Efficiently Computing Approximate Nash Equilibria in Multi-Adversarial Team Games

Prasanna Maddila
Université de Toulouse,
INRAE – MIAT
Castanet-Tolosan, France
siva-sri-prasanna.maddila@inrae.fr

Régis Sabbadin
Université de Toulouse,
INRAE – MIAT
Castanet-Tolosan, France
regis.sabbadin@inrae.fr

Meritxell Vinyals
Université de Toulouse,
INRAE – MIAT
Castanet-Tolosan, France
meritxell.vinyals@inrae.fr

ABSTRACT

Adversarial Team Games (ATG), as introduced by von Stengel and Koller, model strategic interactions in which a team of agents – sharing a common objective but unable to coordinate their actions – faces a single adversary. While computing an exact Nash Equilibrium (NE) in ATGs has been shown to be CLS-complete, a fully polynomial-time approximation scheme (FPTAS) has been developed, enabling the efficient computation of approximate NE in time polynomial in both the natural parameters of the game and the inverse of the approximation error. However, existing results only apply to single-adversary scenarios, leaving the common case of multiple independent adversaries — prevalent in applications such as anti-poaching, robotic planning, and hider-seeker games — largely unexplored. This paper bridges this gap by introducing the Multi-Adversarial Team Game (MATG) framework, a natural generalization of ATGs to scenarios involving several independent adversaries. Our main contribution is to generalize techniques from the single-adversary setting to develop an FPTAS for computing NE in the more general class of MATGs. Beyond our theoretical contributions, we present the first empirical evaluation of this family of algorithms in both ATGs and MATGs, demonstrating the scalability to many adversaries.

KEYWORDS

Multi Adversarial Team Games; Nash Equilibria; FPTAS

ACM Reference Format:
Prasanna Maddila, Régis Sabbadin, and Meritxell Vinyals. 2026. Efficiently Computing Approximate Nash Equilibria in Multi-Adversarial Team Games. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/TBAX6220

1 INTRODUCTION

Adversarial Team Games (ATGs), as introduced by von Stengel and Koller [30], model situations where a team of agents faces a *single* adversary; the team shares a common objective but are unable to coordinate their actions. This inability to coordinate their actions models real-world considerations where communication is prohibited (e.g. by game rules), impractical (e.g. by security) or too expensive (e.g. in large organisations). For ATGs, the literature has studied two main solution concepts: (1) Nash Equilibrium (NE), where no agent can improve their utility by unilaterally deviating from the equilibrium; and (2) Team-Minmax Equilibrium (TME), which is the NE that yields the highest utility for the team.
The TME has been extensively studied due to its appealing properties: it always exists, it is unique (except in degenerate cases), and it can lead to team payoffs arbitrarily higher than those of any other NE. However, computing a TME is intractable (FNP-hard) [3, 14], even for approximate solutions. In contrast, [1] showed that ATGs admit a Fully Polynomial-Time Approximation Scheme (FPTAS) for computing NE. This is a notable result, as only a few multi-player (>2 players) games classes are known to admit polynomial-time algorithms for approximating NE – namely, zero-sum polymatrix games [5], potential games [24], and, since [1], ATGs. However, although [1] identified the extension of ATGs to a broader family of polynomial-time solvable games as an important but non-trivial perspective, no subsequent work has succeeded on pushing this frontier beyond the single-adversary case. It is worth noting that the broader class of two-team games [18], which generalizes ATGs by allowing multiple adversaries that form a second team, have been proven instead computationally intractable.

Against this background, this work focuses on generalizing ATGs by considering multiple *independent* adversaries – a setting that has remained unexplored in the literature despite being very common in real-world scenarios. For instance, in anti-poaching scenarios — a problem modeled by Green Security Games [29, 32, 33] — rangers need to detect several potential poachers within the same protected area [20, 21]. Similarly, in robotic planning [23], different adversaries may control different aspects of the environment. Lastly, in hider-seeker games [13], a team may need to locate not one but multiple independent hiders. In this context, this paper addresses this unexplored multi-adversary setting, focusing on the following research question: *Can we design polynomial-time algorithms for approximating NE in adversarial team games with several independent adversaries?*

Related work.[1,2] For NE, [1] showed that computing an approximate NE (i.e., not necessarily the TME) in ATGs is CLS-complete (even in the case of an ATG with two teammates [2]). Importantly, their constructive proof yielded an algorithm for ϵ-NE with polynomial complexity in all natural game parameters and $1/\epsilon$. Adversarial two-team games [17, 28] are generalisations of ATGs where the adversaries also form a team. [15] analyzed NE computation in two-team zero-sum *polymatrix* games (i.e. with pairwise utilities for

[1] Additional related work is included in Appendix A.
[2] All appendices are available at https://hal.science/hal-04917907.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/TBAX6220

all agents), including a special case with independent adversaries, which they prove is CLS-complete. Other works [6–8, 16, 19, 34] studied ATGs as Markov or Extensive-form games. In contrast, this paper focuses on normal-form MATGs with independent adversaries—a setting not previously studied in either normal-form or sequential extensions, leaving the latter as future work.

Original contributions. Our contributions are three-fold. *First*, we formalise Multi-Adversarial Team Games (MATGs), a generalisation of (single) Adversarial Team Games, enabling the modeling of interactions between a team and multiple independent adversaries. *Second*, building upon the work of [1], we develop a fully polynomial-time approximation scheme (FPTAS) for approximating NE in MATGs. Our algorithm extends the one proposed in [1] for the single-adversary case, preserving its polynomial time complexity in the inverse of the approximation error and the natural parameters of the game, even in the presence of multiple adversaries and so overcoming the *curse of multi-agents* also for the adversarial side. *Finally*, we provide the first empirical evaluation of this family of algorithms for NE computation in both ATGs and MATGs, demonstrating the algorithm's capability to effectively approximate NE in benchmarks with many adversaries.

2 MULTI-ADVERSARIAL TEAM GAMES

A Multi-Adversarial Team Game (MATG), represented in normal form, is defined by a tuple $\Gamma = \left(\mathcal{N}, \mathcal{M}, (\mathcal{A}_i)_{i\in\mathcal{N}}, (\mathcal{B}_j)_{j\in\mathcal{M}}, (U_j)_{j\in\mathcal{M}}\right)$. Γ consists of a finite set of $\mathcal{N} = \{1, .., n\}$ of team agents and a finite set $\mathcal{M} = \{1, .., m\}$ of adversarial agents. Each team member $i \in \mathcal{N}$ has a finite and non empty set of available actions (i.e. pure strategies) $\mathcal{A}_i$, so that $\mathcal{A} := \prod_{i\in\mathcal{N}} \mathcal{A}_i$ denotes the set of all possible action profiles of the team. Similarly, each adversary $j \in \mathcal{M}$ has a finite and nonempty set of actions (i.e. pure strategies) $\mathcal{B}_j$. We denote by $\mathbf{a} = (a_1, \ldots, a_n) \in \mathcal{A}$ the action profile of the team, and by $\mathbf{b} = (\mathbf{b}_1, \ldots, \mathbf{b}_m) \in \mathcal{B} := \prod_{j\in\mathcal{M}} \mathcal{B}_j$ the action profile of the adversaries. Each adversary $j \in \mathcal{M}$ has a payoff function[3] denoted by $U_j : \mathcal{A}\times\mathcal{B}_j \to [0, 1]$. All team members $i \in \mathcal{N}$ have identical payoff functions $U_i : \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ where $U_i(\mathbf{a}, \mathbf{b}) := U_{team}(\mathbf{a}, \mathbf{b}), \forall i \in \mathcal{N}$. The game is zero-sum in the sense $\sum_{j\in\mathcal{M}} U_j(\mathbf{a}, \mathbf{b}_j) + U_{team}(\mathbf{a}, \mathbf{b}) = 0$. We assume that each adversary $j \in \mathcal{M}$ maximises their individual rewards U_j, while the team minimises $\sum_{j\in\mathcal{M}} U_j$.

A team mixed strategy profile is defined as $\mathbf{x} = (\mathbf{x}_i)_{i\in\mathcal{N}}$, where $\mathbf{x}_i \in \mathcal{X}_i = \Delta(\mathcal{A}_i)$ is the team member i's mixed strategy and $\Delta(\mathcal{A}_i)$ is the set of all probability distributions over $\mathcal{A}_i$. Similarly, an adversary mixed strategy profile is defined as $\mathbf{y} = (\mathbf{y}_j)_{j\in\mathcal{M}}$, where $\mathbf{y}_j \in \mathcal{Y}_j = \Delta(\mathcal{B}_j)$ is the adversary member j's mixed strategy. For convenience, we will write $\mathcal{X} := \prod_{i\in\mathcal{N}} \mathcal{X}_i$ and $\mathcal{Y} := \prod_{j\in\mathcal{M}} \mathcal{Y}_j$ for the space of mixed strategy profiles of the team and the adversaries, respectively. Finally, we overload notation so that U_j is not only the payoff function of adversary j but also the mixed extension of such payoff: $U_j(\mathbf{x}, \mathbf{y}_j) = \mathbb{E}_{(\mathbf{a},\mathbf{b}_j)\sim(\mathbf{x},\mathbf{y}_j)}[U_j(\mathbf{a}, \mathbf{b}_j)]$. We will also write poly(Γ) for factors that are polynomial in the natural parameters of the game Γ. As is customary, $-i$ denotes the set containing all the team members except agent i.

[3] Without loss of generality, we assume adversary payoffs belong to $[0, 1]$.

In terms of solution concepts, we focus on computing a set of approximate best responses that form an approximate Nash Equilibrium, defined as follows.

NE-GAP. For each joint strategy $(\mathbf{x}, \mathbf{y})$, we define the Nash Equilibrium Gap (NE-GAP) of some team member's strategy $\mathbf{x}_i$ for some $i \in \mathcal{N}$ (resp. $\mathbf{y}_j$ for some adversary $j \in \mathcal{M}$) as

$$\text{NE-GAP}_i^{team}(\mathbf{x}, \mathbf{y}) = \max_{\mathbf{a}_i\in\mathcal{A}_i} (U_{team}(\mathbf{a}_i, \mathbf{x}_{-i}, \mathbf{y}) - U_{team}(\mathbf{x}, \mathbf{y})) \quad (1)$$

$$\text{NE-GAP}_j^{adv}(\mathbf{x}, \mathbf{y}_j) = \max_{\mathbf{b}_j\in\mathcal{B}_j} (U_j(\mathbf{x}, \mathbf{b}_j) - U_i(\mathbf{x}, \mathbf{y}_j)) \quad (2)$$

The NE-GAP of the joint strategy is then defined as

$$\text{NE-GAP}(\mathbf{x}, \mathbf{y}) = \max\left\{\max_{i\in\mathcal{N}} \text{NE-GAP}_i^{team}(\mathbf{x}, \mathbf{y}), \max_{j\in\mathcal{M}} \text{NE-GAP}_j^{adv}(\mathbf{x}, \mathbf{y}_j)\right\} \quad (3)$$

Approximate best response (ε-BR). A mixed strategy $\mathbf{x}_i$ of team agent i is an ε-best response (ε-BR) to a strategy profile $(\mathbf{x}_{-i}, \mathbf{y})$ iff $\text{NE-GAP}_i^{team}(\mathbf{x}_i, \mathbf{x}_{-i}, \mathbf{y}) \leq \varepsilon$. Similarly, a mixed strategy $\mathbf{y}_j$ of adversary j is an ε-BR to a team strategy profile $\mathbf{x}$ iff $\text{NE-GAP}_j^{adv}(\mathbf{x}, \mathbf{y}_j) \leq \varepsilon$. We denote by $BR_i((\mathbf{x}_{-i}, \mathbf{y}); \varepsilon)$ the set of ε-BR strategies of team agent i to $(\mathbf{x}_{-i}, \mathbf{y})$, and $BR_j(\mathbf{x}; \varepsilon)$ the set of ε-BR strategies of adversary j to $\mathbf{x}$.

ε-Nash Equilibrium. The strategy profile $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$ is an ε-Nash Equilibrium of a MATG Γ for an $\varepsilon \geq 0$ if and only if the strategy of every agent is an ε−BR to the strategies of other agents. Formally, $\forall i \in \mathcal{N}, \mathbf{x}_i \in BR_i((\mathbf{x}_{-i}, \mathbf{y}); \varepsilon)$ and $\forall j \in \mathcal{M}, \mathbf{y}_j \in BR_j(\mathbf{x}; \varepsilon)$. Equivalently, a strategy $(\mathbf{x}, \mathbf{y})$ is an ε-NE iff $\text{NE-GAP}(\mathbf{x}, \mathbf{y}) \leq \varepsilon$.

Table 1: Payoff tables for the first adversary (top) and second adversary (bottom) for a MATG with 2 teammates, 2 adversaries, 2 actions each.

$\mathbf{b}_1$ \ a_1, a_2	0,0	0,1	1,0	1,1
0	0	2/5	2/5	4/5
1	1/5	1/10	1/10	0

$\mathbf{b}_2$ \ a_1, a_2	0,0	0,1	1,0	1,1
0	0	1/5	1/5	2/5
1	3/5	3/10	3/10	0

Example MATG (payoffs shown in Table 1): a team of $n = 2$ agents defends 2 locations against $m = 2$ adversaries. Each adversary has distinct rewards for each location. Capturing an adversary requires both team agents to be in the same cell; if only one is present, the adversary escapes with a reduced reward. The strategies $x_1 = [4/5, 1/5] = x_2, y_1 = [4/5, 1/5], y_2 = [0, 1]$ form a 0-NE (i.e. an exact NE).

3 FPTAS FOR APPROXIMATE NE IN MATGS

This section presents the Multi-Adversarial Team Games Gradient Descent Max (MATG-GDM) algorithm that computes approximate NE in MATGs. It also states our main theoretical result: MATG-GDM computes an ε−NE with computational complexity that is polynomial in the natural parameters of the game and in $1/\varepsilon$. This result is formalised in Theorem 3 and proved in Section 4.

The proposed Algorithm 1 takes as input a MATG Γ, an approximation error $\varepsilon > 0$, a learning rate $\eta > 0$, and some initial team strategy profile $\mathbf{x}^{(0)} \in \mathcal{X}$.

Each iteration starts by calculating the best responses of each adversary $(\mathbf{b}_j^{(t)})_{j\in\mathcal{M}}$, based on the current team strategy profile (Lines 3-5). Next, each team member takes a projected gradient descent step (Lines 6-8) using the adversary best responses. Note that $\Pi_{\mathcal{X}_i}(\cdot)$ denotes the Euclidean projection onto $\mathcal{X}_i$.

Now, based on the updated team strategy $\mathbf{x}^{(t)}$, the adversaries' response is computed by *ExtendNE*($\mathbf{x}^{(t)}$), which returns the values of the vector variables $\mathbf{y} = (\mathbf{y_j}(\mathbf{b}_j))_{j\in\mathcal{M},\mathbf{b}_j\in\mathcal{B}_j}$, solution to the following LP:

$$\begin{aligned}
\max_{\mathbf{y},\{\mathbf{z}_i\}_{i\in\mathcal{N}}} \quad & \sum_{i\in\mathcal{N}} \mathbf{z}_i && (4)\\
\text{s.t.} \quad & \mathbf{z}_i - \sum_{j\in\mathcal{M}} \sum_{\mathbf{b}_j\in\mathcal{B}_j} \mathbf{y_j}(\mathbf{b}_j)\cdot U_j(\mathbf{a}_i,\mathbf{x}_{-i}^{(t)},\mathbf{b}_j) \le 0, \forall i\in\mathcal{N},\ \mathbf{a}_i\in\mathcal{A}_i,\\
& \sum_{\mathbf{b}_j\in\mathcal{B}_j} \mathbf{y_j}(\mathbf{b}_j) = 1, \quad \forall j\in\mathcal{M},\\
& \mathbf{y_j}(\mathbf{b}_j) \ge 0, \quad \forall j\in\mathcal{M},\ \mathbf{b}_j\in\mathcal{B}_j.
\end{aligned}$$

Lastly, MATG-GDM checks if the strategy $(\mathbf{x}^{(t)},\mathbf{y}^{(t)})$ is an ε-NE, terminating if it is the case (Lines 10-11).

Algorithm 1 MATG-Gradient Descent Max

1: **Input:** MATG Γ, approximation error $\varepsilon > 0$, learning rate $\eta > 0$, initial team strategies $\mathbf{x}^{(0)} \in \mathcal{X}$.
2: **repeat**
3:     **for** $j \in \mathcal{M}$ **do**
4:         $\mathbf{b}_j^{(t)} \leftarrow \arg\max_{\mathbf{b}_j\in\mathcal{B}_j} U_j(\mathbf{x}^{(t-1)},\mathbf{b}_j)$
5:     **end for**
6:     **for** $i \in \mathcal{N}$ **do**
7:         $\mathbf{x}_i^{(t)} \leftarrow \Pi_{\mathcal{X}_i}\left(\mathbf{x}_i^{(t-1)} - \eta\nabla_{\mathbf{x}_i}\sum_{j\in\mathcal{M}} U_j(\mathbf{x}^{(t-1)},\mathbf{b}_j^{(t)})\right)$
8:     **end for**
9:     $(\mathbf{y}_j^{(t)})_{j\in\mathcal{M}} \leftarrow ExtendNE(\mathbf{x}^{(t)})$
10: **until** $(\mathbf{x}^{(t)},(\mathbf{y}_j^{(t)})_{j\in\mathcal{M}})$ is an ε-NE
11: **return** $(\mathbf{x}^{(t)},(\mathbf{y}_j^{(t)})_{j\in\mathcal{M}})$

In order to prove that each iteration of the algorithm runs in polynomial time, we assume that the utilities of mixed strategies can be computed in polynomial time. In general multi-player games, the representation of each player's payoffs scales exponentially with the total number of players. We thus restrict our analysis to games with the *Polynomial Expectation Property*, formalised in Assumption 1. This assumption is known to hold for most classes of succintly representable games [25].

Assumption 1 (Polynomial Expectation Property). *For any (mixed) joint strategy profile* $(\mathbf{x},\mathbf{y}_j) \in \mathcal{X}\times\mathcal{Y}_j$, $j \in \mathcal{M}$ *we can compute (exactly) the expectation* $U_j(\mathbf{x},\mathbf{y}_j) = \mathbb{E}_{(\mathbf{a},\mathbf{b}_j)\sim(\mathbf{x},\mathbf{y_j})}U_j(\mathbf{a},\mathbf{b}_j)$ *in time* $\text{poly}(n,\sum_{i\in\mathcal{N}}|\mathcal{A}_i|,|\mathcal{B}_j|,|\mathbf{x}|,|\mathbf{y}_j|)$.

Then, we can show the following proposition:

Proposition 2. *Under Assumption 1, every iteration of MATG-GDM can be computed in time* $poly(\Gamma)$.

Proof. Lines **3-5** compute a best response for each adversary j by computing $\max_{\mathbf{b}_j\in\mathcal{B}_j} U_j(\mathbf{x}^{(t-1)},\mathbf{b}_j)$, taking $|\mathcal{B}_j|$ calls. Therefore, this remains polynomial in Γ.
Lines **6-8** perform a projected gradient descent step for each team member $i \in \mathcal{N}$. By multi-linearity of $U_j(\mathbf{x}^{(t-1)})$, we have that $\frac{\partial}{\partial \mathbf{x}_i(\mathbf{a}_i)} U_j(\mathbf{x}^{(t-1)},\mathbf{b}_j^{(t)}) = U_j(\mathbf{a}_i,\mathbf{x}_{-i}^{(t-1)},\mathbf{b}_j^{(t)})$. This can be computed in polynomial time by Assumption 1. In addition, computing $\Pi_{\mathcal{X}_i}$ takes $O(|\mathcal{A}_i|\log|\mathcal{A}_i|)$ time [31]. Repeating this step for every $i\in\mathcal{N}$ still takes $\text{poly}(\Gamma)$ time.
In Line **9**, *ExtendNE* solves a LP with $\sum_{j\in\mathcal{M}}|\mathcal{B}_j| + n$ variables and $\sum_{i\in\mathcal{N}}|\mathcal{A}_i| + \sum_{j\in\mathcal{M}}|\mathcal{B}_j| + m$ constraints. Moreover, computing each coefficient $U_j(\mathbf{a}_i,\mathbf{x}_{-i}^{(t)},\mathbf{b}_j)$ takes polynomial time under Assumption 1. Therefore, the LP can be constructed and solved in $\text{poly}(\Gamma)$ time.
Line **10** checks whether $(\mathbf{x}^{(t-1)},\mathbf{y}^{(t-1)})$ is an ε–NE. Using the definition of a NE-gap, this can be checked in polynomial time by virtue of Assumption 1. □

We are now ready to formulate our main theoretical result:

Theorem 3 (FPTAS). *Consider any precision* $\varepsilon > 0$. *For any MATG* Γ, *MATG-GDM with a learning rate* $\eta = \Theta(\varepsilon^2)$ *yields an* ε*-NE after at most* $\text{poly}(\Gamma)/\varepsilon^4$ *iterations. Further, by Proposition 2, each iteration of MATG-GDM, under Assumption 1, can be implemented in polynomial time.*

4 PROOF OF THE MAIN THEOREM

In this section, we demonstrate that Algorithm 1, as stated in Theorem 3, is an FPTAS for approximating NE in MATGs. The proof is structured as follows.

First, in Section 4.1 we prove that, for any $\varepsilon \ge 0$, an ε-NE of an MATG Γ can be computed by marginalising an ε-NE of a *correlated-adversaries* transformed game. This *correlated-adversaries* game, denoted Γ^{ca}, is an ATG where all adversaries in the MATG correlate actions and share payoffs i.e. this is equivalent to playing with a single macro-adversary.

Since a *correlated-adversaries* MATG is an ATG, we can use GradientDescentMax (GDM) [1] to compute an ε-approximate NE in time $\text{poly}(\Gamma^{ca})/\varepsilon^4$. However, the action space of the macro-adversary in Γ^{ca} grows exponentially with the number of adversaries in the original MATG, and so, GDM should take exponential time.

We therefore show in Section 4.2 that for *correlated-adversaries* MATG, GDM will terminate after a number of iterations bounded by $\text{poly}(\Gamma)/\varepsilon^4$ (and not $\text{poly}(\Gamma^{ca})/\varepsilon^4$), hence overcoming the exponential dependence on the number of adversaries. Then, in Section 4.3, we show that MATG-GDM applied to Γ produces identical iterates as GDM applied to Γ^{ca}, allowing us to prove Theorem 3.

4.1 Correlated-adversaries transformation

This section establishes that any ε-Nash equilibrium (for $\varepsilon \ge 0$) of the correlated-adversaries transformation of a MATG induces an ε-Nash equilibrium of the original MATG. Let us first define the *correlated-adversaries* MATG (CA-MATG) for a given MATG:

Definition 1 (Correlated-Adversaries MATG). *Given MATG* $\Gamma = \left(\mathcal{N},\mathcal{M},(\mathcal{A}_i)_{i\in\mathcal{N}},(\mathcal{B}_j)_{j\in\mathcal{M}},(U_j)_{j\in\mathcal{M}}\right)$ *we define its CA-MATG*

$\Gamma^{ca} = \left(\mathcal{N}, (\mathcal{A}_i)_{i\in\mathcal{N}}, \mathcal{B}^{ca} = \prod_{j\in\mathcal{M}} \mathcal{B}_j, U^{ca} : \mathcal{A}\times\mathcal{B}^{ca} \to \mathbb{R}\right)$ *where* $U^{ca}(\mathbf{a},\mathbf{b}) = \sum_{j\in\mathcal{M}} U_j(\mathbf{a},\mathbf{b}_j)$ *and* $U^{ca}_{team}(\mathbf{a},\mathbf{b}) = -U^{ca}(\mathbf{a},\mathbf{b})$*, in which all adversaries play correlated strategies and share payoffs additively.*

Table 2: Payoff table for the correlated adversary in the CA-transformation of the MATG in Table 1.

$\mathbf{b}^{ca}$ \ a_1, a_2	0,0	0,1	1,0	1,1
0,0	0	3/5	3/5	6/5
0,1	3/5	7/10	7/10	4/5
1,0	1/5	3/10	3/10	2/5
1,1	4/5	2/5	2/5	0

To illustrate this transformation, Table 2 presents the payoff table of the correlated adversary for the CA-MATG of the MATG in Table 1.

It will also be useful to translate back and forth between correlated strategies of the CA-MATG and independent mixed strategies of adversaries in the original MATG. To do so, we define *correlated* and *marginalised adversary strategies* :

Definition 2 (Correlated adversary strategies). *Given a mixed strategy profile* $(\mathbf{y_j})_{j\in\mathcal{M}} \in \mathcal{Y}$ *for adversaries in an MATG, we define the corresponding correlated adversary strategy as* $\mathbf{y} = \otimes_{j\in\mathcal{M}}\mathbf{y_j}$ *defined as:* $\mathbf{y}(\mathbf{b}) = \prod_{j\in\mathcal{M}} \mathbf{y_j}(\mathbf{b}_j), \ \ \forall \mathbf{b} = (\mathbf{b}_1,\ldots,\mathbf{b}_m) \in \mathcal{B}$.

Definition 3 (Marginalised adversary strategies). *Given a (mixed) correlated strategy* $\mathbf{y} \in \mathcal{Y}^{ca}$ *for adversaries in a CA-MATG, we define the set of marginalised adversary strategies as* $\tilde{\mathbf{y}} = (\tilde{\mathbf{y}}_j)_{j\in\mathcal{M}}$ *defined as:* $\tilde{\mathbf{y}}_j(\mathbf{b}_j) = \sum_{\mathbf{b}_{-j}\in\mathcal{B}_{-j}} \mathbf{y}(\mathbf{b}_j,\mathbf{b}_{-j}), \forall j \in \mathcal{M}, \forall \mathbf{b}_j \in \mathcal{B}_j$.

We first show that for the adversaries, the value of any mixed strategy in a CA-MATG equals the sum of the values of the marginalised strategies in the original MATG.

Proposition 4. *Let a MATG* Γ *and its CA-MATG* Γ^{ca} *be given. Further, let any mixed strategy* $(\mathbf{x},\mathbf{y}) \in \mathcal{X}\times\mathcal{Y}^{ca}$ *of* Γ^{ca} *be given. Then,*

$$U^{ca}(\mathbf{x},\mathbf{y}) = \sum_{j\in\mathcal{M}} U_j(\mathbf{x},\tilde{\mathbf{y}}_j). \tag{5}$$

Proof.

$$\begin{aligned} U^{ca}(\mathbf{x},\mathbf{y}) &= \sum_{j\in\mathcal{M}} U_j(\mathbf{x},\mathbf{y}) \\ &= \sum_{j\in\mathcal{M}}\sum_{\mathbf{a}\in\mathcal{A}} \mathbf{x}(\mathbf{a}) \sum_{\mathbf{b}\in\mathcal{B}^{ca}} \mathbf{y}(\mathbf{b})\cdot U_j(\mathbf{a},\mathbf{b}_j) \\ &= \sum_{\mathbf{a}\in\mathcal{A}} \mathbf{x}(\mathbf{a}) \sum_{j\in\mathcal{M}}\sum_{\mathbf{b}_j\in\mathcal{B}_j}\sum_{\mathbf{b}_{-j}\in\mathcal{B}_{-j}} \mathbf{y}(\mathbf{b}_j,\mathbf{b}_{-j})\cdot U_j(\mathbf{a},\mathbf{b}_j) \\ &= \sum_{\mathbf{a}\in\mathcal{A}} \mathbf{x}(\mathbf{a}) \sum_{j\in\mathcal{M}}\sum_{\mathbf{b}_j\in\mathcal{B}_j} \tilde{\mathbf{y}}_j(\mathbf{b}_j)\cdot U_j(\mathbf{a},\mathbf{b}_j) \\ &= \sum_{j\in\mathcal{M}} U_j(\mathbf{x},\tilde{\mathbf{y}}_j) \end{aligned}$$

□

We also have the following corollary:

Corollary 5. *Given an MATG* Γ*, its CA-MATG* Γ^{ca} *and the mixed strategies* $\left(\mathbf{x}, (\mathbf{y_j})_{j\in\mathcal{M}}\right) \in \mathcal{X}\times\mathcal{Y}$*, we have that*

$$U^{ca}(\mathbf{x},\mathbf{y} = \otimes_{j\in\mathcal{M}}\mathbf{y_j}) = \sum_{j\in\mathcal{M}} U_j(\mathbf{x},\mathbf{y_j}). \tag{6}$$

Proof. This follows from Proposition 4 and by noting that the marginal strategies of $\mathbf{y} = \otimes_{j\in\mathcal{M}}\mathbf{y_j}$ are $(\mathbf{y_j})_{j\in\mathcal{M}}$. □

The next step is to show that for each adversary in an MATG, we can build an ε-BR from a correlated ε-BR in the corresponding CA-MATG.

Proposition 6. *Consider for any* $\varepsilon \geq 0$*, an MATG* Γ *and its CA-MATG* Γ^{ca}*. Let* $\mathbf{x} \in \mathcal{X}$ *be a team strategy profile,* $\mathbf{y} \in \mathcal{Y}^{ca}$ *be a (correlated)* ε*-BR to* $\mathbf{x}$ *in* Γ^{ca} *and* $(\tilde{\mathbf{y}}_j)_{j\in\mathcal{M}}$ *be the set of marginalised strategies of* $\mathbf{y}$*. Then, for all adversaries* $j \in \mathcal{M}$*,* $\tilde{\mathbf{y}}_j$ *is an* ε*-BR of adversary* j *to* $\mathbf{x}$ *in* Γ*.*

Proof. Given that $\mathbf{y}$ is an ε-BR to $\mathbf{x}$ in Γ^{ca} it satisfies:

$$U^{ca}(\mathbf{x},\mathbf{y}) \geq U^{ca}(\mathbf{x},\mathbf{y}') - \varepsilon \quad \forall \mathbf{y}' \in \mathcal{Y}^{ca} \tag{7}$$

which by Proposition 4 is equivalent to:

$$\sum_{j\in\mathcal{M}} U_j(\mathbf{x},\tilde{\mathbf{y}}_j) \geq \sum_{j\in\mathcal{M}} U_j(\mathbf{x},\tilde{\mathbf{y}}'_j) - \varepsilon \quad \forall \mathbf{y}' \in \mathcal{Y}^{ca}, \tag{8}$$

where the $(\tilde{\mathbf{y}}_j)_{j\in\mathcal{M}}$ and $\left(\tilde{\mathbf{y}}'_j\right)_{j\in\mathcal{M}}$ are the marginalised strategies of $\mathbf{y}$ and $\mathbf{y}'$ respectively. Now, for some $k \in \mathcal{M}$, define $\mathbf{y}' = (\mathbf{y}'_k, \tilde{\mathbf{y}}_{-k})$ for any arbitrary deviation $\mathbf{y}'_k \in \mathcal{Y}_k$. Noting that $\mathbf{y}'(\mathbf{b}) = \mathbf{y}'_k(\mathbf{b}_k) \times \prod_{j\in\mathcal{M}\setminus\{k\}} \tilde{\mathbf{y}}_\mathbf{j}(\mathbf{b}_j)$, we have

$$\sum_{j\in\mathcal{M}} U_j(\mathbf{x},\tilde{\mathbf{y}}'_j) = \sum_{j\neq k} U_j(\mathbf{x},\tilde{\mathbf{y}}_j) + U_k(\mathbf{x},\mathbf{y}'_k). \tag{9}$$

Then, by subtracting $\sum_{j\neq k} U_j(\mathbf{x},\tilde{\mathbf{y}}_j)$ in both sides of (8), we get:

$$U_k(\mathbf{x},\tilde{\mathbf{y}}_k) \geq U_k(\mathbf{x},\mathbf{y}'_k) - \varepsilon, \forall \mathbf{y}'_k \in \mathcal{Y}_k. \tag{10}$$

That is, $\tilde{\mathbf{y}}_\mathbf{k}$ is an ε-BR of adversary k to $\mathbf{x}$ in Γ, $\forall k \in \mathcal{M}$. □

Finishing the example given by Table 2, the NE of this CA-MATG is (x_1, x_2, y^{ca}), where $y^{ca} = [0, 4/5, 0, 1/5]$. It is straightforward to verify that the marginalized adversary strategies from y^{ca} are the NE adversary strategies of the original MATG in Table 1.

Finally, we state and prove the main result of this Section.

Theorem 7. *Consider any* $\varepsilon \geq 0$*. Let a MATG* Γ *and let an* ε*-NE* $(\mathbf{x},\mathbf{y})$ *of its CA-MATG,* Γ^{ca}*. Then,* $\mathbf{x}$ *and the marginalised adversary strategies derived from* $\mathbf{y}$*,* $\tilde{\mathbf{y}}$*, form an* ε*-NE of* Γ*.*

Proof. $(\mathbf{x},\mathbf{y})$ is an ε-NE of Γ^{ca} implies that $\forall i \in \mathcal{N} : \mathbf{x}_i \in BR_i((\mathbf{x_{-i}},\mathbf{y});\varepsilon)$ in Γ^{ca}. Since (e.g. from Prop. 4), the values of the team are identical in Γ and Γ^{ca}, $\forall i \in \mathcal{N} : \mathbf{x}_i \in BR_i((\mathbf{x}_{-i},\tilde{\mathbf{y}});\varepsilon)$ in Γ. Then, from Prop. 6, we get that $\forall j \in \mathcal{M} : \tilde{\mathbf{y}}_\mathbf{j} \in BR_j(\mathbf{x};\varepsilon)$ in Γ. □

By Theorem 7, and noting that a CA-MATG is an ATG, GDM can compute an ε-NE for MATGs in poly$(\Gamma^{ca})/\varepsilon^4$ time (Refer Corollary 3.2, [1]). However, the adversary action space in a CA-MATG, $\mathcal{B} = \prod_{j\in\mathcal{M}} \mathcal{B}_j$, grows exponentially with the number of adversaries in the original MATG. Consequently, both the sufficient number of iterations and the per-iteration complexity of GDM become superpolynomial in the size of the original MATG. In the following sections, we show how to overcome these limitations by leveraging the structural properties of CA-MATGs.

4.2 Bounding the number of iterations

This section (in Theorems 10 and 11) extends the proof of [1] – specifically Theorems 3.1 and 3.3 therein – to the case of CA-MATGs. Finally, Corollary 12 states that a polynomial number of iterations (in the size of Γ and in $1/\varepsilon$) of GDM suffices to compute an ε-NE in CA-MATGs.

Consider a MATG Γ and its CA-MATG Γ^{ca}. We define the team's objective function to minimize, ϕ, as:

$$\phi(\mathbf{x}) = \max_{\mathbf{y}\in\mathcal{Y}} U^{ca}(\mathbf{x},\mathbf{y}) = \max_{\mathbf{y}\in\mathcal{Y}} \sum_{j\in\mathcal{M}} U_j(\mathbf{x},\mathbf{y}_j) \tag{11}$$

Next Lemmas establish that U^{ca} is L-Lipschitz continuous and ℓ-smooth (Lemma 8), and that the Moreau envelope of ϕ is bounded by poly(Γ) factors (Lemma 9). Proofs are deferred to Appendix[2] B.1.

Lemma 8. *Consider any MATG Γ and its CA-MATG Γ^{ca}. Then, function $U^{ca}(\mathbf{x},\mathbf{y})$ is L-Lipschitz continuous and ℓ-smooth, with $L = \left(\sum_{j\in\mathcal{M}} \sqrt{\sum_{i\in\mathcal{N}} |\mathcal{A}_i| + |\mathcal{B}_j|}\right)$ and $\ell = \left(m \cdot \sum_{i\in\mathcal{N}} |\mathcal{A}_i| + \sum_{j\in\mathcal{M}} |\mathcal{B}_j|\right)$.*

Then we define the Moreau envelope of function $\phi(\cdot)$ with parameter $\lambda = \frac{1}{2\ell}$ as:

$$\phi_{/2\ell}(\mathbf{x}) = \min_{\mathbf{x}'\in\mathcal{X}} \phi(\mathbf{x}') + \ell||\mathbf{x}-\mathbf{x}'||_2^2 \tag{12}$$

Lemma 9. *Consider any MATG Γ and its CA-MATG Γ^{ca}. Then, $\phi_{/2\ell}(\mathbf{x})$ is bounded by* $\text{poly}(\Gamma) := \text{poly}(m, \sum_{i\in\mathcal{N}} |\mathcal{A}_i|, \sum_{j\in\mathcal{M}} |\mathcal{B}_j|)$.

Next, Theorem 10 refines the proof of the extendability of an approximate stationary point to an ε-NE given in [1] (Theorem 3.1 therein), showing that for a CA-MATG, the approximation depends on constants that are polynomial in the natural parameters of the original MATG Γ. Building on this extendability result, Theorem 11 then establishes that the Moreau envelope evaluated along two consecutive per-iteration solutions of GDM before termination[4] decreases monotonically by a factor[5] $\Omega(\varepsilon^4)$. Then, Corollary 12 concludes that, since the Moreau envelope is bounded by poly(Γ) factors, the maximum number of GDM iterations before termination is polynomial in Γ and $1/\varepsilon$.

Theorem 10. *Consider a MATG Γ and its CA-MATG Γ^{ca}. Let $\mathbf{x}^* \in \mathcal{X}$ be a sufficiently small $(\varepsilon/poly(\Gamma))$-near stationary point of function ϕ. Then, there exists a strategy for the adversary $\mathbf{y}^* \in \mathcal{Y}$ so that $(\mathbf{x}^*, \mathbf{y}^*)$ is an ε-NE in Γ^{ca}.*

Proof. Since Γ^{ca} is an ATG, Theorem 3.1 from [1] applies directly: any sufficiently small $(\varepsilon/\text{poly}(\Gamma^{ca}))$-near stationary point extends to an ε-NE in Γ^{ca}. Crucially, the poly(Γ^{ca}) factor in the theorem arises only from the ℓ-smoothness and L-Lipschitz constants of U^{ca}, which, for a CA-MATG, are bounded by poly(Γ) rather than poly(Γ^{ca}), as established in Lemma 8 above. □

Theorem 11. *Consider any $\varepsilon > 0$, $\mathbf{x}^{(0)} \in \mathcal{X}$ and any MATG Γ and its CA-MATG Γ^{ca}. Then* GDM *applied to $(\Gamma^{ca}, \varepsilon, \eta = \Theta(\varepsilon^2), \mathbf{x}^{(0)})$, satisfies*

$$\phi_{/2\ell}(\mathbf{x}^{(t)}) \le \phi_{/2\ell}(\mathbf{x}^{(t-1)}) - \Omega(\varepsilon^4) \tag{13}$$

for any iteration t prior to termination[4].

Since the proof closely follows the argument of Theorem 3.3 in [1] – additionally showing that in the case of a CA-MATG all poly(Γ^{ca}) factors can be replaced by poly(Γ) factors – we provide a sketch of the proof, with full details deferred to Appendix[2] B.1.

Proof. Since Γ^{ca} is an ATG, Theorem 3.3 from [1] applies directly. Now, the proof in [1] relies solely on: (i) Theorem 10, which shows that an $(\varepsilon/poly(\Gamma))$-near stationary point can be extended to an ε-NE. (ii) the ℓ – smoothness and L-Lipschitz continuity of U^{ca} – proved in Lemma 8 both bounded by $poly(\Gamma)$. □

Corollary 12. *Consider a MATG Γ and its CA-MATG Γ^{ca}. Let $\varepsilon > 0$, and $\mathbf{x}^{(0)} \in \mathcal{X}$. Then* GDM *applied to $(\Gamma^{ca}, \varepsilon, \eta = \Theta(\varepsilon^2), \mathbf{x}^{(0)})$, will terminate after at most* $\text{poly}(\Gamma)/\varepsilon^4$ *iterations.*

Proof. The proof follows from: (1) Theorem 11 states that the Moreau-envelope $\phi_{/2\ell}$ of GDM per-iteration solutions has a decrease per iteration of $\Omega(\varepsilon^4)$ until termination[4]; and (2) from Lemma 9 that states that function $\phi_{/2\ell}$ is bounded by poly(Γ). □

4.3 Per-Iteration solutions of MATG-GDM and GDM

In this section, we show that applying MATG-GDM on Γ yields identical iterates as a similarly initialised GDM algorithm applied to Γ^{ca}. This allows us to finally prove Theorem 3.

Proposition 13. *Consider a MATG Γ and its CA-MATG Γ^{ca}. Let $\varepsilon > 0$, $\eta > 0$ and $\mathbf{x}^{(0)} \in \mathcal{X}$. Then, for any iteration t before termination, the team strategy profiles, $\mathbf{x}^{(t)}$, computed by MATG-GDM applied to $(\Gamma, \varepsilon, \eta, \mathbf{x}^{(0)})$ are identical to those computed by GDM applied to $(\Gamma^{ca}, \varepsilon, \eta, \mathbf{x}^{(0)})$.*

Proof. We proceed by induction on t. The base case ($t = 0$) trivially holds since both algorithms are initialized with the same strategy $\mathbf{x}^{(0)}$. For some inductive step $t \ge 1$, assume the strategies at the previous step are equal (to $\mathbf{x}^{(t-1)}$). At iteration t, both algorithms perform a projected gradient descent step with respect to the same function $\sum_{j\in\mathcal{M}} U_j(\mathbf{x}^{(t-1)}, \mathbf{b}_j^{(t)})$, where $\mathbf{b}_j^{(t)}$ is the adversary j BR to $\mathbf{x}^{(t-1)}$. Notice that

$$\max_{\mathbf{b}\in\mathcal{B}} \sum_{j\in\mathcal{M}} U_j\left(\mathbf{x}^{(t-1)}, \mathbf{b}_j\right) = \sum_{j\in\mathcal{M}} \max_{\mathbf{b}_j\in\mathcal{B}_j} U_j\left(\mathbf{x}^{(t-1)}, \mathbf{b}_j\right). \tag{14}$$

The values $U_j\left(\mathbf{x}^{(t-1)}, \mathbf{b}_j^{(t)}\right)$ are the same in GDM and MATG-GDM at iteration t, and consequently, the resulting updates produced by the projected gradient step are also identical (to $\mathbf{x}^{(t)}$). □

Proposition 14. *Consider a MATG Γ and its CA-MATG Γ^{ca}. Let $\mathbf{x}^{(t)} \in \mathcal{X}$ be any team strategy profile. Let $((\mathbf{y}_j^*)_{j\in\mathcal{M}}, \mathbf{z}^* = (\mathbf{z}_i^*)_{i\in\mathcal{N}})$ be an optimal solution to LP (4) from* ExtendNE$(\mathbf{x}^{(t)})$ *in MATG-GDM.*

[4]Termination here refers to the stopping condition of GDM over a CA-MATG.
[5]$O(\cdot)$, $\Omega(\cdot)$ and $\Theta(\cdot)$ are the usual notations for (tight) upper-bound, lower-bound and upper-and-lower-bound in asymptotic analysis.

Then $(\mathbf{y}^* = \otimes_{j\in\mathcal{M}}\mathbf{y}^*_j, \mathbf{z}^*)$ *is an optimal solution to the following LP (15) from* ExtendNE$(\mathbf{x}^{(t)})$ *in GDM:*

$$\max_{\mathbf{y},(\mathbf{z}_i)_{i\in\mathcal{N}}} \sum_{i\in\mathcal{N}} \mathbf{z}_i \tag{15}$$

$$\text{s.t.}\quad \mathbf{z}_i - \sum_{\mathbf{b}\in\mathcal{B}} \mathbf{y}(\mathbf{b}) \cdot U^{ca}(\mathbf{a}_i, \mathbf{x}^{(t)}_{-i}, \mathbf{b}) \le 0,\ \forall i \in \mathcal{N},\ \mathbf{a}_i \in \mathcal{A}_i$$

$$\sum_{\mathbf{b}\in\mathcal{B}} \mathbf{y}(\mathbf{b}) = 1 \quad \text{and} \quad \mathbf{y}(\mathbf{b}) \ge 0,\ \forall \mathbf{b} \in \mathcal{B}$$

Proof. Let $((\mathbf{y}^*_j)_{j\in\mathcal{M}}, \mathbf{z}^* = (\mathbf{z}^*_i)_{i\in\mathcal{N}})$ be an optimal solution to LP (4). First, remark that $(\mathbf{y}^* = \otimes_{j\in\mathcal{M}}\mathbf{y}^*_j, \mathbf{z}^* = (\mathbf{z}^*_i)_{i\in\mathcal{N}})$ is also a solution of LP (15), since

$$\sum_{\mathbf{b}\in\mathcal{B}} \mathbf{y}^*(\mathbf{b})U^{ca}(\mathbf{a}_i, \mathbf{x}^{(t)}_{-i}, \mathbf{b}) = \sum_{\substack{j\in\mathcal{M}\\ \mathbf{b}_j\in\mathcal{B}_j}} \mathbf{y}^*_j(\mathbf{b}_j)U_j(\mathbf{a}_i, \mathbf{x}^{(t)}_{-i}, \mathbf{b}_j),$$

from Proposition 4. Furthermore, $\mathbf{y}^*$ is a normalized probability distribution over $\mathcal{B}$.

In the other direction, let $(\mathbf{y}, \mathbf{z} = (\mathbf{z}_i)_{i\in\mathcal{N}})$ be any solution of LP (15) and let $(\tilde{\mathbf{y}}_j)_{j\in\mathcal{M}}$ be the marginalized strategies of $\mathbf{y}$. By Proposition 4, we also have

$$\sum_{\mathbf{b}\in\mathcal{B}} \mathbf{y}(\mathbf{b})U^{ca}(\mathbf{a}_i, \mathbf{x}^{(t)}_{-i}, \mathbf{b}) = \sum_{\substack{j\in\mathcal{M}\\ \mathbf{b}_j\in\mathcal{B}_j}} \tilde{\mathbf{y}}_j(\mathbf{b}_j)U_j(\mathbf{a}_i, \mathbf{x}^{(t)}_{-i}, \mathbf{b}_j).$$

Thus, $((\tilde{\mathbf{y}}_j)_{j\in\mathcal{M}}, \mathbf{z} = (\mathbf{z}_i)_{i\in\mathcal{N}})$ satisfies the constraints of LP (4). From this fact, $\sum_{i\in\mathcal{N}} \mathbf{z}^*_i \ge \sum_{i\in\mathcal{N}} \mathbf{z}_i$. In other terms, $(\mathbf{y}^*, \mathbf{z}^*) = (\otimes_{j\in\mathcal{M}} (\mathbf{y}^*_j), (\mathbf{z}^*_i)_{i\in\mathcal{N}})$ is an optimal solution to LP (15). □

We are ready to prove Theorem 3.

Proof. (**Theorem 3**). Let Γ^{ca} be the CA-MATG of Γ. By Corollary 12, for a sufficiently large number of iterations $T = \text{poly}(\Gamma, 1/\varepsilon)$, there exists an iteration $t^* \le T$ of GDM applied to $(\Gamma^{ca}, \varepsilon, \eta, \mathbf{x}^{(0)})$ such that $(\mathbf{x}^{(t^*)}, \mathbf{y}^{(t^*)})$ is an ε-NE of Γ^{ca} – where $\mathbf{y}^{(t^*)}$ is a solution to LP (15) instantiated with $\mathbf{x}^{(t^*)}$. Now, by Proposition 13, MATG-GDM applied to $(\Gamma, \varepsilon, \eta, \mathbf{x}^{(0)})$ computes at iteration t^* the same per-iteration team strategy profile than GDM, i.e. $\mathbf{x}^{(t^*)}$. Let $(\mathbf{y}^{(t^*)}_j)_{j\in\mathcal{M}}$ be the set of adversary strategies computed by MATG-GDM to extend $\mathbf{x}^{(t^*)}$ at iteration t^* – i.e. $\left(\mathbf{y}^{(t^*)}_j\right)_{j\in\mathcal{M}}$ is an optimal solution to LP (4) instantiated with $\mathbf{x}^{(t^*)}$. Then, by Proposition 14, $\otimes_{j\in\mathcal{M}}\mathbf{y}^{(t^*)}_j$ is a solution to LP (15) instantiated with $\mathbf{x}^{(t^*)}$. $(\mathbf{x}^{(t^*)}, \otimes_{j\in\mathcal{M}}\mathbf{y}^{(t^*)}_j)$ is thus an ε-NE of Γ^{ca} and, by Theorem 7, it is also an ε-NE of the original MATG Γ, completing the proof. □

5 IMPLICATION FOR TWO-TEAM ADVERSARIAL GAMES

A Two Team Adversarial Game (TTAG) [1], is a normal-form game defined by the tuple $\Gamma^{TT} = \langle \mathcal{N}, \mathcal{M}, (\mathcal{A}_i)_{i\in\mathcal{N}}, (\mathcal{B}_j)_{j\in\mathcal{M}}, U\rangle$ where a Team A of $n = |\mathcal{N}|$ agents (the minimisers) plays against a Team B of $m = |\mathcal{M}|$ agents (the maximisers). Minimiser $i \in \mathcal{N}$ (resp. maximiser $j \in \mathcal{M}$) has action set $\mathcal{A}_i$ (resp. $\mathcal{B}_j$). The game is *two-team adversarial* in that every maximiser receives identical reward $U_B(\mathbf{a}, \mathbf{b}) = U(\mathbf{a}, \mathbf{b})$, while minimisers try to minimize $U(\mathbf{a}, \mathbf{b})$, i.e. they receive $U_A(\mathbf{a}, \mathbf{b}) = -U(\mathbf{a}, \mathbf{b})$. A mixed strategy of a minimiser $i \in \mathcal{N}$ is a distribution $\mathbf{x}_i \in \Delta(\mathcal{A}_i)$. Similarly, a mixed strategy of a maximiser $j \in \mathcal{M}$ is a distribution $\mathbf{y}_j \in \Delta(\mathcal{B}_j)$. As usual, an ε-NE in a TTAG is a joint mixed strategy $\left((\mathbf{x}_i)_{i\in\mathcal{N}}, (\mathbf{y}_j)_{j\in\mathcal{M}}\right)$ from which no team member or adversary has interest (more than ε) to deviate.

In this Section, we partially address the following remark from [1] regarding the case of correlated adversaries strategies:

Remark 15 (Correlated Adversaries, [1], Remark 3). *Another notable application of having a single adversary is the case where the adversary team has multiple players, but with the twist that the adversaries are allowed to correlate their strategies—i.e., the team is facing a "virtual" player. However, in that case the action space of that virtual player grows exponentially with the number of adversaries m, and so establishing polynomial-time algorithms with m requires further work.*

Let us consider the subclass of TTAGs in which the utility table factorises additively over the maximisers: $U(\mathbf{a}, \mathbf{b}) = \sum_{j\in\mathcal{M}} U_j(\mathbf{a}, \mathbf{b}_j)$.

Let $\Gamma^{TT} = \langle \mathcal{N}, \mathcal{M}, (\mathcal{A}_i)_{i\in\mathcal{N}}, (\mathcal{B}_j)_{j\in\mathcal{M}}, U\rangle$ be a member of this class. From Γ^{TT} we can naturally define MATG $\Gamma = \left(\mathcal{N}, \mathcal{M}, (\mathcal{A}_i)_{i\in\mathcal{N}}, (\mathcal{B}_j)_{j\in\mathcal{M}}, (U_j)_{j\in\mathcal{M}}\right)$ and the corresponding CA-MATG $\Gamma^{ca} = \left(\mathcal{N}, (\mathcal{A}_i)_{i\in\mathcal{N}}, \mathcal{B}^{ca} = \prod_{j\in\mathcal{M}} \mathcal{B}_j, U\right)$.

Note that Γ^{TT} and Γ^{ca} only differ by the fact that adversaries should use independent mixed strategies in Γ^{TT}, while they can use joint mixed strategies in Γ^{ca}. However, using the above results, we can prove that MATG-GDM, applied to Γ can be used to provide an ε-NE of Γ^{TT} in polynomial time:

Proposition 16. *Let* Γ^{TT} *be a two-team adversarial game with maximisers allowed to correlate and* Γ *be the corresponding MATG. MATG-GDM applied to* Γ *provides an* ε*-NE of* Γ^{TT} *in time* $\text{poly}(\Gamma^{TT})/\varepsilon^4$.

A complete proof of Proposition[6] 16 is provided in Appendix[2] B.2.

6 EXPERIMENTAL EVALUATION

We conducted experiments on random MATGs in which adversary payoffs were independently uniformly sampled within the interval $[0, 1]$. Each game configuration, denoted as nvm/a, corresponds to an MATG with n team agents, m adversaries, and a available actions per agent. Reported values correspond to the mean and standard deviation computed over 10 independent instances.

MATG-GDM was implemented in Python 3.11 using the JAX and Optax libraries [4, 10][7], initial team strategies $\mathbf{x}^{(0)}$ were set to uniform random distributions and two learning rates $\eta \in \{10^{-2}, 10^{-3}\}$ were evaluated. Experiments were run on a Linux machine with an Intel i7-12700 processor and 64GB of RAM.

6.1 Comparison with existing solvers

This section compares the capabilities and limitations of existing state-of-the-art solvers in computing NE for MATGs with that of the proposed MATG-GDM algorithm. For this comparison, we consider two representative state-of-the-art algorithms: Iterated Polymatrix Approximation (IPA [12], implemented in Gambit [27]) and the Wilson algorithm [11], implemented in GT-Nash[8].

IPA, an approximate solver, belongs to the class of algorithms that operate on the normal-form game representation. As detailed

[6]Referred to as Corollary B.8 in Appendix.

[7]Code provided at [22].

[8]https://forge.inrae.fr/game-theory-tools-group/gtnash

Table 3: Comparison of IPA, Wilson, and MATG-GDM. # reports the number of instances (out of 10) completed within the 30-minute timeout, and *Runtime* the mean ± standard deviation over the solved instances. NE-Gap is the average ε-NE of IPA solutions; Wilson gap omitted (exact solutions); MATG-GDM reported on two fixed gaps $\varepsilon = \{10^{-3}, 10^{-4}\}$.

Game	IPA (Gambit)			Wilson (GT-Nash)		MATG-GDM (ours) ($\varepsilon = 10^{-3}, \eta = 10^{-2}$)		MATG-GDM (ours) ($\varepsilon = 10^{-4}, \eta = 10^{-3}$)	
	#	Runtime (s)	NE-Gap	#	Runtime (s)	#	Runtime (s)	#	Runtime (s)
2v1/2	10	$(1.78 \pm 0.44) \times 10^{-4}$	$(8.64 \pm 9.83) \times 10^{-8}$	9	0.86 ± 0.31	10	2.74 ± 0.32	10	6.11 ± 2.95
2v1/4	10	$(5.90 \pm 1.28) \times 10^{-4}$	$(1.22 \pm 0.43) \times 10^{-7}$	8	22.27 ± 1.03	10	4.40 ± 1.2	10	30.53 ± 23.87
2v1/6	10	$(1.35 \pm 0.22) \times 10^{-3}$	$(1.40 \pm 0.58) \times 10^{-7}$	5	451.81 ± 448.04	10	6.91 ± 1.33	10	78.42 ± 59.35
2v3/2	10	$(5.33 \pm 1.45) \times 10^{-4}$	$(0.98 \pm 1.26) \times 10^{-7}$	6	40.58 ± 37.14	10	3.41 ± 0.57	10	7.21 ± 4.97
2v3/4	10	$(1.07 \pm 0.32) \times 10^{-2}$	$(1.55 \pm 0.76) \times 10^{-7}$	0	-	10	8.51 ± 3.50	10	89.67 ± 80.12
2v3/6	8	0.20 ± 0.28	$(2.08 \pm 1.15) \times 10^{-7}$	0	-	10	12.57 ± 4.54	10	175.86 ± 113.80
2v6/2	8	$(5.29 \pm 1.41) \times 10^{-3}$	$(2.75 \pm 1.06) \times 10^{-7}$	0	-	10	3.45 ± 0.35	10	7.61 ± 3.81
2v6/4	1	23.81 ± 0.0	$(2.38 \pm 0.0) \times 10^{-7}$	0	-	10	6.07 ± 2.05	10	56.84 ± 84.95
2v6/6	0	-	-	0	-	10	15.08 ± 5.66	10	172.04 ± 110.43

in Appendix[2] C.1, representing a MATG in normal form requires space exponential in the total number of players (i.e., n team members plus m adversaries). Like all solvers relying on the normal-form representation, IPA is severely limited by this exponential growth in representation size. For instance, when using float64 representations for the payoffs, encoding a 3v6/6 MATG in normal form requires approximately 1GB of memory, compared to only 52KB in the original MATG representation.

In contrast to IPA, Wilson is an exact solver for hypergraphical games, a representation which retains the same size as the original MATG representation (See Appendix[2] C.1). However, its computational complexity is doubly-exponential in the maximum number of players in any local game of the hypergraphical representation (Proposition 5.4 in [11]). For an MATG, this corresponds to $n + 1$.

Given the limitations of existing solvers for MATGs, our comparative analysis is restricted to games with the minimal team size ($n = 2$). Within this setting, we evaluate the impact of varying the number of adversaries ($m \in \{1, 3, 6\}$) and the number of actions per player ($a \in \{2, 4, 6\}$). Results are summarized in Table 3.

The first key observation is that both IPA and Wilson fail to scale with respect to either the number of adversaries or the number of actions, as indicated by execution times that grow exponentially in both parameters. In particular, we observe that Wilson, which computes exact NEs, solves most 2v1 instances in roughly 30 seconds; however, it already times out on instances larger than 2v3/4. On the other hand, IPA produces solutions with a negligible gap (NE-Gap $\approx 10^{-7}$) for instances solved before timeout and is much faster than Wilson and MATG-GDM. However, it times out on all 2v6/6 instances.

Finally, MATG-GDM, as an approximate algorithm that guarantees finding an ε-NE in polynomial time, achieves a favorable empirical trade-off between the approximation error ε and runtime. In particular, MATG-GDM finds solutions with a gap of at most $\varepsilon = 10^{-3}$ within seconds, and with a gap of at most $\varepsilon = 10^{-4}$ in a few minutes for all tested instances.

Importantly, unlike other benchmarked solvers, the runtime of MATG-GDM appears independent of the number of adversaries — it depends on the approximation parameter ε, but not on m. As shown in Table 3, for configurations with the same number of actions a and approximation error ε, the difference in runtime between instances with 1 and 6 adversaries is minor (e.g., 2v1/6 instances are solved in 6.9s on average, whereas 2v6/6 require about 15s). Next section analyses whether this scalability trend with respect to the number of adversaries persists in larger instances, where existing solvers are no longer applicable.

6.2 Scalability to many adversaries

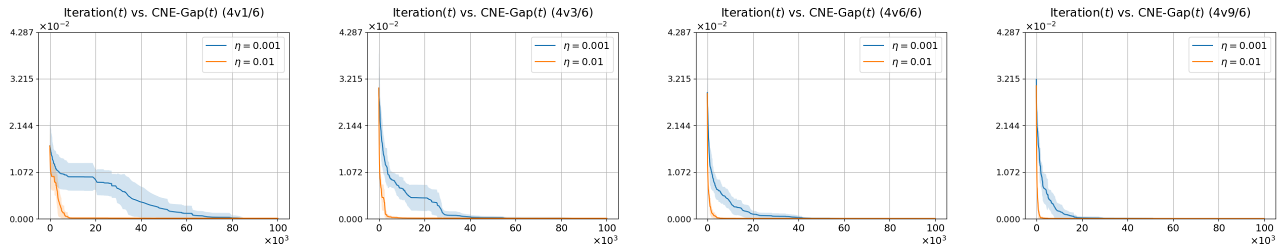
Scalability of MATG-GDM with respect to the number of adversaries was evaluated on large random MATGs with team size $n = 4$ and $a = 6$ actions per agent, and adversaries $m \in \{1, 3, 6, 9\}$. The algorithm was terminated after $T = 100{,}000$ iterations. Additional results for other configurations are provided in Appendix[2] C.

Results are reported in Figure 1 and Table 4. Figure 1 gives an overview of the performance of MATG-GDM across the 100,000 iterations by plotting at each iteration $t \in \{0, \ldots, T\}$ the cumulative minimum NE-GAP, denoted CNE-GAP(t). Formally, given the sequence of iterates $\{(\mathbf{x}^{(t')}, \mathbf{y}^{(t')}) : t' \in \{0, \ldots, T\}\}$ produced by MATG-GDM, the CNE-GAP at time t is defined as

$$\text{CNE-GAP}(t) = \min_{t' \leq t} \text{NE-GAP}(\mathbf{x}^{(t')}, \mathbf{y}^{(t')}). \tag{16}$$

Table 4 reports performance (mean and standard deviations of the termination iteration t^*) for two fixed approximation errors $\varepsilon \in \{10^{-3}, 10^{-4}\}$. Results show that MATG-GDM converges significantly faster for $\eta = 10^{-2}$ compared to $\eta = 10^{-3}$, in all tested configurations. This is likely due to the fact that $\eta = 10^{-2}$ brings (ten times) larger gradient steps than $\eta = 10^{-3}$. However, a risk with choosing large gradient steps is that MATG-GDM oscillates around a Nash equilibrium without never reaching a ε-NE.

Unsurprisingly, our results indicate that smaller target approximation errors (i.e., smaller ε) require a larger number of iterations to reach termination. For instance, Table 4 shows that the number of iterations for $\varepsilon = 10^{-3}$ is substantially lower than for $\varepsilon = 10^{-4}$ across all configurations. Nevertheless, MATG-GDM successfully finds solutions within $\varepsilon = 10^{-4}$ for all tested instances within the maximum iteration limit. Moreover, Figure 1 shows that the NE-GAP



(a) 4v1/6 : CNE-GAP(t) (b) 4v3/6: CNE-GAP(t) (c) 4v6/6: CNE-GAP(t) (d) 4v9/6: CNE-GAP(t)

Figure 1: CNE-GAP(t) (mean and standard deviation) from running MATG-GDM on 10 random instances of 4v1/6, 4v3/6, 4v6/6 and 4v9/6 MATGs.

Table 4: Performance of MATG-GDM on MATGs with $n = 4$ team agents, $a = 6$ actions per player, with varying numbers of adversaries $m \in \{1, 3, 6, 9\}$. The termination iteration t^* gives the mean and standard deviation of iterations required by MATG-GDM to reach ε-NE solutions with $\varepsilon \in \{10^{-3}, 10^{-4}\}$ under two learning rates, $\eta \in \{10^{-2}, 10^{-3}\}$.

Game	MATG-GDM Termination Iteration t^* (mean ± std)			
	NE-Gap$(\mathbf{x}^{(t^*)}, \mathbf{y}^{(t^*)}) \leq 10^{-3}$		NE-Gap$(\mathbf{x}^{(t^*)}, \mathbf{y}^{(t^*)}) \leq 10^{-4}$	
	$\eta = 10^{-2}$	$\eta = 10^{-3}$	$\eta = 10^{-2}$	$\eta = 10^{-3}$
4v1/6	5618 ± 1484	55889 ± 14995	39372 ± 54852	63504 ± 16292
4v3/6	3114 ± 1103	29731 ± 8533	35355 ± 58298	48443 ± 24436
4v6/6	2497 ± 1055	24468 ± 10406	36533 ± 56630	46148 ± 21458
4v9/6	1272 ± 623	12917 ± 6407	4094 ± 2941	23268 ± 13545

approaches zero, indicating that MATG-GDM produces solutions close to optimal within the maximum iteration limit.

Finally, notice that these patterns are largely unaffected by the number of adversaries. Figure 1a, which corresponds to a single adversary, exhibits similar trends to Figure 1d, which corresponds to nine adversaries. Surprisingly, the number of iterations required for termination even decreases significantly with the number of adversaries. For example, in Table 4 the termination iteration for 4v1/6 instances is substantially higher than for 4v9/6 instances. We hypothesize that this is because the gradient step, $\eta \sum_j \nabla_{\mathbf{x}_i} U_j(\cdot, \cdot)$ in MATG-GDM, is proportional to the number of adversaries, and thus behaves as taking multiple gradient steps with one adversary. Overall, the empirical results demonstrate that, unlike existing solvers, MATG-GDM exhibits execution times that scale favorably with the number of adversaries.

7 CONCLUSIONS AND FUTURE WORK

We introduce the *Multi-Adversarial Team Games* (MATG) framework, generalizing the Adversarial Team Games (ATG) framework to scenarios involving multiple independent adversaries. This framework is particularly well-suited to model interactions between a coordinated team of agents in competition with several independent adversaries. Such interactions arise naturally in various domains, including law enforcement problems and security games.

Our main contribution is to show that an ε-Nash equilibrium of an MATG can be computed in time polynomial in the game's natural parameters and in $1/\varepsilon$, overcoming the multi-agent curse on the adversary side. We also present the first empirical evaluation of a polynomial-time algorithm for computing ε-NE in MATGs and ATGs. Our experiments corroborate the theoretical findings and, importantly, demonstrate the practical applicability of MATG-GDM. In particular, experiments show that MATG-GDM scales effectively with the number of adversaries, enabling the computation of approximate NE in MATGs of sizes (e.g., 13 players: 4 team members vs. 9 adversaries) beyond the reach of existing solvers.

Computing even approximate Nash equilibria in general normal form games is a hard problem [9, 26]. Only a few classes of multiplayer games (> 2 players) are known to admit polynomial-time algorithms for approximating Nash equilibrium: zero-sum polymatrix games [5], potential games [24] and Adversarial Team Games (ATG) [1]. The latter notably overcome the curse of multi-agent on the team side, but are limited to a single adversary. Extending the ATG family to a wider family of polynomial-time solvable games is presented as an important, non-trivial perspective by [1]. It is worth noting that the broader class of two-team games [18], which subsumes MATGs, is computationally intractable. Thus, our contribution allows to narrow the uncertainty about the limit of tractability of multiplayer games.

A natural extension of our work is the computation of approximate NE in multi-adversarial extensions of Adversarial Team *Markov* Games [16]. An open question is to determine the conditions under which the independence of adversaries in this Markovian setting enables the efficient computation of approximate NE, potentially leveraging learning algorithms [19].

ACKNOWLEDGMENTS

This work was funded by the French National Research Agency (ANR) under grant ANR-22-CE92-0011-01.

REFERENCES

[1] Ioannis Anagnostides, Fivos Kalogiannis, Ioannis Panageas, Emmanouil-Vasileios Vlatakis-Gkaragkounis, and Stephen Mcaleer. 2023. Algorithms and Complexity for Computing Nash Equilibria in Adversarial Team Games *(EC '23)*. Association for Computing Machinery, New York, NY, USA, 89. https://doi.org/10.1145/3580507.3597776

[2] Ioannis Anagnostides, Ioannis Panageas, Tuomas Sandholm, and Jingming Yan. 2025. The Complexity of Symmetric Equilibria in Min-Max Optimization and Team Zero-Sum Games. *arXiv preprint arXiv:2502.08519* (2025).

[3] Christian Borgs, Jennifer Chayes, Nicole Immorlica, Adam Tauman Kalai, Vahab Mirrokni, and Christos Papadimitriou. 2008. The myth of the folk theorem. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*. 365–372.

[4] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. JAX: composable transformations of Python+NumPy programs. http://github.com/jax-ml/jax

[5] Yang Cai, Ozan Candogan, Constantinos Daskalakis, and Christos Papadimitriou. 2016. Zero-Sum Polymatrix Games: A Generalization of Minmax. *Mathematics of Operations Research* 41, 2 (2016), 648–655. https://doi.org/10.1287/moor.2015.0745

[6] Luca Carminati, Federico Cacciamani, Marco Ciccone, and Nicola Gatti. 2022. A marriage between adversarial team games and 2-player games: Enabling abstractions, no-regret learning, and subgame solving. In *International Conference on Machine Learning*. PMLR, 2638–2657.

[7] Andrea Celli, Marco Ciccone, Raffaele Bongo, and Nicola Gatti. 2019. Coordination in adversarial sequential team games via multi-agent deep reinforcement learning. *arXiv preprint arXiv:1912.07712* (2019).

[8] Andrea Celli and Nicola Gatti. 2018. Computational results for extensive-form adversarial team games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[9] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. 2009. Settling the complexity of computing two-player Nash equilibria. *J. ACM* 56, 3, Article 14 (May 2009), 57 pages. https://doi.org/10.1145/1516512.1516516

[10] DeepMind, Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Antoine Dedieu, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, George Papamakarios, John Quan, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Laurent Sartran, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Miloš Stanojević, Wojciech Stokowiec, Luyu Wang, Guangyao Zhou, and Fabio Viola. 2020. The DeepMind JAX Ecosystem. http://github.com/google-deepmind

[11] Hélène Fargier, Paul Jourdan, and Régis Sabbadin. 2022. A Path-following Polynomial Equations Systems Approach for Computing Nash Equilibria. In *AAMAS '22: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, Auckland, New Zealand, 418–426. https://hal.science/hal-03822343

[12] Srihari Govindan and Robert Wilson. 2004. Computing Nash equilibria by iterated polymatrix approximation. *Journal of Economic Dynamics and Control* 28, 7 (2004), 1229–1241.

[13] Erik Halvorson, Vincent Conitzer, and Ronald Parr. 2009. Multi-Step Multi-Sensor Hider-Seeker Games. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, Craig Boutilier (Ed.). 159–166. http://ijcai.org/Proceedings/09/Papers/037.pdf

[14] Kristoffer Arnsfelt Hansen, Thomas Dueholm Hansen, Peter Bro Miltersen, and Troels Bjerre Sørensen. 2008. Approximability and parameterized complexity of minmax values. In *Internet and Network Economics: 4th International Workshop, WINE 2008, Shanghai, China, December 17-20, 2008. Proceedings 4*. Springer, 684–695.

[15] Alexandros Hollender, Gilbert Maystre, and Sai Ganesh Nagarajan. 2025. The Complexity of Two-Team Polymatrix Games with Independent Adversaries. In *International Conference on Representation Learning*, Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (Eds.), Vol. 2025. 30456–30471. https://arxiv.org/abs/2409.07398

[16] Fivos Kalogiannis, Ioannis Anagnostides, Ioannis Panageas, Emmanouil V. Vlatakis-Gkaragkounis, Vaggos Chatziafratis, and Stelios Andrew Stavroulakis. 2023. Efficiently Computing Nash Equilibria in Adversarial Team Markov Games. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/pdf?id=mjzm6btqgV

[17] Fivos Kalogiannis, Ioannis Panageas, and Emmanouil-Vasileios Vlatakis-Gkaragkounis. 2021. Teamwork makes von neumann work: Min-max optimization in two-team zero-sum games. https://openreview.net/forum?id=UyBxDoukIB

[18] Fivos Kalogiannis, Ioannis Panageas, and Emmanouil V. Vlatakis-Gkaragkounis. 2023. Towards convergence to Nash equilibria in two-team zero-sum games. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/forum?id=4BPFwvKOvo5

[19] Fivos Kalogiannis, Jingming Yan, and Ioannis Panageas. 2024. Learning Equilibria in Adversarial Team Markov Games: A Nonconvex-Hidden-Concave Min-Max Optimization Problem. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 92832–92890. https://arxiv.org/abs/2410.05673

[20] Wai Yee Lam, Chee-Chean Phung, Zainal Abidin Mat, Hamidi Jamaluddin, Charina Pria Sivayogam, Fauzul Azim Zainal Abidin, Azlan Sulaiman, Melynda Ka Yi Cheok, Noor Alif Wira Osama, Salman Sabaan, Abdul Kadir Abu Hashim, Mark Daniel Booton, Abishek Harihar, Gopalasamy Reuben Clements, and Rob Stuart Alexander Pickles. 2023. Using a crime prevention framework to evaluate tiger counter-poaching in a Southeast Asian rainforest. *Frontiers in Conservation Science* 4 (2023). https://doi.org/10.3389/fcosc.2023.1213552

[21] Prasanna Maddila, Casellas Eric, Patrick Chabrier, Régis Sabbadin, and Meritxell Vinyals. 2024. APE: An Anti-poaching Multi-Agent Reinforcement Learning Benchmark. In *Seventeenth European Workshop on Reinforcement Learning*. https://openreview.net/forum?id=JSWRnHC93W

[22] Prasanna Maddila, Régis Sabbadin, and Meritxell Vinyals. 2025. *Multi Adversarial Team Games*. https://doi.org/10.5281/zenodo.18245008

[23] H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. 2003. Planning in the presence of cost functions controlled by an adversary. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 536–543.

[24] Dov Monderer and Lloyd S Shapley. 1996. Potential games. *Games and economic behavior* 14, 1 (1996), 124–143.

[25] Christos H Papadimitriou and Tim Roughgarden. 2008. Computing correlated equilibria in multi-player games. *Journal of the ACM (JACM)* 55, 3 (2008), 1–29.

[26] Aviad Rubinstein. 2016. Settling the Complexity of Computing Approximate Two-Player Nash Equilibria. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. 258–265. https://doi.org/10.1109/FOCS.2016.35

[27] Rahul Savani and Theodore L Turocy. [n.d.]. Gambit: The package for computation in game theory, Version 16.3.1. https://gambitproject.readthedocs.io/en/latest/index.html

[28] Leonard J Schulman and Umesh V Vazirani. 2019. The duality gap for two-team zero-sum games. *Games and Economic Behavior* 115 (2019), 336–345.

[29] Aravind Venugopal, Elizabeth Bondi, Harshavardhan Kamarthi, Keval Dholakia, Balaraman Ravindran, and Milind Tambe. 2021. Reinforcement Learning for Unified Allocation and Patrolling in Signaling Games with Uncertainty. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (Virtual Event, United Kingdom) *(AAMAS '21)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1353–1361.

[30] Bernhard Von Stengel and Daphne Koller. 1997. Team-maxmin equilibria. *Games and Economic Behavior* 21, 1-2 (1997), 309–321.

[31] Weiran Wang and Miguel A Carreira-Perpinán. 2013. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv preprint arXiv:1309.1541* (2013).

[32] Yufei Wang, Zheyuan Ryan Shi, Lantao Yu, Yi Wu, Rohit Singh, Lucas Joppa, and Fei Fang. 2019. Deep reinforcement learning for green security games with real-time information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1401–1408.

[33] Lily Xu, Shahrzad Gholami, Sara McCarthy, Bistra Dilkina, Andrew Plumptre, Milind Tambe, Rohit Singh, Mustapha Nsubuga, Joshua Mabonga, Margaret Driciru, et al. 2020. Stay ahead of Poachers: Illegal wildlife poaching prediction and patrol planning under uncertainty with field test evaluations (Short Version). In *2020 IEEE 36th international conference on data engineering (ICDE)*. IEEE, 1898–1901.

[34] Youzhi Zhang and Bo An. 2020. Computing team-maxmin equilibria in zero-sum multiplayer extensive-form games. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 2318–2325.