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

Due to size limitations, we have made available the supplementary video attachment to this paper, demonstrating the experiments in the anonymized link at https://streamable.com/pngbe8.

A Related Work

Research in Multi-Agent Systems (MAS) in adversarial environments can be classified into four main categories (task-wise): Adversarial Patrol [1]; Adversarial Coverage [2, 3]; Adversarial Formation [4]; and Adversarial Navigation [5, 6, 7]. In most adversarial MAS literature, the adversaries are not artificial intelligent agents [2, 3]. They might be a natural force like wind, fire, rain, or other creatures' aggressive behaviors. Accordingly, the solutions to these problems are path planning avoiding static or dynamical obstacles, formation control avoiding collisions, and so forth [4, 2, 3]. This is particularly applicable to urban search and rescue missions and robots deployed in disaster environments, where the agents are more concerned about unintentional threats such as radiation, clutters, or natural forces such as wind, fire, rain, etc. [8].

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 [9] examined the problem of defending patrol tasks against a sequential attack in a knowledgeable adversarial environment. Prorok [10] studied multi-robot privacy with an adversarial approach. Sanghvi and Sycara [11] identified a swarm vulnerability and studied how an adversary takes advantage of it. Paulos and Kumar [12] describe an architecture for training teams of identical agents in a defense game. As we can see, most of these researches focus on privacy or cyber-adversaries rather than physical adversaries confronting the agents in a Spatio-temporal domain.

Cooperative decision-making among the agents is essential to address the threats posed by intentional physical adversaries. Current researches mainly focus on solving multi-player *pursuit and evasion* game problem [13, 14, 15], which primarily deals with how to guide one or a group of pursuers to catch one or a group of moving evaders [16, 17]. Recent works in this domain concentrate on optimal evasion strategies and task allocation [18, 19] and predictive learning from agents' behaviors [20].

From MAS perspective, existing cooperative decision making models including Markov decision process (MDP) and its variants [21], game-theoretic methods, and swarm intelligence [8]. They mostly involve using Reinforcement Learning (RL) and Recurrent Neural Networks (RNN) to find optimal or sub-optimal action sequences based on current and previous states for achieving independent or transferred learning of decision-making policies [22, 23].

Specifically, the QMIX [22] is a state-of-the-art deep multi-agent RL method that uses a multi-network structure (consisting of a mixing network at the system level) for Q-learning of action policies. They demonstrate that full factorization of value decomposition networks (VDN) is not necessary to be able to extract decentralized policies that are entirely consistent with their centralized counterpart. However, these approaches mainly concern about partial cooperation[1] and do not consider deeper cooperation among all the agents in the MAS to build complex cooperation strategies among agents.

To address these gaps, we propose a new network model called *GUT* for cooperative decision making in MAS. To analyze our approach, we also build a more realistic testing domain - *Explorers and Monsters Game*, which is to achieve a given task with minimum cost while simultaneously considering the adversaries in the environment. Future extension of this game domain includes multiple targets and multi-group Explorers and Monsters.

Like the neural network, *GUT* provides corresponding *Game-Theoretic Computation Units* distributed in each level, which combines agents' tactics in the current situation, the possibility of the previous situation, and relative environment's information. Furthermore, in this paper, we only compare with the decision making part of *QMIX* since our current approach mainly relates to the cooperative decision-making model in MAS. In future work, we will consider augmenting RL, RNN, or various learning methods on top of *GUT* to be able to fully compare with other multi-agent RL methods, including *QMIX* [22] and its superior variant MAVEN [23].

[1] See Sec. E for clarity on the definitions of full and partial cooperation among agents in MAS.

B Game-Theoretic Computation Unit

Fig. 1 describes the structure of a specific *Game-Theoretic Computation Unit* in *GUT*. (a) is the game-theoretic module. Through calculating the *Nash Equilibrium* based on the utility values $(u_{11}, ..., u_{nm})$ of corresponding tactics combination in this module, we can get the probability $(p_1, ..., p_{nm})$ about every combination. Then, according to conditional probability module (b), we can describe the probability of every current possible situation as $(p_{i1}, ..., p_{inm})$. After that, it will move on to next units correspondingly.

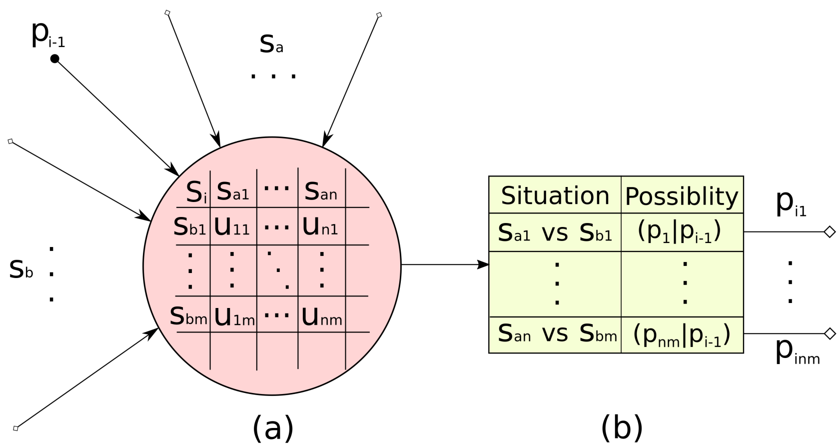

Figure 1: Game-Theoretic Computation Unit

Here,

- p_{i-1} presents the previous situation probability;
- S_i presents current game-theoretic state;
- s_a presents the tactics space $(s_{a1}, ..., s_{an})$ of agent or group a;
- s_b presents the tactics space $(s_{b1}, ..., s_{bm})$ of agent or group b;
- $p_{inm} = (p_{nm}|p_{i-1}), \quad i, n, m \in Z^+$.

C Robot's Needs Hierarchy

In order to quantify the needs [24] of cooperative agents, from statistical perspective, we first formalize the Robot's *Safety, Basic, Capability and Teaming* Needs (Eq. (1). (2). (3). (4).) based on corresponding expectation.

$$Safety\ \ Needs: \quad N_{s_i} = \sum_{i=1}^{s_i} W_i \cdot \mathbb{P}(W_i|P, C); \tag{1}$$

$$Basic\ \ Needs: \quad N_{b_i} = \sum_{i=1}^{b_i} W_i \cdot \mathbb{P}(W_i|P, C, N_{s_i}); \tag{2}$$

$$Capability\ \ \ Needs: \quad N_{c_i} = \sum_{i=1}^{g_i} A_i \cdot \mathbb{P}(A_i|T, P, N_{b_i}); \tag{3}$$

$$Teaming\ \ \ Needs: \quad N_t = \sum_{i=1}^{n} U_i \cdot \mathbb{P}(U_i|P, C, N_{c_i}); \tag{4}$$

Here,

- P and C represent the data of agent's perception and communication separately;

- T represents the task requirement solution space;
- U_i represents the Utility value of agent i in the group;
- W represents corresponding weights;
- A represents the level of agent's each capability;
- b_i represents the size of agent's i's basic needs solution space;
- s_i represents the size of agent's i's safety needs solution space;
- g_i represents the size of agent's i's capability needs solution space;
- n represents the number of agents in the group (explorers or monsters).

In the first level, individual safety needs (Eq. (1)) can express as the *safety expectation*, which can be calculated through its behaviours' weight and corresponding safety possibility based on the information of perception and communication. If individual safety needs goes beyond to a certain level, it will calculate the basic needs (Eq. (2)) which also can be presented as *basic expectation*. After fitting the basic needs, agent similarly can compute its capability needs (Eq. (3)) through the *capability expectation* according to the task requirement and perception data.

Through above analysis, we combine *Utility Theory* to define the intelligent agent's fourth level needs – *Teaming* (Eq. (4)). These represent high level needs for intelligent agent and can be regarded as a kind of motivations or requirements to cooperate with each other achieving some goals or tasks in order to satisfy the individual or group's certain *expectation utilities*. Furthermore, for the highest (fifth) level needs – *Learning*, which can help the individual agent to achieve self-supervised learning based on the learned experiences and lead to self-evolution of the whole system.

D Analysis and Proofs

D.1 GUT 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}}, $i \in n$. 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. For *n*-level GUT, supposing game G_i in level *i*, the size of action space of group A (*z* agents) and B are l_i and m_i correspondingly. For the intentional decision, the *zero-sum* game G_i can be described as Eq. (5):

$$G_i = \{A, B; N_{ti}\}, \quad i \in n; \tag{5}$$

Based on the teaming needs (Eq. (4)), group A's each GUT level N_{ti} can be presented as Eq. (6).

$$N_{ti} = \sum_{i=1}^{z} \mathbb{E}_i(U) = (a_{gk})_{l_i \times m_i}, \quad g \in l_i, k \in m_i, j \in z; \tag{6}$$

According to *Nash's Existence Theorem* [25, 26], it guarantees the existence of a set of mixed strategies (where a pure strategy is chosen at random, subject to some fixed probability) for finite, non-cooperative games of two or more players in which no player can improve his payoff by unilaterally changing strategy. So in every finite games has a *Pure Strategy Nash Equilibrium* or a *Mixed Strategy Nash Equilibrium*, and the process can be formalized as two steps:

a. Compute Pure Strategy Nash Equilibrium

We can present agents' utility matrix as Eq. (7):

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m_i} \\ a_{21} & a_{22} & \cdots & a_{2m_i} \\ \vdots & \vdots & \ddots & \vdots \\ a_{l_i1} & a_{l_i2} & \cdots & a_{l_im_i} \end{bmatrix} \tag{7}$$

The row and column correspond to the utility of agent A and B separately. Through calculating the minimum value list of each row and maximum value list of each column, we can compute the maximum and minimum values of the two lists separately.

$$\min_{1 \le k \le m_i} \max_{1 \le g \le l_i} a_{gk} = \max_{1 \le g \le l_i} \min_{1 \le k \le m_i} a_{gk} \tag{8}$$

If the two value satisfy the Eq. (8), we can get the level *i* Pure Strategy Nash Equilibrium Eq. (9), and corresponding game value Eq. (10).

$$PSNE = (A_{g^*}, B_{k^*}); \tag{9}$$

$$V_{G_i} = a_{g^* k^*} \tag{10}$$

b. Compute Mixed Strategy Nash Equilibrium

The tactics' probability of agent A present as Eq. (11).

$$AX = (x_1, x_2, \ldots, x_{l_i}); \quad x_g \ge 0,\ g = 1, 2, \ldots, l_i; \ \sum_{g=1}^{l_i} x_g = 1 \tag{11}$$

Similarly, we also can conclude agent B's tactics' probability as Eq. (12).

$$BY = (y_1, y_2, \ldots, y_{m_i}); \quad y_k \ge 0,\ k = 1, 2, \ldots, m_i; \ \sum_{k=1}^{m_i} y_k = 1 \tag{12}$$

So we can define *(X, Y)* as *Mixed Situation* in certain status. Then, we can deduce the expected utility of agent A and B Eq. (13) and (14) separately.

$$\mathbb{E}_A(X, Y) = \sum_{g=1}^{l_i} \sum_{k=1}^{m_i} a_{gk} x_g y_k = \mathbb{E}(X, Y); \tag{13}$$

$$\mathbb{E}_B(X, Y) = -\,\mathbb{E}(X, Y) \tag{14}$$

In the Game $G_i = \{A, B; N_{ti}\}$, if we can get all the *Mixed Tactics* of agent A and B as Eq. (15) and (16), we can deduct the G's *Mixed Expansion* as Eq. (17). Then, if we can compute a tactic (X^*, Y^*) satisfying Eq. (18) and (19), we define this tactic is the optimal strategy in current state and the game result is Eq. (20).

$$S_A^* = AX; \tag{15}$$

$$S_B^* = BX; \tag{16}$$

$$G_i^* = \{S_A^*, S_B^*; \mathbb{E}\}; \tag{17}$$

$$\mathbb{E}(X^*, Y) \ge V_{s_A}, \forall\, Y \in S_A^*; \tag{18}$$

$$\mathbb{E}(X, Y^*) \le V_{s_B}, \forall\, X \in S_B^*; \tag{19}$$

$$V_{S_A} = V_{G_i} = V_{S_B} \tag{20}$$

As above discussion, we can express the GUT computation process as corresponding Probabilistic Graphical Models [27] – *Bayesian Network*. Supposing each node is independent, so the **total number of nodes in *GUT* is N** (Eq. (21)), and the current *joint probability distribution* of group A in GUT can be represented as Eq. (22).

$$\begin{aligned} N &= N_1 + N_2 + \ldots + N_n \\ &= 1 + l_1 \times m_1 + (l_1 \times l_2) \times (m_1 \times m_2) + \ldots \\ &+ (l_1 \times l_1 \times \ldots \times l_{n-1}) \times (m_1 \times m_1 \times \ldots \times m_{n-1}) \end{aligned} \tag{21}$$

$$\begin{aligned} \mathbb{P}(X) &= \mathbb{P}(X_1, X_1, \ldots, X_N) \\ &= \mathbb{P}(X_1)\mathbb{P}(X_2|X_1) \ldots \mathbb{P}(X_N|X_1, X_2, \ldots, X_{N-1}) \\ &= \mathbb{P}(X_1)P(X_2|X_1) \ldots \mathbb{P}(X_N|X_{N-(l_1 \times l_2 \times \ldots \times l_{n-1}) \times (m_1 \times m_2 \times \ldots \times m_{n-1})}), \\ &= \prod_i \mathbb{P}_i(X_i|Par_G(X_i)), i \in N \end{aligned} \tag{22}$$

Since *Nash's Existence Theorem* guarantees that every game has at least one Nash equilibrium [28], we get Eq. (23).

$$\mathbb{P}_i(X_i) \neq 0 \implies \prod_i \mathbb{P}_i(X_i|Par_G(X_i)) = \mathbb{P}(X) \neq 0 \tag{23}$$

Low Bound: If the each level Nash Equilibrium calculation in GUT is the Pure Strategy Nash Equilibrium, individual agent can obtain an unique tactic entering into the next level, which means this tactic's possibility is one hundred percent Eq. (24). We also can get corresponding dominant strategy series (s_1, s_2, ... , s_n) in GUT.

$$\mathbb{P}_i(X_i) = 1 \implies \prod_i \mathbb{P}_i(X_i|Par_G(X_i)) = \mathbb{P}(X) \equiv 1 \tag{24}$$

□

D.2 GUT Maximum A Posterior (MAP)

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

$$x^* = \arg\max_{x \in Val(x)} \mathbb{P}(x) \tag{25}$$

Proof. We can simplify an n-level *GUT* as one link *Bayesian Network* Fig. 2, and get the factors of product Eq. (26) (ϕ_{x_i} are the intermediate factors). And the maximum joint probability of *GUT* Decision is equal to get its maximum factors of product. Then through VE (Variable Elimination) [27], we can get the MAP assignment of this *GUT*. The entire process has two steps: 1) Variable elimination Eq. (27); 2) Tracing back to get a joint assignment (x_1^*, x_2^*, ... , x_n^*) Eq. (28). Finally, we can get MAP results of the *GUT* Eq. (25).

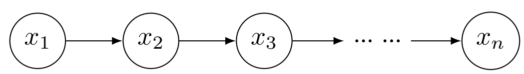

Figure 2: n-level GUT as one link Bayesian Network.

$$\mathbb{P}(x) = \prod_i \mathbb{P}_i(x_i|Par_G(x_i)) = \prod_{x_i} \phi_{x_i} \implies \max \mathbb{P}(x) = \max \prod_{x_i} \phi_{x_i} \tag{26}$$

$$\begin{aligned}
&\textbf{Variable Elimination}: \\
&\textbf{if } X \in Scope[\phi_{x_i}] \textbf{ then } \max_X(\phi_{x_i}, \phi_{x_{i+1}}) = \phi_{x_i} \max_X(\phi_{x_{i+1}}) \implies \\
&\quad \textit{first elimination}: \\
&\quad \max \mathbb{P}(x) = \max_{x_2, x_3, \ldots, x_n} \phi_{x_2}\phi_{x_3}\ldots\phi_{x_n} \max_{x_1} \phi_{x_1} \\
&\quad \textit{second elimination}: \\
&\quad \max \mathbb{P}(x) = \max_{x_3, x_4, \ldots, x_n} \phi_{x_3}\phi_{x_4}\ldots\phi_{x_n} \max_{x_2} \phi_{x_2}\tau_1 \\
&\quad \ldots \ldots \\
&\quad \textit{(n)th elimination}: \\
&\quad \max \mathbb{P}(x) = \max_{x_n} \tau_{n-1}
\end{aligned} \tag{27}$$

$$\begin{aligned}
&\textbf{Tracing Back}: \\
&\quad x_n^* = \arg\max_{x_n} \psi_n(x_n) \\
&\quad x_{n-1}^* = \arg\max_{x_{n-1}} \psi_n(x_n, x_{n-1}) \\
&\quad \ldots \ldots \\
&\quad x_1^* = \arg\max_{x_1} \psi_1(x_n, x_{n-1}, \ldots, x_1)
\end{aligned} \tag{28}$$

$$
\begin{aligned}
&\textbf{Maximum A Posterior}: \\
&x^* = (x_1^*, x_2^*, ..., x_n^*) \textit{ is the MAP assignment}, \\
&\tau_{n-1} \textit{ is the probability of the most probable assignment.}
\end{aligned}
\tag{29}
$$

□

D.3 The Needs in Explorers and Monsters Game

According to above discussion, we assume that Explorers and Monsters have the same speed to move and Monster can not communicate with each other. See Sec. D.3.4 for the list of terms and notations used in this subsection.

D.3.1 Winning Utility Expectation

In the first level, we consider using *Winning Probability* obeying *Bernoulli Distribution* calculates the need based on its utility expectation Eq. (30).

$$
W(t_{ev}, t_{mv}, r_{ev}, r_{mv}, n, m) = (a_1 \frac{a_2 t_{ev} + a_3 r_{ev}}{a_4 t_{mv} + a_5 r_{mv}})^{\frac{m}{n}}; \tag{30}
$$

D.3.2 Energy Utility Expectation

The second Level's utility is described the relative expected energy cost as Eq. (31), (32), (33) and (32). And we consider three parts of energy cost in the whole process: $walking$, $attacking$ and $communication$.

$$
\begin{aligned}
&E(d, v, f, q, n, m, \phi_e, \phi_m) = \\
&b_0 + b_1 \int_{-\infty}^{+\infty} (n - m) e_d(x) p_d(x, d) \mathrm{d}x + b_2 (\sum_{i=1}^{+\infty} n e_{a_e}(i, f) p_{a_m}(j, m\phi_m) \\
&- \sum_{j=1}^{+\infty} m e_{a_m}(j, q) p_{a_e}(i, n\phi_e)) + b_3 \sum_{w=1}^{+\infty} n e_c(w) p_c(w, \frac{d}{v});
\end{aligned}
\tag{31}
$$

$$
e_d(x) = b_{11} x; \tag{32}
$$

$$
e_a(x, y) = b_{12} x y; \tag{33}
$$

$$
e_c(x) = b_{13} x \tag{34}
$$

In the entire $attack - defend$ process, the agent's action distance, the times of attacks and being attacked and the communication times obey $Normal\ Distribution$, $Poisson\ Distribution$ separately. So we can describe their distribution function as Eq. (35), (36), (37) and (38).

$$
p_d(x, d) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-d)^2}{2}}; \tag{35}
$$

$$
p_{a_e}(x, \lambda_e) = \frac{e^{-\lambda_e} \lambda_e^x}{x!}; \tag{36}
$$

$$
p_{a_m}(x, \lambda_m) = \frac{e^{-\lambda_m} \lambda_m^x}{x!}; \tag{37}
$$

$$
p_{a_m}(x, \frac{d}{v}) = \frac{e^{-\frac{d}{v}} (\frac{d}{v})^x}{x!}; \tag{38}
$$

Then we can simplify the Eq. (31) as Eq. (39).

$$
E(d, v, f, q, n, m, \phi_e, \phi_m) = b_0 + b_1 b_{11} (n - m) d + b_2 b_{12} n m (f \phi_m - q \phi_e) + b_3 b_{13} n \frac{d}{v}; \tag{39}
$$

D.3.3 HP Utility Expectation

In the lowest level, we use the expected HP cost to explain the utility as Eq. (40), (41), (42) and (43).

$$H(k, t_e, t_m, r_e, r_m, g, \phi_e, \phi_m) = \\ c_0 + c_1(\sum_{i=1}^{+\infty} kh(t_e, r_e, i)p_{h_m}(i, \phi_m) - \sum_{j=1}^{+\infty} gh(t_m, r_m, j)p_{h_e}(j, \phi_e)); \tag{40}$$

$$h(x, y, z) = \rho z(x + y) \tag{41}$$

$$t_{e,m}(e_{e,m}) = \gamma_{e,m} e_{e,m}; \tag{42}$$

$$r_{e,m}(e_{e,m}) = \delta_{e,m} e_{e,m}; \tag{43}$$

$p_{h_e}(i, \phi_e)$ and $p_{h_m}(j, \phi_m)$ are similar to the Eq. (36) and (37) correspondingly. Through simplifying the Eq. (40), we can get Eq. (44).

$$H(k, t_e, t_m, r_e, r_m, g, \phi_e, \phi_m) = c_0 + c_1\rho[k\phi_m e_e(\gamma_e + \delta_e) - g\phi_e e_m(\gamma_m + \delta_m)]; \tag{44}$$

D.3.4 Terms used in Sec. D.3

- AT and BT present the action space of group A and B correspondingly;
- n and m present the number of Explorers and Monsters separately;
- d presents the group average distance between two opponents;
- v presents the agent's velocity;
- i and j present the times of attacks and being attacked;
- w presents Explorers' communication times;
- f and q present the unit attacking energy cost of both sides agents separately;
- t_{ev} and t_{mv} present average attacking ability levels of both sides separately;
- r_{ev} and r_{mv} present average defending ability levels of both sides separately;
- t_e and t_m present specific agent's attacking ability levels of both sides separately;
- r_e and r_m present specific agent's defending ability levels of both sides separately;
- ϕ_e and ϕ_m present individual agent's size;
- k presents the number of Explorers' attacking simultaneously;
- g presents the number of Monsters' attacking simultaneously;
- a, b, c, ρ, γ and δ present corresponding coefficient;
- e_e and e_m present the current energy level of Explorer and Monster;
- h presents the current HP level of agent;
- p presents the probability corresponding to the different section.

E Relative Definitions and Experimental Setting

In our experiments, we assume that the individual agent's communication range is larger than the sensing range. Every agent can always connect to at least one agent when it communicates (with an assumption of connected communication graph).

Self-interest (Individual Rationality) Individual focuses on the needs and desires (interests) of one's self. From philosophical, psychological, and economic perspectives, it is the motivation for intelligent agents to maximize their benefits from an individual's perspective, which is also *Individual Rationality*.

Group-interest (Collective Rationality) Comparing with *Self-interest*, individual pays more attention about group members' needs, on the other hand, it is *Group-interest*. Through communication and negotiation, they finally get a consensus and agreement fulfilling a solution to maximizing the entire group's needs and rewards, which can be regarded as *Collective Rationality*.

Partial Communication An individual agent only communicates and shares information with agents that are in its sensing range.

Full Communication In this status, agents always keep in touch with each other even when they are not in the sensing range. So, the communication graph can be represented as a completely connected graph, which means that every agent can directly share its information with every other agent.

Noncooperation Individual does not communicate with each other; it makes a decision only depending on its perception and needs. In this situation, the agent only concerns its own benefits representing *Self-interest*.

Partial Cooperation Based on the *Partial Communication*'s information, individuals only cooperate with sensing group members to maximize their needs or minimize the cost.

Full Cooperation According to the *Full Communication*'s data, individual make decision based on the *Group-interest* showing *Collective Rationality*.

E.1 Experiment Setting

Each interaction (trial) in the *Explorers and Monsters Game* last about 40 to 50 minutes in the simulated experiments on a laptop with Intel i7 Processor, GeForce GTX 1050 Ti GPU, and 16GB DDR4 RAM running the OS Ubuntu 18.04.

It is worth noting that the Monsters attacking capability is 3x the Explorers attacking capacity in all the experiments to represent higher capabilities of Monsters (who act based on their self-interests) in preventing the Explorers from their tasks. A video demonstration of the experiments showing sample trials of Explorers and Monsters Game using GUT and QMIX is available at the anonymized link https://streamable.com/pngbe8.

References

[1] Noa Agmon, Sarit Kraus, Gal A Kaminka, and Vladimir Sadov. Adversarial uncertainty in multi-robot patrol. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[2] 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.

[3] 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.

[4] 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.

[5] Ofri Keidar and Noa Agmon. Safety first: Strategic navigation in adversarial environments. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 1581–1583. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[6] Noa Agmon, Yehuda Elmaliah, Yaron Mor, and Oren Slor. Robot navigation with weak sensors. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 272–276. Springer, 2011.

[7] Ofri Keidar and Noa Agmon. Safe navigation in adversarial environments. *Annals of Mathematics and Artificial Intelligence*, 83(2):121–164, 2018.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

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

[17] 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.

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

[19] 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.

[20] 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.

[21] Lantao Yu, Jiaming Song, and Stefano Ermon. Multi-agent adversarial inverse reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7194–7201, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[22] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4295–4304, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[23] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7613–7624. Curran Associates, Inc., 2019.

[24] 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.

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

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

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

[28] 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.