Two-Player Zero-Sum Games with Bandit Feedback

Elif Yılmaz
University of Neuchâtel
Neuchâtel, Switzerland
elif.yilmaz@unine.ch

Christos Dimitrakakis
University of Neuchâtel
Neuchâtel, Switzerland
christos.dimitrakakis@unine.ch

Abstract

We study a two-player zero-sum game (TPZSG) in which the row player aims to maximize their payoff against an adversarial column player, under an unknown payoff matrix estimated through bandit feedback. We propose and analyze two algorithms: ETC-TPZSG, which directly applies Explore-Then-Commit (ETC) to the TPZSG setting and ETC-TPZSG-AE, which improves upon it by incorporating an action pair elimination (AE) strategy that leverages the ε-Nash Equilibrium property to efficiently select the optimal action pair. Our objective is to demonstrate the applicability of ETC in a TPZSG setting by focusing on learning pure strategy Nash Equilibrium. A key contribution of our work is a derivation of instance-dependent upper bounds on the expected regret for both algorithms, which has received limited attention in the literature on zero-sum games. Particularly, after T rounds, we achieve an instance-dependent regret upper bounds of $O(\Delta + \sqrt{T})$ for ETC-TPZSG and $O(\frac{\log(T\Delta^2)}{\Delta})$ for ETC-TPZSG-AE, where Δ denotes the suboptimality gap. Therefore, our results indicate that ETC-based algorithms perform effectively in adversarial game settings, achieving regret bounds comparable to existing methods while providing insights through instance-dependent analysis.

1 Introduction

This paper focuses on finite two-player zero-sum games (TPZSGs), where two players interact in an adversarial setting with strictly opposing objectives, as first analyzed in [31, 32]. These games play a foundational role in game theory and online learning, capturing adversarial interactions in various applications such as economic competition and adversarial machine learning. In such settings, players repeatedly select actions from finite sets, with payoffs determined by a fixed but unknown game matrix.

Studies on learning in games typically use either *full information feedback* where players observe the opponent's actions or the entire payoff matrix after each round or *partial feedback*, also called bandit feedback, where each player observes only their own payoffs without access to the opponent's strategy or the full matrix [7]. In this paper, we study a TPZSG setting with bandit feedback, where players observe each other's actions, and propose two algorithms, ETC-TPZSG and ETC-TPZSG-AE, which minimize regret by learning pure strategies for both players through balancing of exploration and exploitation.

Several existing studies have advanced our understanding of how players learn to play optimally in zero-sum games with bandit feedback. The variant of UCB algorithm for adversarial games under bandit feedback have been analyzed using worst-case regret bounds in [27]. In particular, worst-case analyses lead to all games as equally hard by ignoring the specific structure of the game. Instance-dependent analysis adapts the regret guarantees to the specific properties of a game to provide more practical performance guarantees. More recently, instance-dependent regret bounds

18th European Workshop on Reinforcement Learning (EWRL 2025).

have been investigated for the Tsallis-INF algorithm in TPZSGs [18]. However an understanding of instance-dependent regret in TPZSGs under bandit feedback remains incomplete.

Contributions. This paper investigates the effectiveness of the *Explore-Then-Commit* (ETC) algorithm and extensions in the context of TPZSGs. While ETC has been widely studied in standard multi-armed bandit (MAB) problems, there has been no prior work specifically analyzing its performance to adversarial game settings, which makes this the first work to provide its detailed analysis in such a setting. A key motivation for studying ETC is its algorithmic simplicity allowing for a clear structure.

In particular, we propose two algorithms that adapt the ETC approach to TPZSG setting and present their instance-dependent regret analyses. The first one, ETC-TPZSG, follows the classical ETC framework by uniformly selecting actions for a fixed number of steps and then committing to a pure strategy, achieves an upper bound of $\tilde{O}(\Delta + \sqrt{T})$. The second algorithm, ETC-TPZSG-AE, incorporates an adaptive elimination approach, similar to [6], by sequentially eliminating actions which, with high probability, are not part of an ε-Nash Equilibrium (ε-NE), gives us an upper regret bound of $O(\frac{\log(T\Delta^2)}{\Delta})$. Intuitively, this design is expected to converge more rapidly to the optimal strategy by efficiently reducing uncertainty and narrowing the action spaces of the players. Consequently, despite the simplicity of ETC framework, our results demonstrate that ETC-based approaches can be highly effective in adversarial settings. In both cases, the regret bounds depend on a suitable Δ notion, which varies depending on the type of the regret we are analyzing.

Paper structure. This paper is organized as follows. Section 2 reviews related work. Section 3 introduces preliminaries and formalizes the two-player zero-sum game (TPZSG) setting. In Section 4, we present the ETC in a TPZSG setting, ETC-TPZSG, and analyze its regret. Section 5 introduces an elimination based algorithm, ETC-TPZSG-AE, that leverages the ε-NE property, and provides its regret analysis. Section 6 offers empirical results validating their theoretical performances. Finally, Section 7 discusses our conclusions and future research directions.

2 Related Work

The multi-armed bandit (MAB) problem, has been introduced by [30] and originally described by [29], has been widely explored and has become a crucial framework for online decision-making under uncertainty, as explored in works such as [3, 4, 5, 6, 8, 9]. The MAB problem models scenarios in which a player must repeatedly choose from a set of actions to maximize their own rewards while balancing exploration-exploitation tradeoff, has formalized by [19] and further explored in studies such as [2, 21].

A central challenge is identifying the action with the highest expected reward while minimizing suboptimal choices, requiring efficient exploration and statistical confidence. Several approaches guarantee low error probabilities within finite trials, aiming to minimize regret [1, 16]. Since the player has no prior knowledge of expected rewards, a dedicated exploration phase is essential, has motivated numerous studies such as [10, 26].

The Explore-Then-Commit (ETC) algorithm, which has been introduced by [28], is one of the fundamental approaches in the MAB setting. It explores each action a fixed number of times, then commits to the best-performing action for the remaining rounds. Due to its simplicity and effectiveness, variants of the ETC algorithm are widely used in various decision-making scenarios [17, 34].

Zero-sum games, are a fundamental concept in game theory. They describe competitive scenarios where one player's gain exactly equals the other's loss. First highlighted by [31, 32], their theory and applications have been extensively explored [8, 12, 33]. The Minimax Theorem in [31, 32] guarantees equilibrium existence in TPZSGs, but finding equilibria becomes significantly harder when the payoff matrix is unknown [13, 14, 23, 35].

A more general concept are Nash Equilibria [24, 25]. While these are identical to the minimax equilibria in TZPSGs, they are not as easy to compute in general games [14]. The concept of ε-Nash Equilibrium provides an approximate solution where each player's strategy is within an ε tolerance level. [15] used the ε-NE concept to efficiently approximate equilibrium in two-player games.

In a zero-sum game setting with bandit feedback, players engage in repeated interactions without full knowledge of the payoff matrix, making it essential to estimate the unknown payoffs by previous observations. Various approaches, such as UCB [27] and Tsallis-INF [18], have been proposed to efficiently estimate the unknown payoff matrix while minimizing regret. Learning the game matrix is crucial for converging to equilibrium strategies and ensuring optimal decision-making in adversarial environments. [27] focuses on worst-case regret analysis, while [18] provides an instance-dependent regret analysis which achieves an upper bound of $O(\frac{\log T}{\Delta})$ in the case of a pure strategy equilibrium. Additionally, [22] investigates instance-dependent sample complexity bounds for zero-sum games under bandit feedback while [11] examines convergence rates to NE.

3 Preliminaries

In this section, we define the basic concept of a TPZSG such as payoff matrix, the regret definitions we focus on and the gaps Δ that we use for the analysis.

In a TPZSG, the goals of the players are strictly opposed, which implies that any gain by one player corresponds to an equal loss by the other. This framework provides a fundamental model for adversarial interactions where strategic decision-making under competition is central. The outcome of such a game is fully determined by the strategies chosen by both players and the goal of each player is to maximize their own payoff while minimizing that of their opponent.

Notation. Throughout this paper, $O(\cdot)$ denotes an upper bound up to constant factors.

3.1 Two-Player Zero-Sum Games

In a two-player zero-sum game, consisting of a row (x) player and a column (y) player , the row player aims to maximize their own payoff while the column player attempts to take an action to minimize this payoff.

S_x and S_y are the action sets of the row player and column player, respectively. Thus, the action space of the game is $S_A = S_x \times S_y$, so that the action pairs $(i, j) \in S_A$ where $i \in S_x, j \in S_y$. We also define $m = |S_x|$ as the number of actions for the row player, and $l = |S_y|$ as the number of actions for the column player, while finally $N = ml = |S_A|$ is the total number of action pairs. The game itself is defined through a game matrix A, so that if the players choose the pair (i, j) then the row player obtains a payoff of $A(i, j)$ and the column player $-A(i, j)$.

The row player has a strategy (i.e. a probability distribution) p to select an action $i \in S_x$, while the column player selects an action $j \in S_y$ according to a strategy q.

Definition 3.1. *A pair of mixed strategies (p^*, q^*) is a Nash equilibrium (NE) if for all strategies p and q, it satisfies*

$$V^* \geq p^\top A q^*, \qquad V^* \leq {p^*}^\top A q \tag{1}$$

where $V^* = {p^*}^\top A q^*$ is the value of the game [24]. It means that p^* and q^* are optimal strategies for row and column players, respectively.

There always exists an optimal mixed strategy for both players that guarantees the best possible outcome by the Minimax Theorem [31, 32]. When a mixed strategy assigns a probability of one to a single action and zero to all others, it is referred to as a *pure* strategy. In our setting, we focus on a TPZSG where both players adopt pure strategies.

Definition 3.2 (Pure Nash Equilibria)**.** *A pair (i^*, j^*) is a pure NE if:*

$$i^* = \operatorname*{argmax}_{i \in S_x} \min_{j \in S_y} A(i, j) \quad \textit{and} \quad j^* = \operatorname*{argmin}_{j \in S_y} \max_{i \in S_x} A(i, j),$$

or equivalently

$$A(i, j^*) \leq A(i^*, j^*) \leq A(i^*, j), \qquad \forall i \in S_x, j \in S_y. \tag{2}$$

The condition in (2) ensures that no player has an incentive to deviate to another action. We use $V^* = A(i^*, j^*)$ to denote the value of the game.

3.2 Games with Bandit Feedback

The game proceeds in rounds. At time t, the row player draws an action $i_t \in S_x$ from a strategy p_t, and the column player draws $j_t \in S_y$ from a strategy q_t. The players then observe a noisy payoff r_t with expected value $\mathbb{E}[r_t \mid i_t, j_t] = A(i_t, j_t)$, with the maximizing player obtaining r_t and the minimizing player $-r_t$. We assume that both players observe each other's action.

Since the algorithms do not know A, we use the notation $\hat{A}(i,j)$ to denote the estimated payoff when the action pair (i,j) is played. We express the action pair played in round t by (i^t, j^t). After round t, we estimate each entry of the estimated game matrix by computing the average payoff for each action pair (i,j) as

$$\hat{A}_t(i,j) = \frac{1}{n_{ij,t}} \sum_{s=1}^{t} \mathbb{I}\{(i^s, j^s) = (i,j)\} r_s \tag{3}$$

where $n_{ij,t}$ is the number of times the action pair (i,j) played until round t, r_s's are independently and identically distributed (iid) σ-subgaussian payoffs. We assume that the estimator is unbiased, meaning that $\mathbb{E}[\hat{A}] = A$.

3.3 Gap Notions

To quantify how close a chosen action pair is to the NE, we define suboptimality gaps based on deviations from the equilibrium and the best responses of the players. These gaps capture the extent to which each player could improve their outcome by deviating from their current strategy. For any action $i \in S_x, j \in S_y$, we define the following suboptimality gaps:

$$\Delta_{ij}^{\max} = \max_{i' \in S_x} A(i', j) - A(i,j) \tag{4}$$

$$\Delta_{ij}^{\min} = A(i,j) - \min_{j' \in S_y} A(i, j') \tag{5}$$

$$\Delta_{ij} = \Delta_{ij}^{\max} + \Delta_{ij}^{\min} \tag{6}$$

$$\Delta_{ij}^{*} = A(i^*, j^*) - A(i,j) = V^* - A(i,j) \tag{7}$$

We note that $\Delta_{ij}^{\max} \geq 0$ and $\Delta_{ij}^{\min} \geq 0$, thus $\Delta_{ij} \geq 0$ for any action pair (i,j). However, Δ_{ij}^{*} might be negative.

Using the NE property in (2), for all $i \in S_x$ and $j \in S_y$ we can write the following:

$$A(i^*, j^*) \leq A(i^*, j) \leq \max_{i' \in S_x} A(i', j) \implies A(i^*, j^*) - A(i,j) \leq \max_{i' \in S_x} A(i', j) - A(i,j) \tag{8}$$

$$A(i^*, j^*) \geq A(i, j^*) \geq \min_{j' \in S_y} A(i, j') \implies A(i^*, j^*) - A(i,j) \geq \min_{j' \in S_y} A(i, j') - A(i,j) \tag{9}$$

Then, the inequality (8) implies that $\Delta_{ij}^{*} \leq \Delta_{ij}^{\max}$ and similarly from (9), we have $\Delta_{ij}^{*} \geq -\Delta_{ij}^{\min}$. Therefore, we have $-\Delta_{ij}^{\min} \leq \Delta_{ij}^{*} \leq \Delta_{ij}^{\max}$. On the other hand, since $\Delta_{ij}^{\min} \geq 0$, we can write

$$\Delta_{ij}^{*} \leq \Delta_{ij}^{\max} + \Delta_{ij}^{\min} = \Delta_{ij}. \tag{10}$$

3.4 Regret Notions

To characterize the performance of our algorithms, we consider several regret notions. Let begin with the following formulations, which are the external regret of the max and min player respectively:

$$R_T^{\max} = \max_{i \in S_x} \mathbb{E}\Big[\sum_{t=1}^{T} A(i, j^t) - A(i^t, j^t)\Big]$$

$$R_T^{\min} = \max_{j \in S_y} \mathbb{E}\Big[\sum_{t=1}^{T} A(i^t, j^t) - A(i^t, j)\Big]$$

where i^t and j^t are the actions selected by the row and column player in round t, respectively. $R_T^{\max}$ represents the expected loss due to not selecting the best possible action of row player against column

player's action j while $R_T^{\min}$ captures the expected regret from not choosing the column player's best action for a given action i.

We can now rewrite the regret definitions in terms of the suboptimality gaps:

$$R_T^{\max} = \sum_{(i,j)\in S_A} \Delta_{ij}^{\max} \mathbb{E}[n_{ij,T}], \qquad R_T^{\min} = \sum_{(i,j)\in S_A} \Delta_{ij}^{\min} \mathbb{E}[n_{ij,T}], \tag{11}$$

where $\mathbb{E}[n_{ij,T}]$ denotes the expected number of times the action pair (i, j) is played over T rounds.

In this paper, we analyze two regret notions. The external regret and the Nash regret.

Definition 3.3 (External regret)**.** *The (combined) external regret is given by the following:*

$$R_T = R_T^{max} + R_T^{min} = \sum_{(i,j)\in S_A} \Delta_{ij} \mathbb{E}[n_{ij,T}] \tag{12}$$

External regret quantifies how much worse the players have performed compared to their best actions in hindsight. An alternative regret notion is the Nash regret, which measures the deviation of value from the Nash equilibrium:

Definition 3.4 (Nash regret)**.** *The Nash regret is expressed as*

$$R_T^* = \sum_{(i,j)\in S_A} \Delta_{ij}^* \mathbb{E}[n_{ij,T}]. \tag{13}$$

The Nash regret, as defined here, can be negative, since the minimizing player might play suboptimally. However, if we are able to bound the Nash regret for the maximizing player, then by symmetry, we can also bound it for the minimizing player, thus effectively bounding its absolute value.

4 ETC-TPZSG Algorithm and Regret Analysis

In this section, we extend ETC algorithm to a two-person zero-sum game setting as ETC-TPZSG. Each player selects action from their corresponding finite action sets, which are known. The game is repeated over given time horizon T, and the true payoff matrix A is unknown to both players. Instead, they receive observations of the payoffs based on their chosen actions, which are iid subgaussian random variables.

Let k represent the number of times to explore each action pair $(i, j) \in S_A$. During the exploration phase, each player samples actions to estimate the expected payoffs, which enables us to construct an empirical payoff matrix using the observed payoffs. Then, in commit phase, they play according to the pure NE strategy derived from the estimated payoff matrix. After presenting the ETC-TPZSG algorithm, finally, we analyze its expected regret, measuring the performance due to limited information and suboptimal action pairs during the exploration phase.

As a result, the unknown payoff matrix is approximated by the ETC approach, as in [20] for a bandit setting. The ETC algorithm consists of two phases: an exploration phase, where the player randomly samples actions to gather information with given exploration time k, and a commit phase, where the player selects the empirically best action based on the gathered data. Algorithm 1 presents the implementation of ETC-TPZSG, where each player only observes the payoffs of the action pair they choose during the exploration period, then in the committing phase the algorithm converges to an optimal action pair which refers to the pure NE.

Applying ETC in a zero-sum game setting presents some challenges compared to the standard MAB scenario. In bandit problems, after the exploration phase, a player simply commits to the arm with the highest estimated reward, which is typically effective if exploration is sufficient. However, in a zero-sum game with the pure NE, the objective is not to identify an action with high reward but to find a possibly true NE. Inadequate exploration can lead to poor estimates of payoffs, making it difficult to correctly identify such an equilibrium. In fact, due to inaccurate estimates, a pure NE might not even appear to exist, even when one does in the true game. As a result, the design of ETC approach in zero-sum game setting requires careful balancing of exploration phase.

Algorithm 1 ETC-TPZSG

Input:
S_A: set of action pairs in the game matrix A ▷ S_A includes action pairs (i, j)
m: number of actions for row player
l: number of actions for column player
k: exploration time
T: time horizon ▷ $1 \leq mlk \leq T$
σ^2: subgaussian variance factor
Initialize: $\hat{A} = [0]^{m \times l}$ ▷ $\hat{A}$ is estimated game matrix
In round $t = 0, 1, 2, ..., mlk$; ▷ Exploration Phase
    Explore each action pair (i, j) in S_A k times
    Update $\hat{A}$ using 3
In round $t = mlk + 1, mlk + 2, ..., T$; ▷ Committing Phase
    Play an equilibrium (i^*, j^*) satisfying the following property:

$$\hat{A}(i, j^*) \leq \hat{A}(i^*, j^*) \leq \hat{A}(i^*, j)$$

Theorem 4.1. *The Nash regret of Algorithm 1, when interacting with σ-subgaussian payoffs, is upper bounded as follows:*

$$R_T^* \leq k \sum_{(i,j) \in S_A} \Delta_{ij}^* + (T - Nk) \sum_{(i,j) \in S_A} \Delta_{ij}^* \exp\Big(- \frac{k \Delta_{ij}^2}{16 \sigma^2} \Big) \tag{14}$$

where k is the exploration time per action pair and N is the total number of action pairs.

We provide the proof in Appendix B. The regret comprises two main components, the exploration phase and the loss due to misidentifying the NE. Thus, it is crucial to choose an appropriate exploration time k that balances sufficient exploration and efficient exploitation. Choosing k too large leads to unnecessary exploration while setting it too small increases the risk of making suboptimal decisions. Hence, careful tuning of k is essential to minimize overall regret.

Let assume there are two action pairs (i_1, j) and (i_2, j) in the game such that row player has two actions and column player has one action. In addition, let suppose the NE is at (i_1, j). Then, we have $\Delta = A(i_1, j) - A(i_2, j)$ and we can write the regret simply as

$$R_T^* \leq k\Delta + (T - 2k)\Delta \exp\Big(- \frac{k\Delta^2}{16\sigma^2} \Big) \leq k\Delta + T\Delta \exp\Big(- \frac{k\Delta^2}{16\sigma^2} \Big). \tag{15}$$

Taking the first derivative with respect to k and solve it, we get the following for the exploration time:

$$k = \max\Big\{1, \Big\lceil \frac{16\sigma^2}{\Delta^2} \ln \frac{\Delta^2 T}{16\sigma^2} \Big\rceil \Big\} \tag{16}$$

If Δ is known, we can easily find the necessary exploration time. Therefore, when we put it into the regret bound, it becomes

$$R_T^* \leq \min\Big\{ T\Delta, \Delta + \frac{16\sigma^2}{\Delta} \Big(1 + \max\Big\{ 0, \ln \frac{\Delta^2 T}{16\sigma^2} \Big\} \Big) \Big\} \tag{17}$$

where the first term $T\Delta$ is worst-case regret and for the other, we combine the regret we obtain in (15) with the value of exploration time in (16). Then, focusing on $\max\Big\{0, \ln \frac{\Delta^2 T}{16\sigma^2}\Big\}$, we obtain that $\Delta = \frac{4\sigma}{\sqrt{T}}$ is a critical point. If we put it in the regret bound (17), we have the following:

$$R_T^* \leq \Delta + c\sqrt{T}$$

where $c > 0$ is a some constant. If $\Delta \leq 1$, it becomes

$$R_T^* \leq 1 + c\sqrt{T}. \tag{18}$$

Bounds like (18) are referred to as instance-independent since it only depends on the time horizon T not on the game instance. In the zero-sum game setting, our analysis shows that we get a regret bound comparable to the standard bandit setting in [20], indicating that the ETC-based algorithm performs effectively in adversarial environments and aligns with known results in the literature.

5 ETC-TPZSG-AE Algorithm and Regret Analysis

If a pure NE exists at (i^*, j^*), then it must satisfy the condition given in (2). For the algorithm in the section, we make use of the concept of ε-Nash Equilibrium (ε-NE), which approximately satisfies the standard NE condition. Specifically, an ε-NE satisfies the NE conditions within a tolerance level ε, making it a suitable criterion in learning settings where exact equilibrium identification is hard.

Based on this criterion, we introduce an elimination strategy in Algorithm 2, inspired by [6], to reduce unnecessary exploration of clearly suboptimal action pairs. Specifically, action pairs that do not satisfy the ε-NE property are eliminated from further play, which enables a more efficient learning process by concentrating exploration on approximately optimal action pairs.

Definition 5.1. *The action pair (i^*, j^*) is an ε-Nash Equilibrium (ε-NE) if the following condition holds:*

$$A(i, j^*) - \varepsilon \leq A(i^*, j^*) \leq A(i^*, j) + \varepsilon, \quad \forall i \in S_x, j \in S_y. \tag{19}$$

For the exploration time k of the algorithm, let just go back to the regret bound in (14) and assume that there exists some Δ^* and Δ such that $\Delta^* \geq \Delta^*_{ij}$ and $\Delta \leq \Delta_{ij}$ for all $(i, j) \in S_A$. Then, when we put them into the regret bound and take its derivative with respect to k, we obtain a similar result as in (16); thus, as the exploration time, clearly we can use the following:

$$k = \left\lceil \frac{16\sigma^2}{\Delta^2} \ln\left(\frac{\Delta^2 T}{16\sigma^2}\right) \right\rceil \tag{20}$$

Since the true suboptimality gaps Δ are unknown, the algorithm adopts a decreasing Δ approach across rounds. It is motivated by the intuition that the remaining action pairs become closer to the true NE as suboptimal action pairs are progressively eliminated. Consequently, the suboptimality gaps among the remaining action pairs are expected to decrease over time. In other words, as the rounds progress, we expect that the players get closer to true NE, which means they tend to repeat better action pairs more often. We can afford to use smaller Δ values to explore action pairs near the NE more thoroughly, since smaller Δ leads to a longer exploration time k.

The algorithm schedules Δ_{ij} across rounds using $\hat{\Delta}_t$, which guides both exploration time and the tolerance level for the ε-NE condition. When $\hat{\Delta}_t$ is large, action pairs are played fewer times and more action pairs are kept during elimination. As $\hat{\Delta}_t$ decreases, the algorithm focuses exploration on pairs closer to equilibrium, having already eliminated clearly suboptimal ones.

In order to perform updates over rounds, an initial estimate is required. When the payoffs are bounded within a known interval, such as $[0, 1]$, it is common practice to initialize the estimated suboptimality gap $\hat{\Delta}_t$ with value 1, representing the maximum possible difference between arm rewards as in [6]. However, in our setting, the payoffs are assumed to be σ-subgaussian, which implies that there is no strict upper bound on the suboptimality gaps, we only have $\Delta_{ij} \geq 0$. Since no empirical estimates are available at the initialization step, it is not feasible to use a bound for the expected value of the maximum of σ-subgaussian random variables to guide the choice of $\hat{\Delta}_t$. To address this, we initialize $\hat{\Delta}_t$ with 4σ, which provides a practical starting point for the analysis.

Moreover, we use $\hat{\Delta}_t = 2^{-t+2}\sigma$ in each round t which takes values from 0 to $\lfloor \frac{1}{2} \log_2 \frac{T}{e} \rfloor$. Regarding $\Delta \approx \sqrt{(16\sigma^2/k) \ln(\Delta^2 T/16\sigma^2)}$ from (20) so $\ln(\Delta^2 T/16\sigma^2) \geq 0$, which gives us $\Delta \geq \sqrt{16\sigma^2/T}$. Then, using $2^{-t+2}\sigma \geq \sqrt{16\sigma^2/T}$ and after some calculations, we obtain $t \leq \frac{1}{2} \log_2 T$. The division by e is a technical adjustment, which makes the bound safer in the analysis; otherwise the number of play would be zero in the last round.

If we assume that a unique pure NE exists in the game as before, a well estimated game matrix $\hat{A}$ should also exhibit the properties necessary to support the equilibrium. Specifically, $\hat{A}$ must satisfy the conditions required for the ε-NE in (19) to hold. Based on this principle, the Algorithm 2 systematically eliminates action pairs that fail to satisfy this condition, as such pairs cannot be near to the equilibrium. By reducing the set of action pairs to include only the pairs that meet this criterion, it ensures that the remaining action pairs are the only ones that could potentially be near the NE; thus, it enables to focus on the most relevant action pairs. Consequently, we expect that the algorithm converges to an accurate approximation of the NE, even in the presence of estimation errors in $\hat{A}$.

Algorithm 2 ETC-TPZSG-AE

Input:
S_A: set of action pairs in the game matrix A ▷ S_A includes action pairs (i, j)
m: number of actions for row player
l: number of actions for column player
T: time horizon ▷ $1 \leq ml \ll T$
σ^2: subgaussian variance factor
Initialize: $\hat{\Delta}_0 = 4\sigma$, $S_0 = S_A$ and $\hat{A} = [0]^{m\times l}$ ▷ $\hat{A}$ is estimated game matrix
In round $t = 0, 1, 2, ..., \lfloor \frac{1}{2}\log_2 \frac{T}{e} \rfloor$;
Action Pair Selection:
If $|\mathbf{S_t}| > \mathbf{1}$: ▷ Exploration Phase
Explore each action pair (i, j) in S_t $k_t = \left\lceil \frac{16\sigma^2}{\hat{\Delta}_t^2} \ln\left(\frac{\hat{\Delta}_t^2 T}{16\sigma^2}\right) \right\rceil$ times
Update $\hat{A}$ using (3).
Else: ▷ Committing Phase
Play with a unique action pair in S_t until step T
Action Pair Elimination:
Remove the action pairs (i, j) **NOT** satisfying the following for $\varepsilon_t = \sqrt{\frac{4\sigma^2}{k_t} \ln\left(\frac{\hat{\Delta}_t^2 T}{16\sigma^2}\right)}$:

$$\hat{A}(i', j) - \varepsilon_t \leq \hat{A}(i, j) \leq \hat{A}(i, j') + \varepsilon_t, \forall i', \forall j'$$

Update:
Reset S_{t+1} by *"Action Pair Elimination"* step
$\hat{\Delta}_{t+1} = \frac{\hat{\Delta}_t}{2}$ where $\hat{\Delta}_t = 2^{-t+2}\sigma$

Theorem 5.1. *The Nash regret of Algorithm 2, interacting with σ-subgaussian payoffs, is upper bounded by*

$$R_T^* \leq \sum_{(i,j)\in S_{A_1}} \Delta_{ij}^* \Big(1 + \frac{768\sigma^2}{\Delta_{ij}^2} + \frac{256\sigma^2}{\Delta_{ij}^2} \ln\Big(\frac{\Delta_{ij}^2 T}{256\sigma^2}\Big)\Big) + \sum_{(i,j)\in S_{A_2}} \Big(\Delta_{ij}^* \frac{512\sigma^2}{\lambda^2} + \Delta_{ij}^* T\Big) \quad (21)$$

where $\lambda \geq \sqrt{\frac{16\sigma^2 e}{T}}$ and, $S_{A_1} = \{(i, j) \in S_A : \Delta_{ij} > \lambda\}$ and $S_{A_2} = \{(i, j) \in S_A : 0 < \Delta_{ij} \leq \lambda\}$ are two subsets of the action pairs.

Moreover, if we consider another regret approach in (12), we obtain the following regret bound.

Theorem 5.2. *The external regret incurred by Algorithm 2 when interacting with σ-subgaussian payoffs is bounded as following:*

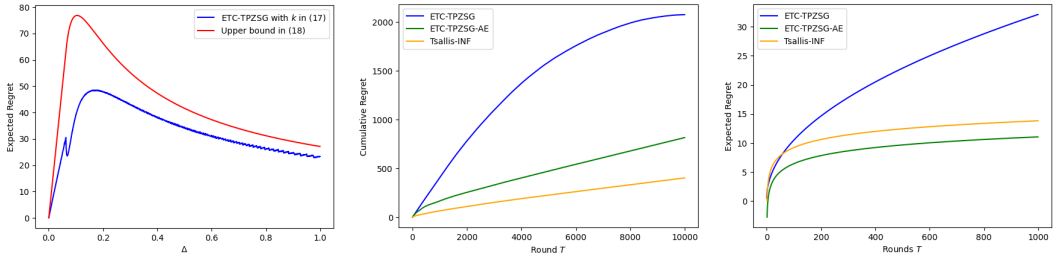
$$R_T \leq \sum_{(i,j)\in S_{A_1}} \Big(\Delta_{ij} + \frac{768\sigma^2}{\Delta_{ij}} + \frac{256\sigma^2}{\Delta_{ij}} \ln\Big(\frac{\Delta_{ij}^2 T}{256\sigma^2}\Big)\Big) + \sum_{(i,j)\in S_{A_2}} \Big(\frac{512\sigma^2}{\lambda} + \lambda T\Big) \quad (22)$$

We simplify the regret bound to $O(\frac{\log(T\Delta^2)}{\Delta})$, as the logarithmic term dominates since by setting $\lambda = \sqrt{\frac{16\sigma^2 e}{T}}$, we ensure that the term λT is bounded by $\sqrt{16\sigma^2 eT}$ which is at most $\frac{e}{\Delta_{ij}}$ when $\Delta_{ij} \leq \lambda$. Our results are consistent with the regret bound derived by [6]. The proofs of the theorems are provided in Appendix C.

In this section, we analyze two distinct regret formulations for Algorithm 2, Nash regret, defined in (3.4), and external regret, defined in (3.3). The first, denoted by R_T^*, evaluates performance against the NE, capturing the cumulative loss relative to the equilibrium strategy. In contrast, the regret analysis by R_T allows us to consider the losses of both players with respect to their individual best responses at each round, providing a broader and more flexible notion of regret. By the inequality $\Delta ij^* \leq \Delta ij$ for all action pairs, it follows that $R_T^* \leq R_T$. This outcome aligns with the interpretation that the expected regret against the NE establishes a more specific comparison and thus, it naturally provides a tighter upper bound on the expected regret.

6 Experiments

In this section, we present a set of simulations to support our theoretical findings and demonstrate the performance of the proposed algorithms. The experiments are conducted with fixed time horizons of $T = 10^3$ and $T = 10^4$. Additional details are provided in Appendix D.



(a) The expected regret of ETC-TPZSG with k in (16) and the upper bound in (17)

(b) The cumulative regrets of ETC-TPZSG, ETC-TPZSG-AE and Tsallis-INF from [18]

(c) The theoretical regret bounds for ETC-TPZSG, ETC-TPZSG-AE and Tsallis-INF from [18]

Figure 1: Regret performance comparison of the algorithms

Figure 1a shows the expected regret bound, averaged over 10^5 simulation runs, of ETC-TPZSG using k in (16) and the upper bound in (17). We consider a setting with $N = 2$, where the row player has two actions, the column player has one, and the first row corresponds to the NE. The suboptimality gap Δ varies from 0 to 1 and the subgaussian parameter is set to $\sigma = 0.5$. The results align with the bound in Theorem (4.1).

In Figure 1b, we compare ETC-TPZSG, ETC-TPZSG-AE, and Tsallis-INF [18], the only existing method with instance-dependent bounds for TPZSGs, based on cumulative regret, computed as the absolute difference between the game value and the payoff from the action played to avoid negative values, averaged over 10^3 runs. The results show that ETC-TPZSG-AE achieves lower regret than ETC-TPZSG, highlighting the effectiveness of its action pair elimination strategy. We use a 2×2 game matrix by Gaussian payoffs and k exploration rounds per action pair for ETC-TPZSG is randomly selected from a predefined list. On the other hand, Figure 1c compares their theoretical regret bounds ($O(\cdot)$ notation) for $\Delta = 0.5$, which aligns with the results in Figure 1b.

7 Discussion

We investigate a TPZSG with bandit feedback, where the payoff matrix is unknown and must be learned through player interactions. This setting is challenging, as players must estimate payoffs while making strategic decisions in an adversarial environment. We adopt the ETC algorithm due to its simplicity, widespread use and lack of prior analysis in this context. We also integrate an elimination based method, allowing the systematic removal of suboptimal action pairs, thereby improving convergence to the equilibrium and reducing unnecessary exploration on suboptimal ones. A key contribution of our work is the derivation of instance-dependent upper bounds on the expected regret for both algorithms, which has received limited attention in the literature on zero-sum games.

While our study focuses on pure strategy learning in a zero-sum game with bandit feedback and provides a theoretical expected regret bounds, several directions remain open for future study. One interesting extension is to consider games where the equilibrium is mixed. While the algorithms provided can be used to identify the support of the equilibrium, another approach should be used to efficiently converge to a mixed equilibrium.

Finally, an important and relatively unexplored direction is fairness in zero-sum games. For example, introducing mechanisms that ensure similar action pairs are explored equally can promote fairness in strategy estimation. This type of fair play mechanism could be essential in TPZSG settings, thus it can promote balanced exploration and improve overall strategy estimation. On the other hand, by integrating fairness based algorithms or constraints, it may be possible to generate game environments that ensure balanced opportunities for all players.

References

[1] Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. In *23rd Conference on Learning Theory*, pages 1–13, 2010.

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

[3] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

[4] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002.

[5] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *36th Annual Foundations of Computer Science*, pages 322–331. IEEE, 1995.

[6] Peter Auer and Ronald Ortner. Ucb revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

[7] Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8(6), 2007.

[8] Avrim Blum and Yishay Monsour. Learning, regret minimization, and equilibria. *Algorithmic Game Theory*, pages 79–102, 2007.

[9] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[10] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *International Conference on Algorithmic Learning theory*, pages 23–37. Springer, 2009.

[11] Yang Cai, Haipeng Luo, Chen-Yu Wei, and Weiqiang Zheng. Uncoupled and convergent learning in two-player zero-sum markov games with bandit feedback. In *Advances in Neural Information Processing Systems*, pages 36364–36406, 2023.

[12] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[13] Constantinos Daskalakis, Alan Deckelbaum, and Anthony Kim. Near-optimal no-regret algorithms for zero-sum games. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 235–254. SIAM, 2011.

[14] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.

[15] Constantinos Daskalakis, Aranyak Mehta, and Christos Papadimitriou. A note on approximate nash equilibria. *Theoretical Computer Science*, 410(17):1581–1588, 2009.

[16] Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. *Advances in Neural Information Processing Systems*, 25, 2012.

[17] Aurélien Garivier, Tor Lattimore, and Emilie Kaufmann. On explore-then-commit strategies. *Advances in Neural Information Processing Systems*, 29, 2016.

[18] Shinji Ito, Haipeng Luo, Taira Tsuchiya, and Yue Wu. Instance-dependent regret bounds for learning two-player zero-sum games with bandit feedback. *arXiv preprint arXiv:2502.17625*, 2025.

[19] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[20] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[21] Francis Maes, Louis Wehenkel, and Damien Ernst. Learning to play k-armed bandit problems. In *4th International Conference on Agents and Artificial Intelligence (ICAART 2012)*, 2012.

[22] Arnab Maiti, Kevin Jamieson, and Lillian Ratliff. Instance-dependent sample complexity bounds for zero-sum matrix games. In *International Conference on Artificial Intelligence and Statistics*, pages 9429–9469. PMLR, 2023.

[23] Eric V Mazumdar, Michael I Jordan, and S Shankar Sastry. On finding local nash equilibria (and only local nash equilibria) in zero-sum games. *arXiv preprint arXiv:1901.00838*, 2019.

[24] John F Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.

[25] John F Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

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

[27] Brendan O'Donoghue, Tor Lattimore, and Ian Osband. Matrix games with bandit feedback. In *Uncertainty in Artificial Intelligence*, pages 279–289. PMLR, 2021.

[28] Vianney Perchet, Philippe Rigollet, Sylvain Chassang, and Erik Snowberg. Batched bandit problems. *The Annals of Statistics*, 44(2):660–681, 2016.

[29] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[30] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

[31] John Von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, 1928.

[32] John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[33] Alan R Washburn. *Two-Person Zero-Sum Games*. Springer, 2014.

[34] Ali Yekkehkhany, Ebrahim Arian, Mohammad Hajiesmaili, and Rakesh Nagi. Risk-averse explore-then-commit algorithms for finite-time bandits. In *2019 IEEE 58th Conference on Decision and Control*, pages 8441–8446. IEEE, 2019.

[35] Martin Zinkevich, Michael Bowling, and Neil Burch. A new algorithm for generating equilibria in massive zero-sum games. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, pages 788–793, 2007.

A Subgaussian Properties

This section collects some supplementary results from [20] that are used repeatedly throughout our proofs. These results concern standard properties of subgaussian random variables such as tail inequalities. They are included here for completeness and easy reference since they play a critical role in our analyses. The statements are not original, we include only the parts that are directly useful for our purposes.

Theorem A.1 ([20, Theorem 5.3])**.** *If X is σ-subgaussian random variable, for any $\epsilon \geq 0$,*

$$\mathbb{P}(X \geq \epsilon) \leq \exp\Big(-\frac{\epsilon^2}{2\sigma^2}\Big). \tag{23}$$

Lemma A.1 ([20, Lemma 5.4]). *If X is σ-subgaussian and X_1 and X_2 are independent with σ_1 and σ_2 subgaussian parameter, respectively, then we can write the followings:*

(a) cX is $|c|\sigma$-subgaussian for all $c \in \mathbb{R}$.

(b) $X_1 + X_2$ is $\sqrt{\sigma_1^2 + \sigma_2^2}$-subgaussian.

B Proof of Theorem 4.1

To analyze the Nash regret of the ETC-TPZSG algorithm, we begin by decomposing the total expected regret into two phases: the exploration phase and the committing phase. During exploration, each action pair is played a fixed number of times k to estimate their mean rewards, potentially incurring regret if suboptimal action pairs are played. Once the algorithm commits to the empirically best action pair, which is the pure NE, the regret accumulates only if this action pair is not the true optimal one. Our goal is to bound the expected regret by quantifying the probability of committing to a suboptimal action pair, which we can call it as a misidentified NE, based on the estimates of their payoffs obtained during exploration phase.

Let (i', j') denote a misidentified NE, so we aim to identify an action pair (i', j'), which appears better than (i^*, j^*) based on the estimated matrix. To analyze the probability of playing with a misidentified NE, we consider whether the following two events, E_1 and E_2, occur:

$$E_1 : \hat{A}(i, j') \leq \hat{A}(i', j'), \quad E_2 : \hat{A}(i', j') \leq \hat{A}(i', j), \quad \forall i \in S_x, \forall j \in S_y$$

and we want to find a bound for $\mathbb{P}(E_1 \cap E_2)$ because these two conditions must be satisfied to consider $(i'.j')$ as a NE.

Since the events E_1 and E_2 are not independent, we consider two different approaches to find an upper bound for $\mathbb{P}(E_1 \cap E_2)$. We then select the approach that gives the tighter bound, as it provides a more accurate estimate of this probability. The first approach uses the fact that $\mathbb{P}(E_1 \cap E_2) \leq \mathbb{P}(E_1)$ and $\mathbb{P}(E_1 \cap E_2) \leq \mathbb{P}(E_2)$, which together imply $\mathbb{P}(E_1 \cap E_2) \leq \sqrt{\mathbb{P}(E_1)\mathbb{P}(E_2)}$. Alternatively, we can use the inequality $\mathbb{P}(E_1 \cap E_2) \leq \min\{\mathbb{P}(E_1), \mathbb{P}(E_2)\}$, which may provide a tighter bound depending on the values of $\mathbb{P}(E_1)$ and $\mathbb{P}(E_2)$. The choice between these approaches depends on which enables us the smaller upper bound.

Let us start by considering the first approach. We can write it as

$$\begin{aligned}
\mathbb{P}(E_1 \cap E_2) &= \mathbb{P}(\hat{A}(i, j') \leq \hat{A}(i', j') \wedge \hat{A}(i', j') \leq \hat{A}(i', j), \forall i, j) && (24)\\
&\leq \mathbb{P}(\hat{A}(i, j') \leq \hat{A}(i', j') \wedge \hat{A}(i', j') \leq \hat{A}(i', j)) && (25)\\
&= \sqrt{\mathbb{P}(\hat{A}(i, j') \leq \hat{A}(i', j'))\, \mathbb{P}(\hat{A}(i', j') \leq \hat{A}(i', j))} && (26)\\
&= \sqrt{\mathbb{P}(\hat{A}(i, j') - \hat{A}(i', j') \leq 0)\, \mathbb{P}(\hat{A}(i', j') - \hat{A}(i', j) \leq 0)} && (27)\\
&\leq \sqrt{\mathbb{P}(\hat{A}(i, j') - \hat{A}(i', j') \leq \Delta^{\max}_{i'j'})\, \mathbb{P}(\hat{A}(i', j') - \hat{A}(i', j) \leq \Delta^{\min}_{i'j'})} && (28)\\
&\leq \sqrt{\exp\Big(-\frac{k(\Delta^{\max}_{i'j'})^2}{4\sigma^2}\Big) \exp\Big(-\frac{k(\Delta^{\min}_{i'j'})^2}{4\sigma^2}\Big)} && (29)\\
&= \sqrt{\exp\Big(-\frac{k[(\Delta^{\max}_{i'j'})^2 + (\Delta^{\min}_{i'j'})^2]}{4\sigma^2}\Big)} && (30)\\
&\leq \sqrt{\exp\Big(-\frac{k(\Delta^{\max}_{i'j'} + \Delta^{\min}_{i'j'})^2}{8\sigma^2}\Big)} && (31)\\
&= \sqrt{\exp\Big(-\frac{k(\Delta_{i'j'})^2}{8\sigma^2}\Big)} && (32)\\
&= \exp\Big(-\frac{k(\Delta_{i'j'})^2}{16\sigma^2}\Big) && (33)
\end{aligned}$$

where $\Delta^{\max}_{i'j'} = \max_i A(i, j') - A(i', j')$, $\Delta^{\min}_{i'j'} = A(i', j') - \min_j A(i', j)$ and $\Delta_{i'j'} = \Delta^{\max}_{i'j'} + \Delta^{\min}_{i'j'}$.

To get (26), we utilize the fact that the probability of the intersection of multiple events is at most that of any individual event, that is, for any collection of events $e_1, e_2, \ldots, e_n$, we have $\mathbb{P}(e_1 \cap e_2 \cap \cdots \cap e_n) \leq \mathbb{P}(e_i)$ for all $i \in 1, \ldots, n$. The inequality (28) holds since $\Delta^{\max}_{i'j'} \geq 0, \Delta^{\min}_{i'j'} \geq 0$, which are true from their definitions. For the step (29), we apply Theorem A.1 with $\sqrt{2\sigma^2/k}$-subgaussian random variable.

At this point, we need to show that the differences $\hat{A}(i,j) - \hat{A}(x,y)$ are $\sqrt{2\sigma^2/k}$-subgaussian where (i,j) and (x,y) are any action pairs in S_A. To establish this, we utilize Lemma A.1, which provides key properties of subgaussian random variables. Specifically, both $\hat{A}(i,j)$ and $\hat{A}(x,y)$ are $\sqrt{\sigma^2/k}$-subgaussian as the average payoffs are computed by the equation (3) and during exploration phase, each action pair is played k times. By the lemma, the difference of two independent subgaussian random variables with parameters $\sqrt{\sigma^2/k}$ is subgaussian with parameter $\sqrt{2\sigma^2/k}$, which follows that $\hat{A}(i,j) - \hat{A}(x,y)$ is $\sqrt{2\sigma^2/k}$-subgaussian for any action pairs (i,j) and (x,y).

Then, to get the last inequality (31), we observe that

$$\frac{\Delta^2_{i'j'}}{2} = \frac{(\Delta^{\max}_{i'j'} + \Delta^{\min}_{i'j'})^2}{2} \leq (\Delta^{\max}_{i'j'})^2 + (\Delta^{\min}_{i'j'})^2$$

where the inequality follows from the fact that $(\Delta^{\max}_{i'j'} - \Delta^{\min}_{i'j'})^2 \geq 0$.

As we mention before, there is an alternative approach to bound $\mathbb{P}(E_1 \cap E_2)$, which means to find a bound for the probability of a misidentified NE. It involves an inequality based on the minimum of the individual probabilities. Similarly, let follow this:

$$\begin{aligned}
\mathbb{P}(E_1 \cap E_2) &= \mathbb{P}(\hat{A}(i,j') \leq \hat{A}(i',j') \wedge \hat{A}(i',j') \leq \hat{A}(i',j), \forall i,j) && (34)\\
&\leq \mathbb{P}(\hat{A}(i,j') \leq \hat{A}(i',j') \wedge \hat{A}(i',j') \leq \hat{A}(i',j)) && (35)\\
&= \min\{\mathbb{P}(\hat{A}(i,j') \leq \hat{A}(i',j')), \mathbb{P}(\hat{A}(i',j') \leq \hat{A}(i',j))\} && (36)\\
&= \min\{\mathbb{P}(\hat{A}(i,j') - \hat{A}(i',j') \leq 0)\, \mathbb{P}(\hat{A}(i',j') - \hat{A}(i',j) \leq 0)\} && (37)\\
&\leq \min\{\mathbb{P}(\hat{A}(i,j') - \hat{A}(i',j') \leq \Delta^{\max}_{i'j'}), \mathbb{P}(\hat{A}(i'j') - \hat{A}(i'j) \leq \Delta^{\min}_{i'j'})\} && (38)\\
&\leq \min\Big\{\exp\Big(-\frac{k(\Delta^{\max}_{i'j'})^2}{4\sigma^2}\Big), \exp\Big(-\frac{k(\Delta^{\min}_{i'j'})^2}{4\sigma^2}\Big)\Big\} && (39)\\
&= \exp\Big(-\frac{k\max\{(\Delta^{\max}_{i'j'})^2, (\Delta^{\min}_{i'j'})^2\}}{4\sigma^2}\Big) && (40)\\
&\leq \exp\Big(-\frac{k[(\Delta^{\max}_{i'j'})^2 + (\Delta^{\min}_{i'j'})^2]}{8\sigma^2}\Big) && (41)\\
&\leq \exp\Big(-\frac{k(\Delta^{\max}_{i'j'} + \Delta^{\min}_{i'j'})^2}{16\sigma^2}\Big) && (42)\\
&\leq \exp\Big(-\frac{k\Delta^2_{i'j'}}{16\sigma^2}\Big) && (43)
\end{aligned}$$

where to get the inequality (41), we use the following:

$$\max\{(\Delta^{\max}_{i'j'})^2, (\Delta^{\min}_{i'j'})^2\} \geq \frac{(\Delta^{\max}_{i'j'})^2 + (\Delta^{\min}_{i'j'})^2}{2} \tag{44}$$

If we assume that their maximum is $\Delta^{\max}_{i'j'}$, then it will imply $(\Delta^{\max}_{i'j'})^2 \geq (\Delta^{\min}_{i'j'})^2$. Similarly, if $\max\{(\Delta^{\max}_{i'j'})^2, (\Delta^{\min}_{i'j'})^2\} = (\Delta^{\min}_{i'j'})^2$, it will give us $(\Delta^{\min}_{i'j'})^2 \geq (\Delta^{\max}_{i'j'})^2$. They are true by assumptions, then, (44) holds.

Thus, both approaches provide the same upper bound for the probability of misidentified NE. In the regret analysis, we will focus on the loss due to playing with suboptimal (or not true NE) action pairs. Thus, after T times playing, $1 \leq Nk \leq T$ where $N = ml$ as the number of action pairs, we can write the Nash regret as

$$R^*_T = \sum_{(i,j) \in S_A} \Delta^*_{ij} \mathbb{E}[n_{ij,T}]$$

where $\mathbb{E}[n_{ij,T}]$ is the expected number of playing the action pair (i, j). We already know that each action pair $(i, j) \in S_A$ is played at least k times for exploration. After the exploration step, we will consider playing the misidentified NE (i', j') instead of real one (i^*, j^*) because we expect the players to select NE action pair as optimal after exploration phase. Since we already have the probability of playing a misidentified NE from (33) or (43), we can write

$$\mathbb{E}[n_{ij,T}] \leq k + (T - Nk) \exp\Big(- \frac{k\Delta_{ij}^2}{16\sigma^2}\Big).$$

Hence, the proof is concluded.

□

C Proof of Theorem 5.1

According to the algorithm, $\sqrt{\frac{16\sigma^2 e}{T}}$ is a critical value for $\hat{\Delta}_t$ since it takes this value in the last round. In this way, let define a threshold parameter as $\lambda \geq \sqrt{\frac{16\sigma^2 e}{T}}$, which gives us two subsets of actions pairs such that $S_{A_1} = \{(i, j) \in S_A : \Delta_{ij} > \lambda\}$ and $S_{A_2} = \{(i, j) \in S_A : 0 < \Delta_{ij} \leq \lambda\}$. Since $\Delta_{ij}^* \leq \Delta_{ij}$ ensures the bounds are preserved, it makes sense to use these action pair sets in the analysis. Thus, we can write the Nash regret as

$$R_T^* = \sum_{(i,j)\in S_{A_1}} \Delta_{ij}^* \mathbb{E}[n_{ij,T}] + \sum_{(i,j)\in S_{A_2}} \Delta_{ij}^* \mathbb{E}[n_{ij,T}]$$

where $\Delta_{ij}^* = A(i^*, j^*) - A(i, j)$. and $n_{ij,T}$ is the total number of times to play action pair (i, j) by time T.

To analyze the Nash regret of the ETC-TPZSG-AE algorithm, it is essential to consider various scenarios that may lead to suboptimal outcomes. Each of these scenarios contributes to the total Nash regret, as they either involve the incorrect elimination of the optimal action pair or the unnecessary selection of suboptimal pairs. The former results in discarding the optimal strategy because of insufficient evidence, while the latter leads to spend more time with action pairs that are unlikely to be part of any near optimal equilibrium.

The algorithm employs a scheduling mechanism for Δ_{ij}, where $\hat{\Delta}_t$ is halved in each round. This approach is necessary because the true values of Δij are unknown to the players. We use $\hat{\Delta}_t$ to determine both the exploration time and the tolerance level in the ε-NE property. Specifically, when $\hat{\Delta}_t$ is large, the algorithm plays each action pair fewer times and keeps more pairs during the elimination step, since the threshold for elimination is relatively loose. As $\hat{\Delta}_t$ decreases, the algorithm allocates more exploration to action pairs that are closer to being part of a NE, because those that are clearly suboptimal have already been eliminated. This adaptive process enables more precise identification of near-optimal strategies by ε-NE condition while minimizing regret.

For each suboptimal action pair (i, j), let $t_{ij} = \min\{t : \hat{\Delta}_t < \Delta_{ij}/2\}$ refer to the earliest round such that $\hat{\Delta}_t < \Delta_{ij}/2$. Using the fact that $\hat{\Delta}_{t+1} = \frac{\hat{\Delta}_t}{2}$ and the definition of t_{ij}, we can write the following:

$$\frac{1}{\hat{\Delta}_{t_{ij}}} \leq \frac{4}{\Delta_{ij}} < \frac{1}{\hat{\Delta}_{t_{ij}+1}} \tag{45}$$

For simplicity of notation, we write $\hat{A}$ instead of $\hat{A}_t$ for the estimated payoff matrix in round t throughout this section. Since we already include additional terms such as ε_t that refers to the tolerance level in round t, it is clear enough which round is being referenced.

We now analyze the regret contributions by considering the following cases:

Case C.1. *A suboptimal action pair (i, j) is not eliminated in round t_{ij} or earlier while the optimal action pair is in the set $S_{t_{ij}}$.*

Since neither some suboptimal nor the optimal action pair is not eliminated, the algorithm fails to discard suboptimal choices. It might converge to an incorrect solution without eliminating suboptimal

actions, but since it still maintains the optimal action pair within the action pair selection set. The regret contribution of this case comes from the fact that the algorithm spends time with a worse choice when a better one is already available.

Let explain briefly in which situations the algorithm eliminates or keep an action pair. For action pair (i, j), if ε-NE property does not hold in round $t = t_{ij}$ which implies that at least one of the inequalities fails, then it will be eliminated in round t_{ij}.

Furthermore, we have $\varepsilon_{t_{ij}} = \sqrt{\frac{4\sigma^2}{k_{t_{ij}}} \ln\left(\frac{\hat{\Delta}_{t_{ij}}^2 T}{16\sigma^2}\right)} \leq \frac{\hat{\Delta}_{t_{ij}}}{2} = \hat{\Delta}_{t_{ij}+1} < \frac{\Delta_{ij}}{4}$. In other words, if there exist $i' \in S_x$ or $j' \in S_y$, such that one of the following is true:

$$I_1 : \hat{A}(i', j) - \hat{A}(i, j) > \varepsilon_{t_{ij}}, \qquad I_2 : \hat{A}(i, j) - \hat{A}(i, j') > \varepsilon_{t_{ij}},$$

then (i, j) is eliminated. The inequality I_1 indicates that there exists an alternative action i' for the row player that offers a higher payoff than action i against the column action j. Similarly, I_2 implies that there exists a better action j' for the column player than action j when playing against the row action i. If either I_1 or I_2 holds, the action pair (i, j) cannot be a part of the NE, as at least one player has an incentive to deviate. Hence, (i, j) is not a NE with high probability.

On the other hand, to keep an action pair (i, j) the following events must both hold:

$$E_1 : \hat{A}(i', j) - \varepsilon_t \leq \hat{A}(i, j), \forall i' \in S_x, \quad E_2 : \hat{A}(i, j) \leq \hat{A}(i, j') + \varepsilon_t, \forall j' \in S_y \tag{46}$$

To calculate the probability of keeping an action pair (i, j), denoted by $\mathbb{P}(E_1 \cap E_2)$, we begin by analyzing the event E_1. Specifically, for an action pair (i, j) which is not a NE, i.e. there exists at least one i' such that $A(i', j) > A(i, j)$, the probability of E_1 can be expressed as follows:

$$\begin{aligned}
\mathbb{P}(E_1) &= \mathbb{P}(\hat{A}(i', j) - \hat{A}(i, j) \leq \varepsilon_t, \forall i' \in S_x) && (47)\\
&\leq \mathbb{P}(\hat{A}(i', j) - \hat{A}(i, j) \leq \varepsilon_t) && (48)\\
&\leq \exp\left(- \frac{\frac{4\sigma^2}{k_t} \ln\left(\frac{\hat{\Delta}_t^2 T}{16\sigma^2}\right)}{\frac{4\sigma^2}{k_t}} \right) && (49)\\
&= \frac{16\sigma^2}{\hat{\Delta}_t^2 T} && (50)
\end{aligned}$$

where $\varepsilon_t = \sqrt{\frac{4\sigma^2}{k_t} \ln\left(\frac{\hat{\Delta}_t^2 T}{16\sigma^2}\right)}$. To derive inequality (48), we use the fact that the probability of the intersection of multiple events is always less than or equal to the probability of any of individual events. Specifically, since we have m actions in S_x, for any collection of events $e_1, e_2, \cdots, e_m$, it holds that $\mathbb{P}(e_1 \cap e_2 \cap \cdots \cap e_m) \leq \mathbb{P}(e_i)$ for all $i = 1, 2, \cdots, m$. In the last inequality (49), we apply Theorem A.1 with $\sqrt{2\sigma^2/k_t}$-subgaussian random variable. The subgaussian parameter is derived by using Lemma A.1 and we note that each action pair is played k_t times.

Similarly, in order to calculate the probability of the event E_2, we can write

$$\mathbb{P}(E_2) = \mathbb{P}(\hat{A}(i, j) - \hat{A}(i, j') \leq \varepsilon_t, \forall j' \in S_y) \leq \frac{16\sigma^2}{\hat{\Delta}_t^2 T}. \tag{51}$$

Therefore, we have the probability of keeping an action pair (i, j) in any round t as

$$\mathbb{P}(E_1 \cap E_2) \leq \min\{\mathbb{P}(E_1), \mathbb{P}(E_2)\} = \frac{16\sigma^2}{\hat{\Delta}_t^2 T}. \tag{52}$$

It means that the probability that a suboptimal action pair (i, j) is not eliminated in round t_{ij} or before is bounded by $\frac{16\sigma^2}{\hat{\Delta}_{t_{ij}}^2 T}$. Then, the regret contribution is simply bounded by a worst case which is $T\Delta_{ij}^*$ for any suboptimal action pair $(i, j) \in S_{A_1}$.

Hence, using the probability of keeping a suboptimal action pair and summing up over all action pairs, we can write their regret contribution as

$$\sum_{(i,j) \in S_{A_1}} \Delta_{ij}^* T \frac{16\sigma^2}{\hat{\Delta}_{t_{ij}}^2 T} \leq \sum_{(i,j) \in S_{A_1}} \Delta_{ij}^* \frac{256\sigma^2}{\Delta_{ij}^2} \tag{53}$$

where we apply inequality (45) to bound it.

Here, we note that we do not need to consider the action pairs in S_{A_2} because, by the design of the algorithm, the gap $\hat{\Delta}_t$ is guaranteed to be at least around λ in the corresponding rounds. If the optimal action pair remains in the game, then all suboptimal action pairs should have already been eliminated by the last round or earlier. This means that suboptimal choices are progressively discarded over time as the algorithm learns. As a result, after all the rounds have been played, only one action pair should remain since we consider the case the optimal action pair (i^*, j^*) remain in the game in addition to elimination of suboptimal ones. This final pair is expected to correspond to the pure NE, representing the best choice for both players with no incentive to change their actions.

Case C.2. *A suboptimal action pair (i, j) is eliminated in round t_{ij} or earlier with the optimal action pair in $S_{t_{ij}}$.*

It enables us to bound the number of times it is played. Once a suboptimal action pair is eliminated and the optimal action pair (i^*, j^*) remains in the game, it no longer contributes to the regret in next rounds. This is because regret arises only when a suboptimal action is chosen instead of the optimal one.

On the other hand, we note that there is no need to consider the action pairs $(i, j) \in S_{A_2}$ because we handle the elimination of suboptimal pairs before the final round. Thus, using (45), each action pair (i, j) is played at most $k_{t_{ij}}$ times such that $k_{t_{ij}} = \left\lceil \frac{16\sigma^2}{\hat{\Delta}^2_{t_{ij}}} \ln\left(\frac{\hat{\Delta}^2_{t_{ij}} T}{16\sigma^2}\right) \right\rceil \leq \left\lceil \frac{256\sigma^2}{\Delta^2_{ij}} \ln\left(\frac{\Delta^2_{ij} T}{256\sigma^2}\right) \right\rceil$.

Then, the regret contribution is expressed by

$$\sum_{(i,j)\in S_{A_1}} \Delta^*_{ij} \left\lceil \frac{256\sigma^2}{\Delta^2_{ij}} \ln\left(\frac{\Delta^2_{ij} T}{256\sigma^2}\right) \right\rceil \leq \sum_{(i,j)\in S_{A_1}} \Delta^*_{ij} \left(1 + \frac{256\sigma^2}{\Delta^2_{ij}} \ln\left(\frac{\Delta^2_{ij} T}{256\sigma^2}\right)\right) \tag{54}$$

$$= \sum_{(i,j)\in S_{A_1}} \Delta^*_{ij} + \Delta^*_{ij} \frac{256\sigma^2}{\Delta^2_{ij}} \ln\left(\frac{\Delta^2_{ij} T}{256\sigma^2}\right) \tag{55}$$

$$\tag{56}$$

Case C.3. *The optimal action pair (i^*, j^*) is eliminated by some suboptimal one (i, j) in round t_* such that $t_{ij} \geq t_*$.*

It leads to a misidentification of the NE, implying that an action pair (i, j) should be better than the optimal one based on the current estimates. By ε-NE property, clearly we have

$$\hat{A}(i^*, j) - \varepsilon_{t_*} \leq \hat{A}(i, j) \leq \hat{A}(i, j^*) + \varepsilon_{t_*}.$$

However, it is not enough to keep a suboptimal action pair (i, j). On the other hand, we note that if the optimal action pair (i^*, j^*) is eliminated in round t_*, it must fail to satisfy ε-NE property as the following:

$$\hat{A}(i, j^*) - \hat{A}(i^*, j^*) > \varepsilon_{t_*} \quad \text{or/and} \quad \hat{A}(i^*, j^*) - \hat{A}(i^*, j) > \varepsilon_{t_*}.$$

That is, under the estimated payoff matrix $\hat{A}$, there exists an action i that is estimated to yield a higher payoff than i^* for the row player, and an action j that is estimated to be more favorable than j^* for the column player.

The optimal action pair can only be eliminated by a suboptimal action pair (i, j) in round t_* such that $t_{ij} \geq t_*$ since action pair (i, j) must remain under consideration at the elimination round of the optimal action pair. This condition also implies that $\hat{\Delta}_{t_{ij}} \leq \hat{\Delta}_{t_*}$ because of the decreasing behavior of $\hat{\Delta}_t$ across rounds in the algorithm. Moreover, it is important to note that the algorithm keeps this action pair (i, j) for at least one additional round following the elimination of the optimal pair as it is identified as a NE. We further observe that all action pairs (i, j) with $t_{ij} < t_*$ have already been eliminated in round t_{ij} or earlier, which we already consider this condition in the previous case.

On the other hand, for the action pairs in S_{A_2}, we assume that $\hat{\Delta}_t < \frac{\lambda}{2}$, which implies that a suboptimal action pair remains in the game until the final round. This scenario emphasizes the regret contribution when the estimated suboptimality gap $\hat{\Delta}_t$ becomes sufficiently small, a case represented

by the action pairs in S_{A_2}. Consequently, these suboptimal action pairs contribute to the overall regret and the analysis accounts for their effect to establish accurate performance guarantees.

If the algorithm keeps a suboptimal action pair (i, j) up to round t_*, the events specified in (46) must hold with parameter ε_{t_*}. Consequently, by applying (52), the probability of keeping a suboptimal action pair can be bounded by $\frac{4N\sigma^2}{\hat{\Delta}_{t_*}^2 T}$. Thus, we can write the regret contribution of this case as

$$\sum_{(i,j)\in S_A} \Delta_{ij}^* T \frac{16\sigma^2}{\hat{\Delta}_{t_*}^2 T} < \sum_{(i,j)\in S_A} \Delta_{ij}^* T \frac{16\sigma^2}{\hat{\Delta}_{t_{ij}+1}^2 T} \tag{57}$$

$$= \sum_{(i,j)\in S_A} \Delta_{ij}^* T \frac{64\sigma^2}{\hat{\Delta}_{t_{ij}}^2 T} \tag{58}$$

$$\leq \sum_{(i,j)\in S_{A_1}} \Delta_{ij}^* \frac{512\sigma^2}{\Delta_{ij}^2} + \sum_{(i,j)\in S_{A_2}} \Delta_{ij}^* \frac{512\sigma^2}{\lambda^2} \tag{59}$$

where $\hat{\Delta}_{t+1} = \frac{\hat{\Delta}_t}{2}$ and we apply (45). We use $\hat{\Delta}_{t+1}$ because the optimal action pair might be eliminated at some round t_* such that $t_* \in [0, \max_{ij} t_{ij}]$. This implies that the optimal action pair can be eliminated by any (i, j) no later than the last round that (i, j) remains in the game. However, we must keep (i, j) at least until the next round, since it is not removed in round t_* and is selected as a misidentified NE.

Case C.4. *A suboptimal action pair (i, j) in the set S_{A_2} remains in the game as a unique action pair in $S_{t_{ij}}$.*

This implies that a suboptimal action pair is played during the committing phase of Algorithm 2 up to step T. Thus, we need to account for an additional regret contribution term given by

$$\sum_{(i,j)\in S_{A_2}} \Delta_{ij}^* T. \tag{60}$$

This regret contribution accounts for scenarios where the algorithm may eliminate all action pairs except one. If the remaining action pair is suboptimal, the algorithm will continue to play with this action pair for the remaining rounds. Since the action pair selection persists up to time step T, resulting in the additional term in the regret bound.

We note that $\Delta_{ij}^* \leq \Delta_{ij}$, which ensures that the regret does not grow excessively, and this relationship allows us to analyze the regret in terms of the defined action sets. As a result, if we sum these regret contributions as mentioned, we can conclude the proof.

□

C.1 Proof of Theorem 5.2

We consider the regret approach consisting of the losses of both players by comparing their best choices to the action played. This regret measures how much worse a player performs compared to their optimal strategy if they had known the opponent's behavior in advance. Clearly we can write the followings:

$$\begin{aligned} R_T &= \sum_{(i,j)\in S_A} \Delta_{ij}^{\max} \mathbb{E}[n_{ij,T}] + \sum_{(i,j)\in S_A} \Delta_{ij}^{\min} \mathbb{E}[n_{ij,T}] \\ &= \sum_{(i,j)\in S_A} \Delta_{ij} \mathbb{E}[n_{ij,T}] \\ &= \sum_{(i,j)\in S_{A_1}} \Delta_{ij} \mathbb{E}[n_{ij,T}] + \sum_{(i,j)\in S_{A_2}} \Delta_{ij} \mathbb{E}[n_{ij,T}] \end{aligned}$$

This implies that the term involving $\Delta_{ij}^{\max}$ characterizes the regret incurred by the maximizing player, whereas the loss of the minimizing player is determined using $\Delta_{ij}^{\min}$. In particular, we utilize the same underlying algorithm, which maintains the same uniform playing strategy and elimination strategy

throughout the process. This consistency allows us to to leverage many elements of the previous regret analysis. Consequently, we can apply the same case scenarios from the proof of Theorem 5.1 as the elimination of action pairs follows the same strategy and the selection process of action pairs remains the same across rounds.

Although our current analysis involves a different notion of regret, consisting of Δ_{ij}, this does not affect the underlying structure of the regret analysis. This ensures that the previous theoretical bounds can be extended by using Δ_{ij} instead of Δ_{ij}^* and we apply $\Delta_{ij}^* \leq \Delta_{ij}$ and $\Delta_{ij} \leq \lambda$ for the action pairs in S_{A_2}, then the result follows.

□

D Experimental Details

In this section, we present additional experiments to compare the performances of the algorithms based on their theoretical bounds and cumulative regret. To calculate the cumulative regret, we use the absolute difference between the value of the game and the payoff of the action played as the regret might otherwise be negative.

In Figure 1b, we compare the performance of the ETC-TPZSG, ETC-TPZSG-AE and Tsallis-INF [18] algorithms. In each run, the exploration time k for each action pair in ETC-TPZSG is randomly selected from a predefined list ranging from 100 to 2500 in increments of 100. The total number of rounds is set to $T = 10^4$. On the other hand, each run uses a new game matrix with a unique pure NE which is generated using Gaussian payoffs with mean zero, standard deviation σ selected from the list $[0.25, 0.5, 0.75, 1]$. Varying σ not only changes the payoff distribution but also affects the exploration time and the tolerance parameter ε used in the elimination phase of ETC-TPZSG-AE. The cumulative regret is averaged over 10^3 simulation runs. As observed, the ETC-TPZSG-AE algorithm consistently outperforms ETC-TPZSG, demonstrating the effectiveness of its elimination strategy in reducing cumulative regret, while Tsallis-INF perform better when compared to ETC-based algorithms. Notably, the ETC-based algorithms offer the advantage of conceptual and implementational simplicity.

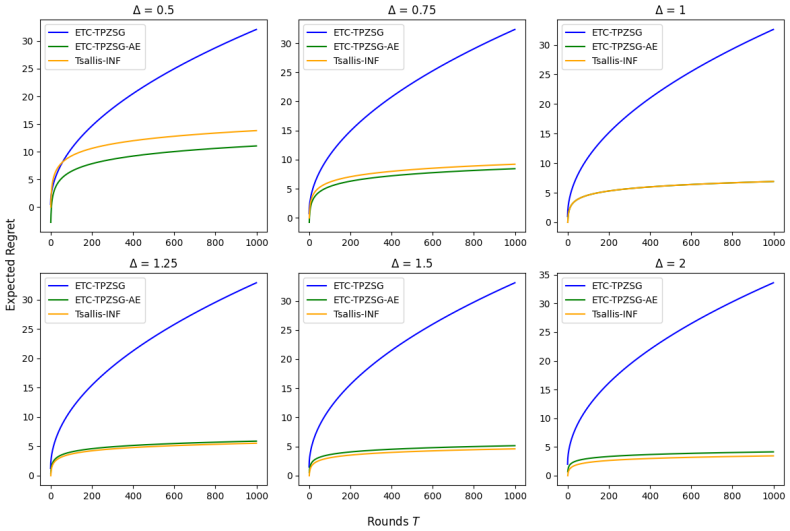


Figure 2: Theoretical expected regret bound comparison between ETC-TPZSG, ETC-TPZSG-AE and Tsallis-INF from [18] using different Δ values

On the other hand, Figure 2 compares the theoretical expected regret bounds (in $O(\cdot)$ notation) of the algorithms across different values of the suboptimality gap Δ. The results show that Tsallis-INF tends to perform better when Δ is large while the ETC-TPZSG-AE algorithm achieves lower regret for smaller values of Δ.