APE: An Anti-poaching Multi-Agent Reinforcement Learning Benchmark

Prasanna Maddila, Eric Casellas, Patrick Chabrier,
Régis Sabbadin and Meritxell Vinyals
Laboratoire MIAT, University of Toulouse, INRAE
Toulouse, France
first_name.surname@inrae.fr

Abstract

Widespread poaching threatens many endangered species today, requiring robust strategies to coordinate ranger patrols and effectively deter poachers within protected areas. Recent research has modelled this problem as a strategic game between rangers and poachers, resulting in anti-poaching becoming a popular application domain within game theory and multi-agent research communities. Unfortunately, the lack of a standard open-source implementation of the anti-poaching game hinders the reproducibility and advancement of current research in the field. This paper aims to fill this gap by providing the first open-source standardised environment for the anti-poaching game. Our contributions are as follows: (1) we formalise anti-poaching as a Partially Observable Stochastic Game; (2) we provide the Anti-Poaching Environment (APE), an open-source Python implementation of a simulator for this game using the PettingZoo API, which is compatible with many existing multi-agent reinforcement learning (MARL) libraries; and (3) we illustrate how to apply deep reinforcement-learning algorithms from the RLlib library, in order to compute cooperative and cooperative-competitive equilibria of APE instances. Our project is published at `https://forgemia.inra.fr/chip-gt/antipoaching`.

1 Introduction

In today's world, endangered species are threatened by widespread poaching, requiring intelligent land patrol strategies, the so-called *anti-poaching* strategies, to effectively detect and prevent such activities [8]. In this paper we refer to *anti-poaching* strategies as the problem[1] of deciding where and when to send ranger squads within a protected area to prevent poaching. Several recent works [19, 20, 22, 25] have applied game theory and multi-agent reinforcement learning (MARL) to model and learn such *anti-poaching* strategies. Unfortunately, despite anti-poaching being a popular application for game theory and MARL, the lack of any standardized open-source implementation of the anti-poaching game itself hinders research efforts, ultimately slowing the development of practical anti-poaching solutions. Also, anti-poaching research faces a unique challenge in comparing with existing work due to strict security and confidentiality constraints. Organizations cannot share patrol data publicly, and research granted privileged access is often prohibited from disclosing it. This raises a critical question: how can we establish a public research benchmark that remains

[1]Not to be confused with the patrol allocation problem, which instead involves determining the *starting* patrolling points for the patrols.

17th European Workshop on Reinforcement Learning (EWRL 2024).

realistic under these confidentiality restrictions? In this work, we tackle this challenge by proposing a public benchmark defined by a range of configurable parameters representing the anti-poaching model. Importantly, the information needed to set these model parameters in a realistic range is far less sensitive than requiring access to raw patrol data.

In the reinforcement learning (RL) community, the development of publicly available benchmarks using standard APIs, namely Gymnasium [18] for the single-agent case (formerly Gym [2]) and PettingZoo [16] for the multi-agent case, is known to have significantly contributed to advance algorithmic developments in the field. The importance of having standarized MARL benchmarks has been stated in several recent works [7, 13] that have proposed similar frameworks yet for other domains, such as predator-prey and target coverage control. The aim of this work is to extend the benefits of such standarised benchmarking to anti-poaching. The main contributions of this paper are the following:

- A formalisation of *anti-poaching* as a partially observed stochastic game (POSG) involving a team of cooperative rangers and several competitive independent poachers.
- A publicly available implementation of the *anti-poaching* game in python as a PettingZoo environment: the *Anti-Poaching Environment (APE).*
- A comprehensive example of using RL algorithms from the RLlib library to compute *cooperative* and *cooperative-competitive* equilibria for instances of the APE game.

This paper is structured as follows. Section 2 reviews related game models and MARL benchmarks. Section 3 formalizes Anti-Poaching as a POSG. Section 4 details the implementation of this game as a Pettingzoo compatible MARL environment and presents a comprehensive example of use with the RLlib library. Section 5 showcases results from applying RLlib RL algorithms in cooperative and cooperative-competitive APE scenarios. Section 6 summarizes the key contributions, current limitations, and potential directions for future research.

2 Related Work

Game-theoretic approaches to anti-poaching Effective anti-poaching strategies rely on a clear understanding of their impact on poaching activity [8]. However, simulation and optimization frameworks from AI can significantly aid the design of such innovative anti-poaching strategies. Some works [14, 1] have explored anti-poaching through patrol *allocation*, often modelled as a Stackelberg Security Game (SSG) where a ranger team (the leader) announces a (potentially stochastic) patrol allocation on a set of targets to deter poachers. In contrast, our work focuses on defining patrolling strategies, where rangers and poachers act simultaneously with no prior knowledge of each other's strategies. Some prior research proposed simpler models for patrolling considering a single defender, prioritizing computational efficiency by using multi-armed bandits [23], two-player zero-sum games [24], and single-team planning [3]. Our work aligns with research that leverages more complex games, namely Partially Observable Stochastic Games (POSGs) or Extensive-Form Games (EFGs), to represent real-world scenarios. [20] introduced a similar[2] zero-sum EFG model, albeit limited to a single defender and a single attacker. Closer to our approach, [19] formulated a zero-sum POSG[3] with multiple rangers and poachers. Both [20] and [19] highlighted the limitations of traditional approaches to tackle such large, complex models and used (deep) multi-agent reinforcement learning (MARL) to learn the patrolling equilibrium strategies. Nevertheless, while such works highlighted the potential of anti-poaching games as a MARL benchmark, their contribution was primarily algorithmic and lacked readily available environments or open-source code. Our work bridges this gap by providing the first standarised MARL benchmark for antipoaching.

Game-theoretic models in other domains. The Anti-Poaching game, as formalized in Section 3, is related to *Adversarial Team Markov Games* (ATMG) [6], which feature a team

[2]Despite the number of agents, there are some other differences regarding the methods of prey detection, trap removal processes and the consideration of footprints.

[3]There are some differences with our model regarding trap removal, the reward function, and including drones with signaling capabilities.

of cooperative agents competing against a single adversarial agent within a fully-observed zero-sum stochastic game framework. Specifically, the Anti-Poaching game can be seen as an instance of a generalisation of ATMG, with: (i) *multiple* non-cooperative adversaries, and (ii) a *partially observed* environment.

Predator-prey games have also been extensively studied in game theory yet they exhibit very different dynamics compared to APE. In predator-prey games, prey are directly captured, often requiring the coordinated actions of multiple predators, leading to implicit collaboration even if they are not part of a team. In contrast, in APE, prey are captured indirectly through traps set by poachers and each ranger squad can capture a poacher if encountered without the need of other rangers. Additionally, in most predator-prey games, the evaders follow predefined policies rather than act as players in the game.

Multi-agent reinforcement learning benchmarks. There are no publicly available MARL benchmarks for anti-poaching. Nevertheless, Table 1 compares existing benchmarks from various MARL domains to APE. We focused our review on open-source standardized[4] MARL benchmarks featuring two types of agents (referred to as Type A and Type B), which allow for multiple agents of each type and whose interactions take place in a grid-based environment. Also, agents in all these benchmarks partially observe the environment in their neighbourhood. Some benchmarks, restricting learning to a single agent type, thereby defining cooperative environments (e.g. [4]) have not been included.

Environment	**Domain**	**Type A agents** Cooperation	**Type B agents** Cooperation	**A vs B** Competition
Simple-tag (MPE) [11]	Predator-prey	Fully-cooperative	Fully-cooperative	Zero-sum
Aquarium [7]	Predator-prey	Independent	Independent	General-sum
MATE [13]	Target coverage	Fully-cooperative	Fully-cooperative	Zero-sum
APE (ours)	**Anti-poaching**	**Fully-cooperative**	**Independent**	**Zero-sum**

Table 1: Comparison among relevant MARL environments.

[11] proposes relevant predator-prey environments that have been included in PettingZoo. For example Simple-tag is a predator-prey environment in which a team of cooperative predators compete with prey in a fully competitive (zero-sum) game. Aquarium [7] is a recent framework which unifies existing predator-prey domains. Here, agents move in a continuous space and the game is general sum.

The closest benchmark to APE is MATE, which simulates target coverage control problems with two types of learning agents: cameras and targets. Like APE, the relation between both types of agents is fully-competitive, resulting in a zero-sum game. Nevertheless, MATE considers that agents of the same type are fully cooperative.

3 The Anti-Poaching Game

The Anti-Poaching game is a finite-horizon, grid-based game between rangers and poachers. The rangers form a cooperative team, but do not communicate or share information. The poachers are independent agents who can place traps inside the grid. The rangers try to capture all the poachers in the grid, while the poachers try to evade detection and recover captured animals from traps they have placed. Each agent has only local, partial observation of her current cell, making this a challenging cooperative-competitive game to solve. We formally define the game as a Partially Observed Stochastic Game (POSG) $(\mathcal{I}, \mathcal{S}, \mathcal{A}, \mathcal{O}, R, T, O, H)$ [26], where $\mathcal{I}$ is the set of players, $\mathcal{S}$ the set of states of the game, $\mathcal{A}$ the set of joint actions of players, $\mathcal{O}$ comprises a set of observation sets (one for each player). In addition R is a set of instant reward functions (one for each player), T is a stochastic transition function between game states, O is a set of observation functions (one for each

[4]All the environments cited in Table 1 are PettingZoo compatible.

player) Finally, the horizon H is the number of decision steps. Solving a POSG means finding a *Nash equilibrium joint policy* for players, given an initial probability distribution over states, denoted ρ.

3.1 Agents and game states

The game is played between a team of cooperative rangers $\mathcal{R} = \{1, \ldots, I\}$ and some independent poachers $\mathcal{P} = \{I + 1, \ldots, I + J\}$ (note that $|\mathcal{R}| = I$ and $|\mathcal{P}| = J$) on a grid of size $\ell \times \ell$ over a finite horizon $[H] = \{1 \ldots, H\}$. The set of agents is thus $\mathcal{I} = \mathcal{R} \cup \mathcal{P}$. The game state at time $t \in [H]$ is a tuple $s^t = (\sigma^t, \tau^t)$, where σ^t is the state of all agents, and τ^t is the state of traps placed in the grid.

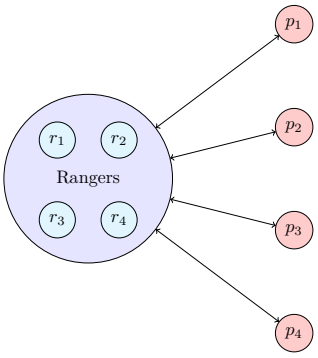

Figure 1: Agents interaction graph visualisation for an Anti-Poaching Game with 4 rangers and 4 poachers.

Agents A ranger's state is simply his location in the grid, defined as the tuple $\sigma_i^t = (m, n), i \in \mathcal{R}$. For poachers, we must also track the number of traps they are currently carrying, η_{trap} (i.e. not placed on the grid) and the number of animals they have recovered from traps so far η_{prey}. Their state is $\sigma_j^t = (m, n, \eta_{trap}, \eta_{prey}), j \in \mathcal{P}$. When a poacher is captured, her state immediately becomes $\sigma_j^t = (-1, -1, 0, 0)$ i.e. she is moved out of the grid to cell $(-1, -1)$, and no longer carries any traps or prey. Rangers are active during the entire game.

Placed Traps Each poacher starts with a fixed number of traps in each game instance. A trap placed in a cell can be either full or empty. Therefore, the state of all placed traps at time t is described as a 3D-array, capturing the number of empty and full traps for all poachers j in each cell (m, n), as $\tau_{j,m,n}^t = (\eta_{E,j}, \eta_{F,j}) \in \mathbb{N}^2$. Here, $\eta_{E,j}$ counts the number of empty traps that agent j has in the cell (m, n), and $\eta_{F,j}$ counts the number of full traps[5].

3.2 Actions

Each ranger $i \in \mathcal{R}$ can move or do nothing at each step. His action space is thus $\mathcal{A}_i = \{\emptyset, \uparrow, \leftarrow, \downarrow, \rightarrow\}$. A poacher $j \in \mathcal{P}$ can additionally place traps if she is carrying any, and thus has action space $\mathcal{A}_j = \{\emptyset, \uparrow, \leftarrow, \downarrow, \rightarrow, place - trap\}$. The joint action for each step is denoted as $a^t = (a_1^t, \ldots, a_{I+J}^t)$, $a^t \in \mathcal{A}$. There is no action $remove - trap$ in this game since we assume that agents will automatically remove a trap if they detect it: A ranger is always interested to remove a trap, while poachers need to inspect if a prey is caught inside.

3.3 Transition Model

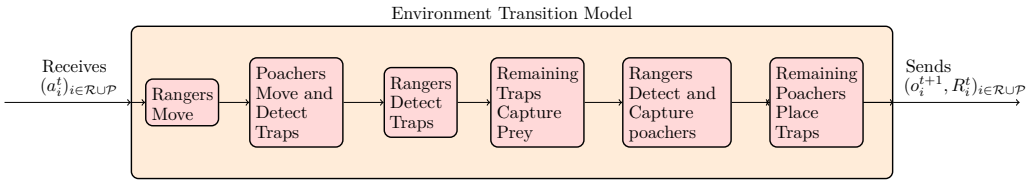

Figure 2: Transition Model for the Anti-Poaching Game. At each step, only the state variables of agents who apply or are affected by an action change value.

In a POSG, the environment receives the joint action $a^t = (a_i^t)_{i \in \mathcal{R} \cup \mathcal{P}}$, and carries out the transition $\langle s^t, a^t, s^{t+1} \rangle$. Using a transition model, we can calculate the transition and its

[5]The total number of (carried and placed) traps of an agent can never increase through time and is thus upper bounded by the number of traps they carried at time $t = 0$.

associated probability $T(s^t, a^t, s^{t+1}) = \mathbb{P}(s^{t+1}|s^t, a^t)$. The proposed model groups agents by action and uses a pre-defined action execution sequence. We assume that any object detected by a ranger is immediately removed from the game. This resolves ambiguous situations where a ranger and poacher detect the same trap. Under the current Anti-Poaching model, a poacher detects (and thus removes) her trap before any ranger can detect it. The model steps occur in the following order:

1. Rangers first transition deterministically to their new individual state.

2. Each poacher moves and reclaims only her traps from the new cell, if any.

3. All rangers now detect traps in their new cell. A ranger may detect each trap independently with probability p_{DT}.

4. Each remaining empty trap then captures an animal with probability p_{CA}.

5. Rangers detect poachers in their new cell. A ranger detects a poacher with probability p_{DP}, and detected poachers are assigned the terminal state $(-1, -1, 0, 0)$.

6. Finally, any remaining poacher who chose to place a trap does so.

Note that since each ranger searches a cell independently, the trap and agent detection probabilities improve with the number of rangers in a cell (Refer Sections A.3.2 and A.3.3). This incentivises rangers to coordinate between searching particular cells and grid exploration.

3.4 Observations

Observations are computed after each transition $\langle s^t, a^t, s^{t+1} \rangle$. Each agent $i \in \mathcal{R} \cup \mathcal{P}$ receives observation $o_i^{t+1} \in \mathcal{O}_i$, which is emitted by the system with probability $O_i(o_i^{t+1}|s^t, a^t, s^{t+1})$. In the Anti-Poaching game, rangers and poachers receive only their own partial observations which may differ from others'. There is no communication since poachers are independent, while rangers avoid any communication which can be intercepted by poachers.

Ranger i observes $o_i = \left(t_{rem}, \sigma_i^{t+1}, s_R, \eta_P, \eta_{capt}, \eta_{cell}\right)$ where t_{rem} is the remaining time till the end of the episode, σ_i^{t+1} is her new state at time $t+1$, $s_R = \{i' \in \mathcal{R}, \sigma_{i'}^{t+1} = \sigma_i^{t+1}\}$ lists all the rangers located in ranger i's new cell and η_P counts the poachers captured in this cell at $t+1$. Lastly, the tuple $\eta_{capt} = \left(\eta_{capt}^E, \eta_{capt}^F\right)$ counts the number of Empty and Full traps recovered from these captured poachers, while the tuple $\eta_{cell} = \left(\eta_{cell}^E, \eta_{cell}^F\right)$ gives the number of Empty and Full traps removed from i's new cell.

Poacher j observes $o_j = \left(t_{rem}, \sigma_j^{t+1}, \eta_R, \eta_P\right)$. (η_R, η_P) are the numbers of rangers and poachers that she detects in her new cell. poachers are indifferent to the identities of other agents since they do not interact with other poachers, and they do not care about the identity of rangers they encounter.

3.5 Rewards

A poacher is rewarded with R_{prey} when she reclaims an animal from a trap. When captured, she incurs a large penalty C_{rem} as well as a penalty $C_{prey} = -R_{prey}$ for each prey she is currently carrying. She also incurs a penalty C_{trap} whenever one of her traps is picked up by a ranger. This can happen when she is captured, or when a ranger detects one of her traps in a cell. The game is zero-sum in the sense that the rewards of the rangers' team is the opposite of the sum of the rewards of the poachers. Furthermore, we assume that the rangers share their rewards equally. Mathematically:

$$\forall (s^t, a^t, s^{t+1}) \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}, \forall i \in \mathcal{R},\ R_i(s^t, a^t, s^{t+1}) = -\frac{1}{|\mathcal{R}|} \sum_{j \in \mathcal{P}} R_j(s^t, a^t, s^{t+1}) \qquad (1)$$

4 The Anti-Poaching Environment (APE) and RLlib integration

In order to use the anti-poaching game framework as a benchmark for multi-agent RL algorithms, we provide an open-source easy to use environment for the PettingZoo API [16]. The environment is lightweight, written in pure Python and has limited dependencies.

Environment Configuration APE is implemented using the PettingZoo API [16], which is designed to simulate learning environments for Multi-Agent Reinforcement Learning (MARL) applications. The Anti-Poaching environment is compatible with multiple MARL libraries out of the box, notably RLlib [10] Stable Baselines-3 [15] and Tianshou [21]. An example use case, where agents select their actions uniformly at random, is given in Listing 1. APE also allows the configuration of multiple parameters like grid size and the probability of various events; the full list of configurable parameters is given in Table 2.

```
from anti_poaching.anti_poaching_game_v0 import anti_poaching

env = anti_poaching.parallel_env(render_mode="rgb")
observations, infos = env.reset()
done = False
while not done:
    action_mask = { agent: observations[agent]["action_mask"]
                    for agent in env.agents }
    actions = {
        agent: env.action_space(agent).sample(action_mask[agent])
        for agent in env.agents
    }
    observations, _, terminations, truncations, _ = env.step(actions)
    env.render()
    done = all(terminations.values() or truncations.values())
```

Listing 1: Example Use of the APE Environment

Training Configuration in RLlib APE's tight integration with RLlib allows for a large number of parameters to be defined for each training. This includes the environment parameters. It also includes various parameters related to the training such as the algorithm to use, the policies followed by rangers or poachers, whether the poachers (or rangers) learn, and other parameters such as the resources allocated.

Training Scenarios The RLlib integration further allows two training scenarios by specifying the set of learning agents. In the *Cooperative scenario*, only rangers learn during the training phase while poachers follow a heuristic policy. Currently, *Random* and *Static* heuristics are provided. An agent using the *Random* heuristic chooses a legal move uniformly at random at each time step. An agent using the *Static* heuristic randomly chooses target cells before the game begins and continuously cycles between them by taking the shortest path. Once at a target cell, she first recovers her trap (and thus any prey) if she has already placed one, and then places a new one. In the *Cooperative-Competitive* scenario, all agents concurrently learn their policies.

Calculating Exploitability Let $\mathcal{I} = \mathcal{R} \cup \mathcal{P}$ and $h_i^t = (a_i^0, o_i^0, \ldots, a_i^{t-1}, o_i^{t-1})$ denote the history of past observations and actions ($h_i^0 =_{def} \emptyset$). A joint mixed strategy profile is $\pi = (\pi_i^t)_{i \in \mathcal{I}, t \in [H]}$, where $\pi_i^t(h_i^t) \in \Delta(\mathcal{A}_i)$ defines a probability distribution over actions. The Exploitability metric measures the incentive for each player i to deviate[17]:

$$EXPL(\pi) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \left[v_{(BR(\pi_{-i}), \pi_{-i}), i}(h_i^0) - v_{\pi, i}(h_i^0) \right] \tag{2}$$

where $\pi_{-i} = (\pi_j)_{j \in \mathcal{I} \setminus \{i\}}$. $v_{\pi,i}(h_i^0)$ denotes the expected sum of rewards (till $t = H$) obtained by agent i when the mixed joint strategy profile π is applied, starting from a random initial state $s^0 \sim \rho$. $BR(\pi_{-i})$ denotes the Best Response of player i against π_{-i}. $EXPL(\pi) = 0$ if and only if π is a Nash equilibrium. Good policies have exploitability close to 0. Exact calculations of $BR(\cdot)$ and $EXPL(\cdot)$ are difficult since they require solving (single-agent) POMDPs. Approximate exploitability computation is implemented as a generic RLlib callback which is called after each evaluation iteration of the main algorithm.

5 Results

This section showcases results of applying RLLib RL algorithms to learn equilibrium strategies for *cooperative* and *cooperative-competitive* APE scenarios.

We compare the Average Ranger Reward (ARR) over the Sampled Environment Timesteps (SETs) for each algorithm. All algorithms are trained for 1 million SETs, and are periodically evaluated every 100,000 SETs. Each evaluation measures each ranger's Rewards per Episode over 100 simulated episodes and reports their ARR. For the Competitive scenario, we also compute the exploitability of the learned policies. Note that each ranger can gain $-\frac{J}{I}C_{rem}$ points at most for capturing poachers[6], which is 100 points per ranger for the default values.

To test the Cooperative training scenario, we use the Policy Gradient (IL-PG) and Proximal Policy Optimization (IL-PPO) algorithms as Independent Learners, and QMIX for cooperative learning. We do not test QMIX in the Competitive training scenario since it requires poachers to be considered as a team for learning and not independent agents.

Episode Horizon (H)	200
Grid dimensions ($\ell \times \ell$)	10×10
Number of rangers/poachers (N/M)	2 / 2
Probability of detection (p_{DP}/p_{DT})	0.2
Probability of animals appearing in a trap (p_{CA})	0.2
Poachers' reward ($R_{prey}/C_{trap}/C_{rem}$)	(1/-1/-100)
Poacher's initial number of traps (n_{traps})	3

Table 2: Default values of all configurable parameters for APE

Default Parameters and Hyperparameters Tuning The configurable parameters for an APE instance are given in Table 2, along with their default values. Due to obvious concerns about security, the majority of papers reviewed in anti-poaching do not make any real data public. However, [5] suggests that the probability of trap detection, p_{DT}, in a tropical forest landscape over a $0.25/km^2$ area, after 60 minutes of search effort, is 0.20 (95% coefficient interval ± 15-25%) irrespective of season, habitat or team. [12] also estimated that in the Nyungwe National Park, the probability of detecting poacher activity during a complete patrol (p_{DP} is "per cell"), was 0.1. To improve the quality of the learned policies, we perform hyper-parameter tuning for each algorithm in each scenario. To tune an algorithm, we launch 100 trials with randomly chosen hyperparameters from a chosen parameter space. Each trial trains an algorithm instance for 250,000 SETs and performs an evaluation iteration at the end. The set of chosen hyperparameters values is provided in the Appendix. For the Cooperative scenario, the quality of a solution is evaluated using ARR. For the Competitive scenario, the trial that minimises the Exploitability metric is considered the best.

Cooperative Training We compare the performance of Independent learners (PPO and PG) and a cooperative algorithm (QMIX) on default APE instances over grids of size $\ell = 5$ and $\ell = 10$ and for Random and Static poacher heuristics. The change in poacher policies is significant; the variance in ARR is much lower when learning against Random agents than against Static agents. This indicates that it is more difficult to learn against Static agents. Note also the effect of the grid size. 2 ranger teams learn to patrol efficiently on the small grid even under partial observations, capturing both poachers most of the time. However, they fail to find more than 1 poacher on average for the larger grid.

Cooperative-competitive Training Each algorithm is again trained for 1 million SETs and evaluated every 100,000 SETs to calculate the ARR. We also calculate the Approximate Exploitability of the joint policy at the end of each evaluation iteration using `ExploitabilityCallback`. We only compare Independent Learner algorithms here since all agents learn concurrently. As before, rangers learn to effectively patrol the smaller grid using

[6]The total reward for capturing all J poachers is $J \times C_{rem}$, shared equally by the I rangers.

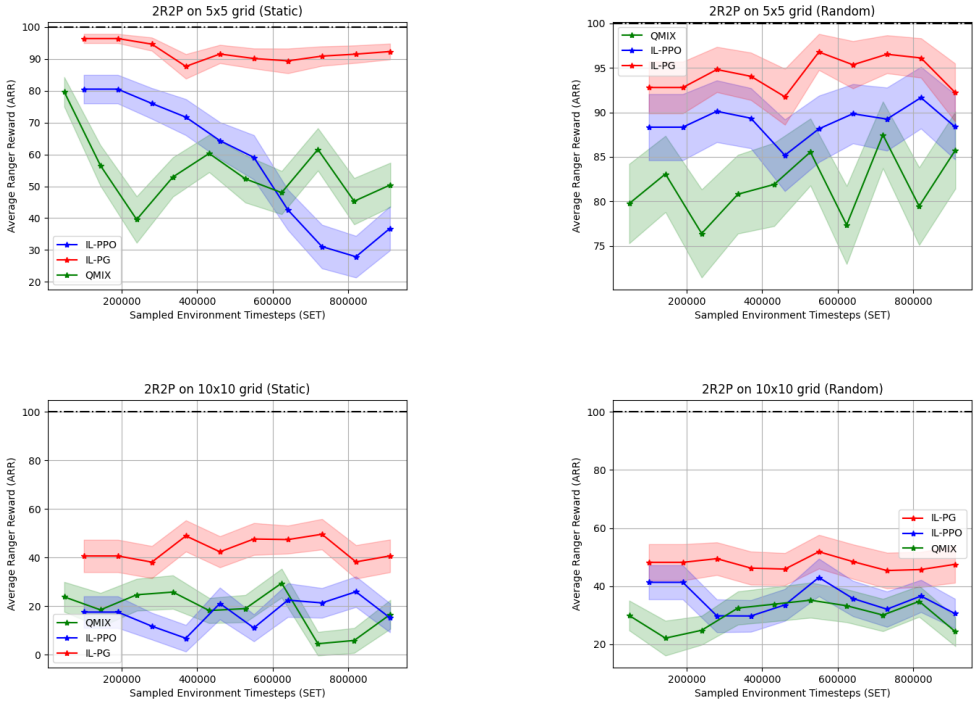

Figure 3: Comparison of Algorithm Performance in the Cooperative Training Scenario over 5×5 and 10×10 grids. The black line represents the reward for capturing all poachers.

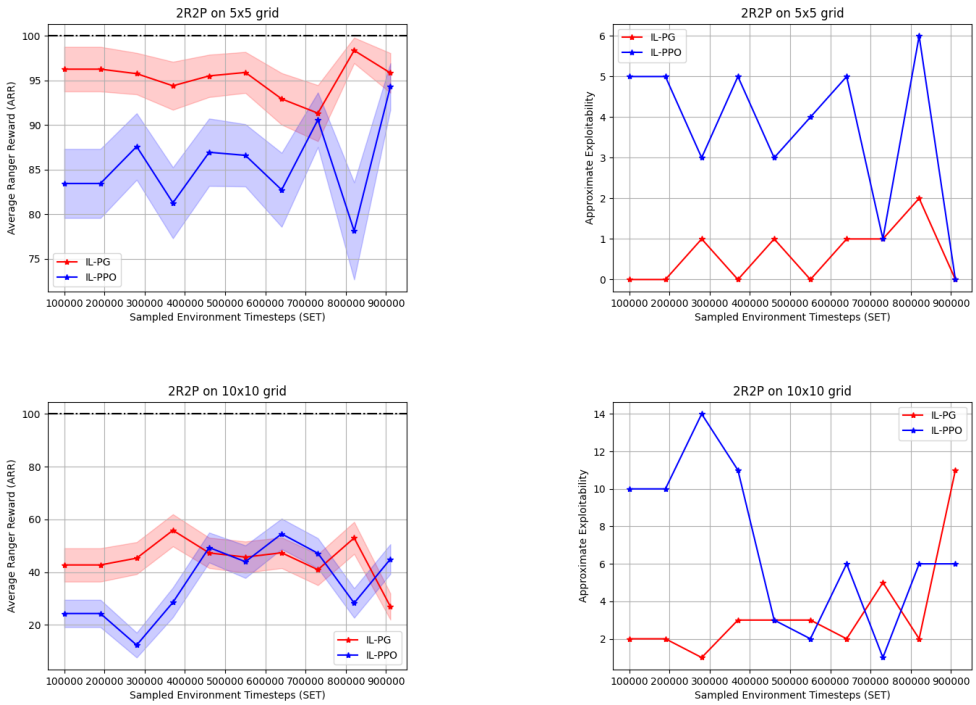

Figure 4: Comparison of Algorithm Performance in the Mixed Training Scenario over 5×5 and 10×10 grids. The black line represents the reward for capturing all poachers.

both IL-PPO and IL-PG, while they find 1 poacher on average on the larger grid. It is interesting to note that the Exploitability is small in both cases ($\sim 5\%$ to $\sim 10\%$), suggesting that the learned policies are robust even against adaptive poachers.

6 Conclusions

This paper introduces APE, the first standardised MARL environment for Anti-Poaching. Anti-poaching is first formalised as a partially observed stochastic game in which a team of rangers are in competition with a set of independent poachers. APE is a PettingZoo-compatible open-source Python implementation of a MARL environment for this game, offering a seamlessly integration with main existing RL libraries. APE is a novel contribution alongside other recent works (e.g. [7, 13]) that have recently proposed similar frameworks in other domains. To demonstrate APE's potential, we provide tests over example scenarios using PPO, PG and QMIX algorithm implementations from the RLlib library.

APE can be extended, in particular in the way partial observability is modelled. As can be seen from the tests, state-of-the-art RL algorithms seem not to improve their performance much through interactions with the environment (except in the 10×10 competitive scenario). Longer training periods did not significantly improve the learned policies either. Since the exploitability of the learned approximate equilibria is small, it looks like the variability of long term reward is small, whatever the rangers' policies. Likely, current observations are highly local and noisy, and thus do not contain enough information on poachers' behaviour and trajectories to allow the rangers to improve their policies through learning. Therefore, we will equip APE with various observation wrappers, so as to provide a means to study the role of observations in learning. We will provide a wrapper to make the problem a fully-observed cooperative-competitive stochastic game, for which equilibrium computation algorithms with guaranteed performance may be developed (e.g. [6] for non-RL approach and a single adversary). We also intend to provide wrappers to model intermediate forms of observability, between the current model and full observability, in order to improve the realism of our case study. Such models could include footstep observations [20], or different kinds of rangers with different observation and action capabilities (i.e. UAV agents [1, 19, 14]).

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

References

[1] Elizabeth Bondi, Hoon Oh, Haifeng Xu, Fei Fang, Bistra Dilkina, and Milind Tambe. To signal or not to signal: Exploiting uncertain real-time information in signaling games for security and sustainability. In *AAAI conference on Artificial Intelligence*, 2020. https://teamcore.seas.harvard.edu/sites/projects.iq.harvard.edu/files/teamcore/files/2020_02_teamcore_aaai_signaluncertainty.pdf.

[2] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[3] Shahrzad Gholami, Sara Mc Carthy, Bistra Dilkina, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, and Aggrey Rwetsiba. Adversary models account for imperfect crime data: Forecasting and planning against real-world poachers (corrected version). In *International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2018)*, 2018.

[4] Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 66–83. Springer, 2017.

[5] Harriet Ibbett, EJ Milner-Gulland, Colin Beale, Andrew DM Dobson, Olly Griffin, Hannah O'Kelly, and Aidan Keane. Experimentally assessing the effect of search effort on snare detectability. *Biological Conservation*, 247:108581, 2020.

[6] Fivos Kalogiannis, Ioannis Anagnostides, Ioannis Panageas, Emmanouil V. Vlatakis-Gkaragkounis, Vaggos Chatziafratis, and Stelios Andrew Stavroulakis. 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.

[7] Michael Kölle, Yannick Erpelding, Fabian Ritz, Thomy Phan, Steffen Illium, and Claudia Linnhoff-Popien. Aquarium: A comprehensive framework for exploring predator-prey dynamics through multi-agent reinforcement learning algorithms. In Ana Paula Rocha, Luc Steels, and H. Jaap van den Herik, editors, *Proceedings of the 16th International Conference on Agents and Artificial Intelligence, ICAART 2024, Volume 1, Rome, Italy, February 24-26, 2024*, pages 59–70. SCITEPRESS, 2024. doi: 10.5220/0012382300003636. URL https://doi.org/10.5220/0012382300003636.

[8] Wai Yee Lam, Chee-Chean Phung, Zainal Abidin Mat, Hamidi Jamaluddin, Charina Pria Sivayogam, Fauzul Azim Zainal Abidin, Azlan Sulaiman, Melynda Ka Yi Cheok, Noor Alif Wira Osama, Salman Sabaan, Abdul Kadir Abu Hashim, Mark Daniel Booton, Abishek Harihar, Gopalasamy Reuben Clements, and Rob Stuart Alexander Pickles. Using a crime prevention framework to evaluate tiger counter-poaching in a southeast asian rainforest. *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.

[9] Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. OpenSpiel: A framework for reinforcement learning in games. *CoRR*, abs/1908.09453, 2019. URL http://arxiv.org/abs/1908.09453.

[10] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael I. Jordan, and Ion Stoica. Rllib: Abstractions for distributed reinforcement learning. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3059–3068. PMLR, 2018. URL http://proceedings.mlr.press/v80/liang18b.html.

[11] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6379–6390, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/68a9750337a418a86fe06c1991a1d64c-Abstract.html.

[12] Jennifer F Moore, Felix Mulindahabi, Michel K Masozera, James D Nichols, James E Hines, Ezechiel Turikunkiko, and Madan K Oli. Are ranger patrols effective in reducing poaching-related threats within protected areas? *Journal of Applied Ecology*, 55(1): 99–107, 2018.

[13] Xuehai Pan, Mickel Liu, Fangwei Zhong, Yaodong Yang, Song-Chun Zhu, and Yizhou Wang. Mate: Benchmarking multi-agent reinforcement learning in distributed target coverage control. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27862–27879. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b2a1c152f14a4b842a9ddb3bd84c62a1-Paper-Datasets_and_Benchmarks.pdf.

[14] Andrew Perrault, Bryan Wilder, Eric Ewing, Aditya Mate, Bistra Dilkina, and Milind Tambe. End-to-end game-focused learning of adversary behavior in security games.

In *AAAI Conference on Artificial Intelligence*. AAAI, 2020. https://teamcore.seas.harvard.edu/sites/projects.iq.harvard.edu/files/teamcore/files/1903.00958.pdf.

[15] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

[16] J Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, et al. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:15032–15043, 2021.

[17] Finbarr Timbers, Nolan Bard, Edward Lockhart, Marc Lanctot, Martin Schmid, Neil Burch, Julian Schrittwieser, Thomas Hubert, and Michael Bowling. Approximate exploitability: Learning a best response, 7 2022. URL https://doi.org/10.24963/ijcai.2022/484. Main Track.

[18] Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. Gymnasium, March 2023. URL https://zenodo.org/record/8127025.

[19] Aravind Venugopal, Elizabeth Bondi, Harshavardhan Kamarthi, Keval Dholakia, Balaraman Ravindran, and Milind Tambe. Reinforcement learning for unified allocation and patrolling in signaling games with uncertainty. In Frank Dignum, Alessio Lomuscio, Ulle Endriss, and Ann Nowé, editors, *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, pages 1353–1361. ACM, 2021. doi: 10.5555/3463952.3464108. URL https://www.ifaamas.org/Proceedings/aamas2021/pdfs/p1353.pdf.

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

[21] Jiayi Weng, Huayu Chen, Dong Yan, Kaichao You, Alexis Duburcq, Minghao Zhang, Yi Su, Hang Su, and Jun Zhu. Tianshou: A highly modularized deep reinforcement learning library. *Journal of Machine Learning Research*, 23(267):1–6, 2022. URL http://jmlr.org/papers/v23/21-1127.html.

[22] Lily Xu. Learning and planning under uncertainty for green security. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4927–4928. ijcai.org, 2021. doi: 10.24963/IJCAI.2021/695. URL https://doi.org/10.24963/ijcai.2021/695.

[23] Lily Xu, Elizabeth Bondi, Fei Fang, Andrew Perrault, Kai Wang, and Milind Tambe. Dual-mandate patrols: Multi-armed bandits for green security. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14974–14982. AAAI Press, 2021. doi: 10.1609/AAAI.V35I17.17757. URL https://doi.org/10.1609/aaai.v35i17.17757.

[24] Lily Xu, Andrew Perrault, Fei Fang, Haipeng Chen, and Milind Tambe. Robust reinforcement learning under minimax regret for green security. In *Uncertainty in Artificial Intelligence*, pages 257–267. PMLR, 2021.

[25] Lily Xu, Esther Rolf, Sara Beery, Joseph R. Bennett, Tanya Y. Berger-Wolf, Tanya Birch, Elizabeth Bondi-Kelly, Justin Brashares, Melissa S. Chapman, Anthony Corso, Andrew

Davies, Nikhil Garg, Angela Gaylard, Robert Heilmayr, Hannah Kerner, Konstantin Klemmer, Vipin Kumar, Lester Mackey, Claire Monteleoni, Paul Moorcroft, Jonathan Palmer, Andrew Perrault, David Thau, and Milind Tambe. Reflections from the workshop on ai-assisted decision making for conservation. *CoRR*, abs/2307.08774, 2023. doi: 10.48550/ARXIV.2307.08774. URL https://doi.org/10.48550/arXiv.2307.08774.

[26] Yaodong Yang and Jun Wang. An Overview of Multi-Agent Reinforcement Learning from Game Theoretical Perspective, March 2021. URL http://arxiv.org/abs/2011.00583. arXiv:2011.00583 [cs].