



HAL
open science

Efficiently computing Nash Equilibria in Multi-Adversarial Team Games

Prasanna Maddila, Sabbadin Régis, Vinyals Meritxell

► **To cite this version:**

Prasanna Maddila, Sabbadin Régis, Vinyals Meritxell. Efficiently computing Nash Equilibria in Multi-Adversarial Team Games. 2025. hal-04917907v2

HAL Id: hal-04917907

<https://hal.science/hal-04917907v2>

Preprint submitted on 31 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Efficiently computing Nash Equilibria in Multi-Adversarial Team Games

Prasanna Maddila¹, Régis Sabbadin¹, and Meritxell Vinyals¹

¹UR 875, MIAT, INRAE, University of Toulouse, Castanet-Tolosan, France

January 31, 2025

Abstract

Adversarial Team Games (ATG), as introduced by von Stengel and Koller, model strategic situations where a team of agents, sharing a common objective but unable to coordinate their actions, faces a single adversary. Recent advances have shown that computing an approximate Nash Equilibrium (NE) in ATGs is computationally tractable, establishing ATGs as one of the rare multi-agent game families where this holds. However, this tractability result is limited to single-adversary scenarios, leaving the common case of multiple independent adversaries — prevalent in applications such as anti-poaching, robotic planning, and hide-seeker games — largely unexplored. This paper addresses this gap by introducing the Multi-Adversarial Team Game (MATG) framework, a natural extension of ATGs to settings with multiple independent adversaries. Our main contribution is to prove that approximate NE computation remains tractable in this generalised setting. We achieve this by designing and analyzing a pseudo fully-polynomial time approximation scheme (pseudo-FPTAS) for computing NE in MATGs. Beyond our theoretical contributions, we provide the first implementation of this family of algorithms for NE computation in both ATGs and MATGs, demonstrating their empirical performance on a set of benchmark problems.

Index terms— Algorithmic Game Theory, Adversarial Team Games, Polytime Nash Equilibrium computation.

1 Introduction

Adversarial Team Games (ATGs) are a successful application of non-cooperative game theory to the real-world. In their mainstream formulation, as introduced by von Stengel and Koller [VK97], ATGs model situations where a team of agents, sharing a common objective but unable to coordinate their actions, faces a *single* adversary. This team inability to coordinate their actions responds to real-world considerations in which team communication is prohibited (e.g. by game rules in Bridge), impractical (e.g. by security) or too expensive (e.g. in large organisations). For ATGs, literature studied two

main solution concepts: (1) Nash Equilibrium (NE), where no agent can obtain higher utility by unilaterally deviating from the equilibrium; and (2) Team-Minmax Equilibrium (TME), which is the best NE for the team (i.e. the one that maximizes the team’s utility among all NE).

Despite its appealing properties, the fact that it has been proved that approximating TME in ATGs is computationally intractable [Bor+08; Han+08] has hindered its practical adoption. In contrast, recent advances [Ana+23] have shown the computational tractability of approximating NE in ATGs - being since one of the few multi-agent games families in which NE approximation is tractable.

However, research has focused on the single-adversary case whereas the case in which the team faces *multiple* adversaries remains unexplored. This multi-adversary setting turns out to be very common in real-world scenarios. For instance, in anti-poaching scenarios — a problem modeled by green security games [Wan+19] — rangers need to detect several potential poachers within the same protected area [Lam+23]. Similarly, in robotic planning [MGB03], different adversaries may control different aspects of the environment. Lastly, in hider-seeker games [HCP09], a team may need to locate not one but multiple independent hidere. In this context¹, this paper addresses this unexplored multi-adversary setting, focusing on the following research questions: (1) *How does transitioning from one to multiple adversaries increase the computational complexity of computing approximate NE in ATGs?* and (2) *Can we design approximate NE algorithms that scale efficiently with the number of independent adversaries?*

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 [Ana+23], we develop a computationally tractable algorithm for approximating NE in MATGs. Our algorithm extends the one proposed in [Ana+23] 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. *Finally*, we corroborate our theoretical findings through extensive numerical experiments, demonstrating the algorithm’s capability to effectively approximate NE in large-scale scenarios 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 $n = |\mathcal{N}|$ team agents and a finite set of $m = |\mathcal{M}|$ 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 ensemble 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 $\mathbf{a} = (a_1, \dots, a_n) \in \mathcal{A}$ the action profile of the team, and

¹Additional related work is included in Appendix.

by $\mathbf{b} = (b_1, \dots, b_m) \in \mathcal{B} := \prod_{j \in \mathcal{M}} \mathcal{B}_j$ the action profile of the adversaries. Each adversary $j \in \mathcal{M}$ has a payoff function denoted by $U_j : \mathcal{A} \times \mathcal{B}_j \rightarrow \mathbb{R}$. The team payoff function is denoted by $U_{team} : \mathcal{A} \times \mathcal{B} \rightarrow \mathbb{R}$, so that the individual payoff of each team member is identical to its teammates: $U_i(\mathbf{a}, \mathbf{b}) = U_{team}(\mathbf{a}, \mathbf{b})$, for all joint action profiles $(\mathbf{a}, \mathbf{b}) \in \mathcal{A} \times \mathcal{B}$ and for all team members $i \in \mathcal{N}$. Further, the team game is zero-sum in the sense that $U_{team}(\mathbf{a}, \mathbf{b}) = -\frac{1}{n} \sum_{j \in \mathcal{M}} U_j(\mathbf{a}, b_j)$.

A team 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 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: $U_j : (\mathbf{x}, \mathbf{y}_j) \in \mathcal{X} \times \mathcal{Y}_j \mapsto E_{(\mathbf{a}, b_j) \sim (\mathbf{x}, \mathbf{y}_j)} [U_j(\mathbf{a}, b_j)]$. We will also write $\text{poly}(\Gamma)$ for factors that are polynomial in the natural parameters of the game.

In terms of solution concepts, we focus on computing a set of approximate best responses that form an approximate Nash Equilibrium, both defined as follows. As is customary, $-i$ denotes the set containing all the agents except agent i .

Approximate best response (ε -BR). Consider a MATG Γ . A 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:

$$\sum_{j \in \mathcal{M}} U_j(\mathbf{x}, \mathbf{y}_j) \leq \sum_{j \in \mathcal{M}} U_j(\mathbf{x}'_i, \mathbf{x}_{-i}, \mathbf{y}_j) + \varepsilon \quad \forall \mathbf{x}'_i \in \mathcal{X}_i \quad (1)$$

We denote by $BR_i((\mathbf{x}_{-i}, \mathbf{y}); \varepsilon)$ the set of ε -BR strategies of team agent i to $(\mathbf{x}_{-i}, \mathbf{y})$. Similarly, a strategy \mathbf{y}_j of adversary j is an ε -BR to a team strategy profile \mathbf{x} iff:

$$U_j(\mathbf{x}, \mathbf{y}_j) \geq U_j(\mathbf{x}, \mathbf{y}'_j) - \varepsilon \quad \forall \mathbf{y}'_j \in \mathcal{Y}_j \quad (2)$$

We denote by $BR_j(\mathbf{x}; \varepsilon)$ the set of ε -BR strategies of adversary j to \mathbf{x} .

Approximate Nash Equilibrium (ε -NE). The strategy profile $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$ is an ε -Nash Equilibrium of a MATG Γ for an approximation guarantee ($\varepsilon \geq 0$) if for every agent, their strategy 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 $\forall j \in \mathcal{M}, \mathbf{y}_j \in BR_j(\mathbf{x}; \varepsilon)$.

	a_1, a_2	0,0	0,1	1,0	1,1
b_1		0	$2/5$	$2/5$	$4/5$
0		0	$2/5$	$2/5$	$4/5$
1		$1/5$	$1/10$	$1/10$	0

	a_1, a_2	0,0	0,1	1,0	1,1
b_2		0	$1/5$	$1/5$	$2/5$
0		0	$1/5$	$1/5$	$2/5$
1		$3/5$	$3/10$	$3/10$	0

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

Consider the MATG illustrated in Figure 1. In this setting, a team of $n = 2$ agents protects 2 locations against $m = 2$ adversaries. Each adversary has distinct rewards

for each location, and capturing an adversary requires both team agents to be present in the same cell. If only one team agent is present, the adversary escapes with a reduced reward. The strategies $x_1 = [\frac{1}{5}, \frac{4}{5}]$, $x_2 = [\frac{1}{5}, \frac{4}{5}]$, $y_1 = [\frac{2}{5}, \frac{3}{5}]$, $y_2 = [1, 0]$ form a NE.

3 A polynomial time algorithm for computing ε -NE in MATGs

In this section, we present the Multi-Adversarial Team Games Gradient Descend Max (MATG-GDM) algorithm, which extends the approach in [Ana+23] for ATGs to handle multiple independent adversaries. We also establish our main theoretical result: MATG-GDM computes an ε -NE with computational complexity that is polynomial in all natural parameters of the game and in $1/\varepsilon$. This result is formalised in Theorem 3.1 and formally proved in Section 4.

The proposed Algorithm 1 takes as input a MATG Γ , an approximation error parameter $\varepsilon > 0$, a learning rate η and a maximum number of iterations T . The first step is to initialize all agents' strategies at an arbitrary point $(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}) \in \mathcal{X} \times \mathcal{Y}$. Then, at each iteration, the algorithm verifies if agents' strategies for that iteration are an ε -NE (Lines 3-5), terminating and returning them if it is the case. In each step $1 \leq t \leq T$, the algorithm computes a best response of each adversary based on the current strategy of the team (Lines 6-8). Next, based on the best response of adversaries, each team member performs a projected gradient descent step (Lines 9-11) in which $\text{Proj}_{\mathcal{X}_i}(\cdot)$ returns the Euclidean projection to the set \mathcal{X}_i . Now, based on the updated strategy of the team $\mathbf{x}^{(t)}$, the response of the adversaries is determined by $\text{ExtendNE}(\mathbf{x}^{(t)})$ that returns the solution of the following LP:

$$\begin{aligned}
& \max_{\mathbf{y}, z} \quad \sum_{i \in \mathcal{N}} z_i & (3) \\
& \text{s.t.} \quad z_i - \sum_{j \in \mathcal{M}} \sum_{b_j \in \mathcal{B}_j} \mathbf{y}_j(b_j) \cdot U_j(a_i, \mathbf{x}_{-i}^{(t)}, b_j) \leq 0, \\
& \quad \forall i \in \mathcal{N}, a_i \in \mathcal{A}_i, \\
& \quad \sum_{b_j \in \mathcal{B}_j} \mathbf{y}_j(b_j) = 1, \quad \forall j \in \mathcal{M}, \\
& \quad \mathbf{y}_j(b_j) \geq 0, \quad \forall j \in \mathcal{M}, b_j \in \mathcal{B}_j.
\end{aligned}$$

This iterative process is repeated until $(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})$ is an ε -NE (Lines 3-5) or until the maximum number of iterations T is reached. In order to prove that the algorithm runs in polynomial time, it is required to assume that utilities of mixed strategies can be computed in polynomial time:

Assumption 3.1 (Polynomial Expectation Property). *For any (mixed) joint strategy profile $(\mathbf{x}, \mathbf{y}_j) \in \mathcal{X} \times \mathcal{Y}_j$, we can compute (exactly) the expectation $U_j(\mathbf{x}, \mathbf{y}_j) = \mathcal{E}_{(\mathbf{a}, b_j) \sim (\mathbf{x}, \mathbf{y})} U_j(\mathbf{a}, b_j)$ 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 corollary:

Algorithm 1: MATG Gradient Descend Max

Input: MATG Γ , approximation error $\varepsilon > 0$, learning rate η , maximum number of iterations T

- 1 Initialize strategies $(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}) \in \mathcal{X} \times \mathcal{Y}$;
- 2 **for** $t \leftarrow 1, 2, \dots, T$ **do**
- 3 **if** $(\mathbf{x}^{(t-1)}, (\mathbf{y}_j^{(t-1)})_{j \in \mathcal{M}})$ is an ε -NE **then**
- 4 **break**
- 5 **end**
- 6 **for** $j \in \mathcal{M}$ **do**
- 7 $b_j^{(t)} \leftarrow \arg \max_{b_j \in \mathcal{B}_j} U_j(\mathbf{x}^{(t-1)}, b_j)$
- 8 **end**
- 9 **for** $i \in \mathcal{N}$ **do**
- 10 $\mathbf{x}_i^{(t)} \leftarrow \text{Proj}_{\mathcal{X}_i} \left(\mathbf{x}_i^{(t-1)} - \eta \nabla_{\mathbf{x}_i} \sum_{j \in \mathcal{M}} U_j(\mathbf{x}^{(t-1)}, b_j^{(t)}) \right)$
- 11 **end**
- 12 $(\mathbf{y}_j^{(t)})_{j \in \mathcal{M}} \leftarrow \text{ExtendNE}(\mathbf{x}^{(t)})$
- 13 **end**
- 14 **return** $(\mathbf{x}^{(t)}, (\mathbf{y}_j^{(t)})_{j \in \mathcal{M}})$

Corollary 3.1. *Under Assumption 3.1, every iteration of Algorithm 1 can be computed in polynomial time.*

Proof. Lines 3-5 require to check whether $(\mathbf{x}^{(t-1)}, \mathbf{y}^{(t-1)})$ is an ε -NE, which in turn requires to verify that the strategy profile of every agent is an ε -BR. This can be checked in polynomial time by virtue of Assumption 3.1.

Lines 6-8 require to compute a best response for each adversary j , a computation that can be trivially performed by computing $\max_{\mathbf{b} \in \mathcal{B}} U(\mathbf{x}^{(t-1)}, \mathbf{b})$, which under Assumption 3.1 takes polynomial time. By repeating this step for every adversaries $j \in \mathcal{M}$ the time remains polynomial.

Lines 9-11 perform a projected gradient descent step for each team member $i \in \mathcal{N}$. By definition, $U_j(\mathbf{x}^{(t-1)}, b_j^{(t)}) = \sum_{a_i \in \mathcal{A}_i} x_i^{(t-1)}(a_i) U_j(a_i, \mathbf{x}_{-i}^{(t-1)}, b_j^{(t)})$. Thus, by multilinearity, $\frac{\partial}{\partial x_i(a_i)} U_j(\mathbf{x}^{(t-1)}, b_j^{(t)}) = U_j(a_i, \mathbf{x}_{-i}^{(t-1)}, b_j^{(t)})$ which by virtue of Assumption 3.1 can be computed in polynomial time. In addition, the $\text{Proj}_{\mathcal{X}_i}$ operator only requires to normalize a probability distribution over \mathcal{A}_i and can be computed exactly in nearly-linear time on $|\mathcal{A}_i|$. Repeating this step for every $i \in \mathcal{N}$ still takes polynomial time in the size of the MATG.

Finally in Line 12, ExtendNE solves a LP with $\sum_{j \in \mathcal{M}} |\mathcal{B}| + n$ variables and $\sum_{i \in \mathcal{N}} |\mathcal{A}_i| + \sum_{j \in \mathcal{M}} |\mathcal{B}| + m$ constraints. Moreover, the computation of each coefficient $U_j(a_i, \mathbf{x}_{-i}^{(t-1)}, b_j)$ is polynomial in the number of natural parameters of the game under Assumption 3.1. Therefore, the LP can be constructed and solved in time polynomial in the size of Γ . \square

We are now ready to formulate our main theoretical result:

Theorem 3.1 (Pseudo-FPTAS). *Consider any precision $\varepsilon > 0$. For any MATG Γ , Algorithm 1 with a sufficiently small learning rate $\eta = O(\varepsilon^2)$ yields an ε -NE after a maximum number of iterations $T = \text{poly}(\Gamma)/\varepsilon^4$. Further, under Assumption 3.1, every iteration of the algorithm can be implemented in polynomial time.*

4 Proof of the main Theorem

In this section, we demonstrate that Algorithm 1, as stated in Theorem 3.1, computes an ε -NE in pseudo-polynomial time. The proof is structured as follows:

First, in Section 4.1 we prove that, for any $\varepsilon \geq 0$, an ε -NE of a MATG can be computed by marginalising an ε -NE of a correlated-adversaries transformed game. This *correlated-adversaries* game 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 a (single-adversary) ATG, we can use any existing algorithm designed for ATGs to compute NE in MATGs. In particular, GradientDescentMax (GDM) as proposed in [Ana+23] computes ε -approximate NEs in ATGs with polynomial complexity in all natural game parameters and $1/\varepsilon$. However, the action space of the adversary in a correlated-adversaries MATG grows exponentially with the number of adversaries in the original MATG. Hence, a naive application of the algorithm in [Ana+23] takes exponential time. Instead our proof builds on the following steps.

- In Section 4.2, we exploit the underlying structure of a correlated-adversaries MATG to show that the different steps of GDM, when applied to a correlated-adversaries MATG, are equivalent to the steps proposed in Algorithm 1, Section 3. In particular, all steps are polynomial in the size of the original MATG.
- Then, in Section 4.3, we show that the maximum number of iterations of Algorithm 1 in Section 3 needed to find an ε -NE remains polynomial in the problem size.

4.1 Correlated-adversaries transformation

In this section, we show that any ε -NE of a MATG Γ is also an ε -NE of the equivalent ATG obtained by assuming that the adversaries in the MATG *correlate* their strategies. We call this ATG *correlated adversaries (ca)-transformation*. In the other direction, we show (in Theorem 4.1) that from any ε -NE of the ca-transformation we can compute an ε -NE of the original MATG by computing *marginalizations* of the correlated strategy over the strategies sets of the adversaries of the MATG.

Let us first define the *correlated-adversaries transformation* of a MATG:

Definition 4.1 (Correlated-adversaries MATG transformation).

Given a MATG $\Gamma(\mathcal{N}, \mathcal{M}, (\mathcal{A}_i)_{i \in \mathcal{N}}, (\mathcal{B}_j)_{j \in \mathcal{M}}, (U_j)_{j \in \mathcal{M}})$ we define its correlated-adversaries (ca)-transformation $\Gamma^{ca}(\mathcal{N}, (\mathcal{A}_i)_{i \in \mathcal{N}}, \mathcal{B}^{ca} = \prod_{j \in \mathcal{M}} \mathcal{B}_j, U^{ca} : \mathcal{A} \times \mathcal{B}^{ca} \rightarrow \mathfrak{R})$

where $U^{ca}(\mathbf{a}, \mathbf{b}) = \sum_{j \in \mathcal{M}} U_j(\mathbf{a}, b_j)$ and $U_{team}^{ca}(\mathbf{a}, \mathbf{b}) = -U^{ca}(\mathbf{a}, \mathbf{b})$, in which all adversaries play correlated strategies and share payoffs additively.

$b^{ca} \backslash a_1, a_2$	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

Figure 2: Payoff table for the correlated adversary in the ca-transformation of the MATG in Fig. 1.

For example, Figure 2 presents the payoff table of the correlated adversary for the ca-transformation of the MATG in Figure 1.

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

Definition 4.2 (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 strategy as $\mathbf{y} = \otimes_{j \in \mathcal{M}} \mathbf{y}_j$ defined as: $\mathbf{y}(\mathbf{b}) = \prod_{j \in \mathcal{M}} \mathbf{y}_j(b_j)$, $\forall \mathbf{b} \in \mathcal{B}$.*

Definition 4.3 (Marginalised adversary strategies). *Given a (mixed) correlated strategy $\mathbf{y} \in \mathcal{Y}^{ca}$ for adversaries in a ca-transformation, we define the set of marginalised strategies of adversaries as $\tilde{\mathbf{y}} = (\tilde{\mathbf{y}}_j)_{j \in \mathcal{M}}$ defined as: $\tilde{\mathbf{y}}_j(b_j) = \sum_{\mathbf{b}_{-j} \in \mathcal{B}_{-j}} \mathbf{y}(b_j, \mathbf{b}_{-j})$, $\forall j \in \mathcal{M}, \forall b_j \in \mathcal{B}_j$.*

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 4.1. *Let a MATG Γ and its (ca)-transformation 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, $U^{ca}(\mathbf{x}, \mathbf{y}) = \sum_{j \in \mathcal{M}} U_j(\mathbf{x}, \tilde{\mathbf{y}}_j)$.*

Proof.

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

□

We also have the following corollary:

Corollary 4.1. *Given an MATG Γ , its (ca)-transformation Γ^{ca} and the mixed strategies $(\mathbf{x}, (\mathbf{y}_j)_{j \in \mathcal{M}}) \in \mathcal{X} \times \mathcal{Y}$, we have that*

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

Proof. Since $\tilde{\mathbf{y}}_j$ is the marginal strategy of \mathbf{y} for adversary j , by applying Prop. 4.1 we have $\tilde{\mathbf{y}}_j = \mathbf{y}_j$. □

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-transformation.

Proposition 4.2. *Consider any $\varepsilon \geq 0$. Let a MATG Γ and its (ca)-transformation Γ^{ca} . Let $\mathbf{x} \in \mathcal{X}$ be a team strategy profile and let $\mathbf{y} \in \mathcal{Y}^{ca}$ be a (correlated) ε -BR to \mathbf{x} in Γ^{ca} . Then, $\forall j \in \mathcal{M}$ $\tilde{\mathbf{y}}_j$, where $\tilde{\mathbf{y}}_j$ is the marginalised strategy of \mathbf{y} over \mathcal{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:

$$U^{ca}(\mathbf{x}, \mathbf{y}) \geq U^{ca}(\mathbf{x}, \mathbf{y}') - \varepsilon \quad \forall \mathbf{y}' \in \mathcal{Y}^{ca} \quad (4)$$

which by Proposition 4.1 is equivalent to:

$$\sum_{j \in \mathcal{M}} U_j(\mathbf{x}, \tilde{\mathbf{y}}_j) \geq \sum_{j \in \mathcal{M}} U_j(\mathbf{x}, \tilde{\mathbf{y}}'_j) - \varepsilon \quad \forall \mathbf{y}' \in \mathcal{Y}^{ca}, \quad (5)$$

where the $(\tilde{\mathbf{y}}_j)_{j \in \mathcal{M}}$ and $(\tilde{\mathbf{y}}'_j)_{j \in \mathcal{M}}$ are the marginalised strategies of \mathbf{y} and \mathbf{y}' respectively.

Now, fix $k \in \mathcal{M}$ and define $\mathbf{y}' = \mathbf{y}'_k \otimes (\otimes_{j \neq k} \tilde{\mathbf{y}}_j)$ for any arbitrary $\mathbf{y}'_k \in \mathcal{Y}_k$. Note that, $\mathbf{y}'(\mathbf{b}) = \mathbf{y}'_k(b_k) \times \prod_{j \in \mathcal{M}_{-k}} \tilde{\mathbf{y}}_j(b_j)$. Then, we have

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

Then, by subtracting $\sum_{j \neq k} U_j(\mathbf{x}, \tilde{\mathbf{y}}_j)$ in both sides of (5), we get:

$$U_k(\mathbf{x}, \tilde{\mathbf{y}}_k) \geq U_k(\mathbf{x}, \mathbf{y}'_k) - \varepsilon, \quad \forall \mathbf{y}'_k \in \mathcal{Y}_k. \quad (6)$$

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

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

Theorem 4.1. *For any $\varepsilon \geq 0$, consider an MATG Γ and its (ca)-transformation, Γ^{ca} . For any ε -NE $(\mathbf{x}^*, \mathbf{y}^*)$ of Γ^{ca} , \mathbf{x}^* and the marginalised adversary strategies $\tilde{\mathbf{y}}^*$, derived from \mathbf{y}^* , form an ε -NE of Γ .*

Proof. First, since $(\mathbf{x}^*, \mathbf{y}^*)$ is an ε -NE of Γ^{ca} , this 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. 4.1), 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 . Lastly, from Prop. 4.2, we get that $\forall j \in \mathcal{M} : \tilde{\mathbf{y}}_j^* \in BR_j(\mathbf{x}; \varepsilon)$ in Γ . \square

Thus, the NE of the ca-transformation in the example of Figure 2 is (x_1, x_2, y^{ca}) , where $y^{ca} = [\frac{2}{5}, 0, \frac{3}{5}, 0]$. It is straightforward to verify that the marginalized adversary strategies from y^{ca} are the NE adversary strategies of the original MATG in Figure 1.

4.2 Equivalence of MATG-GDM and GDM

Theorem 4.2. *Consider a MATG Γ and its (ca)-transformation Γ^{ca} . Let $\varepsilon > 0$, $\eta > 0$ and $T > 0$. MATG-GDM applied to $(\Gamma, \varepsilon, \eta, T)$ is equivalent to GDM applied to $(\Gamma^{ca}, \varepsilon, \eta, T)$.*

Proof. For any iteration, $0 \leq t \leq T$, let $(\mathbf{x}^{(t)}, (\mathbf{y}_j^{(t)})_{j \in \mathcal{M}})$ be the strategies computed by Algorithm 1 applied to $(\Gamma, \varepsilon, \eta, T)$. We prove that $(\mathbf{x}^{(t)}, \otimes_{j \in \mathcal{M}} \mathbf{y}_j^{(t)})$ correspond to the strategies computed by GDM applied to $(\Gamma^{ca}, \varepsilon, \eta, T)$ at the same iteration t .

Nash Equilibrium Check (Lines 3–5). Algorithm 1 verifies whether $(\mathbf{x}^{(t)}, (\mathbf{y}_j^{(t)})_{j \in \mathcal{M}})$ is an ε -NE. By Theorem 4.1, $(\mathbf{x}^{(t)}, (\mathbf{y}_j^{(t)})_{j \in \mathcal{M}})$ is an ε -NE of Γ iff $(\mathbf{x}^{(t)}, \otimes_{j \in \mathcal{M}} \mathbf{y}_j^{(t)})$ is an ε -NE of Γ^{ca} , establishing equivalence.

BR computation (Lines 6–8). Algorithm 1 computes a BR strategy for each adversary w.r.t. $\mathbf{x}^{(t)}$ in Γ . By Prop. 4.2, the correlated strategy of a set of adversaries' BR strategies to $\mathbf{x}^{(t)}$ in Γ is a BR strategy of the correlated adversary in Γ^{ca} to $\mathbf{x}^{(t)}$, establishing equivalence.

Projected gradients (Lines 9–11). The gradient projection step equivalence follows directly from the fact that by Prop. 4.1, for any $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}^{ca}$: $U^{ca}(\mathbf{x}, \mathbf{y}) = \sum_{j \in \mathcal{M}} U_j(\mathbf{x}, \tilde{\mathbf{y}}_j)$, where $\tilde{\mathbf{y}}_j$ is the marginal of \mathbf{y} over \mathcal{Y}_j .

ExtendNE procedure (Line 12). Algorithm 1 solves the LP (3), whereas GDM instead solves the following LP, in variables $\mathbf{y} = (\mathbf{y}(\mathbf{b}))_{\mathbf{b} \in \mathcal{B}}$ and $z = (z_i)_{i \in \mathcal{N}}$:

$$\begin{aligned} \max_{\mathbf{y}, z} \quad & \sum_{i \in \mathcal{N}} z_i & (7) \\ \text{s.t.} \quad & z_i - \sum_{\mathbf{b} \in \mathcal{B}} \mathbf{y}(\mathbf{b}) \cdot U^{ca}(a_i, \mathbf{x}_{-i}^{(t)}, \mathbf{b}) \leq 0 \quad \forall i \in \mathcal{N}, a_i \in \mathcal{A}_i, \\ & \sum_{\mathbf{b} \in \mathcal{B}} \mathbf{y}(\mathbf{b}) = 1 \quad \text{and} \quad \mathbf{y}(\mathbf{b}) \geq 0 \quad \forall \mathbf{b} \in \mathcal{B}. \end{aligned}$$

Let (\mathbf{y}', z') be a feasible solution of this LP. Then, $\tilde{\mathbf{y}}' = (\tilde{\mathbf{y}}'_j)_{j \in \mathcal{M}}$ where $\tilde{\mathbf{y}}'_j$ is the marginalised strategy of \mathbf{y}' over \mathcal{Y}_j . Then, by Prop 4.1, we have that $\forall i \in \mathcal{N}$, $a_i \in \mathcal{A}_i$:

$$U^{ca}(a_i, \mathbf{x}_{-i}^{(t)}, \mathbf{y}') = \sum_{j \in \mathcal{M}} U_j(a_i, \mathbf{x}_{-i}^{(t)}, \tilde{\mathbf{y}}'_j), \quad (8)$$

Hence,

$$\sum_{\mathbf{b} \in \mathcal{B}} \mathbf{y}'(\mathbf{b}) U^{ca}(a_i, \mathbf{x}_{-i}^{(t)}, \mathbf{b}) = \sum_{j \in \mathcal{M}} \sum_{b_j \in \mathcal{B}_j} \tilde{\mathbf{y}}'_j(b_j) U_j(a_i, \mathbf{x}_{-i}^{(t)}, b_j).$$

Thus, $\tilde{\mathbf{y}}'$ satisfies the constraints of LP (3), and is a feasible solution. Using Eq 8, we can show the reverse argument as well: given $\tilde{\mathbf{y}}'$, we can construct the correlated strategy $\mathbf{y}' = \otimes_j \tilde{\mathbf{y}}'_j$ to produce a feasible solution for LP (7).

Consequently, LP (3) should also have an optimal solution, and noting that z' is the same for both feasible solutions, the values of the optimal solutions should also be the same. \square

We have proved that the steps of Algorithm 1 are equivalent to those of Anagnostides' algorithm. Furthermore, by Corollary 3.1, each iteration of Algorithm 1 can be computed in polynomial time (under Assumption 3.1).

To complete the argument that an ε -NE of a MATG can be computed in pseudo-polynomial time, it remains to establish that the number of iterations T of MATG-GDM remain polynomial in the size of Γ and $1/\varepsilon$. This will be the focus of the next section.

4.3 Convergence to an ε -NE in pseudo-polynomial number of iterations

This section extends the proof of [Ana+23]- specifically Theorem 3.3 and Corollary 3.2 therein — to the case of MATGs, showing that a polynomial number of iterations (in the size of Γ and in $1/\varepsilon$) of Algorithm 1 is sufficient to compute an ε -NE.

In doing so, we also partially address the following remark from [Ana+23]:

Remark 4.1 (Correlated Adversaries, [Ana+23]). *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 first define the functions $\phi(\cdot) : \mathcal{X} \rightarrow \mathbb{R}$ and $V_{\mathbf{x}}(\cdot) : \mathcal{X} \rightarrow \mathbb{R}$ as follows.

$$\phi(\mathbf{x}) = \max_{\mathbf{y} \in \mathcal{Y}} \sum_{j \in \mathcal{M}} U_j(\mathbf{x}, \mathbf{y}_j) \quad (9)$$

$$V_{\mathbf{x}}(\mathbf{x}') = \phi(\mathbf{x}') + \ell \|\mathbf{x} - \mathbf{x}'\|_2^2 \quad (10)$$

where ℓ is the smoothness parameter of $\sum_{j \in \mathcal{M}} U_j(\mathbf{x}', \mathbf{y}_j)$. Lastly, let $\text{prox}_{\phi/(2\ell)}(\mathbf{x}; \varepsilon)$ be any ε -approximate solution in value to the program $\min_{\mathbf{x}' \in \mathcal{X}} V_{\mathbf{x}}(\mathbf{x}')$.

Theorem 4.3. *Consider any precision $\varepsilon > 0$ and any MATG Γ , and let function $g : \mathcal{X} \rightarrow \mathbb{R}$ be defined as*

$$g(\mathbf{x}, \varepsilon) = V_{\mathbf{x}}(\text{prox}_{\phi/(2\ell)}(\mathbf{x}; \varepsilon)) \quad (11)$$

Then MATG-GDM applied to (Γ, ε) , with a sufficiently small learning rate $\eta = O(\varepsilon^2)$ satisfies

$$g(\mathbf{x}^{(t)}, \varepsilon) < g(\mathbf{x}^{(t-1)}, \varepsilon) - \Omega(\varepsilon^4)$$

for any iteration t until termination.

Proof. Let Γ^{ca} be the ca-transformation of Γ . Since Γ^{ca} is an ATG, Theorem 3.3 from [Ana+23] applies to it directly. Furthermore, the proof relies solely on the ℓ -smoothness and L -Lipschitz continuity of U^{ca} , as well as on the learning rate η , to establish a lower bound on the decrease in value of $g(\mathbf{x}^{(t)}) - g(\mathbf{x}^{(t-1)})$ (and thus the number of steps of the algorithm). Crucially, the proof does not depend on the size of \mathcal{B}^{ca} (the joint action space of adversaries). This implies that decrease in value is independent of the number of adversaries. Thus, the proof applies to the ca-transformation, and adversarial strategies that result in the same value of $g(\mathbf{x}^{(t)})$ can be obtained by marginalizing the corresponding adversaries strategies in the ca-transformation. \square

Analogously to the ATG case, Theorem 4.3 leads to a pseudo-FPTAS for approximate NE computation, as formalised in Theorem 3.1. This follows from the fact that g is bounded by a factor polynomial in $\max_{j,\mathbf{a},\mathbf{b}_j} U_j(\mathbf{a}, \mathbf{b}_j)$ (rather than in the "size" of U_j , which is polynomial in the number of bits required to represent this value). Consequently, the maximum number of iterations is bounded by a factor $\text{poly}(\Gamma, \max_{j,\mathbf{a},\mathbf{b}} U_j(\mathbf{a}, \mathbf{b})) / \varepsilon^4$.

5 Experimental Evaluation

In this section, we evaluate empirically the MATG-GDM algorithm. We conducted experiments on different configurations of random games as well as on network security games [Jai+11]. In both cases, we address two key questions: a) Is Algorithm 1 scalable? b) with respect to which parameters? The answers to these questions determine the algorithm's practical usability. Experiments were implemented in Python 3.11 using the JAX and Optax libraries [Bra+18; Dee+20].

Each configuration $\{n\} \times \{m\} / \{a\}$ is read as an MATG with n team agents, m adversaries and a actions per agent.

To evaluate the quality of the solutions found, we calculate the NE-GAP, which measures the maximum unilateral deviation in value from equilibrium (i.e. a strategy \mathbf{x}, \mathbf{y} is an ε -NE if $\text{GAP}(\mathbf{x}, \mathbf{y}) \leq \varepsilon$). In a MATG, the NE-GAP is defined as the maximum of the Team-GAP (T-GAP) and the Adversaries-GAP (A-GAP), specified as follows:

$$\begin{aligned} \text{T-GAP}(\mathbf{x}, \mathbf{y}) &= \max_{i \in \mathcal{N}, a_i \in \mathcal{A}_i} \sum_{j \in \mathcal{M}} (U_j(a_i, \mathbf{x}_{-i}, \mathbf{y}_j) - U_j(\mathbf{x}, \mathbf{y}_j)) \\ \text{A-GAP}(\mathbf{x}, \mathbf{y}) &= \max_{j \in \mathcal{M}, b_j \in \mathcal{B}_j} U_j(\mathbf{x}, b_j) - U_j(\mathbf{x}, \mathbf{y}_j) \\ \text{GAP}(\mathbf{x}, \mathbf{y}) &= \max(\text{T-GAP}(\mathbf{x}, \mathbf{y}), \text{A-GAP}(\mathbf{x}, \mathbf{y})) \end{aligned}$$

5.1 Random Games

Experimental Setup We first test various configurations using randomly generated instances of MATGs with rewards drawn uniformly from the interval $[0, 1]$. All strategies were initialised randomly.

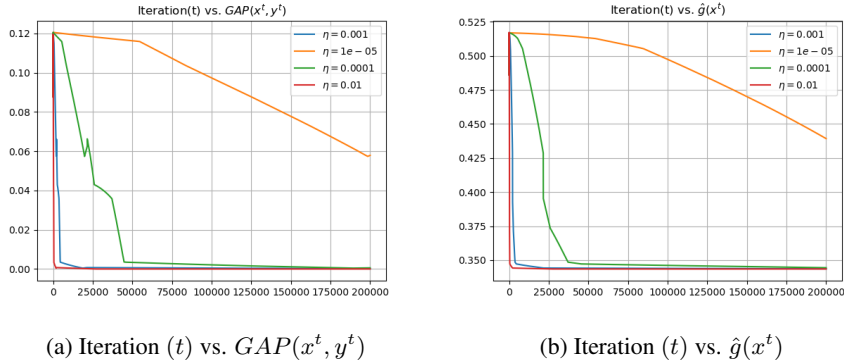


Figure 3: Results of MATG-GDM on a 2v2 MATG instance with 3 actions executed over $T = 2 \times 10^5$ timesteps with different learning rates.

Configuration $\{n\}v\{m\}/\{a\}$	$\eta = 0.001$		$\eta = 0.0001$	
	Best Timestep (t^*)	$GAP(x^{t^*}, y^{t^*})$	Best Timestep (t^*)	$GAP(x^{t^*}, y^{t^*})$
3v1/6	1200 ± 461	0.009 ± 0.002	1052 ± 674	0.014 ± 0.006
3v3/6	3615 ± 4216	0.007 ± 0.003	1225 ± 621	0.015 ± 0.002
3v6/6	6248 ± 6443	0.004 ± 0.001	1645 ± 1037	0.015 ± 0.004
4v1/6	944 ± 572	0.005 ± 0.001	1043 ± 562	0.007 ± 0.002
4v3/6	606 ± 576	0.005 ± 0.001	1122 ± 605	0.006 ± 0.001
4v6/6	960 ± 527	0.005 ± 0.002	1104 ± 581	0.007 ± 0.002

Table 1: Average performance (over 10 trials) of the MATG-GDM algorithm on different configurations of Random games for two different learning rates.

Results Figure 3 first illustrates the effect of the learning rate on MATG-GDM for a 2v2/3. Figure 3(a) plots the NE-GAP of the solution found at each iteration of the algorithm. Figure 3(b) plots $\hat{g}(x^t; \varepsilon)$, an approximation of $g(x^t; \varepsilon)$, obtained by taking $x^{t+1} \approx \text{prox}_{\phi/2\ell}(x^t; \varepsilon)^2$. We observe that $GAP(x^t, y^t)$ does not decrease monotonically, whereas $\hat{g}(x^t)$ does, as prescribed by the theory.

Table 1 reports the average performance of the MATG-GDM algorithm for random graphs with team sizes $n \in \{3, 4\}$ competing against $m \in \{1, 3, 6\}$ adversaries, with the number of actions a fixed to 6. Results per configuration are averaged over 10 independent trials run up to $T = 2 \cdot 10^4$ iterations. Performance is compared across two different learning rates, $\eta \in \{10^{-3}, 10^{-4}\}$. MATG-GDM was terminated early if $\hat{g}(x^t)$ stayed within a 10^{-3} -length neighbourhood for over 500 timesteps.

The results show that MATG-GDM is effective in practice, consistently finding high-quality approximate solutions close to equilibrium, with an average gap of in the range 0.004 – 0.01. Moreover, these solutions are typically identified at an iteration

²Refer Section 3.3, [PB+14]

Configuration $\{n\} \vee \{m\} / \{a\}$	$\eta = 0.001$		$\eta = 0.0001$	
	Best Timestep (t^*)	$GAP(x^{t^*}, y^{t^*})$	Best Timestep (t^*)	$GAP(x^{t^*}, y^{t^*})$
3v1/16	5331 ± 537	0.00004 ± 0.00002	4106 ± 5515	0.057 ± 0.023
3v3/16	2580 ± 655	0.0008 ± 0.001	12768 ± 3371	0.0069 ± 0.013
3v6/16	2470 ± 639	0.0003 ± 0.0003	7399 ± 489	0.004 ± 0.004
4v1/16	6690 ± 1536	0.0002 ± 0.0005	3313 ± 3191	0.053 ± 0.034
4v3/16	2852 ± 475	0.0003 ± 0.0003	15334 ± 4337	0.006 ± 0.006
4v6/16	2372 ± 879	0.0008 ± 0.0007	6294 ± 2367	0.006 ± 0.002

Table 2: Average performance (over 10 trials) of the MATG-GDM algorithm on different configurations of Network Security games for two different learning rates.

(t^*) significantly lower than the maximum number of allowed iterations ($T = 2 \cdot 10^4$). We also observe that the gap of the best solution found as well as the iteration at which such solution was found does not vary significantly neither across different problem configurations nor across the two tested learning rates ($\eta \in \{0.001, 0.0001\}$). This shows that the algorithm is scalable even in configurations with multiple adversaries. Additional experiments are provided in the Supplementary material.

5.2 Network Security Games

Description of Game and Experimental Setup For the second evaluation, we solve Network Security games (NetSec) on a graph. In this game, a team of n agents has to protect k nodes against m adversaries. Each adversary has distinct rewards for each node i.e. they have different objectives. They receive a positive reward by visiting a node unprotected by a team agent, and 0 otherwise. As in Section 5.1, each configuration is launched with 10 randomly generated instances, and the GAP and \hat{g} statistics are measured.

Results Table 2 reports the average performance of MATG-GDM over NetSec games with $n \in \{3, 4\}$ team agents against $m \in \{1, 3, 6\}$ adversaries with $a = 16$ actions each. MATG-GDM converges to good solutions on these larger instances, with GAP typically in the range $\{10^{-3}, 10^{-4}\}$. While convergence time scales with the number of actions, the algorithm usually converges well within $T = 2 \cdot 10^4$ iterations. In conclusion, MATG-GDM remains practical for these instances as well.

6 Conclusions and future work

In this paper, we introduced the *Multi-Adversarial Team Games (MATG)* framework, which generalizes 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 (e.g. anti-poaching) and strategic board games (e.g. bridge).

Our main theoretical contribution has been to show, by extending the results of [Ana+23], that a ε -Nash equilibrium of a MATG can be computed in time polynomial in the game's natural parameters and in $1/\varepsilon$. In addition, we provided the first implementation of a polynomial-time algorithm allowing to compute ε -NE in MATGs (and by extension in ATGs). The efficiency of this algorithm has been illustrated on benchmark problems, corroborating our theoretical findings.

A natural extension of this work concerns the study of multi-adversarial extensions of Adversarial Team *Markov* Games [Kal+23]. In particular, we hypothesize that the independence of adversaries in such *Multi-Adversarial Team Markov Games* may also enable the efficient computation of approximate equilibrium strategies, possibly with the mobilisation of learning algorithms [KYP24].

References

- [VK97] Bernhard Von Stengel and Daphne Koller. “Team-maxmin equilibria”. In: *Games and Economic Behavior* 21.1-2 (1997), pp. 309–321.
- [MGB03] H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. “Planning in the presence of cost functions controlled by an adversary”. In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 2003, pp. 536–543.
- [Bor+08] Christian Borgs et al. “The myth of the folk theorem”. In: *Proceedings of the fortieth annual ACM symposium on Theory of computing*. 2008, pp. 365–372.
- [Han+08] Kristoffer Arnsfelt Hansen et al. “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. 2008, pp. 684–695.
- [HCP09] Erik Halvorson, Vincent Conitzer, and Ronald Parr. “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*. Ed. by Craig Boutilier. 2009, pp. 159–166. URL: <http://ijcai.org/Proceedings/09/Papers/037.pdf>.
- [Jai+11] Manish Jain et al. “A double oracle algorithm for zero-sum security games on graphs”. In: *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, May 2-6, 2011, Volume 1-3*. Ed. by Liz Sonenberg et al. IFAAMAS, 2011, pp. 327–334. URL: <http://portal.acm.org/citation.cfm?id=2030518%5C&CFID=69153967%5C&CFTOKEN=38069692>.

- [PB+14] Neal Parikh, Stephen Boyd, et al. “Proximal algorithms”. In: *Foundations and trends® in Optimization* 1.3 (2014), pp. 127–239.
- [Bra+18] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. 2018. URL: <http://github.com/google/jax>.
- [Wan+19] Yufei Wang et al. “Deep reinforcement learning for green security games with real-time information”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 1401–1408.
- [Dee+20] DeepMind et al. *The DeepMind JAX Ecosystem*. 2020. URL: <http://github.com/google-deepmind>.
- [Ana+23] Ioannis Anagnostides et al. *Algorithms and Complexity for Computing Nash Equilibria in Adversarial Team Games*. 2023. arXiv: 2301.02129 [cs.GT]. URL: <https://arxiv.org/abs/2301.02129>.
- [Kal+23] Fivos Kalogiannis et al. “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, 2023. URL: <https://openreview.net/pdf?id=mjzm6btqgV>.
- [Lam+23] Wai Yee Lam et al. “Using a crime prevention framework to evaluate tiger counter-poaching in a Southeast Asian rainforest”. In: *Frontiers in Conservation Science* 4 (2023). ISSN: 2673-611X. DOI: 10.3389/fcosc.2023.1213552. URL: <https://www.frontiersin.org/articles/10.3389/fcosc.2023.1213552>.
- [KYP24] Fivos Kalogiannis, Jingming Yan, and Ioannis Panageas. “Learning Equilibria in Adversarial Team Markov Games: A Nonconvex-Hidden-Concave Min-Max Optimization Problem”. In: *arXiv preprint arXiv:2410.05673* (2024).

Efficiently computing Nash Equilibria in Multi-Adversarial Team Games

Appendix

In this appendix, we first go over additional related work to Multi-Adversarial Team Games. For completeness, we also cite supplementary theorems referred to in the main paper. Finally, we provide additional experiments for random instances of MATGs to better illustrate the performance of MATG-GDM over a variety of configurations.

A Additional related works

Related Work *Single-adversary team games.* The Team-Maxmin Equilibrium (TME), a celebrated solution concept for ATGs which dates back to the pioneering work of von Stengel and Koller [VK97], serves as a prominent solution for ATGs. A TME is a Nash Equilibrium (NE) with three key properties: (1) it always exists; (2) it is unique¹; and (3) it maximizes the team’s utility. Furthermore, [Bas+17] demonstrated that the team’s payoff in a NE (other than TME) can be arbitrarily smaller than its payoff in a TME, motivating efforts to approximate TMEs efficiently.

Unfortunately, TMEs are computationally intractable (FNP-hard), as proven in [Bor+08; Han+08], even in the case of a ATG with two teammates. Moreover, [Han+08] (with corrections by [Bas+17]) showed that expressing MEs may require irrational probabilities, complicating their computation and practical adoption.

Despite these challenges, practical algorithms for approximating TMEs have been proposed [Bas+17; ZA20b]. In particular, some works [ZS22; Bas+17] focus on deriving the team’s strategy from a *correlated* TME (CTME) that considers that team members can synchronize their actions (i.e. can jointly decide and execute their strategy). While CTMEs can be computed via linear programming (LP) by reducing the problem to a two-player zero-sum game, the size of the team’s player action space in such reduction is exponential in the size of the team and therefore approximation algorithms [McA+23] for games with many team members have been proposed in practice. Nevertheless, it is known that the inefficiency gap between TME and CTME team’s payoffs can

¹Except in degenerate cases, thereby avoiding equilibrium selection issues.

be, in the worst-case, exponential in the size of the team [Jia+13; Bas+17]. [Bas+17] provided the first experimental evaluation of TME algorithms, benchmarking²: (1) global optimization using an off-the-shelf MINLP solver; (2) an algorithm that reconstructs the team’s mixed strategy profile from a CTME (computed via the corresponding LP); (3) the ϵ -additive approximation from [Han+08] based on simple strategies; and (4) the *IteratedLP* algorithm, which iteratively updates the team’s strategy profile by replacing the strategy of the team member that obtained the best improvement on the 2-player zero-sum game between her and the adversary³. Their results highlight the superior performance of global optimization up to the limits in which it does not run out of memory⁴, while *IteratedLP* emerged as a scalable alternative with competitive results. [ZA20b] extended the double oracle (DO) framework [MGB03] to TME computation in ATGs. As common in DO methods, the worst-case runtime can exceed direct TME computation but it empirically performs well in practice. Seeking tractable alternatives, [Ana+23] analyzed the complexity of approximating NEs (not necessarily the TME) in ATGs, showing it is CLS-complete. Their constructive proof yielded an algorithm for ϵ -approximate NEs with polynomial complexity in all natural game parameters and $1/\epsilon$. Our work extends this algorithm to multi-adversarial ATGs (MATGs), showing its complexity remains polynomial and depends on $\sum_{j \in \mathcal{M}} |A_j|$ instead of $\prod_{j \in \mathcal{M}} |A_j|$, overcoming the *curse of multi-agents* also for the adversarial side. Moreover, we provide the first empirical evaluation of such algorithms, since [Ana+23] lacked any empirical evaluation for the single-adversary setting.

Two-team adversarial games. Extensions of ATGs to multi-adversarial settings have been explored in adversarial two-team games [KPV21; SV19], where adversaries also form a team. However, our focus is on independent adversaries. [HMN24] analyzed NE computation in two-team zero-sum *polymatrix* games, including a special case with independent adversaries that they prove CLS-complete. However, their setting is different since it assumes pairwise utility functions, while here we study the extension to multiple adversaries in the original version of the game with utilities defined between each adversary and the entire team.

Adversarial Team Markov games. Finally, other works [CG18; ZA20a; Cel+19; Car+22; Kal+23; KYP24] have studied ATGs in their sequential version, i.e. as a Markov game or an extensive-form game. 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.

²In their experiments they vary the team’s size and the number of player’s action in the ranges [2, 4] and [5, 150] respectively.

³Fixing the strategy of the other teammates to the ones in the current team’s strategy.

⁴Global optimization runs for team size equal 2 (any number of actions), equal 3 (up to 45 actions) and equal 4 (up to 20 actions).

B Theorems referred to in the paper

[Ana+23] have shown that the computation of an ϵ -Nash equilibrium in a single-adversary team game can be computed in a number of iterations of Algorithm 1 (restricted to the case of a single adversary), which grows polynomially in the size of Γ and in $1/\epsilon$.

As a first step, let us recall Theorems 3.2 and 3.3 and Corollary 3.2 of [Ana+23], in the case of ATG:

Theorem B.1 (Anagostides, 2023, Theorem 3.2).

Let function $V_{\mathbf{x}} : \mathcal{X} \rightarrow \mathbb{R}$ be defined as $V_{\mathbf{x}}(\mathbf{x}') = \max_{\mathbf{y}} U(\mathbf{x}', \mathbf{y}) + \ell \|\mathbf{x} - \mathbf{x}'\|_2$.

Then, there exists a $\text{poly}(n, \sum_{i=1}^n |\mathcal{A}_i|, |\mathcal{B}|, \log(1/\epsilon))$ -time algorithm $\text{Alg} : \mathcal{X} \rightarrow \mathcal{X}$ such that

$$\min_{\mathbf{x}'} V_{\mathbf{x}}(\mathbf{x}') \leq V_{\mathbf{x}}(\text{Alg}(\mathbf{x})) \leq \min_{\mathbf{x}'} V_{\mathbf{x}}(\mathbf{x}') + \epsilon.$$

Theorem B.2 (Anagostides, 2023, Theorem 3.3). Consider any precision $\epsilon > 0$, and let

$$g : \mathcal{X} \ni \mathbf{x} \rightarrow \max_{\mathbf{y}' \in \mathcal{Y}} U(\text{prox}_{\phi/2\ell}(\mathbf{x}; O(\epsilon^4)), \mathbf{y}') + \|\mathbf{x} - \text{prox}_{\phi/2\ell}(\mathbf{x}; O(\epsilon^4))\|_2^2.$$

For any adversarial team game $\Gamma(n, U)$, $\text{GRADIENTDESCENTMAX}(\Gamma, \epsilon)$ with a sufficiently small learning rate $\eta = O(\epsilon^2)$ satisfies

$$g(\mathbf{x}^{(t)}) < g(\mathbf{x}^{(t-1)}) - \Omega(\epsilon^4)$$

for any iteration t until termination.

Corollary B.1 (Anagostides, 2023, Corollary 3.2, Pseudo-FPTAS for Approximate Nash). Consider any precision $\epsilon > 0$. For any adversarial team game $\Gamma(n, U)$, $\text{GRADIENTDESCENTMAX}(\Gamma, \epsilon)$ with a sufficiently small learning rate $\eta = O(\epsilon^2)$ yields an ϵ -approximate Nash equilibrium after a sufficiently large $T = \text{poly}(\Gamma/\epsilon^4)$. Further, under Assumption 3.1, every iteration of $\text{GRADIENTDESCENTMAX}$ can be implemented in polynomial time.

C Additional experiments

Instance of 3v3/6 In Figure 1, we illustrate a random instance of a 3v3/6 configuration with learning rates $\eta \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. We observe again that in both graphs, the gradient steps follow the same trajectory, but smaller learning rates dilate the curve. Furthermore, $GAP(x^t, y^t)$ does not monotonically decrease. It is more interesting to note that for the largest learning rate $\eta = 10^{-2}$, both GAP and \hat{g} oscillate in the neighbourhood once converged. These patterns held true for all further testing.

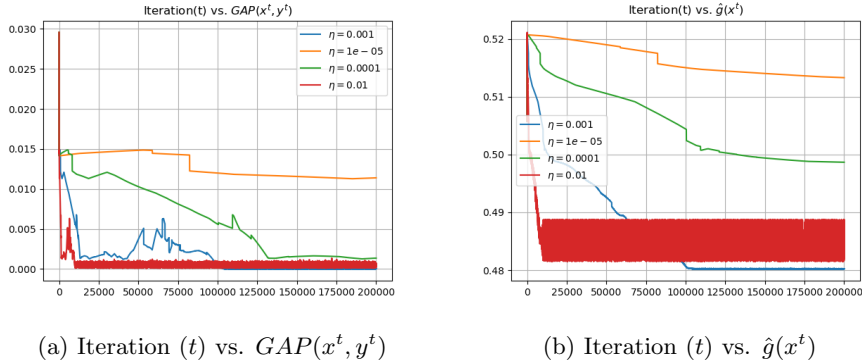


Figure 1: Example of a 3v3 MATG Instance with 6 actions for all agents, executed with different learning rates over $T = 2 \times 10^5$ timesteps.

Random Games In this section, we describe further experiments on Random Games. The tested configurations were $n \in \{3, 4\}$ players against $m \in \{1, 3, 6\}$ adversaries, with $k \in \{4, 6, 8\}$ actions each. Each configuration was tested with two different learning rates $\eta \in \{10^{-3}, 10^{-4}\}$. The results in Table 1 are the averages over 10 random instances for each configuration.

References

- [VK97] Bernhard Von Stengel and Daphne Koller. “Team-maxmin equilibria”. In: *Games and Economic Behavior* 21.1-2 (1997), pp. 309–321.
- [MGB03] H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. “Planning in the presence of cost functions controlled by an adversary”. In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 2003, pp. 536–543.
- [Bor+08] Christian Borgs et al. “The myth of the folk theorem”. In: *Proceedings of the fortieth annual ACM symposium on Theory of computing*. 2008, pp. 365–372.
- [Han+08] Kristoffer Arnsfelt Hansen et al. “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. 2008, pp. 684–695.
- [Jia+13] Albert Xin Jiang et al. “Defender (mis) coordination in security games”. In: (2013).

Game Configuration	$\eta = 0.001$		$\eta = 0.0001$	
	Best Timestep (t^*)	Minimum $GAP(x^{t^*}, y^{t^*})$	Best Timestep (t^*)	Minimum $GAP(x^{t^*}, y^{t^*})$
3v1/4	2077 \pm 2784	0.014 \pm 0.004	750 \pm 749	0.020 \pm 0.005
3v3/4	5527 \pm 4757	0.011 \pm 0.006	1200 \pm 599	0.026 \pm 0.010
3v6/4	4031 \pm 2046	0.010 \pm 0.005	909 \pm 675	0.031 \pm 0.012
3v1/6	1200 \pm 461	0.009 \pm 0.002	1052 \pm 674	0.014 \pm 0.006
3v3/6	3615 \pm 4216	0.007 \pm 0.003	1225 \pm 621	0.015 \pm 0.002
3v6/6	6248 \pm 6443	0.004 \pm 0.001	1645 \pm 1037	0.015 \pm 0.004
3v1/8	1219 \pm 648	0.006 \pm 0.001	888 \pm 725	0.010 \pm 0.003
3v3/8	2238 \pm 1296	0.008 \pm 0.002	1375 \pm 439	0.012 \pm 0.002
3v6/8	2180 \pm 1031	0.005 \pm 0.001	1327 \pm 398	0.013 \pm 0.003
4v1/4	1109 \pm 452	0.008 \pm 0.002	813 \pm 663	0.009 \pm 0.002
4v3/4	1243 \pm 658	0.012 \pm 0.004	1073 \pm 580	0.014 \pm 0.004
4v6/4	1542 \pm 1104	0.009 \pm 0.002	1062 \pm 647	0.013 \pm 0.004
4v1/6	944 \pm 572	0.005 \pm 0.001	1043 \pm 562	0.007 \pm 0.002
4v3/6	606 \pm 576	0.005 \pm 0.001	1122 \pm 605	0.006 \pm 0.001
4v6/6	960 \pm 527	0.005 \pm 0.002	1104 \pm 581	0.007 \pm 0.002
4v1/8	967 \pm 381	0.003 \pm 0.001	815 \pm 438	0.003 \pm 0.001
4v3/8	1388 \pm 222	0.003 \pm 0.0007	1122 \pm 605	0.006 \pm 0.001
4v6/8	1116 \pm 415	0.003 \pm 0.0006	1355 \pm 428	0.004 \pm 0.001

Table 1: Table comparing configurations (10 random instances per configuration) over different learning rates using Algorithm 1. Note that the stopping criteria was kept identical.

- [Bas+17] Nicola Basilico et al. “Team-maxmin equilibrium: efficiency bounds and algorithms”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017.
- [CG18] Andrea Celli and Nicola Gatti. “Computational results for extensive-form adversarial team games”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.
- [Cel+19] Andrea Celli et al. “Coordination in adversarial sequential team games via multi-agent deep reinforcement learning”. In: *arXiv preprint arXiv:1912.07712* (2019).
- [SV19] Leonard J Schulman and Umesh V Vazirani. “The duality gap for two-team zero-sum games”. In: *Games and Economic Behavior* 115 (2019), pp. 336–345.
- [ZA20a] Youzhi Zhang and Bo An. “Computing team-maxmin equilibria in zero-sum multiplayer extensive-form games”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 02. 2020, pp. 2318–2325.

- [ZA20b] Youzhi Zhang and Bo An. “Converging to team-maxmin equilibria in zero-sum multiplayer games”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 11033–11043.
- [KPV21] Fivos Kalogiannis, Ioannis Panageas, and Emmanouil-Vasileios Vlatakis-Gkaragkounis. “Teamwork makes von neumann work: Min-max optimization in two-team zero-sum games”. In: (2021).
- [Car+22] Luca Carminati et al. “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. 2022, pp. 2638–2657.
- [ZS22] Brian Hu Zhang and Tuomas Sandholm. “Team correlated equilibria in zero-sum extensive-form games via tree decompositions”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 5. 2022, pp. 5252–5259.
- [Ana+23] Ioannis Anagnostides et al. *Algorithms and Complexity for Computing Nash Equilibria in Adversarial Team Games*. 2023. arXiv: 2301.02129 [cs.GT]. URL: <https://arxiv.org/abs/2301.02129>.
- [Kal+23] Fivos Kalogiannis et al. “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, 2023. URL: <https://openreview.net/pdf?id=mjzm6btqgV>.
- [McA+23] Stephen McAleer et al. “Team-PSRO for learning approximate TMECor in large team games via cooperative reinforcement learning”. In: *Advances in Neural Information Processing Systems 36* (2023), pp. 45402–45418.
- [HMN24] Alexandros Hollender, Gilbert Maystre, and Sai Ganesh Nagarajan. “The Complexity of Two-Team Polymatrix Games with Independent Adversaries”. In: *arXiv preprint arXiv:2409.07398* (2024).
- [KYP24] Fivos Kalogiannis, Jingming Yan, and Ioannis Panageas. “Learning Equilibria in Adversarial Team Markov Games: A Nonconvex-Hidden-Concave Min-Max Optimization Problem”. In: *arXiv preprint arXiv:2410.05673* (2024).