A Game-Theoretic Utility Network for Cooperative Multi-Agent Decisions in Adversarial Environments

Anonymous Author(s)
Affiliation
email

Abstract

Many underlying relationships among multi-agent systems (MAS) in various scenarios, especially agents working on dangerous, hazardous, and risky situations, can be represented in terms of game theory. In adversarial environments, the adversaries can be intentional or unintentional based on their needs and motivations. Agents will adopt suitable decision-making strategies to maximize their current needs and minimize their expected costs. In this paper, we propose a new network model called Game-Theoretic Utility Tree (GUT) to achieve cooperative decision-making for MAS in adversarial environments combining the core principles of game theory, utility theory, probabilistic graphical models (PGM). Through calculating multi-level Game-Theoretic computation units, GUT can decompose high-level strategies into executable lower levels. Then, we design a game-like exploration task to validate our model against a cooperative decision-making algorithm based on the state-of-the-art QMIX approach. Also, we implement different predictive models for MAS working with incomplete information to estimate adversaries' state. Our experimental results demonstrate that the GUT significantly enhances cooperation among MAS to successfully complete the assigned tasks with lower costs and higher winning probabilities against adversaries.

1 Introduction

Natural systems have been the key inspirations in the design, study, and analysis of Multi-Agent Systems (MAS) and Swarm Robotics [1, 2, 3, 4]. Cooperation in MAS can maximize global system utility and guarantee sustainable development for each group member [5, 6]. Also, it is essential to perceive the environment and recognize the threats and adversaries in the environment cooperatively among all agents in the MAS team. An **Adversary** in the environment impairs the ability of the individual agents and the MAS as a whole to achieve their global and local tasks [7, 8, 9].

Recently, researchers have been able to combine the disciplines of Robotics and adversarial MAS into *Adversarial Robotics* focusing on autonomous agents operating in adversarial environments [10, 11, 12, 7, 13]. Following the examples from Information Systems [14], we can classify an adversary based on its needs and motivations into two general categories: **intentional** (such as an enemy or intelligent opponent agent, which actively impairs the MAS needs and capabilities) and **unintentional** (like obstacles and weather, which might passively threaten MAS abilities) adversary. In this paper, we refer to them as *deliberate* **Monsters** and *accidental* **Obstacles**, respectively.

Related Work Most of past and current research focus on the unintentional adversaries in the environment, such as path planning avoiding static or dynamical obstacles, formation control avoiding collisions and so forth [15, 10, 11]. This is particularly applicable to urban search and rescue missions and robots deployed in disaster environments. The agents are more concerned about unintentional threats such as radiation, clutters, or natural forces such as wind, fire, rain, etc. [16].

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

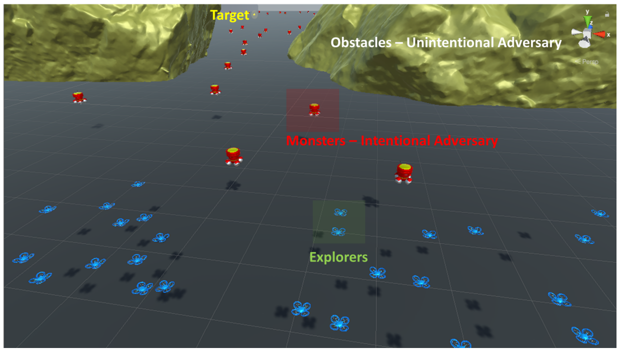


Figure 1: An illustrative game scenario where the paths to a target of the Explorer agents are blocked by Monster agents.

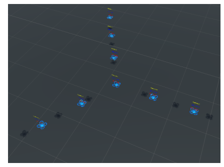


Figure 2: Patrolling

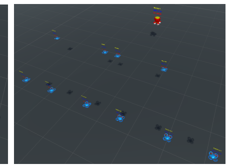


Figure 4: Defending

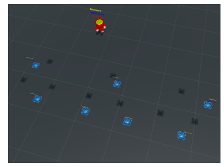


Figure 3: Attacking

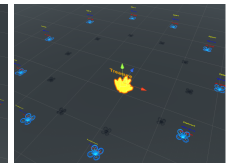


Figure 5: Circling

On the other hand, there is little research done in studying confrontational strategies, preventive control, and behaviors to mitigate intentional adversaries (or enemy agents). For instance, Lin [17] examined the problem of defending patrol tasks against a sequential attack in a knowledgeable adversarial environment. Prorok [18] studied multi-robot privacy with an adversarial approach. Sanghvi and Sycara [12] identified a swarm vulnerability and studied how an adversary takes advantage of it. Paulos and Kumar [19] describe an architecture for training teams of identical agents in a defense game. As we can see, most of these research focuses on the privacy or cyber-adversaries rather than physical adversaries confronting the agents in a Spatio-temporal domain.

Cooperative decision-making among the agents[1] is essential to address the threats posed by intentional physical adversaries. This is similar to the multi-player *pursuit and evasion* game problem [20, 7, 21, 22]. Recent works include optimal evasion strategies and task allocation [23, 24, 25] and predictive learning from agents' behaviors [26]. However, it is more realistic and useful for the MAS to achieve a given task with minimum cost while simultaneously considering the adversaries. Unfortunately, to the best of our knowledge, none of the methods focus on this domain.

Contributions To address the gaps in the literature, we introduce a new network model called *Game-Theoretic Utility Tree (GUT)* to achieve cooperative decision-making for MAS in adversarial environments compromising both intentional and unintentional adversaries. GUT combines the principles and merits of *Game Theory* [27, 28], *Utility Theory* [29, 30], *Probabilistic Graphical Model (PGM)* [31, 32], and *Tree Structure* exploiting its hierarchy to mitigate the threats posed by intentional adversarial agents. We also propose a simple but efficient algorithm called "*Adapting The Edge*," which combines individual adapting behaviors and group cooperation together to tackle static unintentional adversaries as well. To evaluate our method, we present a new *Explorers and Monsters Game* domain (Fig. 1) to demonstrate the efficacy of our approach and analyze the performance of explorers to reach their targets against the state-of-the-art QMIX [33] based decision models.

2 Problem Statement

In our *Explorers and Monsters Game*, there are n Explorers tasked to explore the environment and collect rewards (reaching treasure locations p_{tr}) in an uncertain environment and each agent with the (health) *HP* cost $h_i(i \in n)$. There are m Monsters and k Obstacles distributed in the environment. Explorers i $(i \in n)$ and Monsters j $(j \in m)$ have strategy space $S_{e_i}(s_1, s_2, ..., s_p)$ and $S_{m_j}(s_1, s_2, ..., s_q),\ p, q \in Z^+$,, respectively. Every strategy has corresponding *actions* to execute $s(at_1, at_2, ..., at_r), r \in Z^+$. Monsters will adopt (known) strategies that reduces the HP cost of the Explorers. Supposing the group of explorer's successful possibility (win rate) getting the treasure is W, our problem can be described as finding a suitable strategy for explorers S_e in the solution space under the premise of maximizing the W to minimize the explorer group's *HP* cost, formulated as:

$$\begin{aligned} &\arg\max_{S_e} \quad W(S_e|S_m, k, p_{tr}); \\ &\mathit{subject\ to} \quad \min \sum_{i=1,j=1}^{n,m} h_i(S_{e_i}|S_{m_j}, k, p_{tr}). \end{aligned} \tag{1}$$

[1]We use the terms Agents and Robots interchangeably. For more related work, see Appendix. A.

3 *GUT* Approach Overview

We introduce *Game-Theoretic Utility Tree (GUT)* (see Fig. 6), a new network for solving the *Explorers and Monsters Game* through cooperative decision-making of Explorers in adversarial environments. In each level, GUT distributes various ***Game-Theoretic Computation Units*** (see Appendix. B for more information), which combines agents' tactics in the current condition, the possibility of the previous condition, and relative environment's information. It also provides different *Utility Function* to quantify related agents' needs. Through calculating the *Game-Theoretic Computation Unit*, an individual agent can get the corresponding possibility for all the solutions in the current unit. Then, it will iterate this process until it computes the leaf units at the lowest level. According to the PGM and relative Maximum A Posterior (MAP) theory [31], an agent can always find a suitable solution set with a maximum joint probability in the entire solution space. This allows GUT decompose high-level strategies into executable lower levels.

In our *Explorers and Monsters Game*, we build a three-level *GUT* to elaborate it. The first level (high-level) determines whether or not attacking (or defending). The second level is to figure out the specific agent to be attacked (or defended). The third level (lowest level) decides how the agents should group themselves to adapt to the current situation.

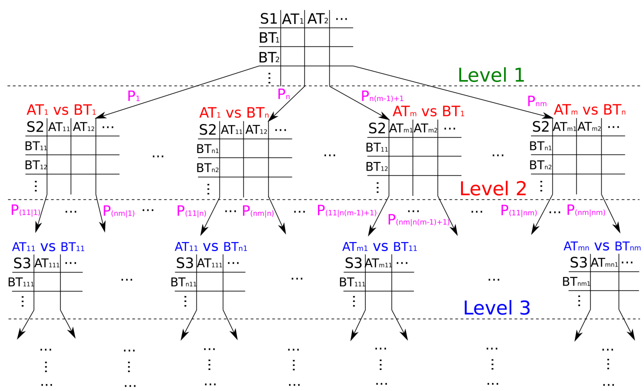


Figure 6: General Individual Robot's *GUT*

More specifically, in the first level we define Explorers and Monsters both have two strategies: *Attack* and *Defend*, which are represented through *Triangle* (Fig. 3) and *Regular Polygon* (Fig. 4) formation shapes, respectively. According to the payoff matrix (Table. 1) in a zero-sum game, they can calculate the strategy which can fit the current situation. Based on the precondition in the first level, they need to decide attacking or to defend against the specific Monster agent. For example, we assume that in the attacking model, Explorers and Monsters have two kinds of behaviors: attacking the *nearest* agent or the attacking the agent with the *lowest* or the *lowest* ability in the formation. Similarly, the defending model also can choose to defend against the *nearest* agent or based on the opponents attacking ability. Through the payoff matrix in Table. 2, they can confirm the target sequence. Finally, in the lowest level, with the tactics payoff matrix in Table. 3, individuals can calculate the final tactics, such as the number of groups in Explorers and the behavior of following others or not in Monsters.

Through this process, we decompose the individual agent's strategy into three levels, and each level focuses on different utilities corresponding to different needs or requirements. In order to simplify this calculation process, across every level, we respectively use Winning probability (W), the relative expected cost of energy ($E(e)$) and HP (health power) ($E(hp)$) – the expected utility difference of both sides – to measure every level's utility. In the first level, the utility presents the Winning probability, which relates to the current perceived adversaries' number and individual attacking and defending ability. In the second level, we consider the relative expected energy cost to describe this level's utility, which depends on the agents' distribution and numbers. In the lowest level, we use the relative expected HP cost representing the utility caused by the individual and group's current information, such as the number of groups and agent's current energy level.

For the unintentional adversaries, we design the *Adapting The Edge* algorithm (Alg. 2), which can help individual agents tackling static unintentional adversaries and adapting their edge's trajectory until it finds a suitable route to the goal point. In this process, through the communication and information sharing with other agents, an individual can select the direction having less possibility of a potential collision with the unintentional adversaries to move. In our scenarios, the two mountains represent the unintentional adversaries, and explorers need to find a path passing through them.

In this game, the Explorer's group into *Patrol Formation* (see Fig. 2) to detect the unknown world at the beginning, through tackling various threats in the past, they will select the shortest path to

Utility \ EB MB	Attack	Defend
Attack	W_{AA}	W_{DA}
Defend	W_{AD}	W_{DD}

Table 1: Level 1 (Attack/Defend)- Explorer & Monster Tactics Payoff Matrix.

Utility \ EB MB	Nearest	A Lowest	A Highest
Nearest	$E(e)_{NN}$	$E(e)_{A_L N}$	$E(e)_{A_H N}$
A Lowest	$E(e)_{NA_L}$	$E(e)_{A_L A_L}$	$E(e)_{A_H A_L}$
A Highest	$E(e)_{NA_H}$	$E(e)_{A_L A_H}$	$E(e)_{A_H A_H})$

Table 2: Level 2 (Who to Attack/Defend)- Explorer & Monster Tactics Payoff Matrix.

Utility \ EB MB	One Group	Two Group	Three Group
Independent	$E(hp)_{1I}$	$E(hp)_{2I}$	$E(hp)_{3I}$
Dependent	$E(hp)_{1D}$	$E(hp)_{2D}$	$E(hp)_{3D}$

Table 3: Level 3 (How to Attack/Defend) - Explorer & Monster Tactics Payoff Matrix.

Algorithm 1: Explorer's Collective Strategy Using *GUT* Model in *Explorers and Monsters Game*.

Input: Explorers' and Monsters' states
Output: formation shape s; current attacking target t; number of groups g.

```
set state = "level one";
while the changing of Monster's number == True And Monster's number != 0 do
    if state=="level one" then
        Compute the Nash Equilibrium;
        Get the most feasible formation shape s;
        state = "level two"
    else if state=="level two" And s != Null then
        Compute the Nash Equilibrium;
        Get the most feasible attacking target t;
        state = "level three"
    else if state=="level three" And s, t != Null then
        Compute the Nash Equilibrium;
        Get the most feasible number of groups g;
if Monster's number == 0 then
    s = "Patrol";
    g = 1;
return s, t, g
```

the treasure (Fig. 5). In the whole process, explorers present a kind of global behaviors performing *Collective Rationality* and caring about *Group interest*. In contrast, each Monster shows *Self-interest* and does not cooperate with other Monsters.

4 Formalization and Algorithms

In this section, through formalizing *Robot's Need Hierarchy* involving basic, safety, and capability needs (see Appendix. C for formalization of needs), we can define the adversary and formalize the decision process of intentional and unintentional adversaries in our *Explorers and Monsters Game*.

Definition 1 (Adversary)**.** *For certain state $s_1 \in S$ and a group of agents R_1 given the action series a_{1i} $\in A$ fulfilling task T. Supposing without any interruption the maximum teaming needs is max(N_1(s_1, a_{1i})). Considering another groups agents R_2 involving with reaction series a_{2k}, if Eq. (2) is satisfied, it can be defined R_2 as **Adversary** to R_1. In additional, if R_2's corresponding expected needs is not equal to current needs (Eq. (3)), it will be regarded as **Intentional Adversary**. Otherwise, we consider as **Unintentional Adversary** (Eq. (4)).*

$$max(N_1(s_1, a_{1i})) > max(N_{12}(s_{12}, a_{1j})), \quad i, j, k \in Z^+; \tag{2}$$
$$E(N_{21}|s_{21}, a_{2k}) \neq N_2; \tag{3}$$
$$E(N_{21}|s_{21}, a_{2k}) = N_2. \tag{4}$$

4.1 Intentional Adversaries Decision

Theorem 1 (GUT Decision)**.** *For n level GUT, group A and intentional adversaries B have corresponding zero-sum game G_i(A,B;N_{ti}), $i \in n$ (Fig. 6). Through computing the Nash Equilibrium of G_i in each level, A has at least one dominant strategy series (s_1, s_2, ... , s_n) in GUT.*

Proof. see Appendix. D.1 □

Below we also can guarantee an *n* levels *GUT* having the Maximum A Posterior (MAP).

Corollary 1 (GUT MAP)**.** *Supposing the joint probability of a GUT is P(x) = P(x_1, x_2, ... , x_n), there is unique x^* that maximize:*

$$x^* = \underset{x \in Val(x)}{\arg\max}\, P(x) \tag{5}$$

Proof. see Appendix. D.2 □

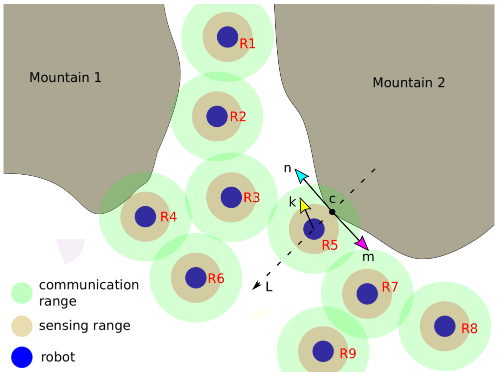


Figure 7: Illustration of "Adapt The Edge" algorithm for tackling (unintentional) obstacles.

Algorithm 2: Adapting The Edge

Input: Explorers' and Mountains' states
Output: moving direction r and distance Δd

while *The nearest collision point* $c \,! = Null$ **do**
  calculate the number n and m of non-collision agents in both side of the line l passing through c and perpendicular c's tangent;
  if $n > m$ **then**
    r = n side in line l;
    Δd = one step of agent's movement;
  else if $n == m$ **then**
    agent stop;
  else if $n < m$ **then**
    r = m side in line l;
    Δd = one step of agent's movement;
return r = current position to goal point, Δd

The *Utility Function* is critical to the design to determine whether or not the agent can calculate reasonable tactics. According to the discussion in Sec. 4 about the *Explorers and Monsters Game*, in this step, we first assume each robot has two primary utility attributes: *Energy Level* and *HP*, then analyze the three-level GUT from different needs perspective. See Appendix. D.3 for more analysis.

Through calculating the corresponding expected utility, agents can get the Nash Equilibrium [34, 35] according to each level's payoff matrix. After computing the entire *GUT*, it needs to combine each level's tactics and execute the integrated strategy in the planning level. Alg. 1 depicts the entire decision process.

4.2 Unintentional Adversaries Decision

When Explorers perceive the mountains (obstacles - static unintentional adversaries), they need to utilize the limited information available among all agents by communicating the perceived information. In our experiment, the scenario illustrated in Fig. 7. There are nine robots, and robots $R4$ and $R5$ detect the mountain. For robot $R5$, to avoid a collision, it needs to switch the current direction k to tangent's direction of the nearest collision point c. Since it has two directions, n and m, according to $R5$'s state, it should select the direction n, which has more non-collision robots currently.

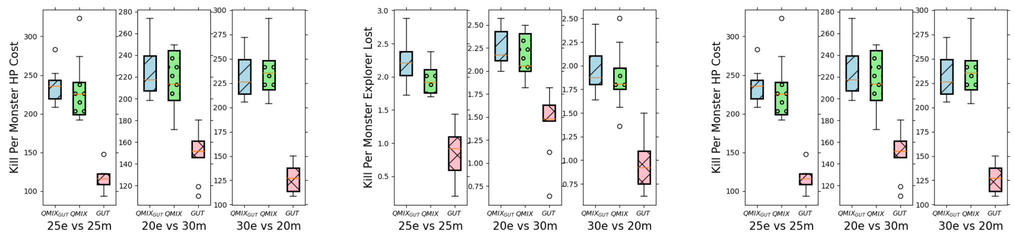
Specifically, considering the line L, which is vertical to a tangent through the point c, is the boundary. For the direction n, there are five robots but only four robots $R1$, $R2$, $R3$ and $R6$ do not have collision. We also can see the non-collision robots' number is three in the direction of m. So currently, $R5$ will select direction n to move certain distance Δd, then adjust the direction to the goal point to go forward and loop the entire process until it perceives no unintentional adversaries on its routing. Combining the two kinds of decision, we present the entire decision process as Alg. 2. For simplicity, we only select the maximum feasible solution at each level, entering the next level.

5 Evaluation through Simulations

Considering cross-platform, scalability, efficiency, and extendability of the simulation, we chose "Unity" [36] game engine to simulate the *Explorers and Monsters Game* and selected Gamebit [37] toolkit[2] for calculating each level's Nash Equilibrium. We evaluate *GUT* from two different perspectives; *Interaction Experiments* to demonstrate the performance of Explorer's cooperative strategies, and *Information Prediction* to demonstrate the cooperation when the perception of Monster's state is not accurate, thus using predictive models.

In the entire experiments, we suppose each Explorer has the same energy and HP levels initially, and every moving step will cost 0.015% energy. Also, every communication round and per time attacking will cost 0.006% and 0.01% energy, respectively. If Explorer is being attacked by Monster, it will cost 0.15% HP per time. For the Monster, per time attacking energy and per time attacked HP cost are 0.03% and 0.05%, respectively.

[2]Gamebit is an open-source collection of tools for doing computation in game theory and can build, analyze, and explore game models



(a) Explorer Average HP Cost (b) Kill Per Monster - Explorer Lost (c) Kill Per Monster - HP Cost

Figure 8: The Performance of Interaction Experiments with Different Proportion in Corresponding Scenarios.

5.1 Interaction Experiments

In this section, according to a cooperative decision-making algorithm based on the state-of-the-art QMIX [33, 38], we design three kinds of cooperative styles against *GUT* in different interactive and communication form as follow (for relative definition and experimental settings, see Appendix. E):

QMIX [*Partial Cooperation + Partial Communication*] QMIX [33] is a state-of-the-art value-based method applied to reinforcement learning in MAS. Here, we only focus on the decision making part of QMIX, which considers the global benefit yielding the same result as a set of individual rewards. This allows each agent to participate in a decentralized execution solely by choosing greedy actions for its rewards. Accordingly, we assume that each Explorer can cooperate, communicate, and share information with its perceiving explorers. Then based on the number of its observing explorers and monsters, it will calculate its winning rate to select attacking or defending the *hp lowest* Monsters.

QMIX$_{GUT}$ [*Noncooperation + Partial Communication*] In this situation, individual Explorer using *GUT* computes the winning utility based on its perceiving and partial communication information as above discussion. However, they do not get the consistency to attack the specific Monster or defend.

GUT [*Full Cooperation (collective Rationality) + Full Communication*] For our architecture, we assume each Explorer working in a full communication mode, which means that every group member can share its information. Then through the *GUT* calculating and decomposing their strategies to executable level, they finally get a consistent decision in the distributed system.

In these experiments, we do not involve unintentional adversaries Fig. 9 and consider three different proportion (M/E) between monster and explorer as follow: *20 explorers vs 30 monsters*, *25 explorers vs 25 monsters* and *30 explorers vs 20 monsters*. In this experiment, we do not involve the *Unintentional Adversary*(obstacles) and assume that explorers can get monsters' current state if they perceive them. For each scenario, we conduct ten simulation trials for each proportion in same environment setting. Fig. 8 shows three scenarios' results with different proportion. In this experiment, *GUT* had the best performance compared with other cases. The QMIX and QMIX$_{GUT}$ have the similar *explorer average HP cost* results in Fig. 8(a), but in Fig. 8(b) *num. of explorers lost for killing a monster* and Fig. 8(c) *HP cost for killing a monster*, the QMIX has some advantage comparing with QMIX$_{GUT}$. In the winning rate comparison Table. 4, it also reflects the similar result.

Through this experiment, it does not only show that cooperation can conduce to decrease the cost or increase the benefits in MAS but also contribute to overcoming the more challenging task. More importantly, *GUT* help agents represent more complex group behaviors and strategies, such as forming various shapes, separating different groups, to adapt the dynamically changing environment in MAS cooperation, which mainly improves the system performance.

From another perspective, communication plays an important role when agents solve the conflict and have an agreement in cooperation [13]. For the QMIX$_{GUT}$ and QMIX, each explorer communication information pattern includes small data such as the number of its observing explorers and monsters corresponding to the naive behaviors attacking or defending Monster directly. However, for GUT, the communication data is more comprehensive, aiding the agents to build a complex relationship of situation and action spaces at each level, then choose a suitable tactic combination to adapt it.

Generally speaking, like a neural network, *GUT* involves the *Game-Theoretic Computation Units* distributed at each level. Through the previous level's output as an input and combining with current

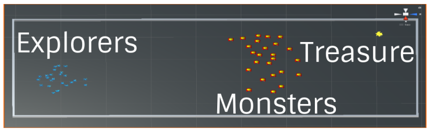


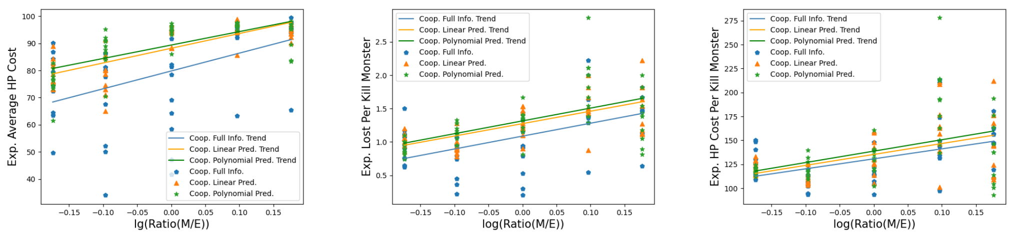
Figure 9: Only Intentional Adversary.

Winning Rate / APP / PRD	QMIX$_{GUT}$	QMIX	GUT
20e vs 30m	40%	50%	70%
25e vs 25m	90%	100%	100%
30e vs 20m	100%	100%	100%

Table 4: Winning Rate Comparison.

Ra	Com			Incom L			Incom Poly		
	WR	C$_{s_e/w}$	C$_{s_{hp}/w}$	WR	C$_{s_e/w}$	C$_{s_{hp}/w}$	WR	C$_{s_e/w}$	C$_{s_{hp}/w}$
With Intentional Adversary									
20:30	70%	1077.37	2649.80	30%	2367.14	6306.90	30%	2726.64	6216.44
20:25	90%	818.63	2027.98	50%	1414.84	3807.23	40%	1375.83	4824.64
25:25	100%	1211.09	1772.00	90%	1432.06	2606.12	80%	1789.15	2949.89
25:20	100%	1414.35	1739.78	100%	1449.07	1960.45	100%	1472.46	2177.52
30:20	100%	1608.18	2241.09	100%	2041.85	2370.76	100%	1961.86	2271.48
With Unintentional Adversary and Intentional Adversary									
25:25	100%	1443.85	2110.91	70%	2144.10	3143.63	60%	2451.37	3742.98

Table 5: System Utility Comparison. Ra: Ratio of Explorers to Monsters, WR: Winning Rate, C$_{s_e/w}$: system average energy cost winning a round, C$_{s_{hp}/w}$: system average HP cost winning a round.



(a) Explorer Average HP Cost (b) Kill Per Monster - Explorer Lost (c) Kill Per Monster - HP Cost

Figure 10: The Individual Explorer's Performance with Different Predictive Models Only Intentional Adversary.

information, at this layer, agents will calculate several game units based on corresponding to action space with some possibility. Iterating this process in the entire network, agents will decompose its strategy into several executable levels, then select the most probable tactic set to execute.

5.2 Information Prediction

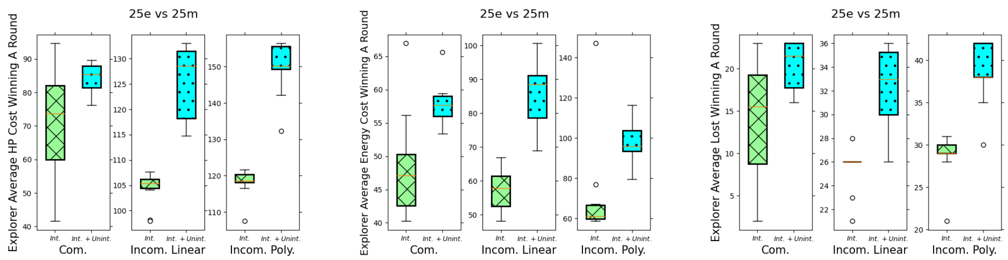
We design two kinds of scenarios to analyze the individual and system performance. One is ***Complete Information***, which means that if Explorer can perceive the adversary, it will know the opponent's status, such as unit attacking energy cost and energy level, and vice versa. The other is ***Incomplete Information***, and agents can not gain an opponent's status and information when they perceive them.

Predictive Model For incomplete information, we implement two simple *Machine Learning* prediction models *Linear Regression* and *Polynomial Regression* to estimate adversary's relative parameters as comparison.

We take regressors as individual *HP* cost to predict opponent unit attacking energy cost E_{uc} and agent's current energy E_{el}. In the *Linear Regression*, we use agent's unit *HP* cost and average system *HP* cost to predict opponent unit attacking energy cost and energy level correspondingly. The model can be presented as Formula $E_{uc} = HP_{uc} \times \beta_{uc_0} + \varepsilon$ and $E_{el} = 100 - HP_{asc} \times \beta_{asc_0} + \varepsilon$. For the *Polynomial Regression* situation, the models are shown as $E_{uc} = HP_{uc}^2 \times \beta_{uc_2} + HP_{uc} \times \beta_{uc_1} + \varepsilon$ and $E_{el} = 100 - HP_{asc}^2 \times \beta_{asc_2} - HP_{asc} \times \beta_{asc_1} + \varepsilon$. Here, β is corresponding regression coefficients($\beta_{uc_{0,1,2}}$ = {0.08, 0.03, 0.0001}, $\beta_{asc_{0,1,2}}$ = {0.03, 0.0003, 0.00001}), ε presents the error following the normal distribution $\mathcal{N}(0, 1)$.

a. Environments with only intentional adversary In this part, we consider five proportions of Explorers and Monsters (M/E) distributing in the scenario randomly. For each model, we also conduct ten trials for each rate with the same experimental setting. From an individual perspective, Fig. 10 shows that comparing with the result trend of *Complete Information* (ground truth), *Linear Regression* model has more accuracy than *Polynomial Regression* model. From system perspective (Table. 5), the winning rate and system average energy/HP cost also show a similar result.

Through these experiments, we can notice that a suitable predictive model plays a vital role in shrinking the difference between the final results and ground truth. In a more realistic scenario, agents would be in an *Incomplete Information* scenario, so they need to estimate the opponent's state from



(a) Explorer Average HP Cost/w (b) Explorer Average Energy Cost/w (c) Explorer Average Loss/w

Figure 11: The Individual Performance with Different Predictive Models Considering Unintentional Adversary.

indirect information in the adversarial environment. Besides, the predictive model's parameters also require to change according to different scenarios, which means that an agent needs to learn from its or system performance and adapt parameters corresponding to the situation.

b. Environments with unintentional adversary In this setting, we consider a more complex scenario (Fig. 1), which involves the unintentional adversary (obstacle) and monsters adopting the *QMIX* to make their individual decision. We fix the number of Explorers (E=25) and Monsters (M=25) and also conduct ten trails for each predictive model. Through the individual performance shown in Figs. 11(a) and 11(b), we can notice that the individual average HP and energy cost winning a round increases a lot when unintentional adversaries are involved. From the system perspective, Fig. 11(c) shows that the entire group will lose more agents to win a game concerning about the unintentional adversaries. Also, Table. 5 reveals that unintentional adversaries lead to a decrease in the winning rate and an increase in the system cost with the same condition to win one game. Whether this effect can be attributed to the unintentional adversary or the Monster's *QMIX* strategy needs to be investigated.

More importantly, comparing with the ground truth (complete info.), the linear regression model presents higher accuracy (winning rate, individual and system cost) than the polynomial regression model, which also adds evidence to our discussion above.

These experiments and evaluations show that the predictive model and parameter estimation are both crucial problems in *GUT*. It not only involves suitable mathematical models but also needs to consider various learning methods to improve the efficiency and accuracy of decision making concerning more complex scenarios in MAS, which is an avenue for future work.

6 Conclusions

Our work introduces a new network model called *GUT - Game-Theoretic Utility Tree* to mimic an intelligent agent decision process for achieving multi-agent cooperation in adversarial environments. We define the *Adversarial Environment* from agent's needs perspective and classify adversaries into *unintentional* and *intentional*. GUT builds a hierarchical relationship between individual behaviors and intentional adversaries and helps the entire MAS represent more complex cooperative behaviors to adapt to various scenarios. Further, to tackle the static unintentional adversaries (obstacles), we present the *Adapting The Edge* algorithm. Finally, we validate our approach through extensive simulation experiments with a *Explorers and Monsters Game* design and compare GUT against a cooperative decision-making algorithm based on the state-of-the-art QMIX approach. The results demonstrate GUT's high performance in enabling superior cooperation among multiple agents and tacking both intentional and unintentional adversaries to achieve the global objectives.

GUT leaves room for improvements. For example, *GUT* can be upgraded to include structure learning from different scenarios. We are also interested in designing appropriate utility functions, building suitable predictive models, and optimizing parameters in our future work. Besides, implementing *GUT* in real robots is also an interesting and challenging problem, which can help us to develop more robust computation models for the entire system.

7 Broader Impact

In this paper, the authors introduce *GUT*, a new network model for achieving cooperative decision-making for multi-agent and multi-robot systems in adversarial environments.

GUT could be applied to a wide range of applications, including AI, defense, robotics, wireless network, network security, differential privacy, computer vision and so forth. From MAS perspective, the agents have to exhibit an awareness of the environment not only at an individual agent level but also at a system level, where computational game theory provide useful examples of the study in the area of machine behaviour [16, 39]. Our research could be used to build a *Game-Theoretic Utility Neural Networks*, which establish a hierarchical relationship between individual behaviours and environmental information and help the entire system representing more complex cooperative behaviours to adapt to various scenarios.

While there will be important impacts resulting from the use of *GUT* in general, here we focus on the impact of using our method to provide solutions for adversarial environments. There are many benefits to using such a approach, such as increasing the integration in decision-planning applications. This can help enhance cooperation among MAS to successfully complete the assigned tasks with lower costs and higher winning probabilities against adversaries. The potential risks could include: (i) inefficient network structure in the use of *GUT* could cause duplicate computations and increase computation complexity which may lead to issues in time-critical applications; (ii) unsuitable predictive models and parameters could increase the risk of uncertainty and error for its outputs; (iii) in Multi-Robot System (MRS), the risk of network security will also play an important role.

We see opportunities for research applying *GUT* to beneficial purposes, such as using various learning methods to optimize structure, integrating with the self-supervised reinforcement learning, improving algorithmic efficiency and prediction accuracy. To mitigate the risks associated with network security, we encourage research to understand the impacts of using *GUT* in particular real-world scenarios.

References

[1] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *From natural to artificial swarm intelligence*. Oxford University Press, Inc., 1999.

[2] Christian Blum and Daniel Merkle. Swarm intelligence. *Swarm Intelligence in Optimization; Blum, C., Merkle, D., Eds*, pages 43–85, 2008.

[3] Jeff S Shamma. *Cooperative control of distributed multi-agent systems*. Wiley Online Library, 2007.

[4] Marco Dorigo, Mauro Birattari, and Manuele Brambilla. Swarm robotics. *Scholarpedia*, 9(1):1463, 2014.

[5] Jiaying Shen, Xiaoqin Zhang, and Victor Lesser. Degree of local cooperation and its implication on global utility. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 546–553. IEEE Computer Society, 2004.

[6] Roxana Rădulescu, Patrick Mannion, Diederik M Roijers, and Ann Nowé. Multi-objective multi-agent decision making: a utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems*, 34(1):10, 2020.

[7] Timothy H Chung, Geoffrey A Hollinger, and Volkan Isler. Search and pursuit-evasion in mobile robotics. *Autonomous robots*, 31(4):299, 2011.

[8] Myungsoo Jun and Raffaello D'Andrea. Path planning for unmanned aerial vehicles in uncertain and adversarial environments. In *Cooperative control: models, applications and algorithms*, pages 95–110. Springer, 2003.

[9] Noa Agmon, Sarit Kraus, and Gal A Kaminka. Uncertainties in adversarial patrol. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 1267–1268. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[10] Noa Agmon, Gal A Kaminka, and Sarit Kraus. Multi-robot adversarial patrolling: facing a full-knowledge opponent. *Journal of Artificial Intelligence Research*, 42:887–916, 2011.

[11] Roi Yehoshua and Noa Agmon. Adversarial modeling in the robotic coverage problem. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 891–899. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[12] Navyata Sanghvi, Sasanka Nagavalli, and Katia Sycara. Exploiting robotic swarm characteristics for adversarial subversion in coverage tasks. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 511–519. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[13] Qin Yang, Zhiwei Luo, Wenzhan Song, and Ramviyas Parasuraman. Self-Reactive Planning of Multi-Robots with Dynamic Task Assignments. In *IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS) 2019*, 2019. Extended Abstract.

[14] Mouna Jouini, Latifa Ben Arfa Rabai, and Anis Ben Aissa. Classification of security threats in information systems. *Procedia Computer Science*, 32:489–496, 2014.

[15] Yaniv Shapira and Noa Agmon. Path planning for optimizing survivability of multi-robot formation in adversarial environments. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4544–4549. IEEE, 2015.

[16] Yara Rizk, Mariette Awad, and Edward W Tunstel. Decision making in multiagent systems: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, 10(3):514–529, 2018.

[17] Efrat Sless Lin, Noa Agmon, and Sarit Kraus. Multi-robot adversarial patrolling: Handling sequential attacks. *Artificial Intelligence*, 274:1–25, 2019.

[18] Hehui Zheng, Jacopo Panerati, Giovanni Beltrame, and Amanda Prorok. An adversarial approach to private flocking in mobile robot teams. *arXiv preprint arXiv:1909.10387*, 2019.

[19] James Paulos, Steven W Chen, Daigo Shishika, and Vijay Kumar. Decentralization of multiagent policies by learning what to communicate. *arXiv preprint arXiv:1901.08490*, 2019.

[20] Rene Vidal, Shahid Rashid, Cory Sharp, Omid Shakernia, Jin Kim, and Shankar Sastry. Pursuit-evasion games with unmanned ground and aerial vehicles. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 3, pages 2948–2955. IEEE, 2001.

[21] Andreas Kolling and Stefano Carpin. Multi-robot pursuit-evasion without maps. In *2010 IEEE International Conference on Robotics and Automation*, pages 3045–3051. IEEE, 2010.

[22] Peng Cheng. A short survey on pursuit-evasion games. *Department of Computer Science, University of Illinois at Urbana-Champaign*, 2003.

[23] William Lewis Scott III. *Optimal evasive strategies for groups of interacting agents with motion constraints*. PhD thesis, PhD thesis, Princeton University, 2017.

[24] Venkata Ramana Makkapati and Panagiotis Tsiotras. Optimal evading strategies and task allocation in multi-player pursuit–evasion problems. *Dynamic Games and Applications*, pages 1–20, 2019.

[25] Jung Soon Jang and Claire Tomlin. Control strategies in multi-player pursuit and evasion game. In *AIAA guidance, navigation, and control conference and exhibit*, page 6239, 2005.

[26] Shashwat Shivam, Aris Kanellopoulos, Kyriakos G Vamvoudakis, and Yorai Wardi. A predictive deep learning approach to output regulation: The case of collaborative pursuit evasion. *arXiv preprint arXiv:1909.00893*, 2019.

[27] John F Nash et al. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

[28] Roger B Myerson. *Game theory*. Harvard university press, 2013.

[29] Peter C Fishburn. Utility theory for decision making. Technical report, Research analysis corp McLean VA, 1970.

[30] Mykel J Kochenderfer. *Decision making under uncertainty: theory and application*. MIT press, 2015.

[31] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[32] Michael I Jordan. An introduction to probabilistic graphical models, 2003.

[33] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: monotonic value function factorisation for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1803.11485*, 2018.

[34] John Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[35] Albert Xin Jiang and Kevin Leyton-Brown. A tutorial on the proof of the existence of nash equilibria. *University of British Columbia Technical Report TR-2007-25. pdf*, 14, 2009.

[36] Unity Game Engine. Unity game engine-official site. *Online][Cited: October 9, 2008.] http://unity3d. com*, pages 1534–4320, 2008.

[37] Richard D McKelvey, Andrew M McLennan, and Theodore L Turocy. *Gambit: Software tools for game theory*. Version 0.2006. 01.20, 2006.

[38] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. In *Advances in Neural Information Processing Systems*, pages 7611–7622, 2019.

[39] Iyad Rahwan, Manuel Cebrian, Nick Obradovich, Josh Bongard, Jean-François Bonnefon, Cynthia Breazeal, Jacob W Crandall, Nicholas A Christakis, Iain D Couzin, Matthew O Jackson, et al. Machine behaviour. *Nature*, 568(7753):477, 2019.