Near-Optimal Pure Exploration in Matrix Games: A Generalization of Stochastic Bandits & Dueling Bandits

Arnab Maiti* **Ross Boczar*** **Kevin Jamieson** **Lillian J. Ratliff**
University of Washington
{arnabm2,rjboczar,ratliffl}@uw.edu, jamieson@cs.washington.edu

Abstract

We study the sample complexity of identifying the pure strategy Nash equilibrium (PSNE) in a two-player zero-sum matrix game with noise. Formally, we are given a stochastic model where any learner can sample an entry (i, j) of the input matrix $A \in [-1, 1]^{n \times m}$ and observe $A_{i,j} + \eta$ where η is a zero-mean 1-sub-Gaussian noise. The aim of the learner is to identify the PSNE of A, whenever it exists, with high probability while taking as few samples as possible. Zhou et al. (2017) presents an instance-dependent sample complexity lower bound that depends only on the entries in the row and column in which the PSNE lies. We design a near-optimal algorithm whose sample complexity matches the lower bound, up to log factors. The problem of identifying the PSNE also generalizes the problem of pure exploration in stochastic multi-armed bandits and dueling bandits, and our result matches the optimal bounds, up to log factors, in both the settings.

1 INTRODUCTION

Pure exploration is a well-studied topic in stochastic multi-armed bandits. Given a set of arms, the aim is to identify the arm with the highest mean with high probability while sequentially sampling the arms. A practically motivated version of this problem is *dueling bandits* where the learner pulls two arms and observes the winner among them. The objective is to identify the Condorcet winner: the arm that has probability greater than $1/2$ of winning against any other arm. The dueling bandit problem has also been studied from the perspective of pure exploration. A generalization of dueling bandits are matrix games which model scenarios where multiple strategic agents are involved and are competing against each other in a stochastic environment.

Consider a hypothetical situation in which firms F_1 and F_2 both offer the same product. Typically, when both firms set an identical price, the customer base is evenly distributed between them. However, if one firm sets its price higher than the other, it is likely to experience a reduced share of the customer base. This presents a strategic challenge for the firm: should it opt for a lower price p_1 or a higher price p_2? If the firms are non-cooperative, the optimal thing for each firm to do is to set the lower price p_1 otherwise they will most probably lose customers. Such a strategic scenario in a stochastic environment can be formally captured by a two-player zero-sum game defined by a matrix A as follows. Consider $A = [0, 0.25; -0.25, 0]$ where $A_{i,j}$ denotes the expected increase in the fraction of customers visiting firm F_1 if first decides to charge p_i and F_2 decides to charge p_j. Both firms setting a low price p_1 corresponds to the entry $(1, 1)$ and such an entry is referred to as pure strategy Nash equilibrium (PSNE) in game theory.

Formally, an entry is a PSNE of A if it is the highest entry in its column and lowest entry in its row. Although, a PSNE need not exist in every matrix A, there are some conditions under which they exist (see Duersch et al. (2012); Radzik (1991); Shapley (1964)). For instance, Duersch et al. (2012) showed that a symmetric two-player zero-sum game has a pure strategy equilibrium if and only if it is not a generalized rock-paper-scissors matrix. In this paper we generalize both stochastic multi-armed bandits and dueling bandits by studying the problem of identifying a PSNE, whenever it exists, in matrix games with noise.

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

1.1 Problem Setting

We consider two problems settings.

Matrix Games. In this problem setting, there is an arbitrary input matrix $A \in [-1,1]^{n\times m}$ which is unknown to the learner. The learner can sample an entry (i,j) of A and observe the random variable $X_{i,j} = A_{i,j} + \eta$ where η is a zero-mean 1-sub-Gaussian noise. The aim is to design a δ-PAC learner, that is, a learner which has the following property: if the learner stops after a finite time τ and returns an entry $(\widehat{i},\widehat{j})$ then $(\widehat{i},\widehat{j})$ is a PSNE of A with probability at least $1-\delta$. The objective is to minimize the number of samples drawn from A.

Dueling Bandits. In dueling bandits setting, there are K arms and a matrix $\mathsf{P} \in [0,1]^{K\times K}$. The matrix P has the property that for all i,j we have $\mathsf{P}_{i,j} + \mathsf{P}_{j,i} = 1$. In each round t, a pair of arms i,j are pulled (also known as a duel) and a winner X_t is declared where $\mathsf{P}(X_t = i) = 1 - \mathsf{P}(X_t = j) = \mathsf{P}_{i,j}$. An arm $i_\star$ is called a Condorcet winner if for all $j \in [K] \setminus \{i_\star\}$ we have $\mathsf{P}_{i_\star,j} > 1/2$. In pure exploration, one aims to design a learner that identifies the Condorcet winner, whenever it exists, with probability at least $1-\delta$ by performing as few duels as possible. Note that dueling bandits is a special case of a matrix game with $A = \mathsf{P}$. Here the PSNE of A is equal to (i_*, i_*) where i_* is the Condorcet winner.

1.2 Contributions

Zhou et al. (2017) initiated the study of identifying the PSNE where they showed that any δ-PAC learner requires $\Omega(\mathsf{H}_1 \log(1/\delta))$ samples to find the PSNE $(i_\star, j_\star)$, whenever it exists, where

$$\mathsf{H}_1 = \sum_{i\neq i_\star} \frac{1}{(A_{i_\star,j_\star} - A_{i,j_\star})^2} + \sum_{j\neq j_\star} \frac{1}{(A_{i_\star,j} - A_{i_\star,j_\star})^2}.$$

In this paper, we design a δ-PAC learner (Algorithm 3) that achieves the sample complexity of $\mathsf{H}_1 \log(1/\delta)$, up to log factors.

Recall that dueling bandits is a special case of matrix games. Haddenhorst et al. (2021) showed that any Condorcet δ-PAC learner requires $\sum_{i\neq i_\star} \Delta_{i,i_\star}^{-2} \log(1/\delta)$ where $\Delta_{i,i_\star} = \mathsf{P}_{i_\star,i} - 1/2$. In the notation of matrix games, this sample complexity is equal to $\frac{1}{2}\mathsf{H}_1 \log(1/\delta)$. As our algorithm achieves $\mathsf{H}_1 \log(1/\delta)$, up to log factors, it is a near-optimal Condorcet learner. To the best of our knowledge, this is the first algorithm to achieve this for Condorcet winner identification, without making additional strong assumptions (see the discussion on Dueling Bandits in Section 1.3).

Beyond the theoretical contributions, we also benchmark our algorithm against strong baselines in Section 3 and demonstrate that our algorithm is also superior empirically.

1.3 Related Work

Matrix games. As just described Zhou et al. (2017) showed that identifying the PSNE of a zero-sum matrix game requires a sample complexity of $\Omega(\mathsf{H}_1 \cdot \log(1/\delta))$. They also provided an algorithm based on LUCB that achieves an upper bound of $O(\mathsf{H}_1 \log(\mathsf{H}_1/\delta) + \frac{nm}{\tilde{\Delta}})$ where $\tilde{\Delta}$ is a matrix-dependent parameter. This upper bound is sub-optimal, and can be far worse than the lower bound. For instance, when δ and the gaps $|A_{i,j} - A_{i_\star,j_\star}|$ are constants, the upper bound scales as $O(nm)$ where as the lower bound is $\Omega(n+m)$. Recently, Maiti et al. (2023a) studied the problem of finding ε-Nash Equilibrium in $n \times 2$ games and provided near-optimal instance-dependent bounds. The instance-dependent parameters include the gaps between the entries of the matrix, and difference between the value of the game and reward received from playing a sub-optimal row. Later, Maiti et al. (2023b) extended the techniques of Maiti et al. (2023a) to identify the support of the Nash equilibrium in arbitrary $n \times m$ games. However, the bounds in their paper are sub-optimal.

An important class of learners for matrix games utilizes no-regret algorithms for adversarial bandits (cf. Freund and Schapire (1999)). Specifically, initialize two independent copies of $\mathsf{Exp3\text{-}IX}$ where the first selects rows to maximize reward, and the other selects columns to minimize loss, where both are fed $A_{i,j} + \eta$. Due to the fact that both algorithms enjoy an external regret bound of $\sqrt{KT}$ for a game played for T time steps with K arms, it can be shown that with constant probability, the average of the plays converges to an ϵ-approximate Nash equilibrium as soon as T exceeds $\frac{n+m}{\epsilon^2}$, ignoring log factors. Consequently, this guarantees that the PSNE can be identified with constant probability once T exceeds $\frac{n+m}{\Delta_{\min}^2}$ where

$$\Delta_{\min} = \min\{\min_{i\neq i_\star} A_{i_\star,j_\star} - A_{i,j_\star}, \min_{j\neq j_\star} A_{i_\star,j} - A_{i_\star,j_\star}\}.$$

The Tsalis-Inf algorithm of Zimmert and Seldin (2021) is a no-regret algorithm for adversarial bandits that also enjoys some instance-dependent guarantees under certain favorable conditions. While we are not aware of any analysis of this algorithm in the matrix game setting, our experiments will show that while it is a stronger baseline than $\mathsf{Exp3\text{-}IX}$, its empirical performance is far inferior to our algorithm.

Stochastic Multi-Armed Bandits. Best arm identification in the fixed confidence setting is a well-studied topic (see Appendix B for the problem formulation). One of the earlier works in Even-

Dar et al. (2002) introduces the Successive Elimination algorithm that achieved an upper bound of $\sum_{i\neq i_\star} \Delta_i^{-2}\log(n\Delta_i^{-2}/\delta)$ where $i_\star$ is the best arm and $\Delta_i = \mu_{i_\star} - \mu_i$. A lower bound of $\sum_{i\neq i_\star} \Delta_i^{-2}\log(1/\delta)$ was then established by Mannor and Tsitsiklis (2004) (later it was refined and simplified by Kaufmann et al. (2016)). The lower upper confidence bound (LUCB) algorithm designed by Kalyanakrishnan et al. (2012) achieved an upper bound of $\sum_{i\neq i_\star} \Delta_i^{-2}\log(\sum_{j\neq i_\star}\Delta_j^{-2}/\delta)$. Eventually, an upper bound of $\sum_{i\neq i_\star} \Delta_i^{-2}\log(\log(\Delta_i^{-2})/\delta)$ was achieved by the Exponential-gap Elimation algorithm of Karnin et al. (2013) and the lil'UCB algorithm of Jamieson et al. (2014). Jamieson et al. (2014) also showed that this upper bound is optimal for two-armed bandits. Later Chen et al. (2017) provided even tighter upper bounds in the general case based on entropy-like terms determined by the gaps.

Dueling Bandits. The dueling bandits problem is a well-studied variant of stochastic bandits (cf. Bengs et al. (2021) for a survey). The aim here is to identify the best arm (according to some rule) by noisy pairwise comparisons. This paper focuses on the Condorcet winner: the arm (assuming it exists) that has a probability greater than 1/2 of beating every other arm. Haddenhorst et al. (2021) provided a lower bound of $\sum_{i\neq i_\star} \Delta_{i,i_\star}^{-2} \equiv \frac{1}{2}\mathsf{H}_1$ to identify the Condorcet winner $i_\star$, whenever it exists, where $\Delta_{i,i_\star} = \mathsf{P}_{i_\star,i} - 1/2$ is the probability that arm $i_\star$ beats arm i, minus 1/2. They also provided an upper bound of $\frac{K}{\Delta^2_{\min,all}}$, up to log factors, where $\Delta_{\min,all} = \min_{i\neq j}|\mathsf{P}_{i,j} - 1/2|$.

In an effort to hit the lower bound, prior works introduced a number of strong assumptions to make the problem easier. The strong assumptions include total order, strong stochastic transitivity (SST) and stochastic triangle inequality (STI). Under total ordering, Mohajer et al. (2017) achieved an upper bound of $\frac{K\log\log K}{\min_{i\neq i_\star}\Delta^2_{i,i_\star}}$. Finally, Ren et al. (2020) achieved an upper bound of $\sum_{i\neq i_\star} \Delta_{i,i_\star}^{-2} \equiv \frac{1}{2}\mathsf{H}_1$ that matches the lower bound up to log factors, but under the strong assumptions of the total ordering setting, SST and STI. We emphasize that our work does *not* make these strong assumptions, yet still achieves the optimal sample complexity of H_1, up to log factors.

Regret minimization for dueling bandits has also been studied. Recently Saha and Gaillard (2022) achieved the optimal regret bound of $\sum_{i\neq i_\star}\frac{\log T}{\Delta_{i,i_\star}}$ with the help of Tsallis-inf from Zimmert and Seldin (2021). This immediately implies an upper bound of $\frac{1}{\min_{i\neq i_\star}\Delta_{i,i_\star}}\sum_{i\neq i_\star}\frac{1}{\Delta_{i,i_\star}}$ with constant probability, up to log factors. In the notation of matrix games, this sample complexity is at least H_1, and potentially $\Delta_{\min}^{-1}$ times larger than H_1 for certain instances (see the dueling bandit instance of Section 3). To the best of our knowledge, this is the best known sample complexity bound in this setting prior to our work.

2 FIXED CONFIDENCE NEAR-OPTIMAL ALGORITHM

Consider a matrix $A \in [-1,1]^{n\times m}$ which has a PSNE $(i_\star, j_\star)$. Let $\Delta_{i,j} = |A_{i,j} - A_{i_\star,j_\star}|$ for any i,j. Let $\mathsf{H}_1 = \sum_{i\neq i_\star}\frac{1}{\Delta^2_{i,j_\star}} + \sum_{j\neq j_\star}\frac{1}{\Delta^2_{i_\star,j}}$. Let us assume that $A_{i,j_\star} < A_{i_\star,j_\star} < A_{i_\star,j}$ for any $(i,j)\neq(i_\star,j_\star)$, otherwise H_1 is not well-defined. Now we state the main result.

Theorem 1. *There is an algorithm (Algorithm 3) that, with probability at least $1-\delta$, takes at most $c_0\cdot\mathsf{H}_1\cdot\log(\frac{nm\log(\mathsf{H}_1)}{\delta})\cdot\log(nm)$ samples from the input matrix A and returns the PSNE $(i_\star, j_\star)$. Here c_0 is an absolute constant.*

For the sake of simplicity of presentation, let us assume that n and m are powers of two. Let $\Delta_g := \left(\frac{1}{n+m-2}\left(\sum_{i\neq i_\star}\frac{1}{\Delta^2_{i,j_\star}} + \sum_{j\neq j_\star}\frac{1}{\Delta^2_{i_\star,j}}\right)\right)^{-1/2}$. Our goal is to identify $(i_\star, j_\star)$ with a sample complexity that scales as $(n+m-2)\Delta_g^{-2} = \mathsf{H}_1$, ignoring log factors.

We divide the analysis into multiple parts. First, in Section 2.1, we design an algorithm using a guess of Δ_g. Next, in Section 2.2, we describe the details of the procedures ColMidVal and RowMidVal that were used in Section 2.1. In Section 2.3, we state the main algorithm which has a sample complexity of $(n+m-2)\Delta_g^{-2}$, ignoring log factors. Finally, in Section 2.4, we address the special case of dueling bandits. In the interest of space, we refer the reader to Appendix E for a few omitted calculations.

2.1 Algorithm with a guess of Δ_g

In this section, we aim to design an algorithm to identify the PSNE $(i_\star, j_\star)$ when we are given a guess of the parameter Δ_g. Informally, the algorithm proceeds in a logarithmic number of stages. In each stage it aims to eliminate either half of the sub-optimal rows or sub-optimal columns. Suppose at one particular stage the algorithm aims to eliminate half of the sub-optimal rows. The algorithm takes samples from each column and computes a value near the median of the column entries. Then the algorithm chooses the column $\hat{j}$ which has the lowest such value. Then the algorithm samples all the entries of the column $\hat{j}$ sufficiently and removes those rows whose corresponding entries in column $\hat{j}$ are lower than median of the column entries.

Before formally describing our algorithm, we state the guarantees of the two subroutines ColMidVal and

Algorithm 1 $\mathsf{MidSearch}(A, \Delta, \delta)$

```
$\mathcal{X}_1 \leftarrow$ rows of $A$, $\mathcal{Y}_1 \leftarrow$ columns of $A$
for $t = 1, 2, \dots$ do
    if $\max\{|\mathcal{X}_t|, |\mathcal{Y}_t|\} = 2$ then
        Sample every element in $\mathcal{X}_t \times \mathcal{Y}_t$ $\frac{n+m-2}{2} \cdot \frac{50\log(16/\delta)}{\Delta^2}$ times and return the PSNE of the sub-matrix formed by $\mathcal{X}_t \times \mathcal{Y}_t$.
    else if $|\mathcal{X}_t| \geq |\mathcal{Y}_t|$ then
        Let $\mathcal{C}_j$ denote the elements of column $j$ in the sub-matrix formed by $\mathcal{X}_t \times \mathcal{Y}_t$.
        $\varepsilon_t \leftarrow \frac{1}{9}\left(\frac{|\mathcal{X}_t|}{n+m-2}\right)^{1/2} \cdot \Delta$
        Set $\hat{b}_{t,x}(j) \leftarrow \mathsf{ColMidVal}(\mathcal{C}_j, \varepsilon_t, \frac{\delta}{2m^2n^2})$ for each $j \in \mathcal{Y}_t$.
        $\hat{\jmath} \leftarrow \arg\min_{j\in\mathcal{Y}_t} \hat{b}_{t,x}(j)$ and $\mathcal{Y}_{t+1} \leftarrow \mathcal{Y}_t$
        Sample every entry in $\{(i, \hat{\jmath}) : i \in \mathcal{X}_t\}$ $\frac{n+m-2}{|\mathcal{X}_t|} \cdot \frac{162\log(4n^2m^2/\delta)}{\Delta^2}$ times and compute the empirical average $\bar{A}_{i,j}$ of these samples.
        $\mathcal{X}_{t+1} \leftarrow \{i_1, \dots, i_{|\mathcal{X}_t|/2}\} \subset \mathcal{X}_t = \{i_1, \dots, i_{|\mathcal{X}_t|}\}$ where $\bar{A}_{i_1,\hat{\jmath}} \geq \cdots \geq \bar{A}_{i_{|\mathcal{X}_t|},\hat{\jmath}}$.
    else if $|\mathcal{X}_t| < |\mathcal{Y}_t|$ then
        Let $\mathcal{R}_i$ denote the elements of row $i$ in the sub-matrix formed by $\mathcal{X}_t \times \mathcal{Y}_t$.
        $\varepsilon_t \leftarrow \frac{1}{9}\left(\frac{|\mathcal{Y}_t|}{n+m-2}\right)^{1/2} \cdot \Delta$
        Set $\hat{b}_{t,y}(i) \leftarrow \mathsf{RowMidVal}(\mathcal{R}_i, \varepsilon_t, \frac{\delta}{2m^2n^2})$, for each $i \in \mathcal{X}_t$.
        $\hat{\imath} \leftarrow \arg\max_{i\in\mathcal{X}_t} \hat{b}_{t,y}(i)$ and $\mathcal{X}_{t+1} \leftarrow \mathcal{X}_t$
        Sample every entry in $\{(\hat{\imath}, j) : j \in \mathcal{Y}_t\}$ $\frac{n+m-2}{|\mathcal{Y}_t|} \cdot \frac{162\log(4n^2m^2/\delta)}{\Delta^2}$ times and compute the empirical average $\bar{A}_{i,j}$ of these samples.
        $\mathcal{Y}_{t+1} \leftarrow \{j_1, \dots, j_{|\mathcal{Y}_t|/2}\} \subset \mathcal{X}_t = \{j_1, \dots, j_{|\mathcal{Y}_t|}\}$ where $\bar{A}_{\hat{\imath},j_1} \leq \cdots \leq \bar{A}_{\hat{\imath},j_{|\mathcal{Y}_t|}}$.
    end if
end for
```

$\mathsf{RowMidVal}$ from Section 2.2 that will be used in our algorithm. Consider $\varepsilon, \delta > 0$. Given a set of n arms $\mathcal{A}$ with means $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$, $\mathsf{ColMidVal}$ outputs a value $\hat{b} \in [\mu_{n/2} - \varepsilon, \mu_{n/4+1} + \varepsilon]$ with probability $1-\delta$ by using at most $O(\frac{1}{\varepsilon^2}\log\left(\frac{1}{\delta}\right))$ samples. Similarly, $\mathsf{RowMidVal}$ outputs a value $\hat{b} \in [\mu_{3n/4} - \varepsilon, \mu_{n/2+1} + \varepsilon]$ with probability $1-\delta$ by using at most $O(\frac{1}{\varepsilon^2}\log\left(\frac{1}{\delta}\right))$ samples. We refer the reader to Algorithm 1 for a formal description of our algorithm.

Now we begin the analysis of Algorithm 1. First, we state the following proposition that provides a lower bound on the gaps $\Delta_{i,j_\star}$. The lower bound later allows us to argue that a constant fraction of the entries in the optimal column $j_\star$ is well below $A_{i_\star,j_\star}$, and this fact plays a crucial role in lowering the number of samples required in each stage.

Proposition 1. *Consider a subset $S = \{i_1, \dots, i_\ell\} \subseteq [n]$ such that $\Delta_{i_1,j_\star} \leq \cdots \leq \Delta_{i_\ell,j_\star}$. Let $\ell \geq 2$. Then for any $s \geq \lceil \ell/4 \rceil + 1$, $\Delta_{i_s,j_\star} \geq \frac{1}{2}\left(\frac{|S|}{n+m-2}\right)^{1/2}\Delta_g$.*

Proof. Consider an index $s \geq \lceil \ell/4 \rceil + 1$. Now we have the following:

$$\begin{aligned}\left(\frac{|S|}{n+m-2}\right)^{-1}\Delta_g^{-2} &= \frac{1}{|S|}\left(\sum_{i\neq i_\star}\frac{1}{\Delta_{i,j_\star}^2} + \sum_{j\neq j_\star}\frac{1}{\Delta_{i_\star,j}^2}\right)\\ &\geq \frac{1}{|S|}\sum_{i\in S\setminus\{i_\star\}}(\Delta_{i,j_\star})^{-2}\\ &\geq \frac{1}{|S|}\sum_{k=2}^{\lceil \ell/4\rceil+1}(\Delta_{i_k,j_\star})^{-2}\\ &\geq \frac{\lceil \ell/4\rceil}{|S|}\cdot\Delta_{i_s,j_\star}^{-2}\\ &\geq (2\Delta_{i_s,j_\star})^{-2},\end{aligned}$$

where the second to last inequality holds since $\Delta_{i_s,j_\star} \geq \Delta_{i_k,j_\star}$ for all $k \leq \lceil \ell/4 \rceil + 1$, and the last inequality is due to the fact that $|S| = \ell$. $\square$

Similarly, we have the following proposition.

Proposition 2. *Consider a subset $S = \{j_1, \dots, j_\ell\} \subseteq [m]$ such that $\Delta_{i_\star,j_1} \leq \cdots \leq \Delta_{i_\star,j_\ell}$. Let $\ell \geq 2$. Then for any $s \geq \lceil \ell/4 \rceil + 1$, $\Delta_{i_\star,j_s} \geq \frac{1}{2}\left(\frac{|S|}{n+m-2}\right)^{1/2}\Delta_g$.*

Next, we establish the sample complexity of our algorithm in the following lemma.

Lemma 1. *The sample complexity of the procedure $\mathsf{MidSearch}(A, \Delta, \delta)$ is $c \cdot \frac{n+m-2}{\Delta^2} \cdot \log\left(\frac{nm}{\delta}\right) \cdot \log(nm)$ where c is an absolute constant.*

Proof. Let $\mathcal{T}_1 := \{t : |\mathcal{X}_{t+1}| = \frac{|\mathcal{X}_t|}{2}\}$ and $\mathcal{T}_2 := \{t : |\mathcal{Y}_{t+1}| = \frac{|\mathcal{Y}_t|}{2}\}$, and consider an iteration $t \in \mathcal{T}_1$. Recall that $\varepsilon_t = \frac{1}{9}\left(\frac{|\mathcal{X}_t|}{n+m-2}\right)^{1/2} \cdot \Delta$. Due to Lemma 3, $\mathsf{ColMidVal}(\mathcal{C}_j, \varepsilon_t, \frac{\delta}{2m^2n^2})$ takes $\frac{c\cdot\log(2n^2m^2/\delta)}{\varepsilon_t^2}$ samples where c is an absolute constant. Hence, the total number of samples taken in the iteration t is upper bounded by $|\mathcal{Y}_t| \cdot \frac{n+m-2}{|\mathcal{X}_t|} \cdot \frac{81c\cdot\log(2n^2m^2/\delta)}{\Delta^2} + |\mathcal{X}_t| \cdot \frac{n+m-2}{|\mathcal{X}_t|} \cdot \frac{162\log(4n^2m^2/\delta)}{\Delta^2} \leq (c+1)\cdot(n+m-2)\cdot\frac{162\log(4n^2m^2/\delta)}{\Delta^2}$, since $|\mathcal{Y}_t| \leq |\mathcal{X}_t|$. By an identical argument, this bound applies to each iteration $t \in \mathcal{T}_2$.

Finally consider iteration t such that $\max\{|\mathcal{X}_t|, |\mathcal{Y}_t|\} = 2$. The total number of samples taken in this iteration t is upper bounded by $4 \cdot \frac{n+m-2}{2} \cdot \frac{50\log(16/\delta)}{\Delta^2}$.

We now get the desired sample complexity as $|\mathcal{T}_1| = \log_2(n) - 1$ and $|\mathcal{T}_2| = \log_2(m) - 1$. $\square$

Finally, we establish the correctness of our algorithm in the following lemma. The proof of the lemma makes use of the fact that the entries around the median of column $j_\star$ are far below $A_{i_\star,j_\star}$. The sampling done from each column ensures that the entries around the median of column $\hat{j}$ are also far below $A_{i_\star,j_\star}$. This along with the fact that $A_{i_\star,\hat{j}} \geq A_{i_\star,j_\star}$ ensures that we don't eliminate the row $i_\star$.

Lemma 2. *If $\Delta \leq \Delta_g$, then the procedure* MidSearch(A,Δ,δ) *returns $(i_\star, j_\star)$ with probability at least $1-\delta$.*

Proof. Let us consider the case when $\min\{n,m\} \geq 4$. The other cases can be proved analogously. Let

$\mathcal{T}_1 := \{t : |\mathcal{X}_{t+1}| = \frac{|\mathcal{X}_t|}{2}\}$ and $\mathcal{T}_2 := \{t : |\mathcal{Y}_{t+1}| = \frac{|\mathcal{Y}_t|}{2}\}$.

Consider an iteration $t \in \mathcal{T}_1$ such that $|\mathcal{X}_t| \geq 4$ and let us assume that $i_\star \in \mathcal{X}_t$ and $j_\star \in \mathcal{Y}_t$. Observe that $\mathcal{Y}_{t+1} = \mathcal{Y}_t$. We now show that with probability at least $1-\frac{\delta}{nm}$, we have $i_\star \in \mathcal{X}_{t+1}$. Let $\Delta_t = \left(\frac{|\mathcal{X}_t|}{n+m-2}\right)^{1/2}\Delta_g$. Recall the definition of ε_t and observe that $\varepsilon_t \leq \Delta_t/9$.

Consider an index $j \in \mathcal{Y}_t$. Let us relabel the indices in $\mathcal{X}_t$ as $\{i_1^{(j)}, \ldots, i_{|\mathcal{X}_t|}^{(j)}\}$ such that $A_{i_1^{(j)},j} \geq \ldots \geq A_{i^{(j)}_{|\mathcal{X}_t|},j}$. Let G_j be the event that $\mathbf{b}_{t,x}(j) \in [A_{i^{(j)}_{|\mathcal{X}_t|/2},j} - \varepsilon_t, A_{i^{(j)}_{|\mathcal{X}_t|/4+1},j} + \varepsilon_t]$. Due to Lemma 4, event G_j holds with probability at least $1 - \frac{\delta}{2m^2n^2}$.

Let us assume that G_j holds for all $j \in \mathcal{Y}_t$. This happens with with probability at least $1-\frac{\delta}{2n^2m}$ due to a union bound. As event $G_{j_\star}$ holds, we have the following due to Proposition 1:

$$\begin{aligned}\mathbf{b}_{t,x}(j_\star) &\leq A_{i^{(j_\star)}_{|\mathcal{X}_t|/4+1},j_\star} + \frac{\Delta_t}{9}, \\ &\leq A_{i_\star,j_\star} - \frac{\Delta_t}{2} + \frac{\Delta_t}{9} = A_{i_\star,j_\star} - \frac{7\Delta_t}{18}.\end{aligned}$$

Recall the definition of $\hat{j}$. Since event $G_{\hat{j}}$ holds, we have the following for all $\frac{|\mathcal{X}_t|}{2} \leq s \leq |\mathcal{X}_t|$:

$$A_{i_s^{(\hat{j})},\hat{j}} \leq \mathbf{b}_{t,x}(\hat{j}) + \frac{\Delta_t}{9} \leq \mathbf{b}_{t,x}(j_\star) + \frac{\Delta_t}{9} \leq A_{i_\star,j_\star} - \frac{5\Delta_t}{18}.$$

Let us assume that for all $i \in \mathcal{X}_t$, $|\bar{A}_{i,\hat{j}} - A_{i,\hat{j}}| \leq \frac{\Delta_t}{9}$. This happens with probability at least $1-\frac{\delta}{2nm^2}$ due to the sub-Gaussian tail bound and union bound. We now have the following for all $\frac{|\mathcal{X}_t|}{2} \leq s \leq |\mathcal{X}_t|$:

$$\bar{A}_{i_s^{\hat{j}},\hat{j}} \leq A_{i_s^{\hat{j}},\hat{j}} + \frac{\Delta_t}{9} \leq A_{i_\star,j_\star} - \frac{3\Delta_t}{18}.$$

This implies that $|\{i \in \mathcal{X}_t : \bar{A}_{i,\hat{j}} \leq A_{i_\star,j_\star} - \frac{3\Delta_t}{18}\}| \geq \frac{|\mathcal{X}_t|}{2}$. Recall that $\mathcal{X}_{t+1} = \{i_1, \ldots, i_{|\mathcal{X}_t|/2}\} \subset \mathcal{X}_t = \{i_1, \ldots, i_{|\mathcal{X}_t|}\}$ where $\bar{A}_{i_1,\hat{j}} \geq \cdots \geq \bar{A}_{i_{|\mathcal{X}_t|},\hat{j}}$. Now observe that $\bar{A}_{i_\star,\hat{j}} \geq A_{i_\star,\hat{j}} - \frac{\Delta_t}{9} \geq A_{i_\star,j_\star} - \frac{\Delta_t}{9}$. Hence, $i_\star \in \mathcal{X}_{t+1}$.

Analogously we can show that for an iteration $t \in \mathcal{T}_2$ such that $|\mathcal{Y}_t| \geq 4$, $i_\star \in \mathcal{X}_t$ and $y_\star \in \mathcal{Y}_t$, with probability at least $1-\frac{\delta}{nm}$ we have $i_\star \in \mathcal{X}_{t+1}$ and $y_\star \in \mathcal{Y}_{t+1}$.

Observe that the algorithm terminates at a fixed iteration $t_\star = \log_2(nm) - 1$. Let $p_t = \mathsf{P}(i_\star \in \mathcal{X}_t, j_\star \in \mathcal{Y}_t | i_\star \in \mathcal{X}_{t-1}, j_\star \in \mathcal{Y}_{t-1})$. Now we have the following due to the chain rule and Bernoulli's inequality,

$$\mathsf{P}(i_\star \in \mathcal{X}_{t_\star}, j_\star \in \mathcal{Y}_{t_\star}) = \prod_{t=2}^{t_\star} p_t \geq \left(1 - \frac{\delta}{nm}\right)^{t_\star} \geq 1 - \delta/2.$$

Hence, with probability at least $1-\delta/2$ we have an iteration $t_\star$ such that $\mathcal{X}_{t_\star} = \{i_\star, \hat{i}\}$ and $\mathcal{Y}_{t_\star} = \{j_\star, \hat{j}\}$. Under the assumption of the existence of such an iteration $t_\star$, we now show that with probability at least $1-\delta/2$ we return $(i_\star, j_\star)$. Let $\Delta_{t_\star} = \left(\frac{|\mathcal{X}_{t_\star}|}{n+m-2}\right)^{1/2}\Delta_g$. For all $(i,j) \in \mathcal{X}_{t_\star} \times \mathcal{Y}_{t_\star}$, let $\bar{A}_{i,j}$ denote the empirical average of the $\frac{n+m-2}{|\mathcal{X}_{t_\star}|} \cdot \frac{50\log(16/\delta)}{\Delta_g^2}$ samples. Let us assume that for all $(i,j) \in \mathcal{X}_{t_\star} \times \mathcal{Y}_{t_\star}$, $|\bar{A}_{i,j} - A_{i,j}| \leq \frac{\Delta_{t_\star}}{5}$. This happens with probability at least $1-\delta/2$ due to sub-gaussian tail bound and union bound. Now observe that $A_{i_\star,j_\star} - \frac{\Delta_{t_\star}}{5} \leq \bar{A}_{i_\star,j_\star} \leq A_{i_\star,j_\star} + \frac{\Delta_{t_\star}}{5}$. Due to Proposition 2, we have $\bar{A}_{i_\star,\hat{j}} \geq A_{i_\star,\hat{j}} - \frac{\Delta_{t_\star}}{5} \geq A_{i_\star,j_\star} + \frac{\Delta_{t_\star}}{2} - \frac{\Delta_{t_\star}}{5} > \bar{A}_{i_\star,j_\star}$. Similarly due to Proposition 1, we have $\bar{A}_{\hat{i},j_\star} \leq A_{\hat{i},j_\star} + \frac{\Delta_{t_\star}}{5} \leq A_{i_\star,j_\star} - \frac{\Delta_{t_\star}}{2} + \frac{\Delta_{t_\star}}{5} < \bar{A}_{i_\star,j_\star}$. Hence, we return $(i_\star, j_\star)$. $\square$

2.2 ColMidVal & RowMidVal Subroutine

Here, we define ColMidVal (Algorithm 2) which solves the subproblem of finding a value near the median of means of n arms; RowMidVal is defined analogously in Appendix C along with its guarantees.

Consider $\varepsilon, \delta > 0$. Given a set of n arms $\mathcal{A}$ with means $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$, we show that ColMidVal outputs a value $\mathbf{b} \in [\mu_{n/2} - \varepsilon, \mu_{n/4+1} + \varepsilon]$ with probability $1-\delta$ by using at most $O(\frac{1}{\varepsilon^2}\log\left(\frac{1}{\delta}\right))$ samples. We assume that n is a multiple of 4 for simplicity.

We first establish the sample complexity of ColMidVal in the following lemma.

Lemma 3. ColMidVal$(\mathcal{A},\varepsilon,\delta)$ *requires at most $\frac{c\log(1/\delta)}{\varepsilon^2}$ samples.*

Proof. As k, δ_1 are constants, the total number of sam-

Algorithm 2 $\mathsf{ColMidVal}(\mathcal{A}, \varepsilon, \delta)$

1: $\ell \leftarrow \lceil 14\log(\frac{1}{\delta})\rceil$, $\delta_1 \leftarrow 0.05$, $\delta_2 \leftarrow 0.05$, $k \leftarrow 108\log(\frac{4}{\delta_2})$ and $z \leftarrow \frac{k}{3}+1$.
2: **for** $i = 1$ to ℓ **do**
3: $\quad \mathcal{B}_i \leftarrow \{k$ arms sampled independently and uniformly at random from $\mathcal{A}\}$.
4: $\quad$ Sample each arm $j \in \mathcal{B}_i$ $\frac{2\log(\frac{2k}{\delta_1})}{\varepsilon^2}$ times and compute the empirical mean $\hat{\mu}_j$.
5: $\quad v_i \leftarrow z$-th highest value in $\{\hat{\mu}_j : j \in \mathcal{B}_i\}$
6: **end for**
7: **return** the median of $\{v_i : i \in [\ell]\}$

ples is

$$14\log\left(\frac{1}{\delta}\right)\cdot k\cdot\left\lceil\frac{2\log(\frac{2k}{\delta_1})}{\varepsilon^2}\right\rceil \le \frac{c\log(1/\delta)}{\varepsilon^2}$$

for an absolute constant c. □

Next, we establish the correctness of $\mathsf{ColMidVal}$. The proof follows from the standard median of means argument.

Lemma 4. *Given n arms $\mathcal{A}$ with means $\mu_1 \ge \mu_2 \ge \cdots \ge \mu_n$, $\mathsf{ColMidVal}(\mathcal{A}, \varepsilon, \delta)$ outputs a value $\hat{b} \in [\mu_{n/2} - \varepsilon, \mu_{n/4+1} + \varepsilon]$ with probability at least $1-\delta$.*

Proof. Consider an iteration i. Let $n_{1,i}$ and $n_{2,i}$ be the number of times arms 1 through $n/4$ and arms $n/4+1$ through $n/2$ were drawn for $\mathcal{B}_i$, respectively; that is, the counts of occurrences corresponding to the first and second quantiles of highest arm means. We then say the event E_i holds if

- $\frac{7k}{40} \le n_{1,i} \le \frac{k}{3}$ and $\frac{7k}{40} \le n_{2,i} \le \frac{k}{3}$, and
- for all $j \in \mathcal{B}_i$, $|\hat{\mu}_j - \mu_j| \le \varepsilon$.

By a standard sub-Gaussian tail bound and union bound, we have that $|\hat{\mu}_j - \mu_j| \le \varepsilon$ simultaneously for all $j \in \mathcal{B}_i$ with probability at least $1-\delta_1$. This implies that any sampled arm not in the first quantile has an empirical mean of at most $\mu_{n/4+1} + \varepsilon$. Similarly any sampled arm from the first and second quantile has an empirical mean of at least $\mu_{n/2} - \varepsilon$. Next, observe that $\mathsf{E}[n_{1,i}] = \mathsf{E}[n_{2,i}] = k/4$. Using the Chernoff bound and union bound, we have $\frac{7k}{40} \le |n_{1,i}| \le \frac{k}{3}$ and $\frac{7k}{40} \le |n_{2,i}| \le \frac{k}{3}$ with probability at least $1-\delta_2$. Thus, by a union bound, $\mathsf{P}[E_i] \ge 1-\delta_1-\delta_2 = \frac{9}{10}$.

We first claim that if E_i holds, then $v_i \in [\mu_{n/2} - \varepsilon, \mu_{n/4+1} + \varepsilon]$. We see that the z-th highest empirical mean is in the interval $[\mu_{n/2} - \varepsilon, \mu_{n/4+1} + \varepsilon]$ if there are *at most* $z-1$ empirical means greater than $\mu_{n/4+1} + \varepsilon$ and there are *at least* z empirical means greater than or equal to $\mu_{n/2} - \varepsilon$. Noting that $z = \frac{k}{3}+1$, on E_i, the first condition occurs as $n_{1,i} \le k/3 = z-1$, and the second condition occurs as $n_{1,i} + n_{2,i} \ge \frac{7k}{20} > z$.

Finally, let X_i be the indicator variable $\mathbb{1}_{E_i}$, and let $Z = \sum_{i=1}^{\ell} X_i$ be the total number of times E_i occurs. If $Z > \ell/2$, then the median of $\{v_i : i \in [\ell]\}$ lies in $[\mu_{n/2} - \varepsilon, \mu_{n/4+1} + \varepsilon]$. Observing that $\mathsf{E}[Z] \ge \frac{9\ell}{10}$, where $\ell = \lceil 14\log(\frac{1}{\delta})\rceil$, by the Chernoff bound we have $\mathsf{P}[Z \ge 0.54\ell] \ge 1-\delta$. Hence, $\mathsf{ColMidVal}(\mathcal{A}, \varepsilon, \delta)$ returns a value $\hat{b} \in [\mu_{n/2} - \varepsilon, \mu_{n/4+1} + \varepsilon]$ with probability at least $1-\delta$. □

2.3 Algorithm without the knowledge of Δ_g

First we describe the guarantees of a procedure based on stochastic bandit algorithms that is used to verify whether an entry $(\hat{\imath}, \hat{\jmath})$ is a PSNE of A or not.

Lemma 5. *Given an entry $(\hat{\imath}, \hat{\jmath})$ and parameters Δ and δ, there is a procedure Verify possessing the following properties with probability $1-\delta$:*

- *If $\Delta \le \Delta_g$ and $(\hat{\imath}, \hat{\jmath}) = (i_\star, j_\star)$, then $(\hat{\imath}, \hat{\jmath})$ is "accepted" as the PSNE of A by Verify.*
- *If $(\hat{\imath}, \hat{\jmath}) \neq (i_\star, j_\star)$, then $(\hat{\imath}, \hat{\jmath})$ is "not accepted" as the PSNE of A by Verify.*

Moreover, Verify terminates after taking $c \cdot \mathsf{H}_\star \cdot \log(\frac{\log(\mathsf{H}_\star)}{\delta})$ samples from A where c is an absolute constant and $\mathsf{H}_\star = \frac{n+m-2}{\Delta^2}$.

We refer the reader to Appendix D for more details.

Algorithm 3 is a meta-algorithm for finding $(i_\star, j_\star)$ with high probability without the knowledge of Δ_g. Theorem 1 establishes the guarantees for Algorithm 3.

Algorithm 3

1: **Given:** Probability error term δ.
2: **for** $t = 1, 2, \ldots$ **do**
3: $\quad \Delta_t \leftarrow \frac{1}{2^{t-1}}$ and $\delta_t \leftarrow \frac{\delta}{4t^2}$
4: $\quad (i_t, j_t) \leftarrow \mathsf{MidSearch}(A, \Delta_t, \delta_t)$
5: $\quad$ **if** $\mathsf{Verify}(A, i_t, j_t, \delta_t, \Delta_t)$="accepted" **then**
6: $\quad\quad$ **Return** (i_t, j_t) as the PSNE
7: $\quad$ **end if**
8: **end for**

Proof of Theorem 1. Recall that $\Delta_g = (\frac{n+m-2}{\mathsf{H}_1})^{1/2}$. Consider an iteration $t \le \lceil\log_2(1/\Delta_g)\rceil+1$. If $(i_t, j_t) \neq (i_\star, j_\star)$ then, due to Lemma 5, (i_t, j_t) is not returned as the PSNE with probability at least $1 - \frac{\delta}{4t^2}$. Consider the iteration $t_\star = \lceil\log_2(1/\Delta_g)\rceil+2$. Observe that $\Delta_{t_\star} \le \frac{1}{2^{\lceil\log_2(1/\Delta_g)\rceil}} = \Delta_g$. Due to Lemma 2, with probability at least $1 - \frac{\delta}{4t_\star^2}$, we have $(i_{t_\star}, j_{t_\star}) = (i_\star, j_\star)$. Due to Lemma 5, the pair $(i_\star, j_\star)$ is accepted and returned as a PSNE with probability at least $1 - \frac{\delta}{4t_\star^2}$.

Hence, with probability at least $1 - \frac{\delta}{4t_\star^2} - \sum_{t=1}^{\infty} \frac{\delta}{4t^2} > 1 - \delta$ the algorithm terminates by the iteration $t_\star = \lceil \log_2(1/\Delta_g) \rceil + 2$ and returns $(i_\star, j_\star)$ as the PSNE.

Let us assume that the algorithm terminates by the iteration $t_\star = \lceil \log_2(1/\Delta_g) \rceil + 2$. Due to Lemma 1 and Lemma 5, the number of samples in any iteration t is upper bounded by $c \cdot \frac{n+m-2}{\Delta_t^2} \cdot \log\left(\frac{nm \log(\frac{1}{\Delta_t})}{\delta_t}\right) \cdot \log(nm)$ where c is an absolute constant. The sample complexity τ of Algorithm 3 is upper bounded as follows:

$$
\begin{aligned}
\tau &\leq \sum_{t=1}^{t_\star} c \cdot \frac{n+m-2}{\Delta_t^2} \cdot \log\left(\frac{nm \log(\frac{1}{\Delta_t})}{\delta_t}\right) \cdot \log(nm), \\
&\leq \sum_{t=1}^{t_\star} c \cdot 4^t (n+m-2) \cdot \log\left(\frac{4t^3 nm}{\delta}\right) \cdot \log(nm), \\
&\leq c \cdot (n+m-2) \cdot \log\left(\frac{4t_\star^3 \cdot nm}{\delta}\right) \cdot \log(nm) \cdot \sum_{t=1}^{t_\star} 4^t,
\end{aligned}
$$

where the second inequality holds since $\Delta_t = \frac{1}{2^{t-1}}$ and $\delta_t = \frac{\delta}{4t^2}$. Using the fact that $\sum_{t=1}^{t_\star} 4^t \leq \frac{4^{t_\star+1}}{3}$, we further deduce that

$$
\begin{aligned}
\tau &\leq \frac{4c}{3} \cdot (n+m-2) \cdot \log\left(\frac{4t_\star^3 \cdot nm}{\delta}\right) \cdot \log(nm) \cdot 2^{2t_\star}, \\
&\leq \frac{256c}{3} \cdot \frac{n+m-2}{\Delta_g^2} \cdot \log\left(\frac{4t_\star^3 \cdot nm}{\delta}\right) \cdot \log(nm),
\end{aligned}
$$

where the last inequality is due to $t_\star \leq \log_2(1/\Delta_g) + 3$. The claim in the theorem follows from the fact that $\Delta_g = (\frac{n+m-2}{\mathbf{H_1}})^{1/2}$. □

2.4 Dueling Bandits

Recall that in the context of dueling bandits defined by the matrix P, we have $\Delta_{i,j} = |1/2 - \mathsf{P}_{i,j}|$. The lower bound to identify the Condorcet winner $i_\star$, whenever it exists, scales roughly as $\sum_{i \neq i_\star} \Delta_{i,i_\star}^{-2}$. Now, an instance of dueling can be interpreted as a two-player zero-game on a matrix A where $A = \mathsf{P}$. It is straightforward to see that the PSNE of A is $(i_\star, i_\star)$. Algorithm 3 achieves the bound of $\sum_{i \neq i_\star} \Delta_{i,i_\star}^{-2}$, up to log factors.

3 EXPERIMENTS

To illustrate the performance of MidSearch, we introduce the following class of $d \times d$ reward matrices, where we use bold to denote a vector of repeated entries:

$$
A_{\mathsf{hard}}(\Delta_{\min}, \beta) := \left[\begin{array}{c|c|c}
0.5 & 0.5 + \Delta_{\min} & (\mathbf{0.5} + \boldsymbol{\beta})^\top \\
\hline
0.5 - \Delta_{\min} & \multicolumn{2}{c}{\multirow{2}{*}{$0.5I + U$}} \\
\mathbf{0.5} - \boldsymbol{\beta} & &
\end{array}\right],
$$

where U is the matrix with all ones on the (strictly) upper triangular entries and $0 < \Delta_{\min} < \beta$. These instances are difficult for the class of learners that play two no-regret learners against themselves, as the second row and second column have high and low sums, respectively, causing $A_{2,2}$ to potentially appear optimal. Note, this is also a dueling bandit instance, which is a special case of a matrix game.

As baselines, we compare MidSearch[1] to Tsallis-INF (Zimmert and Seldin, 2021), Exp3-IX (Neu, 2015), LUCB-G (Zhou et al., 2017), and uniform random sampling for different matrices in A_{hard}. Each algorithm is given a fixed budget T of samples that it can draw from the input matrix and we measure its percentage of identifying the PSNE for the given value of T. Code to replicate these experiments is available at `https://github.com/aistats2024-noisy-psne/midsearch`; supplementary experimental results can be found in Appendix F.

Let $H_1 = \frac{d-2}{\beta^2} + \frac{1}{\Delta_{\min}^2}$. Figure 1 shows that even with a budget of $50H_1$ samples, the Exp3-IX, LUCB-G[2], and Uniform algorithms already begin to degrade in performance at $d = 32$, returning the correct PSNE on 282, 121, and 99 of the 300 trials, respectively (Exp3-IX degrades slightly slower, worsening to 150/300 at $d = 128$).

Thus, for larger matrices, we only compare MidSearch and Tsallis-INF. In Figure 2 we plot the performance of MidSearch vs. Tsallis-INF for instances of $A_{\mathsf{hard}}(\Delta_{\min}, 0.1)$ with $d = 1024$, varying only $\Delta_{\min}$. We see that after $T = 50H_1$ samples, MidSearch remains close to perfect for the $d = 1024$ matrix, while the accuracy of Tsallis-INF decreases as $\Delta_{\min}$ decreases from 0.1 to 0.0125.

Note that the sample complexity for Uniform is $d^2\Delta_{\min}^{-2}$, LUCB-G is no worse than $d\Delta_{\min}^{-2} + d^2$, Exp3-IX is no worse than $d\Delta_{\min}^{-2}$, Tsallis-INF is no worse

[1] The subroutines ColMidVal and RowMidVal are wasteful. We optimize them for empirical performance by using an aggressive procedure similar to successive halving. Furthermore, a fixed budget of T is achieved for MidSearch by setting $\Delta \approx \left(\frac{n+m}{T}\right)^{1/2}$ and $\delta \approx$ constant.

[2] For LUCB-G, we set $\delta \approx$ constant and return the PSNE it identifies by using at most T samples. We take the PSNE of the empirical mean matrix (if it exists) if the budget has been reached.

Figure 1: Plot for each algorithm on the $A_{\text{hard}}(0.05, 0.1)$ instance with $N = 300$ trials and with a budget of $50H_1$ samples; the y-axis shows what percentage of the time each algorithm's output was correct at different checkpoints during each run. We also overlay the corresponding Wilson binomial confidence interval ($p = 95\%$).

Figure 2: Plot of MidSearch vs. Tsallis-INF for varying $\Delta_{\min}$ parameter. Each algorithm was run for $N = 300$ trials on the $A_{\text{hard}}(\Delta_{\min}, 0.1)$ instance ($d = 1024$) with a budget of $50H_1$ samples.

than $d\,\Delta_{\min}^{-1} + \Delta_{\min}^{-2}$; MidSearch is no worse than $H_1 = d\,\Delta^{-2} + \Delta_{\min}^{-2}$. Based on these experiments, we conjecture that these bounds are indeed tight.

4 CONCLUSION

In this paper we generalized stochastic bandits and dueling bandits by studying the problem of identifying the PSNE in two-player zero-sum games, whenever it exists. We designed an algorithm that matches the lower bound from Zhou et al. (2017), up to log factors. To the best of our knowledge, this is also the first algorithm that achieves near-optimal guarantees for identifying Condorcet winner in dueling bandits. Our work leads to the following interesting problems. First, can one design a δ-PAC learner that has optimal bounds even in the log factors, similar to the extensive work in multi-armed bandits? Although PSNE is not guaranteed to exist in every matrix, a mixed strategy Nash Equilibrium is always guaranteed to exist. The optimal sample complexity to identify an ε-Nash Equilibrium in an interesting open problem (see Maiti et al. (2023a) for more discussion). Last-iterate convergence in matrix games has also gained significant interest in recent years (see Wei et al. (2021)). An important open problem here is to study last-iterate convergence in matrix games with noisy bandit feedback. Finally, can one design an algorithm for dueling bandits that simultaneously has optimal sample complexity for identifying the Condorcet winner, when it exists, and optimal regret guarantees with respect to the Condorcet winner (see Saha and Gaillard (2022) for the de nition and near-optimal result for regret), up to log factors.

ACKNOWLEDGEMENTS

This work was supported in part by NSF TRIPODS CCF Award #2023166 and a Northrop Grumman University Research Award.

References

Viktor Bengs, Róbert Busa-Fekete, Adil El Mesaoudi-Paul, and Eyke Hüllermeier. Preference-based online learning with dueling bandits: A survey. The Journal of Machine Learning Research, 22(1):278 385, 2021.

Lijie Chen, Jian Li, and Mingda Qiao. Towards instance optimal bounds for best arm identi cation. In Conference on Learning Theory, pages 535 592. PMLR, 2017.

Peter Duersch, Jörg Oechssler, and Burkhard C Schipper. Pure strategy equilibria in symmetric two-player zero-sum games. International Journal of Game Theory, 41:553 564, 2012.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Pac bounds for multi-armed bandit and markov decision processes. In Computational Learning Theory: 15th Annual Conference on Computational Learning Theory, COLT 2002 Sydney, Australia, July 8 10, 2002 Proceedings 15, pages 255 270. Springer, 2002.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. Games and Economic Behavior, 29(1-2):79 103, 1999.

Björn Haddenhorst, Viktor Bengs, Jasmin Brandt, and Eyke Hüllermeier. Testi cation of condorcet winners in dueling bandits. In Uncertainty in Arti cial Intelligence, pages 1195 1205. PMLR, 2021.

Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In Conference on Learning Theory, pages 423 439. PMLR, 2014.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In ICML, volume 12, pages 655 662, 2012.

Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In International conference on machine learning, pages 1238 1246. PMLR, 2013.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best arm identi cation in multi-armed bandit models. Journal of Machine Learning Research, 17:1 42, 2016.

Arnab Maiti, Kevin Jamieson, and Lillian Ratli. Instance-dependent sample complexity bounds for zero-sum matrix games. In International Conference on Arti cial Intelligence and Statistics, pages 9429 9469. PMLR, 2023a.

Arnab Maiti, Kevin Jamieson, and Lillian J Ratli. Logarithmic regret for matrix games against an adversary with noisy bandit feedback. arXiv preprint arXiv:2306.13233, 2023b.

Shie Mannor and John N Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. Journal of Machine Learning Research, 5(Jun):623 648, 2004.

Soheil Mohajer, Changho Suh, and Adel Elmahdy. Active learning for top-k rank aggregation from noisy comparisons. In International Conference on Machine Learning, pages 2488 2497. PMLR, 2017.

Gergely Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. Advances in Neural Information Processing Systems, 28, 2015.

Tadeusz Radzik. Saddle point theorems. International Journal of Game Theory, 20:23 32, 1991.

Wenbo Ren, Jia Liu, and Ness Shro. The sample complexity of best-k items selection from pairwise comparisons. In International Conference on Machine Learning, pages 8051 8072. PMLR, 2020.

Aadirupa Saha and Pierre Gaillard. Versatile dueling bandits: Best-of-both world analyses for learning from relative preferences. In International Conference on Machine Learning, pages 19011 19026. PMLR, 2022.

Lloyd Shapley. Some topics in two-person games. Advances in game theory, 52:1 29, 1964.

Chen-Yu Wei, Chung-Wei Lee, Mengxiao Zhang, and Haipeng Luo. Linear last-iterate convergence in constrained saddle-point optimization. In International Conference on Learning Representations, 2021.

Yichi Zhou, Jialian Li, and Jun Zhu. Identify the nash equilibrium in static games with random payo s. In International Conference on Machine Learning, pages 4160 4169. PMLR, 2017.

Julian Zimmert and Yevgeny Seldin. Tsallis-inf: An optimal algorithm for stochastic and adversarial bandits. The Journal of Machine Learning Research, 22(1):1310 1358, 2021.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with speci cation of all dependencies, including external libraries. [Not Applicable]
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]
3. For all gures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear de nition of the speci c measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Not Applicable]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identi able information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Technical Lemmas

Lemma 6 (sub-Gaussian tail bound)**.** Let $X_1, X_2, \ldots, X_n$ be i.i.d samples from a 1-sub-Gaussian distribution with mean μ. Then we have the following for any $0 < \delta < 1$:

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| \geq \sqrt{\frac{2\log(2/\delta)}{n}}\right] \leq \delta$$

Lemma 7 (Chernoff Bound)**.** Let $X_1, X_2, \ldots, X_n$ be i.i.d samples from a Bernoulli distribution with mean μ. Then we have the following for any $0 < \epsilon < 1$:

$$\mathbb{P}\left[\frac{1}{n}\sum_{i=1}^{n} X_i \geq (1+\epsilon)\mu\right] \leq e^{-\frac{n\mu\epsilon^2}{3}} \quad \text{and} \quad \mathbb{P}\left[\frac{1}{n}\sum_{i=1}^{n} X_i \leq (1-\epsilon)\mu\right] \leq e^{-\frac{n\mu\epsilon^2}{2}}$$

B Stochastic Multi-Armed Bandit

In stochastic Multi-Armed Bandit (MAB), there are n arms where each arm i is associated with a sub-distribution D_i whose mean is μ_i. When an arm i is pulled (or sampled) by a learner, it observes the random variable $X_i \sim D_i$. The arms with the highest mean is referred as the best-arm. In pure exploration, one aims to design a learner that aims to find the best arm with probability at least $1-\delta$ by pulling as few arms as possible.

Let $i_\star = \max_i \mu_i$ and $\Delta_i = \mu_{i_\star} - \mu_i$. Optimal bounds to identify the best-arm scales roughly as $\sum_{i \neq i_\star} \frac{1}{\Delta_i^2}\log(1/\delta)$. Now, an instance of MAB can be interpreted as a two-player zero-game on a matrix $A \in \mathbb{R}^{n \times 1}$ where $A_{i,1} = \mu_i$. It is easy to observe that the PSNE of A is $(i_\star, 1)$. It is easy to observe that our algorithm essentially becomes a variant of Sequential Halving for this special case of MAB and achieves the bound of $\sum_{i \neq i_\star} \frac{1}{\Delta_i^2}\log(1/\delta)$, up to log factors.

C RowMidVal Subroutine

Here, we define RowMidVal which solves the subproblem of finding a value near the median of means of n arms.

Consider $\varepsilon, \delta > 0$. Given a set of n arms A with means $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$, RowMidVal outputs a value $b \in [\mu_{n/4+1} - \varepsilon, \mu_{n/2} + \varepsilon]$ with probability $1-\delta$ by using at most $O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ samples. The proof of this claim follows from an analogous argument as that of ColMidVal.

Algorithm 4 RowMidVal (A, ε, δ)

- $\ell \leftarrow \lceil 14\log(\frac{1}{\delta})\rceil$, $\delta_1 \leftarrow 0.05$, $\delta_2 \leftarrow 0.05$, $k \leftarrow \lceil 108\log(\frac{4}{\delta_2})\rceil$ and $z \leftarrow \frac{k}{3} + 1$.
- **for** $i = 1$ to ℓ **do**
  - $B_i \leftarrow$ k arms sampled independently and uniformly at random from A.
  - Sample each arm $j \in B_i$ for $\frac{2\log(\frac{2k}{\delta_1})}{\varepsilon^2}$ times and compute the empirical mean $\hat{b}_j$.
  - $v_i \leftarrow$ z-th lowest value in $\{\hat{b}_j : j \in B_i\}$
- **end for**
- **return** the median of $\{v_i : i \in [\ell]\}$

D Verification algorithm

In this section, we focus on the following subproblem:

Subproblem: Given an entry $(\hat{i}, \hat{j})$ and a parameter Δ, do the following with probability $1-\delta$:

- If $\Delta \leq \Delta_g$ and $(\hat{i}, \hat{j}) = (i_\star, j_\star)$, then $(\hat{i}, \hat{j})$ is **accepted** as the PSNE of A

- If $(\hat{i}, \hat{j}) \neq (i_\star; j_\star)$, then $(\hat{i}, \hat{j})$ is not accepted as the PSNE of A.

Consider a stochastic multi-armed bandit instance with means $_1 \quad _2 \quad ::: \quad _n$. Let $H_0 = \sum_{i=2}^{n} \frac{1}{(_1 \quad _i)^2}$. Let Alg be a best arm identi cation algorithm that does the following:

- If the best arm is unique, Alg nds the best arm by using at most $c_1 \; H_0 \; \log(\frac{\log H_0}{})$ samples with probability at least $1 \quad =2$ where c_1 is an absolute constant.
- If the best arm is not unique, Alg does not terminate with probability at least $1 \quad =2$.

See Karnin et al. (2013) for one such algorithm. We use Alg to design a procedure called Verify that achieves the desired guarantees for the above the sub-problem.

Algorithm 5 Verify $(A; \hat{i}, \hat{j}; \; ; \;)$

1: Let $_1$ be a multi-armed bandit instance with means $_i = A_{i;\hat{j}}$ for all $i \; 2 \; [n]$
2: Let $_2$ be a multi-armed bandit instance with means $_j = \quad A_{\hat{i};j}$ for all $j \; 2 \; [m]$
3: $H_\star \quad \frac{n+m \quad 2}{2}$
4: Run Alg on the bandit instance $_1$ and terminate it after it takes $c_1 \; H_\star \; \log(\frac{\log H_\star}{})$ samples.
5: If Alg terminates without returning arm $\hat{i}$ as the best arm, then return not accepted .
6: Run Alg on the bandit instance $_2$ and terminate it after it takes $c_1 \; H_\star \; \log(\frac{\log H_\star}{})$ samples.
7: If Alg terminates without returning arm $\hat{j}$ as the best arm, then return not accepted .
8: Return accepted .

We rst establish the sample complexity of Verify in the following lemma.

Lemma 8. Verify takes at most $2c_1 \; H_\star \; \log(\frac{\log H_\star}{})$ samples before terminating

Proof. The lemma trivially follows from lines 4 and 6 of Verify . □

We prove the following proposition that is useful to establish the correctness of our algorithm

Proposition 3. If $(\hat{i}, \hat{j}) \neq (i_\star; j_\star)$, then either $A_{\hat{i},\hat{j}} < A_{i_\star;\hat{j}}$ or $A_{\hat{i},\hat{j}} > A_{\hat{i};j_\star}$.

Proof. Let us assume that $A_{\hat{i},\hat{j}} \quad A_{i_\star;\hat{j}}$ and $A_{\hat{i},\hat{j}} \quad A_{\hat{i};j_\star}$. This implies that $A_{i_\star;\hat{j}} \quad A_{\hat{i};j_\star}$ which is a contradiction as $A_{\hat{i};j_\star} < A_{i_\star;j_\star} < A_{i_\star;\hat{j}}$. □

The following lemma establishes the correctness of Verify .

Lemma 9. Verify does the following with probability $1 \quad$:

- If $\quad _g$ and $(\hat{i}, \hat{j}) = (i_\star; j_\star)$, then $(\hat{i}, \hat{j})$ is accepted as the PSNE of A
- If $(\hat{i}, \hat{j}) \neq (i_\star; j_\star)$, then $(\hat{i}, \hat{j})$ is not accepted as the PSNE of A.

Proof. Let us rst consider the case when $\quad _g$ and $(\hat{i}, \hat{j}) = (i_\star; j_\star)$. Recall the bandit instances $_1$ and $_2$. Let $H_x = \sum_{i \, 2 \, [n] nf i_\star g} \frac{1}{(A_{i_\star;j_\star} \quad A_{i;j_\star})^2}$ and $H_y = \sum_{j \, 2 \, [m] nf j_\star g} \frac{1}{(A_{i_\star;j} \quad A_{i_\star;j_\star})^2}$. Observe that $H_x + H_y = \frac{n+m \quad 2}{^2_g}$ $\frac{n+m \quad 2}{2} = H_\star$. Now observe that $i_\star$ is the best arm of the instance $_1$ and $j_\star$ is the best arm of the instance $_2$. Hence, Alg does the following with probability $1 \quad$:

- For the instance $_1$, Alg returns $i_\star$ as the best arm after taking at most $c_1 \; H_x \; \log(\frac{\log H_x}{})$ samples.
- For the instance $_2$, Alg returns $i_\star$ as the best arm after taking at most $c_1 \; H_y \; \log(\frac{\log H_y}{})$ samples.

Hence, we accept $(\hat{i}, \hat{j})$ with probability at least $1 \quad$.

Next, we consider the case when $(\hat{i}, \hat{j}) \neq (i_\star; j_\star)$. Due to Proposition 3, either $\hat{i}$ is not the best arm of the instance $_1$ or $\hat{j}$ is not the best of arm of the instance $_2$. Hence, with probability $1 \quad$, $(\hat{i}, \hat{j})$ is not accepted . □

E Omitted Calculations

Here we state the calculations omitted from various proofs in the main body.

Omitted Calculations from the proof of Lemma 4. Recall that we generate the set B_i by sampling k arms independently and uniformly at random from A where $k = d108\log(\frac{4}{\ {}^2})e = 474$. Hence, $\frac{k}{3} + 1$ is an integer equal to 159 and $\frac{7k}{20} > 165$. Also recall that $E[n_{1;i}] = E[n_{2;i}] = k{=}4$. First we have the following due to the Cherno bound:

$$
\begin{aligned}
P\ n_{1;i}\quad \frac{7k}{40} &= P\ n_{1;i}\quad 1\ \ \frac{3}{10}\quad \frac{k}{4} \\
&= P\ n_{1;i}\quad 1\ \ \frac{3}{10}\quad E[n_{1;i}] \\
&\quad \exp\ \frac{E[n_{1;i}]\ \ (0{:}3)^2}{2} \\
&= \exp\ \frac{k\ \ 0{:}09}{4\ \ 2} \\
&\quad \exp\ \frac{108\log(\frac{4}{\ {}^2})\ \ 0{:}09}{4\ \ 2}! \\
&= \exp\ \ 1{:}215\ \log\ \frac{4}{\ {}^2} \\
&< \exp\ \ \log\ \frac{4}{\ {}^2} = \frac{\ {}^2}{4}
\end{aligned}
$$

Next we have the following due to the following due to the Cherno bound

$$
\begin{aligned}
P\ n_{1;i}\quad \frac{k}{3} &= P\ n_{1;i}\quad 1 + \frac{1}{3}\quad \frac{k}{4} \\
&= P\ n_{1;i}\quad 1 + \frac{1}{3}\quad E[n_{1;i}] \\
&\quad \exp\ \frac{E[n_{1;i}]\ \ (1{=}3)^2}{3} \\
&= \exp\ \frac{k}{4\ \ 3\ \ 9} \\
&\quad \exp\ \frac{108\log(\frac{4}{\ {}^2})}{108}! \\
&= \exp\ \ \log\ \frac{4}{\ {}^2} = \frac{\ {}^2}{4}
\end{aligned}
$$

Similarly, we can show using Cherno bound that $\frac{7k}{40}\ \ j\ n_{2;i}\,j\ \ \frac{k}{3}$ occurs with probability at least $1\ \ \frac{\ {}^2}{2}$.

Recall the de nition of Z. Also recall that $` = d14\log(\frac{1}{\ })e$ and $E[Z]\ \ \frac{9`}{10}$. Now we have the following due to the

Cherno bound:

$$
\begin{aligned}
P[Z \quad 0{:}54`] = P \quad n_{1;i} \quad 1 \quad \frac{2}{5} \quad \frac{9`}{10} \\
P \quad n_{1;i} \quad 1 \quad \frac{2}{5} \quad E[Z] \\
\exp \quad \frac{E[Z] \quad (0{:}4)^2}{2} \\
\exp \quad \frac{9` \quad 0{:}16}{10 \quad 2} \\
\exp \quad \frac{9 \quad 14 \log(\frac{1}{}) \quad 0{:}16}{10 \quad 2} \\
= \exp \quad 1{:}008 \quad \log \quad \frac{1}{} \\
< \exp \quad \log \quad \frac{1}{} \quad =
\end{aligned}
$$

□

Omitted calculations in the proof of Theorem 1. Using Lemma 1 we get that at any iteration t, MidSearch uses at most $c_1 \quad \frac{n+m \quad 2}{\frac{2}{t}} \quad \log(\frac{nm}{t}) \quad \log(nm)$ samples where c_1 is an absolute constant. Using Lemma 5 we get that at any iteration t, Verify uses at most $c_2 \quad \frac{n+m \quad 2}{\frac{2}{t}} \quad \log\left(\frac{\log \quad \frac{n+m}{\frac{2}{t}}}{t}\right)$ where c_2 is absolute constant. As $n; m \quad 1$, we have $n \quad 1 \quad \frac{m}{2m \quad 1}$ which implies $n + m \quad 2nm$. Now we have the following:

$$
\begin{aligned}
\log\left(\frac{\log \quad \frac{n+m}{\frac{2}{t}}}{t}\right) \quad & \log\left(\frac{\log \quad \frac{2nm}{\frac{2}{t}}}{t}\right) \\
& \log\left(\frac{\log(2nm) + 2\log \quad \frac{1}{t}}{t}\right) \\
& \log\left(\frac{2nm + 2\,nm \log \quad \frac{1}{t}}{t}\right) \qquad \text{(as } \log(x) < x\text{)} \\
& = \log\left(\frac{2nm \log \quad \frac{e}{t}}{t}\right)
\end{aligned}
$$

Now the total number of samples taken at iteration t is upper bounded as follows:

$$
\begin{aligned}
= c_1 \quad \frac{n+m \quad 2}{\frac{2}{t}} \quad \log \quad \frac{nm}{t} \quad \log(nm) + c_2 \quad \frac{n+m \quad 2}{\frac{2}{t}} \quad \log\left(\frac{\log \quad \frac{n+m}{\frac{2}{t}}}{t}\right) \\
c_1 \quad \frac{n+m \quad 2}{\frac{2}{t}} \quad \log \quad \frac{nm}{t} \quad \log(nm) + c_2 \quad \frac{n+m \quad 2}{\frac{2}{t}} \quad \log\left(\frac{2nm \log \quad \frac{e}{t}}{t}\right) \\
c \quad \frac{n+m \quad 2}{\frac{2}{t}} \quad \log\left(\frac{nm \log \quad \frac{1}{t}}{t}\right) \quad \log(nm)
\end{aligned}
$$

where c is an absolute constant. □

F Additional Plots

In this section, we perform few additional experiments. We consider the hard instance $A_{\text{hard}}(\;_{\min};\;)$ described in Section 3 with $d = 128; 256; 512$. As shown in Section 3 both MidSearch and Tsallis-INF perform far better than Exp3-IX, LUCB-G and uniform random sampling. Hence, we compare the performance of MidSearch and Tsallis-INF here by measuring the probability of identifying the PSNE. First in Figure 3, we x $= 0{:}1$ and vary $_{\min}$ and compare the performance of MidSearch and Tsallis-INF. We observe that MidSearch has a better performance than Tsallis-INF. Next in Figure 4, we x $_{\min} = 0{:}01$ and vary and compare the performance of MidSearch and Tsallis-INF. We observe that MidSearch has a better performance than Tsallis-INF. Finally in Figure 5, we vary both and $_{\min}$ and compare the performance of MidSearch and Tsallis-INF. We observe that MidSearch has a better performance than Tsallis-INF.

Figure 3: Plot of MidSearch vs. Tsallis-INF for varying $_{\min}$ parameter. Each algorithm was run for $N = 300$ trials on the $A_{\text{hard}}(\;_{\min}; 0{:}1)$ instance with a budget of $50H_1$ samples.

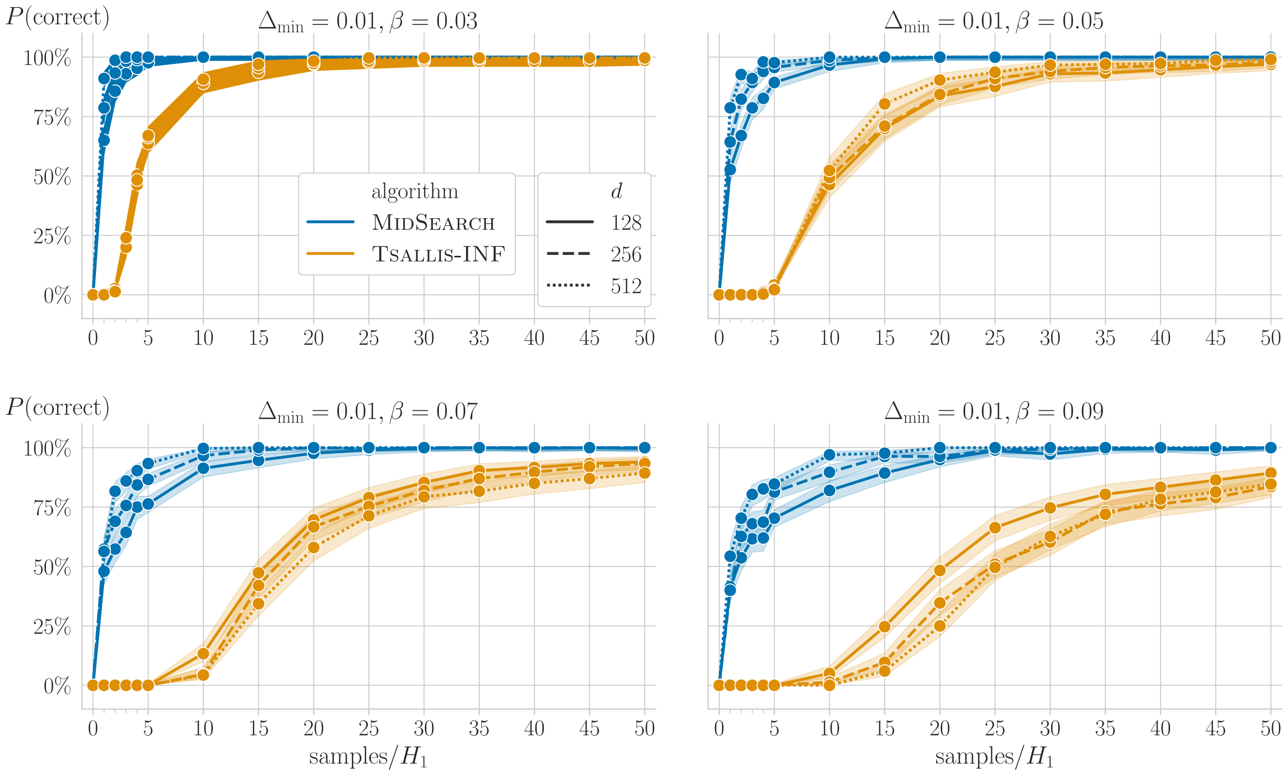

Figure 4: Plot of MidSearch vs. Tsallis-INF for varying β parameter. Each algorithm was run for $N = 300$ trials on the $A_{\mathsf{hard}}(0.01, \beta)$ instance with a budget of $50H_1$ samples.

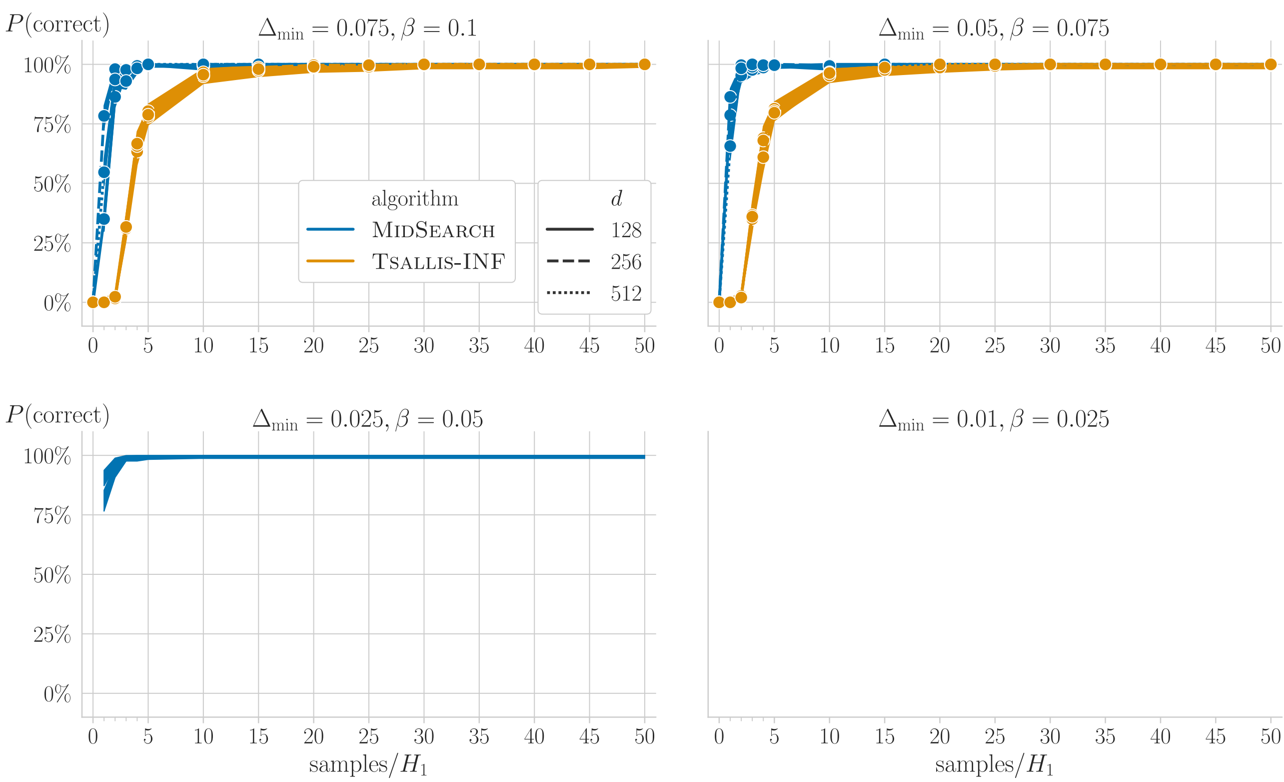

Figure 5: Plot of MidSearch vs. Tsallis-INF for varying both $\Delta_{\min}$ and β parameters. Each algorithm was run for $N = 300$ trials on the $A_{\mathsf{hard}}(\Delta_{\min}, \beta)$ instance with a budget of $50H_1$ samples.