that

$$\begin{aligned} u_i(\tau_i(\mathbf{x}^\circ)) - u_i\Big(\rho(\tau_i(\mathbf{x}^{(j')}))\Big) &\geqslant \frac{2}{3}\omega \\ \left|u_i(\tau_i(\mathbf{x}^{(j')})) - u_i\Big(\rho(\tau_i(\mathbf{x}^{(j')}))\Big)\right| &< \frac{1}{3}\omega. \end{aligned} \tag{19}$$

To see why Equ. 19 is true, let $\mathbf{v}$ be the closest point to $\rho(\tau_i(\mathbf{x}^{(j')}))$ that is not in $\mathcal{B}_{\epsilon_0}(\tau_i(\mathbf{x}^\circ))$. From Equ. 13, $u_i(\tau_i(\mathbf{x}^\circ)) - u_i(\mathbf{v}) \geqslant \omega$. For large enough j', from (17), $\|\rho(\tau_i(\mathbf{x}^{(j')})) - \mathbf{v}\| < \epsilon_2$. Equ. 19 then follows from applying Equ. 16.

For all $\mathbf{x}_1 \in \mathcal{B}_{\epsilon_2}(\tau_i(\mathbf{x}^\circ))$, from Equ. 16,

$$|u_i(\tau_i(\mathbf{x}^\circ)) - u_i(\mathbf{x}_1)| < \frac{1}{3}\omega.$$

Hence, for all $\mathbf{x}_1 \in \mathcal{B}_{\epsilon_2}(\tau_i(\mathbf{x}^\circ))$,

$$\begin{aligned}
&u_i(\mathbf{x}_1) - u_i(\tau_i(\mathbf{x}^{(j')})) \\
&= u_i(\mathbf{x}_1) - u_i(\tau_i(\mathbf{x}^\circ)) + u_i(\tau_i(\mathbf{x}^\circ)) - u_i\Big(\rho(\tau_i(\mathbf{x}^{(j')}))\Big) \\
&\quad + u_i\Big(\rho(\tau_i(\mathbf{x}^{(j')}))\Big) - u_i(\tau_i(\mathbf{x}^{(j')})) + u_i(\tau_i(\mathbf{x}^{(j')})) - u_i(\tau_i(\mathbf{x}^{(j')})) \\
&> -\frac{1}{3}\omega + \frac{2}{3}\omega - \frac{1}{3}\omega \\
&= 0.
\end{aligned}$$

Since the intersection $\bar{\mathcal{X}}_i(\mathbf{x}^{(j')}) \cap \mathcal{B}_{\epsilon_2}(\tau_i(\mathbf{x}^\circ))$ is non-empty due to Equ. 18 and $\tau_i(\mathbf{x}^{(j')}) \notin \mathcal{B}_{\epsilon_2}(\tau_i(\mathbf{x}^\circ))$ since $\tau_i(\mathbf{x}^{(j')})$ is outside the ϵ_0 open ball around $\tau_i(\mathbf{x}^\circ)$ and $\epsilon_0 > \epsilon_2$, there exists $\mathbf{x}' \in \bar{\mathcal{X}}_i(\mathbf{x}^{(j')}) \cap \mathcal{B}_{\epsilon_2}(\tau_i(\mathbf{x}^\circ))$ such that $u_i(\mathbf{x}') > u_i(\tau_i(\mathbf{x}^{(j')}))$ which is a contradiction since $\tau_i(\mathbf{x}^{(j')})$ is the unique maximizer of u_i in $\bar{\mathcal{X}}_i(\mathbf{x}^{(j')})$. We thus conclude that τ_i is continuous. Since $f_i(\mathbf{x}) = u_i(\mathbf{x}) - u_i(\tau_i(\mathbf{x}))$ and both u_i and τ_i are continuous, f_i is continuous.

What remains is to prove the claim Equ. 14

$$\lim_{j\to\infty} \|\tau_i(\mathbf{x}^{(j)}) - \rho(\tau_i(\mathbf{x}^{(j)}))\| = 0.$$

$$\begin{aligned}
\|\tau_i(\mathbf{x}^{(j)}) - \rho(\tau_i(\mathbf{x}^{(j)}))\|_2 &= \Big\|\Big(\mathbf{0}_i, (\mathbf{x}_{-i}^{(j)} - \mathbf{x}_{-i}^\circ)\Big)\Big\|_2 \\
&\leqslant \Big\|\Big((\mathbf{x}_i^{(j)} - \mathbf{x}_i^\circ), (\mathbf{x}_{-i}^{(j)} - \mathbf{x}_{-i}^\circ)\Big)\Big\|_2 \\
&= \|\mathbf{x}^{(j)} - \mathbf{x}^\circ\|_2.
\end{aligned}$$

By our construction of $(\mathbf{x}^{(j)})_{j=1}^\infty$, $\lim_{j\to\infty} \|\mathbf{x}^{(j)} - \mathbf{x}^\circ\| = 0$. Since any two norms in $\mathbb{R}^d$ are equivalent, $\lim_{j\to\infty} \|\mathbf{x}^{(j)} - \mathbf{x}^\circ\|_2 = 0$. Since $\|\mathbf{x}^{(j)} - \mathbf{x}^\circ\|_2 \geqslant \|\tau_i(\mathbf{x}^{(j)}) - \rho(\tau_i(\mathbf{x}^{(j)}))\|_2$ for all j, it follows that $\lim_{j\to\infty} \|\tau_i(\mathbf{x}^{(j)}) - \rho(\tau_i(\mathbf{x}^{(j)}))\|_2 = 0$. By norm equivalence again, $\lim_{j\to\infty} \|\tau_i(\mathbf{x}^{(j)}) - \rho(\tau_i(\mathbf{x}^{(j)}))\| = 0$ which completes the proof of the claim and concludes the proof of the lemma. ■

Lemma 4 *The set $\mathcal{X}\backslash\tau^{-1}(\mathcal{B})$ is compact.*

Proof Since

$$\mathcal{X}\backslash\tau^{-1}(\mathcal{B}) = \bigcup_{i\in\mathcal{A}} \mathcal{X}\backslash\tau_i^{-1}(\mathcal{B}),$$

to prove that $\mathcal{X}\backslash\tau^{-1}(\mathcal{B})$ is compact, we only need to show that $\mathcal{X}\backslash\tau_i^{-1}(\mathcal{B})$ is compact. Define

$$\tilde{\tau}_i^{-1}(\mathcal{B}) \coloneqq \{\mathbf{x} \in \mathbb{R}^d \mid \tau_i(\mathbf{x}) \in \mathcal{B}\}$$

where $\mathbb{R}^d \supset \mathcal{X}$. Since τ_i is a continuous function and $\mathcal{B}$ is an open set, the preimage $\tilde{\tau}_i^{-1}(\mathcal{B})$ of $\mathcal{B}$ under τ_i in $\mathbb{R}^d$ is an open set, i.e., its complement $(\tilde{\tau}_i^{-1}(\mathcal{B}))^c$ is a closed set. Since each $\mathcal{X}_i$ is compact, $\mathcal{X}$ is compact, and so $\mathcal{X}\backslash\tilde{\tau}_i^{-1}(\mathcal{B}) = \mathcal{X} \cap (\tilde{\tau}_i^{-1}(\mathcal{B}))^c$ is a compact set. Furthermore, $\mathcal{X}\backslash\tilde{\tau}_i^{-1}(\mathcal{B}) = \mathcal{X}\backslash\tau_i^{-1}(\mathcal{B})$. Hence, $\mathcal{X}\backslash\tau_i^{-1}(\mathcal{B})$ is a compact set. ■

Lemma 5 *The number of reported strategy profiles that fall outside $\tau^{-1}(\mathcal{B})$ is sublinear. Let $\mathcal{X}_T \coloneqq \{\mathbf{x}_t\}_{t=1}^T$ be the set of reported strategy profiles until iteration T. Then,*

$$\lim_{T\to\infty} \frac{1}{T} |\mathcal{X}_T \cap \big(\mathcal{X}\backslash\tau^{-1}(\mathcal{B})\big)| = 0.$$

Proof We can rewrite R_T as

$$\begin{aligned}
R_T &:= -\sum_{t=1}^{T} \min_{i\in\mathcal{A}} f_i(\mathbf{x}_t) \\
&= -\sum_{\mathbf{x}\in\mathcal{X}_T\cap\tau^{-1}(\mathcal{B})} \min_{i\in\mathcal{A}} f_i(\mathbf{x}) - \sum_{\mathbf{x}\in\mathcal{X}_T\cap(\mathcal{X}\setminus\tau^{-1}(\mathcal{B}))} \min_{i\in\mathcal{A}} f_i(\mathbf{x}) \\
&\geqslant -\sum_{\mathbf{x}\in\mathcal{X}_T\cap\tau^{-1}(\mathcal{B})} \min_{i\in\mathcal{A}} f_i(\mathbf{x}) - \epsilon'_f|\mathcal{X}_T\cap\left(\mathcal{X}\setminus\tau^{-1}(\mathcal{B})\right)| \\
&\geqslant \epsilon'_f|\mathcal{X}_T\cap\left(\mathcal{X}\setminus\tau^{-1}(\mathcal{B})\right)|.
\end{aligned}$$

Since R_T is sublinear,

$$\begin{aligned}
\lim_{T\to\infty}\frac{R_T}{T} &= 0 \\
\lim_{T\to\infty}\frac{1}{T}\epsilon'_f|\mathcal{X}_T\cap\left(\mathcal{X}\setminus\tau^{-1}(\mathcal{B})\right)| &= 0 \\
\lim_{T\to\infty}\frac{1}{T}|\mathcal{X}_T\cap\left(\mathcal{X}\setminus\tau^{-1}(\mathcal{B})\right)| &= 0.
\end{aligned}$$

■

Lemma 6 *If $\gamma_T < \mathcal{O}(T)$, then*

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})}\mathbb{1}_{\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)\in\mathcal{X}\setminus\mathcal{B}} = 0.$$

Proof Let

$$\epsilon_u := \min_{\mathbf{x}\in\tau^{-1}(\mathcal{B})}\min_{i\in\mathcal{A}}\left(u_i(\tau_i(\mathbf{x})) - \max_{\mathbf{x}'_i\in\{\mathbf{x}'_i\in\mathcal{X}_i|(\mathbf{x}'_i,\mathbf{x}_{-i})\notin\mathcal{B}\}} u_i((\mathbf{x}'_i,\mathbf{x}_{-i}))\right).$$

Since $\tau_i(\mathbf{x})$ is unique by Assumption 2 and $\mathbf{x}_t\in\tau^{-1}(\mathcal{B})$ implies that $\tau_i(\mathbf{x}_t)\in\mathcal{B}$ for all $i\in\mathcal{A}$, $\epsilon_u > 0$.

Define

$$r_u(\mathbf{x}_t) := u_{j_t}(\tau_{j_t}(\mathbf{x}_t)) - u_{j_t}(\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)).$$

Hence, if $\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)\in\mathcal{X}\setminus\mathcal{B}$,

$$r_u(\mathbf{x}_t)\geqslant\epsilon_u.$$

$$\begin{aligned}
\sum_{t=1}^{T}\mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})}r_u(\mathbf{x}_t) &= \sum_{t=1}^{T}\mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})}\mathbb{1}_{\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)\in\mathcal{B}}\,r_u(\mathbf{x}_t) \\
&\quad+\sum_{t=1}^{T}\mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})}\mathbb{1}_{\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)\in\mathcal{X}\setminus\mathcal{B}}\,r_u(\mathbf{x}_t) \\
&\geqslant \epsilon_u\sum_{t=1}^{T}\mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})}\mathbb{1}_{\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)\in\mathcal{X}\setminus\mathcal{B}}.
\end{aligned}$$

From Lemma 7, since $\gamma_T < \mathcal{O}(T)$, $\sum_{t=1}^{T} r_u(\mathbf{x}_t)$ is sublinear, i.e., $\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_u(\mathbf{x}_t) = 0$. Hence,

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})} r_u(\mathbf{x}_t) \leqslant \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_u(\mathbf{x}_t) = 0$$

$$\lim_{T\to\infty} \frac{1}{T}\epsilon_u \sum_{t=1}^{T} \mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})} \mathbb{1}_{\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)\in\mathcal{X}\setminus\mathcal{B}} \leqslant \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})} r_u(\mathbf{x}_t) = 0$$

$$\lim_{T\to\infty} \frac{1}{T}\epsilon_u \sum_{t=1}^{T} \mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})} \mathbb{1}_{\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)\in\mathcal{X}\setminus\mathcal{B}} = 0$$

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{\mathbf{x}_t\in\tau^{-1}(\mathcal{B})} \mathbb{1}_{\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)\in\mathcal{X}\setminus\mathcal{B}} = 0.$$

■

Lemma 7 *If $\gamma_T < \mathcal{O}(T)$, then $\sum_{t=1}^{T} r_u(\mathbf{x}_t)$ is sublinear, i.e.,*

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} r_u(\mathbf{x}_t) = 0.$$

Proof The proof is similar to GP-UCB's proof (Srinivas et al., 2010):

$$\begin{aligned}
\sum_{t=1}^{T} r_u(\mathbf{x}_t) &= \sum_{t=1}^{T} u_{j_t}(\tau_{j_t}(\mathbf{x}_t)) - u_{j_t}(\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)) \\
&\leqslant \sum_{t=1}^{T} \widehat{u}_{j_t,t-1}(\tau_{j_t}(\mathbf{x}_t)) - \breve{u}_{j_t,t-1}(\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)) \\
&\leqslant \sum_{t=1}^{T} \widehat{u}_{j_t,t-1}(\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)) - \breve{u}_{j_t,t-1}(\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)) \\
&= \sum_{t=1}^{T} 2\beta_t \sigma_{t-1}(\hat{\tau}_{j_t,t-1}(\mathbf{x}_t)) \\
&\leqslant \sum_{t=1}^{T} 2\beta_t \sigma_{t-1}(\tilde{\mathbf{x}}_t) \\
&\leqslant 2\beta_T\sqrt{4(T+2)\gamma_T} = \mathcal{O}(\beta_T\sqrt{T\gamma_T})
\end{aligned}$$

where the last inequality follows from Lemma 10. Since $\gamma_T < \mathcal{O}(T)$, $\sum_{t=1}^{T} r_u(\mathbf{x}_t)$ is sublinear. ■

A.4 Proof of Lemma 2

Lemma 2 *With probability of at least $1-\delta$, for all $i \in \mathcal{A}$, $\mathbf{x}^{\triangle} \in \mathbf{X}^{\triangle}$, and $t \in \mathbb{Z}^{+}$,*

$$\breve{f}_{i,t-1}(\mathbf{x}^{\triangle}) \leqslant f_i(\mathbf{x}^{\triangle}) \leqslant \widehat{f}_{i,t-1}(\mathbf{x}^{\triangle})$$

and

$$\widehat{f}_{i,t-1}(\mathbf{x}^{\triangle}) - \breve{f}_{i,t-1}(\mathbf{x}^{\triangle}) \leqslant 2\beta_t \left(\sigma_{t-1}(\bar{\mathbf{x}}) + \sigma_{t-1}(\mathbf{x}^i)\right).$$

Proof From Lemma 9, with probability of at least $1-\delta$, for all $i \in \mathcal{A}$, pure strategies $\mathbf{x} \in \mathbf{X}$, and $t \in \mathbb{Z}^{+}$, the following holds:

$$\breve{u}_{i,t-1}(\mathbf{x}) \leqslant u_i(\mathbf{x}) \leqslant \widehat{u}_{i,t-1}(\mathbf{x})$$

To extend these inequalities to any mixed strategy profile $\mathbf{x}^\triangle \in \mathbf{X}^\triangle$, we write the expected utility and its upper and lower confidence bounds in terms of vector inner products:

$$\begin{aligned}\breve{u}_{i,t-1}(\mathbf{x}^\triangle) &= \mathbf{p}(\mathbf{x}^\triangle)^\top \breve{\mathbf{u}}_{i,t-1} \\ &\leqslant \mathbf{p}(\mathbf{x}^\triangle)^\top \mathbf{u}_i = u_{i,t-1}(\mathbf{x}^\triangle) \\ &\leqslant \mathbf{p}(\mathbf{x}^\triangle)^\top \widehat{\mathbf{u}}_{i,t-1} = \widehat{u}_{i,t-1}(\mathbf{x}^\triangle)\end{aligned}$$

For the first statement of the lemma, using these inequalities,

$$\begin{aligned}\breve{f}_{i,t-1}(\mathbf{x}^\triangle) &:= \breve{u}_{i,t-1}(\mathbf{x}^\triangle) - \max_{\mathbf{x}_i \in \mathcal{X}_i} \widehat{u}_{i,t-1}(\mathbf{x}_i, \mathbf{x}^\triangle_{-i}) \\ &\leqslant u_i(\mathbf{x}^\triangle) - \max_{\mathbf{x}_i \in \mathcal{X}_i} \widehat{u}_{i,t-1}(\mathbf{x}_i, \mathbf{x}^\triangle_{-i}) \\ &\leqslant u_i(\mathbf{x}^\triangle) - \max_{\mathbf{x}_i \in \mathcal{X}_i} u_i(\mathbf{x}_i, \mathbf{x}^\triangle_{-i}) = f_i(\mathbf{x}^\triangle) \\ &\leqslant \widehat{u}_{i,t-1}(\mathbf{x}^\triangle) - \max_{\mathbf{x}_i \in \mathcal{X}_i} u_i(\mathbf{x}_i, \mathbf{x}^\triangle_{-i}) \\ &\leqslant \widehat{u}_{i,t-1}(\mathbf{x}^\triangle) - \max_{\mathbf{x}_i \in \mathcal{X}_i} \breve{u}_{i,t-1}(\mathbf{x}_i, \mathbf{x}^\triangle_{-i}) = \widehat{f}_{i,t-1}(\mathbf{x}^\triangle).\end{aligned}$$

For the second statement of the lemma, we first bound the difference in the upper and lower confidence bounds of the expected utility of any mixed strategy profile $\mathbf{x}^\triangle$:

$$\begin{aligned}\widehat{u}_{i,t-1}(\mathbf{x}^\triangle) - \breve{u}_{i,t-1}(\mathbf{x}^\triangle) &= \mathbf{p}(\mathbf{x}^\triangle)^\top \widehat{\mathbf{u}}_{i,t-1} - \mathbf{p}(\mathbf{x}^\triangle)^\top \breve{\mathbf{u}}_{i,t-1} \\ &= \mathbf{p}(\mathbf{x}^\triangle)^\top (\widehat{\mathbf{u}}_{i,t-1} - \breve{\mathbf{u}}_{i,t-1}) \\ &= \mathbf{p}(\mathbf{x}^\triangle)^\top (2\beta_t \boldsymbol{\sigma}_{t-1}) \\ &\leqslant 2\beta_t \max_{\mathbf{x} \in \text{supp}(\mathbf{x}^\triangle)} \sigma_{t-1}(\mathbf{x}) \qquad (20)\end{aligned}$$

where $\boldsymbol{\sigma}_{t-1} \in \mathbb{R}^{\prod_{i=1}^n m_i}$ is a vector consisting of the GP posterior standard deviation of each pure strategy profile in the domain, i.e., $\sigma_{t-1}(\mathbf{x})$ for each of the m pure strategy profiles $\mathbf{x} \in \breve{\mathcal{X}}$.

$$\begin{aligned}\widehat{f}_{i,t-1}(\mathbf{x}^\triangle) - \breve{f}_{i,t-1}(\mathbf{x}^\triangle) &= \widehat{u}_{i,t-1}(\mathbf{x}^\triangle) - \breve{u}_{i,t-1}(\mathbf{x}^\triangle) + \max_{\mathbf{x}_i \in \mathcal{X}_i} \widehat{u}_{i,t-1}(\mathbf{x}_i, \mathbf{x}^\triangle_{-i}) - \max_{\mathbf{x}_i \in \mathcal{X}_i} \breve{u}_{i,t-1}(\mathbf{x}_i, \mathbf{x}^\triangle_{-i}) \\ &\leqslant \widehat{u}_{i,t-1}(\mathbf{x}^\triangle) - \breve{u}_{i,t-1}(\mathbf{x}^\triangle) + \widehat{u}_{i,t-1}(\mathbf{q}_i, \mathbf{x}^\triangle_{-i}) - \breve{u}_{i,t-1}(\mathbf{q}_i, \mathbf{x}^\triangle_{-i}) \\ &\overset{(i)}{\leqslant} 2\beta_t \max_{\mathbf{x} \in \text{supp}(\mathbf{x}^\triangle)} \sigma_{t-1}(\mathbf{x}) + 2\beta_t \max_{\mathbf{x}' \in \text{supp}(\mathbf{q}_i, \mathbf{x}^\triangle_{-i})} \sigma_{t-1}(\mathbf{x}') \\ &= 2\beta_t \left(\max_{\mathbf{x} \in \text{supp}(\mathbf{x}^\triangle)} \sigma_{t-1}(\mathbf{x}) + \max_{\mathbf{x}' \in \text{supp}(\mathbf{q}_i, \mathbf{x}^\triangle_{-i})} \sigma_{t-1}(\mathbf{x}') \right) \\ &= 2\beta_t \left(\sigma_{t-1}(\bar{\mathbf{x}}) + \sigma_{t-1}(\mathbf{x}^i) \right)\end{aligned}$$

where $\mathbf{q}_i$, $\bar{\mathbf{x}}$ and $\mathbf{x}^i$ are defined in Equ. 9, Equ. 10 and Equ. 8 respectively, and (i) follows from Equ. 20. ■

A.5 Proof of Theorem 3

Theorem 3 *With probability of at least $1-\delta$, the sequence of reported strategies $(\mathbf{x}_t^\triangle)_{t=1}^T$ selected by UCB-MNE (Algo. 2) incurs a mixed Nash regret bounded by*

$$R_T^\triangle \leqslant \mathcal{O}\left(\beta_T \sqrt{T \gamma_T}\right).$$

Proof The mixed Nash regret is bounded by

$$\begin{aligned}
R_T^{\triangle} &\coloneqq \sum_{t=1}^{T} -\min_{i\in\mathcal{A}} f_i(\mathbf{x}_t^{\triangle}) \\
&\overset{(i)}{\leqslant} \sum_{t=1}^{T} -\min_{i\in\mathcal{A}} \breve{f}_{i,t-1}(\mathbf{x}_t^{\triangle}) \\
&= \sum_{t=1}^{T} -\breve{f}_{j_t,t-1}(\mathbf{x}_t^{\triangle}) \\
&\overset{(ii)}{=} \sum_{t=1}^{T} \tilde{u}_{j_t,t-1}(\mathbf{x}_t^{\triangle}) - \max_{\mathbf{x}_{j_t}\in\mathcal{X}_{j_t}} \tilde{u}_{j_t,t-1}(\mathbf{x}_{j_t}, \mathbf{x}_{-j_t}^{\triangle}) - \breve{f}_{j_t,t-1}(\mathbf{x}_t^{\triangle}) \\
&\leqslant \sum_{t=1}^{T} \widehat{u}_{j_t,t-1}(\mathbf{x}_t^{\triangle}) - \max_{\mathbf{x}_{j_t}\in\mathcal{X}_{j_t}} \tilde{u}_{j_t,t-1}(\mathbf{x}_{j_t}, \mathbf{x}_{-j_t}^{\triangle}) - \breve{f}_{j_t,t-1}(\mathbf{x}_t^{\triangle}) \\
&\leqslant \sum_{t=1}^{T} \widehat{u}_{j_t,t-1}(\mathbf{x}_t^{\triangle}) - \max_{\mathbf{x}_{j_t}\in\mathcal{X}_{j_t}} \breve{u}_{j_t,t-1}(\mathbf{x}_{j_t}, \mathbf{x}_{-j_t}^{\triangle}) - \breve{f}_{j_t,t-1}(\mathbf{x}_t^{\triangle}) \\
&= \sum_{t=1}^{T} \widehat{f}_{j_t,t-1}(\mathbf{x}_t^{\triangle}) - \breve{f}_{j_t,t-1}(\mathbf{x}_t^{\triangle}) \\
&\overset{(iii)}{\leqslant} \sum_{t=1}^{T} 2\beta_t \left(\sigma_{t-1}(\bar{\mathbf{x}}) + \sigma_{t-1}(\mathbf{x}^i)\right) \\
&\leqslant \sum_{t=1}^{T} 4\beta_t \left(\max_{\mathbf{x}\in\{\bar{\mathbf{x}}_t, \mathbf{x}_t^{j_t}\}} \sigma_{t-1}(\mathbf{x})\right) \\
&\overset{(iv)}{\leqslant} 4\beta_T\sqrt{4(T+2)\gamma_T} = \mathcal{O}\left(\beta_T\sqrt{T\gamma_T}\right)
\end{aligned}$$

where $j_t \coloneqq \operatorname{argmin}_{i\in\mathcal{A}} \breve{f}_{i,t-1}(\mathbf{x}_t^{\triangle})$ and $\tilde{u}_{j_t,t-1}$ is a random function such that $\breve{u}_{j_t,t-1}(\mathbf{x}) \leqslant \tilde{u}_{j_t,t-1}(\mathbf{x}) \leqslant \widehat{u}_{j_t,t-1}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$. (i) follows from Lemma 2, (ii) follows as the reported mixed strategy profile $\mathbf{x}_t^{\triangle}$ was chosen as a computed mixed NE based on a sampled $\tilde{u}_{j_t,t-1}$, hence $\tilde{u}_{j_t,t-1}(\mathbf{x}_t^{\triangle}) - \max_{\mathbf{x}_{j_t}\in\mathcal{X}_{j_t}} \tilde{u}_{j_t,t-1}(\mathbf{x}_{j_t}, \mathbf{x}_{-j_t}^{\triangle}) = 0$, (iii) follows from Lemma 2 again, and (iv) follows from Lemma 10 and the algorithm's choice of pure strategy profiles $\tilde{\mathbf{x}}_t$ sampled at. ■

A.6 Proof of Proposition 1

Proposition 1 *Let $k^{\triangle}$ be a positive semidefinite kernel defined as $k^{\triangle}(\mathbf{x}_j^{\triangle}, \mathbf{x}_{j'}^{\triangle}) \coloneqq \mathbf{p}(\mathbf{x}_j^{\triangle})^{\top}\tilde{\mathbf{K}}\mathbf{p}(\mathbf{x}_{j'}^{\triangle})$. Then, a mixed-space GP model with prior mean $\mathbf{0}$ and kernel $k^{\triangle}$ gives the predictive mean of any $u_i(\mathbf{x}^{\triangle})$ and covariance between any $u_i(\mathbf{x}_j^{\triangle})$ and $u_i(\mathbf{x}_{j'}^{\triangle})$ to be equal to those obtained with a linear transformation of the pure-space GP predictive distribution over $\mathbf{u}_i$ with prior mean $\mathbf{0}$ and kernel k.*

Proof We first verify that $k^{\triangle}$ is a positive semidefinite kernel. Since k is a positive semidefinite kernel, $\tilde{\mathbf{K}}$ is a positive semidefinite symmetric matrix and has the Cholesky decomposition $\tilde{\mathbf{K}} = \mathbf{L}\mathbf{L}^{\top}$. $k^{\triangle}(\mathbf{x}_j^{\triangle}, \mathbf{x}_{j'}^{\triangle})$ can then be written as

$$\begin{aligned}
k^{\triangle}(\mathbf{x}_j^{\triangle}, \mathbf{x}_{j'}^{\triangle}) &\coloneqq \mathbf{p}(\mathbf{x}_j^{\triangle})^{\top}\tilde{\mathbf{K}}\mathbf{p}(\mathbf{x}_{j'}^{\triangle}) \\
&= \mathbf{p}(\mathbf{x}_j^{\triangle})^{\top}\mathbf{L}\mathbf{L}^{\top}\mathbf{p}(\mathbf{x}_{j'}^{\triangle}) \\
&= (\mathbf{L}^{\top}\mathbf{p}(\mathbf{x}_j^{\triangle}))^{\top}(\mathbf{L}^{\top}\mathbf{p}(\mathbf{x}_{j'}^{\triangle})) \\
&= \langle \phi(\mathbf{x}_j^{\triangle}), \phi(\mathbf{x}_{j'}^{\triangle}) \rangle
\end{aligned}$$

where $\phi(\cdot) \coloneqq \mathbf{L}^{\top}\mathbf{p}(\cdot)$. $k^{\triangle}(\mathbf{x}_j^{\triangle}, \mathbf{x}_{j'}^{\triangle})$ is thus a positive semidefinite kernel since it can be written as the inner product of $\phi(\mathbf{x}_j^{\triangle})$ and $\phi(\mathbf{x}_{j'}^{\triangle})$ for any $\mathbf{x}_j^{\triangle}$ and $\mathbf{x}_{j'}^{\triangle}$ and some map ϕ (Schölkopf and Smola, 2002, pp. 34).

This proof will rely throughout on the following fact: If $\mathbf{u}_i$ is distributed according to $\mathcal{N}(\mathbf{m}, \mathbf{S})$ for some mean vector $\mathbf{m}$ and covariance matrix $\mathbf{S}$, since $u_i(\mathbf{x}^\triangle) = \mathbf{p}(\mathbf{x}^\triangle)^\top \mathbf{u}_i$, the expected utility $u_i(\mathbf{x}^\triangle)$ will be distributed according to $\mathcal{N}(\mathbf{p}(\mathbf{x}^\triangle)^\top \mathbf{m}, \mathbf{p}(\mathbf{x}^\triangle)^\top \mathbf{S} \mathbf{p}(\mathbf{x}^\triangle))$ since it is a linear transformation of a Gaussian.

We next show that for any agent i, the mixed-space GP prior mean of any $u_i(\mathbf{x}^\triangle)$ and covariance between any $u_i(\mathbf{x}_j^\triangle)$ and $u_i(\mathbf{x}_{j'}^\triangle)$ with $k^\triangle$ is the same as that obtained through a linear transformation of the pure-space GP prior mean and covariance on $\mathbf{u}_i$ with k.

The mixed-space GP prior mean function on any $u_i(\mathbf{x}_j^\triangle)$ is 0 by assumption. Since the pure-space GP prior mean function on $\mathbf{u}_i$ is also $\mathbf{0}$ by assumption, trivially, $0 = \mathbf{p}(\mathbf{x}^\triangle)^\top \mathbf{0}$ and the prior means are equal as desired. For the prior covariance between any $u_i(\mathbf{x}_j^\triangle)$ and $u_i(\mathbf{x}_{j'}^\triangle)$, observe that

$$\begin{aligned}\begin{bmatrix} u_i(\mathbf{x}_j^\triangle) \\ u_i(\mathbf{x}_{j'}^\triangle) \end{bmatrix} &= \begin{bmatrix} \mathbf{p}(\mathbf{x}_j^\triangle) & \mathbf{p}(\mathbf{x}_{j'}^\triangle) \end{bmatrix}^\top \mathbf{u}_i \\ &\sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{p}(\mathbf{x}_j^\triangle) & \mathbf{p}(\mathbf{x}_{j'}^\triangle) \end{bmatrix}^\top \tilde{\mathbf{K}} \begin{bmatrix} \mathbf{p}(\mathbf{x}_j^\triangle) & \mathbf{p}(\mathbf{x}_{j'}^\triangle) \end{bmatrix}\right) \end{aligned} \tag{21}$$

since the pure-space GP prior over $\mathbf{u}_i$ is $\mathcal{N}(\mathbf{0}, \tilde{\mathbf{K}})$. The prior covariance is then given by the $(1,2)$-th element of the covariance matrix of the distribution in Equ. 21, i.e.,

$$\begin{aligned}\mathrm{Cov}(u_i(\mathbf{x}_j^\triangle), u_i(\mathbf{x}_{j'}^\triangle)) &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}^\top \begin{bmatrix} \mathbf{p}(\mathbf{x}_j^\triangle) & \mathbf{p}(\mathbf{x}_{j'}^\triangle) \end{bmatrix}^\top \tilde{\mathbf{K}} \begin{bmatrix} \mathbf{p}(\mathbf{x}_j^\triangle) & \mathbf{p}(\mathbf{x}_{j'}^\triangle) \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= \mathbf{p}(\mathbf{x}_j^\triangle)^\top \tilde{\mathbf{K}} \mathbf{p}(\mathbf{x}_{j'}^\triangle) \\ &= k^\triangle(\mathbf{x}_j^\triangle, \mathbf{x}_{j'}^\triangle)\end{aligned}$$

which is the prior covariance given by the mixed-space GP model with kernel $k^\triangle$ as desired.

Finally, we show that, for any agent i, the mixed-space GP posterior mean of any $u_i(\mathbf{x}^\triangle)$ and covariance between any $u_i(\mathbf{x}_j^\triangle)$ and $u_i(\mathbf{x}_{j'}^\triangle)$ with $k^\triangle$ is the same as that obtained through a linear transformation of the pure-space GP posterior mean and covariance on $\mathbf{u}_i$ with k, both conditioned on the same set of observations of pure strategy profiles. Suppose the pure-space GP model is conditioned on the set of observations $\mathcal{D}_t \coloneqq (\tilde{\mathbf{x}}_j, y_{i,j})_{j=1}^t$. Note that each pure strategy profile $\tilde{\mathbf{x}}_j$ has a unique representation $\tilde{\mathbf{x}}_j^\triangle$ in the mixed strategy profile space, at which all agents simply place all probability mass on their strategy in $\tilde{\mathbf{x}}_j$. The set of observations $\mathcal{D}_t$ in pure space thus has an equivalent representation in mixed space $\mathcal{D}_t^\triangle \coloneqq (\tilde{\mathbf{x}}_j^\triangle, y_{i,j})_{j=1}^t$ on which the mixed-space GP model is conditioned.

The pure-space GP model gives the posterior distribution over $\mathbf{u}_i$ to be

$$\begin{aligned}\mathbf{u}_i &\sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \\ \boldsymbol{\mu}_{i,t} &\coloneqq \mathbf{k}_t(\tilde{\mathcal{X}})^\top (\mathbf{K}_t + \lambda \mathbf{I})^{-1} \mathbf{y}_{i,t} \\ \boldsymbol{\Sigma}_t &\coloneqq \tilde{\mathbf{K}} - \mathbf{k}_t(\tilde{\mathcal{X}})^\top (\mathbf{K}_t + \lambda \mathbf{I})^{-1} k_t(\tilde{\mathcal{X}}) \\ [\mathbf{k}_t(\tilde{\mathcal{X}})]_{j\ell} &\coloneqq k(\tilde{\mathbf{x}}_j, \tilde{\mathbf{x}}^{(\ell)}),\ \mathbf{k}_t(\tilde{\mathcal{X}}) \in \mathbb{R}^{t \times m} \\ [\mathbf{K}_t]_{j\ell} &\coloneqq k(\tilde{\mathbf{x}}_j, \tilde{\mathbf{x}}_\ell),\ \mathbf{K}_t \in \mathbb{R}^{m \times m}.\end{aligned}$$

The mixed-space GP model gives the posterior mean of any $u_i(\mathbf{x}^\triangle)$ to be

$$\begin{aligned}\mu_i(\mathbf{x}^\triangle) &= \mathbf{k}_t^\triangle(\mathbf{x}^\triangle)^\top (\mathbf{K}_t^\triangle + \lambda \mathbf{I})^{-1} \mathbf{y}_{i,t} \\ [\mathbf{k}_t^\triangle(\mathbf{x}^\triangle)]_j &\coloneqq k^\triangle(\mathbf{x}^\triangle, \tilde{\mathbf{x}}_j^\triangle),\ \mathbf{k}_t^\triangle(\mathbf{x}^\triangle) \in \mathbb{R}^t \\ [\mathbf{K}_t^\triangle]_{j\ell} &\coloneqq k^\triangle(\tilde{\mathbf{x}}_j^\triangle, \tilde{\mathbf{x}}_\ell^\triangle),\ \mathbf{K}_t^\triangle \in \mathbb{R}^{m \times m}.\end{aligned} \tag{22}$$

Define the matrix $\mathbf{P}_t \in \mathbb{R}^{m \times t}$:

$$\mathbf{P}_t \coloneqq \begin{bmatrix} \mathbf{p}(\tilde{\mathbf{x}}_1^\triangle) & \mathbf{p}(\tilde{\mathbf{x}}_2^\triangle) & \cdots & \mathbf{p}(\tilde{\mathbf{x}}_t^\triangle) \end{bmatrix} \tag{23}$$

$\mathbf{k}_t^{\triangle}(\mathbf{x}^{\triangle})$ can then be written as

$$\begin{aligned}\mathbf{k}_t^{\triangle}(\mathbf{x}^{\triangle}) &= \begin{bmatrix} k^{\triangle}(\mathbf{x}^{\triangle}, \tilde{\mathbf{x}}_1^{\triangle}) \\ k^{\triangle}(\mathbf{x}^{\triangle}, \tilde{\mathbf{x}}_2^{\triangle}) \\ \vdots \\ k^{\triangle}(\mathbf{x}^{\triangle}, \tilde{\mathbf{x}}_t^{\triangle}) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{p}(\mathbf{x}^{\triangle})^\top \tilde{\mathbf{K}} \mathbf{p}(\tilde{\mathbf{x}}_1^{\triangle}) \\ \mathbf{p}(\mathbf{x}^{\triangle})^\top \tilde{\mathbf{K}} \mathbf{p}(\tilde{\mathbf{x}}_2^{\triangle}) \\ \vdots \\ \mathbf{p}(\mathbf{x}^{\triangle})^\top \tilde{\mathbf{K}} \mathbf{p}(\tilde{\mathbf{x}}_t^{\triangle}) \end{bmatrix} \\ &= \mathbf{P}_t^\top \tilde{\mathbf{K}} \mathbf{p}(\mathbf{x}^{\triangle}) \end{aligned} \tag{24}$$

Note that when $\tilde{\mathbf{x}}_j^{\triangle}$ and $\tilde{\mathbf{x}}_{j'}^{\triangle}$ are pure strategy profiles and hence admit the representations $\tilde{\mathbf{x}}_j$ and $\tilde{\mathbf{x}}_{j'}$, $k^{\triangle}(\tilde{\mathbf{x}}_j^{\triangle}, \tilde{\mathbf{x}}_{j'}^{\triangle}) = k(\tilde{\mathbf{x}}_j, \tilde{\mathbf{x}}_{j'})$ since we have shown that the prior covariance for any two mixed strategy profiles is the same via both pure- and mixed-space GP models. We thus have that

$$\mathbf{K}_t^{\triangle} = \mathbf{K}_t. \tag{25}$$

From Equ. 24, Equ. 25 and Lemma 8, the mixed-space GP posterior mean (Equ. 22) becomes

$$\begin{aligned}\mu_i(\mathbf{x}^{\triangle}) &= \mathbf{k}_t^{\triangle}(\mathbf{x}^{\triangle})^\top (\mathbf{K}_t^{\triangle} + \lambda \mathbf{I})^{-1} \mathbf{y}_{i,t} \\ &= \mathbf{p}(\mathbf{x}^{\triangle})^\top \tilde{\mathbf{K}} \mathbf{P}_t (\mathbf{K}_t + \lambda \mathbf{I})^{-1} \mathbf{y}_{i,t} \\ &= \mathbf{p}(\mathbf{x}^{\triangle})^\top \mathbf{k}_t(\tilde{\mathcal{X}})^\top (\mathbf{K}_t + \lambda \mathbf{I})^{-1} \mathbf{y}_{i,t} \\ &= \mathbf{p}(\mathbf{x}^{\triangle})^\top \boldsymbol{\mu}_{i,t}\end{aligned}$$

which is the posterior mean given by a linear transformation of the pure-space GP posterior distribution as desired.

The mixed-space GP model gives the posterior covariance between any $u_i(\mathbf{x}_j^{\triangle})$ and $u_i(\mathbf{x}_{j'}^{\triangle})$ to be

$$\begin{aligned}\mathrm{Cov}(u_i(\mathbf{x}_j^{\triangle}), u_i(\mathbf{x}_{j'}^{\triangle})) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}^\top \Bigg(&\begin{bmatrix} k^{\triangle}(\mathbf{x}_j^{\triangle}, \mathbf{x}_j^{\triangle}) & k^{\triangle}(\mathbf{x}_j^{\triangle}, \mathbf{x}_{j'}^{\triangle}) \\ k^{\triangle}(\mathbf{x}_{j'}^{\triangle}, \mathbf{x}_j^{\triangle}) & k^{\triangle}(\mathbf{x}_{j'}^{\triangle}, \mathbf{x}_{j'}^{\triangle}) \end{bmatrix} - \\ &\begin{bmatrix} \mathbf{k}_t^{\triangle}(\mathbf{x}_j^{\triangle}) & \mathbf{k}_t^{\triangle}(\mathbf{x}_{j'}^{\triangle}) \end{bmatrix}^\top (\mathbf{K}_t^{\triangle} + \lambda \mathbf{I})^{-1} \begin{bmatrix} \mathbf{k}_t^{\triangle}(\mathbf{x}_j^{\triangle}) & \mathbf{k}_t^{\triangle}(\mathbf{x}_{j'}^{\triangle}) \end{bmatrix} \Bigg) \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ = k^{\triangle}(\mathbf{x}_j^{\triangle}, \mathbf{x}_{j'}^{\triangle}) - \mathbf{k}_t^{\triangle}(\mathbf{x}_j^{\triangle})^\top (\mathbf{K}_t^{\triangle} + \lambda \mathbf{I})^{-1} \mathbf{k}_t^{\triangle}(\mathbf{x}_{j'}^{\triangle}).\end{aligned}$$

Again using Equ. 24, Equ. 25 and Lemma 8, the mixed-space GP posterior covariance becomes

$$\begin{aligned}\mathrm{Cov}(u_i(\mathbf{x}_j^{\triangle}), u_i(\mathbf{x}_{j'}^{\triangle})) &= k^{\triangle}(\mathbf{x}_j^{\triangle}, \mathbf{x}_{j'}^{\triangle}) - \mathbf{k}_t^{\triangle}(\mathbf{x}_j^{\triangle})^\top (\mathbf{K}_t^{\triangle} + \lambda \mathbf{I})^{-1} \mathbf{k}_t^{\triangle}(\mathbf{x}_{j'}^{\triangle}) \\ &= \mathbf{p}(\mathbf{x}_j^{\triangle})^\top \tilde{\mathbf{K}} \mathbf{p}(\tilde{\mathbf{x}}_{j'}^{\triangle}) - \mathbf{p}(\mathbf{x}_j^{\triangle})^\top \tilde{\mathbf{K}} \mathbf{P}_t (\mathbf{K}_t + \lambda \mathbf{I})^{-1} \mathbf{P}_t^\top \tilde{\mathbf{K}} \mathbf{p}(\mathbf{x}_{j'}^{\triangle}) \\ &= \mathbf{p}(\mathbf{x}_j^{\triangle})^\top \left(\tilde{\mathbf{K}} - \tilde{\mathbf{K}} \mathbf{P}_t (\mathbf{K}_t + \lambda \mathbf{I})^{-1} \mathbf{P}_t^\top \tilde{\mathbf{K}} \right) \mathbf{p}(\mathbf{x}_{j'}^{\triangle}) \\ &= \mathbf{p}(\mathbf{x}_j^{\triangle})^\top \left(\tilde{\mathbf{K}} - \mathbf{k}_t(\tilde{\mathcal{X}})^\top (\mathbf{K}_t + \lambda \mathbf{I})^{-1} \mathbf{k}_t(\tilde{\mathcal{X}}) \right) \mathbf{p}(\mathbf{x}_{j'}^{\triangle}) \\ &= \mathbf{p}(\mathbf{x}_j^{\triangle})^\top \boldsymbol{\Sigma}_t p(\mathbf{x}_{j'}^{\triangle})\end{aligned}$$

which is the $(1, 2)$-th element of the posterior covariance matrix of $u_i(\mathbf{x}_j^{\triangle})$ and $u_i(\mathbf{x}_{j'}^{\triangle})$ given by a linear transformation of the pure-space GP posterior distribution as desired and completes the proof.

■

Lemma 8 *With $\mathbf{P}_t$ defined as in Equ. 23, $\mathbf{k}_t(\tilde{\mathcal{X}}) = \mathbf{P}_t^\top \tilde{\mathbf{K}}$.*

Proof Since the sampled decisions $\tilde{\mathbf{x}}_j^{\triangle}$ for all $1 \leqslant j \leqslant t$ are pure strategy profiles, by definition of $\mathbf{p}(\cdot)$, $\mathbf{p}(\tilde{\mathbf{x}}_j^{\triangle})$ is a one-hot vector. Specifically, $\mathbf{p}(\tilde{\mathbf{x}}_j^{\triangle}) \in \mathbb{R}^m$ has a 1 in the $s(j)$-th element and 0 everywhere else, where $s(j)$ is a map such that $\tilde{\mathbf{x}}_j^{\triangle}$ is equivalent to the $s(j)$-th element in $\tilde{\mathcal{X}}$. $\mathbf{P}_t$ is thus a selection matrix of the form

$$\mathbf{P}_t = \begin{bmatrix} \mathbf{e}_{s(1)} & \mathbf{e}_{s(2)} & \cdots & \mathbf{e}_{s(t)} \end{bmatrix}$$

where each $\mathbf{e}_{s(j)}$ is a one-hot vector as described above. $\mathbf{P}_t^\top \tilde{\mathbf{K}}$ is then a $\mathbb{R}^{t \times m}$ matrix such that the j-th row of $\mathbf{P}_t^\top \tilde{\mathbf{K}}$ is equal to the $s(j)$-th row of of $\tilde{\mathbf{K}}$, i.e.,

$$\begin{aligned} [\mathbf{P}_t^\top \tilde{\mathbf{K}}]_{j\ell} &= k(\tilde{\mathbf{x}}^{(s(j))}, \tilde{\mathbf{x}}^{(\ell)}) \\ &= k(\tilde{\mathbf{x}}_j, \tilde{\mathbf{x}}^{(\ell)}) \end{aligned}$$

which is the definition of $\mathbf{k}_t(\tilde{\mathcal{X}})$ as desired. ■

A.7 Other lemmas

Lemma 9 ((Chowdhury and Gopalan, 2017) Theorem 2) *Let $\beta_t := B + \sigma\sqrt{2\left(\gamma_{t-1} + 1 + \ln(1/\delta)\right)}$ where B is the upper bound of the RKHS norms of each u_i. With probability of at least $1-\delta$, for all $i \in \mathcal{A}$, $\mathbf{x} \in \mathcal{X}$, and $t \in \mathbb{Z}^+$,*

$$|\mu_{i,t-1}(\mathbf{x}) - u_i(\mathbf{x})| \leqslant \beta_t \sigma_{t-1}(\mathbf{x})$$

where $\mu_{i,t-1}$ and σ_{t-1} are defined in Equ. 1 and Equ. 2 with $\lambda = 1 + \eta$ and $\eta := 2/T$.

Lemma 10 ((Chowdhury and Gopalan, 2017) Lemma 4) *Let $(\hat{\mathbf{x}}_t)_{t=1}^{\tau}$ be a sequence of strategies that the algorithm samples at. Then*

$$\sum_{t=1}^{\tau} \sigma_{t-1}(\hat{\mathbf{x}}_t) \leqslant \sqrt{4(\tau+2)\gamma_\tau}\,.$$

B COMPUTATIONAL DETAILS

B.1 Bilevel optimization for computing pure NE in continuous general-sum games

In the standard BO setting, to (approximately) optimize acquisition functions over continuous decision variable sets, one typically uses (sample-inefficient) black-box optimization algorithms such as DIRECT (Jones et al., 1993) or gradient-based optimization (e.g., L-BFGS-B) with multiple starts. In the pure NE setting, we encounter multiple bilevel optimization problems:

1. To compute a ground-truth NE, we need to solve $\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}} \min_{i\in\mathcal{A}} f_i(\mathbf{x})$ (Equ. 3).
2. To compute the reported strategy profile in UCB-PNE (Algo. 1), we need to solve $\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}} \min_{i\in\mathcal{A}} \widehat{f}_{i,t-1}(\mathbf{x})$ (Equ. 4).
3. To compute the reported strategy profile for baselines PE and BN (Sec. 6), we need to solve $\operatorname{argmax}_{\mathbf{x}\in\mathcal{X}} \min_{i\in\mathcal{A}}(\mu_{i,t-1}(\mathbf{x}) - \max_{\mathbf{t}_i\in\mathcal{X}_i}(\mu_{i,t-1}(\mathbf{t}_i, \mathbf{x}_{-i})))$ where $\mu_{i,t-1}$ is agent i's GP posterior mean (Equ. 1) in iteration t.

These are bilevel optimization problems as the inner functions have a maximization operation. In principle, one can nest any black-box optimization algorithm within a second black-box optimization algorithm to compute these quantities to any desired accuracy. In practice, doing so incurs a large computational burden. In our experiments, we use DIRECT as the outer algorithm and a simple algorithm we name 'sample-and-shrink' as the inner algorithm. In sample-and-shrink, we randomly choose 2^{13} points in a Sobol sequence within the function's bounds and find the point with the maximum function value. We then shrink the function's bounds by a factor of 4 around this point and choose another 2^{13} points in a Sobol sequence within these smaller bounds. We then return the point with the maximum function value. We find that this choice of outer and inner algorithms strikes a good balance between accuracy and computation time. For case 1, we use DIRECT with 200 iterations. For cases 2 and 3 (which need to be computed once every BO iteration), we use DIRECT with 50 iterations.

B.2 Computing mixed NE in general-sum games

While our theory for mixed NE is applicable to n-agent games, in our experiments, we focus on two-agent general-sum games due to the relative simplicity of computation. We refer the reader to of Shoham and Leyton-Brown (2008, Sec. 4.2) for a detailed exposition on the computation of mixed NE in two-player general sum games. In brief, one may use the Lemke-Howson algorithm by formulating the problem as a linear complementarity problem (Lemke and Howson, 1964), or the support enumeration method (SEM) (Porter et al., 2008). While both Lemke-Howson and SEM have a worst-case time complexity exponential in an agent's number of strategies, SEM has been shown to be faster than Lemke-Howson in some experimental settings (Porter et al., 2008), hence we adopt SEM in our experiments. SEM leverages the fact that computing a mixed NE reduces to a linear feasibility program given that the support of the mixed NE is known. It then simply iterates over all possible supports (smartly pruning some based on conditional domination) and tests each support by attempting to solve the linear feasibility program. In our implementation, we observe that caching the found supports from previous BO iterations and iterating over those first reduces the computation time significantly. We hypothesize that this speedup is due to the GP posterior means not changing much towards the end of the iterations and hence a mixed NE is likely be found with the same support. We use uniform random sampling to sample each $\tilde{\mathbf{u}}_{i,t-1}$.

C EXPERIMENTAL DETAILS

C.1 Acquisition functions

In this section, we declare the hyperparameters used for each acquisition function.

C.1.1 UCB-PNE

We use $\beta_t = 2$ for all $t \leqslant T$. We use DIRECT (Jones et al., 1993) to compute both the reporting strategy profile (bilevel optimization, refer to App. B.1 for more details) and the exploring strategy profile (100 DIRECT iterations).

C.1.2 Probability of Equilibrium (Picheny et al., 2019)

Following the experimental setup in Al-Dujaili et al. (2018), we discretize each agent's strategy set into 32 strategies (31 in previous work). This discretization is uniformly randomly sampled in each iteration, as we found that using a fixed discretization leads to poor performance.

C.1.3 BN (Al-Dujaili et al., 2018)

We use $\gamma = 2$. For the inner maximization, we use the Monte Carlo method (with 1000 samples) to approximate the regret instead of the closed-form method as the previous work showed empirically that the Monte Carlo strategy performed slightly better. For the outer maximization, we use DIRECT (Jones et al., 1993) with 100 iterations. Following the previous work, the ϵ-greedy policy uses an exploring probability $\epsilon = 0.05$ (not to be confused with ϵ in the main paper).

C.1.4 UCB-MNE

We use $\beta_t = 2$ for all $t \leqslant T$. Refer to App. B.2 for details on computing the reported mixed strategy.

C.2 Utility (objective) functions

C.2.1 Synthetic Random Functions (RF)

We construct a two-player general sum game by sampling two random 2-D functions from a GP prior and using them as utility functions. Each agent's strategy is in $[-1.0, 1.0]$. The GP prior used a squared exponential kernel with lengthscales $[0.5, 0.5]$. The learner uses the kernel of the GP prior for the GPs modeling the utilities. We used noise standard deviation $\sigma = 0.001$ and 5 initial samples where the decisions are chosen uniformly at random. To test the acquisition functions' ability to handle games with non-zero ϵ_*, we chose 5 RNG seeds such that the resultant games had $\epsilon_* \geqslant 0.05$. Specifically, we used seeds 4, 19, 20, 70, and 102 for the NumPy and TensorFlow RNGs.

C.2.2 Generative Adversarial Networks: 1-D Data Manifold (GAN)

Generative adversarial networks (GANs) are a class of generative models that consist of a generator and a discriminator. The generator maps realizations of a random variable to generated data, while the discriminator is trained to distinguish between real and generated data. The generator is trained to fool the discriminator by producing generated data as close to the real data as possible. Through this process, we obtain a generative model that produces realistic data. The desired generator and discriminator parameters form a Nash equilibrium of a two-player general sum game (Salimans et al., 2016).

We evaluate our algorithms on a simple GAN setup inspired by the experiments in Fedus et al. (2018). The real data lies on a 1-D manifold in 2-D space with probability distribution $p_{\mathbf{w}}$ obtained by multiplying some uniformly randomly sampled true parameters $\mathbf{w} \in [-1, 1]^2$ with samples from $\mathcal{N}(0, 1)$. The generator generates data on a 1-D manifold in the same way with probability distribution $p_{\mathbf{g}}$ and has parameters $\mathbf{g} \in [-1, 1]^2$. The goal is to learn the true parameters, i.e., have $\mathbf{g} = \mathbf{w}$. The discriminator is a simple binary classifier $D_{\mathbf{v}} : \mathbb{R}^2 \to [0, 1]$ that outputs the probability that the input is real. $D_{\mathbf{v}}$ is parameterized by $\mathbf{v} \in [-1, 1]^3$ and has the form

$$
\begin{aligned}
D(\mathbf{z}) &:= s(\mathbf{z}^\top V \mathbf{z}) \\
V &:= \begin{bmatrix} v_1 & \frac{1}{2}v_2 \\ \frac{1}{2}v_2 & v_3 \end{bmatrix} \\
s(z) &:= \frac{1}{1 + e^{-z}}
\end{aligned}
$$

where $\mathbf{z} \in \mathbb{R}^2$. The utility function of the discriminator is the expected log-likelihood of the binary classification task on the real and fake data, which is the negative of the usual discriminator loss (Goodfellow et al., 2014):

$$
u_D := \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{w}}} \left[\log D(\mathbf{z})\right] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{g}}} \left[\log(1 - D(\mathbf{z}))\right]
$$

The utility function of the generator is the negative of what Fedus et al. (2018) terms the non-saturating loss:

$$
u_G := \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{g}}} \left[\log D(\mathbf{z})\right].
$$

A pure NE exists at $\mathbf{g} = \mathbf{w}$ and $\mathbf{v} = \mathbf{0}$, because when $\mathbf{g} = \mathbf{w}$, the discriminator utility becomes

$$
u_D^* = \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{w}}} \left[\log D(\mathbf{z}) + \log(1 - D(\mathbf{z}))\right]
$$

and any function that assigns $D(\mathbf{z}) = 0.5$ everywhere on $\text{supp}(p_{\mathbf{w}})$ maximizes u_D^*. This condition is satisfied by $\mathbf{v} = \mathbf{0}$ as it assigns $D(\mathbf{z}) = 0.5$ everywhere. In this case, u_G is the same for all possible values of $\mathbf{g}$ and thus we have an NE. This problem thus has $\epsilon_* = 0$ in the pure NE setting. The learner used a squared exponential kernel with lengthscales $[0.5, 0.5, 2.0, 2.0, 2.0]$ for the GPs modeling the utilities. We used noise standard deviation $\sigma = 0.001$ and 5 initial samples where the decisions are chosen uniformly at random. We used seeds $0 - 4$ for the NumPy and TensorFlow RNGs.

C.2.3 Binary Classifiers: Additive Adversarial Attacks and Defenses (BCAD)

We evaluate our algorithms on finding NE in additive adversarial attacks and defenses on binary classifiers as described in Pal and Vidal (2020). We have a binary classifier that assigns the label $\text{sgn}(g(\mathbf{z}))$ to each point $\mathbf{z} \in \mathbb{R}^2$ where

$$
\begin{aligned}
g(\mathbf{z}) &:= \frac{1}{4}z_1^2 - \frac{1}{2}z_1 z_2 - \frac{1}{4}z_2^2 \\
\nabla g(\mathbf{z}) &= \frac{1}{2}\begin{bmatrix} z_1 - z_2 \\ -z_1 - z_2 \end{bmatrix}.
\end{aligned}
$$

The attacker's strategy is a vector field of perturbations with ℓ_2 norm less than some margin ϵ. Concretely, the attacker perturbs each point $\mathbf{z}$ with the function $\mathbf{a}_{\mathbf{v}} : \mathbb{R}^2 \to \mathbb{R}^2$, parameterized with $\mathbf{v} \in \mathbb{R}^3$:

$$
\begin{aligned}
\mathbf{a}_{\mathbf{v}}(\mathbf{z}) &:= -\epsilon \cdot \text{sgn}(g(\mathbf{z})) \frac{\mathbf{b}_{\mathbf{v}}(\mathbf{z})}{\|\mathbf{b}_{\mathbf{v}}(\mathbf{z})\|} \\
\mathbf{b}_{\mathbf{v}}(\mathbf{z}) &:= \begin{bmatrix} v_1 z_1 + v_2 z_2 \\ v_3(z_1 + z_2) \end{bmatrix}.
\end{aligned}
$$

The defender's strategy is a constant perturbation $\mathbf{d} \in \mathbb{R}^2, \|\mathbf{d}\| \leqslant \epsilon$. The perturbed point $\tilde{\mathbf{z}}$ is then both attacker's and defender's perturbations added to $\mathbf{z}$:

$$\tilde{\mathbf{z}} \coloneqq \mathbf{z} + \mathbf{a}_{\mathbf{v}}(\mathbf{z}) + \mathbf{d}.$$

The attacker's utility given a particular point $\mathbf{z}$ is

$$u_A(\mathbf{z}) \coloneqq \begin{cases} 1 \text{ if } \operatorname{sgn}(g_L(\mathbf{z}, \mathbf{z})) \neq \operatorname{sgn}(g_L(\mathbf{z}, \tilde{\mathbf{z}})) \\ -1 \text{ otherwise.} \end{cases}$$
$$g_L(\mathbf{z}, \mathbf{z}') \coloneqq g(\mathbf{z}) + \nabla g(\mathbf{z})^\top (\mathbf{z}' - \mathbf{z}).$$

The attacker gains utility if the added perturbations change the sign of $\mathbf{z}$ using a linear approximation of g around $\mathbf{z}$. The attacker's utility is then their expected utility under the distribution of $\mathbf{z}$ $p_{\mathbf{z}}$:

$$u_A \coloneqq \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}} \left[u_A(\mathbf{z}) \right].$$

The defender's utility is $-u_A$ which makes this a zero-sum game. The defender gains utility if the chosen $\mathbf{d}$ maintains the sign of $\mathbf{z}$ using a linear approximation of g around $\mathbf{z}$ in expectation. We choose $p_{\mathbf{z}}$ to be a uniform distribution over a randomly sampled rectangle with area 1 within the bounds $[-1, 1]^2$. We approximate the expectations with Monte Carlo sampling.

From Pal and Vidal (2020), under the defined utility function, choosing $\mathbf{a}$ according to the Fast Gradient Method (FGM) and $\mathbf{d}$ according to Randomized Smoothing results in an NE. FGM is recovered when $\mathbf{v} = (1/2, -1/2, -1/2)$ and Randomized Smoothing selects some $\mathbf{d}_* \in \mathbb{R}^2, \|\mathbf{d}_*\| \leqslant \epsilon$ which is also our domain for $\mathbf{d}$, hence the NE is achievable. This problem thus also has $\epsilon_* = 0$ in the pure NE setting. The learner used a squared exponential kernel with lengthscales $[1.5, 0.5, 1.0, 0.5, 0.5]$ for the GPs modeling the utilities. We used noise standard deviation $\sigma = 0.001$ and 5 initial samples where the decisions are chosen uniformly at random. We used seeds $0 - 4$ for the NumPy and TensorFlow RNGs.

C.3 Computation time

C.3.1 Pure NE

Table 1: Mean and standard error of computation times of each acquisition function in each game in the pure NE setting in CPU seconds.

	RF	GAN	BCAD
PE (without reporting)	43.38 ± 0.76	99.66 ± 0.42	102.73 ± 0.20
PE (with reporting)	11770.84 ± 30.71	10980.22 ± 12.86	12312.04 ± 17.72
BN (without reporting)	270.15 ± 56.96	380.67 ± 21.08	454.75 ± 18.20
BN (with reporting)	12214.34 ± 17.78	11242.32 ± 66.51	12632.13 ± 24.40
UCB-PNE no-exp	5796.41 ± 7.45	6638.85 ± 9.52	9658.40 ± 13.67
UCB-PNE	5848.52 ± 22.72	6670.05 ± 9.34	9659.24 ± 11.50

Table 1 shows the computation times in CPU seconds of the acquisition functions in each game in the pure NE setting for the hyperparameters used. These times were obtained using GPs with a dataset size of 100 and measured over 5 different acquisition function computations. Majority of the computation time arises from the bilevel optimization B.1 used to compute the reported strategy profile in our learning setting. UCB-PNE was designed with this requirement to report a strategy profile in mind and hence does not have a variant without reporting.

C.3.2 Mixed NE

Table 2 shows the computation times in CPU seconds of the acquisition functions in each game in the mixed NE setting for the hyperparameters used. These times were obtained by measuring the mean time per iteration in each of the experiments (used to compute Fig 5), then taking the mean and standard error over the 5 different RNG seeds. We chose to measure computation time for mixed NE in this way (compared to fixing a dataset) as it better illustrates the variability of computation time when using SEM to compute a potential mixed NE. All the acquisition functions tested use SEM to compute a potential mixed NE and differ only in the choice of sampled pure strategy profile. We observe that the computation spent

Table 2: Mean and standard error of computation times of each acquisition function in each game in the mixed NE setting in CPU seconds.

	RF	GAN	BCAD
Random	0.34 ± 0.16	248.14 ± 150.52	10.14 ± 3.78
Max entropy	0.78 ± 0.59	234.49 ± 137.38	1.39 ± 0.26
UCB-MNE no-exp	0.30 ± 0.12	287.30 ± 162.12	4.72 ± 1.24
UCB-MNE	0.30 ± 0.09	239.89 ± 145.61	9.10 ± 3.32

on the sampled strategy profile is minimal compared to the computation spent on SEM. The standard errors for mixed NE are much larger in proportion to the mean compared to in pure NE due to SEM: Some sampled utilities from the GPs take much longer to compute a mixed NE for using SEM as it heavily depends on the size of the support of a mixed NE for those sampled utilities. As such, there is large variability in the computation time per iteration.

C.4 Implementation

The experiments were implemented primarily with Python, NumPy (Harris et al., 2020), and GPflow (Matthews et al., 2017). Refer to the code repository for the full list of Python packages used.

C.5 Hardware

The experiments were run on the following hardware configurations:

1. $2\times$ AMD EPYC 7543 32-Core Processors, 256 GiB RAM, Ubuntu 20.04.4 LTS.
2. $2\times$ AMD EPYC 7352 24-Core Processors, 256 GiB RAM, Ubuntu 20.04.4 LTS.
3. $2\times$ Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz, 256 GiB RAM, Ubuntu 20.04.4 LTS.
4. $2\times$ Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz, 256 GiB RAM, Ubuntu 20.04.4 LTS.