entropy

MDPI

Article

Co-Evolution of Predator-Prey Ecosystems by Reinforcement Learning Agents

Jeongho Park [1,†], Juwon Lee [1,†], Taehwan Kim [1,†], Inkyung Ahn [2] and Jooyoung Park [1,*]

[1] Department of Control and Instrumentation Engineering, Korea University, 2511 Sejong-ro, Sejong-City 30019, Korea; seanpark0107@korea.ac.kr (J.P.); saero94j@korea.ac.kr (J.L.); kteaw0110@korea.ac.kr (T.K.)

[2] Department of Mathematics, College of Science and Technology, Korea University, 2511 Sejong-ro, Sejong-City 30019, Korea; ahnik@korea.ac.kr

* Correspondence: parkj@korea.ac.kr; Tel.: +82-10-9003-1810

† These authors contributed equally to this work.

Abstract: The problem of finding adequate population models in ecology is important for understanding essential aspects of their dynamic nature. Since analyzing and accurately predicting the intelligent adaptation of multiple species is difficult due to their complex interactions, the study of population dynamics still remains a challenging task in computational biology. In this paper, we use a modern deep reinforcement learning (RL) approach to explore a new avenue for understanding predator-prey ecosystems. Recently, reinforcement learning methods have achieved impressive results in areas, such as games and robotics. RL agents generally focus on building strategies for taking actions in an environment in order to maximize their expected returns. Here we frame the co-evolution of predators and preys in an ecosystem as allowing agents to learn and evolve toward better ones in a manner appropriate for multi-agent reinforcement learning. Recent significant advancements in reinforcement learning allow for new perspectives on these types of ecological issues. Our simulation results show that throughout the scenarios with RL agents, predators can achieve a reasonable level of sustainability, along with their preys.

Keywords: predator and prey; population; co-evolution; ecosystem; reinforcement learning

Citation: Park, J.; Lee, J.; Kim, T.; Ahn, I.; Park, J. Co-Evolution of Predator-Prey Ecosystems by Reinforcement Learning Agents. *Entropy* **2021**, *23*, 461. https://doi.org/10.3390/e23040461

Academic Editor: Giulia De Masi

Received: 17 March 2021
Accepted: 12 April 2021
Published: 13 April 2021

1. Introduction

The problem of addressing predator-prey interactions is an important field in ecology, and finding a reasonable population model for a predator-prey ecosystem is particularly important for understanding its dynamic features. Many researchers have studied population models with evolutionary dispersal perspectives, such as dispersal depending on other species [1–4] and starvation-driven diffusion depending on resources [5–10]. It is known that both the interaction between different species and the response of a species to its environment are necessary to develop a more realistic dispersal model for biological species [11,12]. For decades, researchers have developed dispersal theory based on the surrounding environment as an influential element [12], and the environment affecting a particular species includes elements, such as other interacting species. Because various species usually migrate to a region to find a more favorable habitat that provides sufficient food and/or better conditions for survival, an understanding of dispersal strategy is critically important to the study of species evolution. For a general explanation of discrete and continuous models on dispersal evolution, we refer the reader to References [12–16] and references therein. Many observations in nature demonstrate that the emergence of a predator or prey can induce the directed movement of a species; thus, researchers have proposed and investigated several mathematical models along these lines [17–21]. Despite the modeling and/or predictive capacity inherent within these mathematical approaches, the study of population dynamics still remains a challenging task in computational biology because analyzing and accurately predicting the intelligent adaptation of interacting

species is difficult, and it is often desirable to employ an agent-based perspective to shed more light on the topic.

The agent-based approach (e.g., References [22,23]) is based on mutually interacting agents via prescribed rules in a simulated environment, and can efficiently and conveniently describe individual and mutual behaviors together. Contrary to the traditional tools of population dynamics modeling, the agent-based approach does not resort to directly modeling equilibrium through mathematics. Nevertheless, Monte Carlo simulation combined with the agent-based approach in two-dimensional space can often reveal more diverse and detailed spatiotemporal patterns arising in the domains under consideration. When the state transition rules and rewards are not stipulated in advance, as is typical in domains where reinforcement learning is applied, training is realized by means of interactions with the environment, which includes the responses of other agents, which enables learning agents to discover and evolve toward a better policy. Recently, reinforcement learning (RL) [24] has achieved impressive advances in areas, such as games [25,26]. RL agents generally focus on building strategies for taking actions in an environment in order to maximize expected rewards. Here we frame the co-evolution of predators and prey in an ecosystem as allowing agents to learn and evolve toward better ones in a manner appropriate for multi-agent learning. This leads us to new perspectives on the problem at hand, thanks to recent significant advancements in reinforcement learning. In this paper, we use a multi-agent version of reinforcement learning, which is often called MARL (multi-agent reinforcement learning) to understand and model the co-evolution process of predator-prey ecosystems. In general, multi-agent reinforcement learning is known to be much more challenging than single-agent cases, because agents in an environment need to learn together.

Many recent works have undertaken related approaches. As an important MARL application domain, one may consider games, such as Go and StarCraft. In this domain, MARL brings about breakthroughs exhibiting superhuman performance for complex environments (e.g., Atari, Go, chess, shogi, and StarCraft) [27,28]. Visually complex and challenging tasks were considered and addressed with great success by means of the MuZero algorithm [27], based on combining a tree-based search with a trained model, and utilizing self-play for each board game domain. StarCraft2 is considered by AlphaStar [28], which utilizes the strategies of multi-agent reinforcement learning and imitation learning. Similar to Muzero, AlphaStar also conducts self-play for training, and despite the game's complexity, it turns out that AlphaStar reached the Grandmaster level for all races (Terran, Zerg, Protoss). Hahn et al. [29] considered swarms consisting of multi-agent individuals. For these individuals' objectives, a multi-agent reinforcement learning (MARL) method based on Deep Q-Networks (DQN) is utilized to execute a foraging task. Ritz et al. [30] applied reinforcement learning to train a predator in a single predator and a multi prey system, in which a predator evolves by means of RL and interacts with preys, which are non-RL agents. In their works, a version of DQN was utilized for the RL framework with long-term reward discounting and stacked observations. Phan et al. [31] also considered a MARL method for a multi-agent system, in which they proposed a novel approach called Stable Emergent Policy (STEP) approximation. The STEP approximation method is trained by means of a decentralized planning approach, in which a simulator is used for executing the planning. After the decentralized planning procedure is over, the trained policy is reintegrated into the method. Hahn et al. [32] proposed Swarm Emergent Learning Fish (SELFish), a reinforcement learning approach to evolve prey animals based on predation. For the multi-agent concept, every individual agent optimizes its own behavior without any centralization or integration. This leads to an emergent flocking behavior. Gabor et al. [33] proposed a hybrid adversarial learner, in which one hybrid component is a reinforcement learning mechanism for problem solving, and the other components is an evolutionary algorithm for finding instances of problems. While this method was simulated with the problem scenario of a smart factory, it is almost identical to the predator-prey problem, since the smart factory problem is defined in a grid environment, and the goal

of the agent is find a particular workstation. Hüttenrauch et al. [34] proposed a new RL state representation for swarm systems, and utilized ad-hoc feature spaces of the mean embedding based on histograms and radial basis functions, along with communication protocols. In Reference [35], Blasius et al. applied power analysis and bivariate wavelet analysis to quantify some statistical associations among the dynamics of predator and prey densities. The problem of formalizing the co-evolution of predators and preys as RL was recently introduced in Wang et al. [36], who claim that predators' RL ability contributed to the stability of an ecosystem and helped predators attain more reasonable behavior patterns of coexistence with their prey; the RL effect of prey on its own population was not as successful as that of predators, and increased the risk of extinction of the predators. Their subsequent work [37] adopted neural networks and presented a similar RL-based evolution mechanism for predator-prey ecosystems, with discretized features for states. The methodologies used in these works are somewhat similar in essence to those of the present paper, but there are some distinctive differences in the final results, which are to be detailed in the discussion section below. Finally, for a comprehensive survey of recent progress in multi-agent reinforcement learning, the reader is referred to, e.g., Reference [38].

In this paper, we utilize the perspective of recent multi-agent reinforcement learning (MARL) approaches, and present a MARL-based description of co-evolution mechanisms in predator-prey ecosystems, in which agents shows biologically plausible approximation of their co-evolution over multiple generations in nature. Specifically, we present a simple procedure for deriving co-evolution of agents with imperfect observations in predator-prey ecosystems by directly optimizing for equilibrium policies via reinforcement learning. We empirically show the policies this procedure yields and show that they demonstrate sound equilibrium properties. In addition, we empirically show that some dynamic properties of real predator-prey systems can be replicated with simulations.

The remaining sections of the paper are organized as follows: In Section 2, we describe the environment for simulations, along with learning agents. We explain the environment rules, and observations and rewards that characterize the learning agents. In addition, we introduce our multi-agent RL approach for agents' learning policies through interaction with simulations. In Sections 3 and 4, we describe the quantitative experiments and simulation results for the ecosystem under consideration, and we provide empirical results that validate the effectiveness of the RL agents in yielding reasonable outcomes. In Section 5, we analyze and discuss the qualitative behavior of the resulting agents. Finally, we conclude in Section 6, and sketch out some directions for future research.

The main contributions of this paper can be summarized as follows:

- First, we presented a simulation environment and learning agents that features predator-prey dynamics. In particular, we presented a novel procedure for co-evolution of predator-prey ecosystems by multi-agent reinforcement learning agents. More specifically, in the steps of the procedure for training predator and prey agents, we relied on a strategy, in which agents first compute approximate best response at each iteration, and then they try to strengthen their policies for responding well against the approximate best response. We also observed that the learned agent behaviors are somewhat ecologically plausible, in that they conform somewhat closely to results found in ecology research, e.g., the cycles in Lotka-Volterra equations [39,40].
- Second, we showcase some emergent features: RL-driven policies are qualitatively different from baseline random policies, yielding good solutions for both predators and preys. Moreover, RL-driven policies perform robustly with effective sustainability in the face of different initial conditions and/or environment sizes.
- Finally, we empirically show that throughout the scenarios resulting from co-evolution via multi-agent RL, predators found a reasonable means of survival, along with their preys, with a reduced risk of extinction.

2. Methods

The central concern of this paper is to examine the co-evolution of predator-prey ecosystems with an RL perspective, and this section presents a framework for studying the problem through numerical simulations with learning agents. In the following, we first describe the environment under consideration for interacting populations of predators and preys. The environment is a virtual ecosystem, in which predator and prey agents deal with partially observable states without explicit mutual communication. Next, we present reinforcement-learning-based reasoning about the emerging co-evolution in the predator-prey environment, which may be interpreted as similarity to what occurs in natural predator-prey ecosystems over a multitude of generations. Further related observations are to be provided later in the discussion section.

2.1. Environment Rules

In this subsection, we provide a detailed description of the environment for the virtual ecosystem under consideration, and its underlying dynamics for simulating predator-prey ecosystems. The environment is a grid world (see, e.g., Reference [36]) organized as a two-dimensional space in a quadrilateral form, with a lattice structure [41] consisting of $N \times N$ cells. We consider the $N = 50$ scenario in the simulations, and other N scenarios will also be considered later in the discussion section. The lattice cells can be taken by predators or preys, or may remain empty. At the beginning of each episode, a certain number of cells are chosen randomly as the initial locations of agents. Spacial boundaries of the environment are assumed to be periodic (see, e.g., Reference [41]), in order to deal with a space similar to unbounded cases. Predators and preys are both agents capable of learning. This subsection focuses only on the simulation aspects of the agents (e.g., rules of the environment, and observations of agents), which will be necessary for performing Monte Carlo simulations for the environment. A visualization of the environment under consideration is shown in Figure 1.

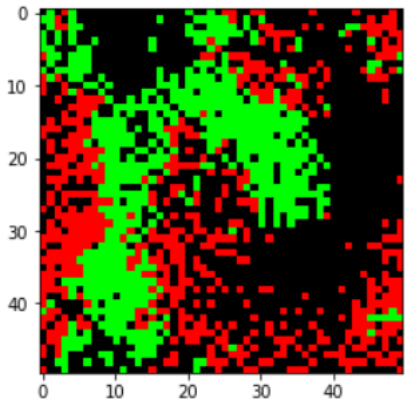

Figure 1. A visualization of the environment used for studying predator-prey ecosystems. The predators and prey agents are colored red and green, respectively.

- Dynamics of predators: A predator X can move to an adjacent cell which is not occupied by other agents. If the adjacent cell is already occupied by agents, two different rules are invoked, depending on the situation. If the adjacent cell is occupied by a prey organism, the predator can move and eat the prey in the cell while reproducing an offspring in the original cell with a success probability b_X. This reproduction follows the Bernoulli distribution with success probability b_X, which is denoted by $Bernoulli(b_X)$. If the outcome of the Bernoulli reproduction is a failure, then the predator simply moves on and eats the prey without reproducing offspring. If the adjacent cell to which X intends to move is already occupied by another predator, predator X cannot move and, thus, remains in the original cell. In this environment, it is assumed that all predators have the same maximum permissible starvation level, T_X. Every time a

step in the simulation passes without the predator eating, the predator's starvation level increases, and when its starvation level reaches the maximum permissible level, T_X, then the predator is removed from the environment, yielding an empty cell.

- Dynamics of prey animals: A prey organism Y can move to an adjacent cell which is not occupied, and is able to reproduce an offspring with a success probability b_Y. If the outcome of the Bernoulli reproduction is a failure, then the prey moves to the next cell without reproducing offspring. When the adjacent cell to which Y intends to move is occupied by a predator, the prey Y will be eaten by the predator as a result of the movement. If the adjacent cell to which Y intends to move is already occupied by another prey organism, the prey animal Y cannot move, and remains in its original cell. In this environment, it is assumed that all the prey organisms have the same maximum age, T_Y. With the passage of each simulation step, the prey age increases, and, when its age reaches T_Y, the prey organism is removed from the environment, yielding an empty cell.
- Observations of agents: In this environment, each agent conducts its moves based on its perception of its neighborhood, which is the square scope with the size $r \times r$. The agent spatial observations are padded as needed, when their observation window extends beyond the grid world. We consider the $r = 5$ case in simulations here. Each cell in the neighborhood is either empty or occupied by an agent (i.e., a predator or prey); hence, the number of agents at each cell cannot exceed one.
- Rewards: The agents need to interact with an antagonistic definition of rewards, in that a desirable result for one class of agents is undesirable for the other class, for which we define the reward functions of agents as follows:

$$\begin{aligned} Reward_{predator} &= \begin{cases} +1, \text{if a predator captures a prey animal as a result of its action} \\ 0, \text{otherwise} \end{cases} \\ Reward_{prey} &= \begin{cases} -1, \text{if a prey animal is captured as a result of its action} \\ 0, \text{otherwise} \end{cases} \end{aligned} . \quad (1)$$

Finally, we assume that there is no explicit communication among agents in the framework.

2.2. Policies of Predators and Preys

Agents navigate the environment in order to hunt or avoid being hunted. The action space of the agents includes nine actions for moving to an adjacent cell or remaining at the same location (Figure 2). As mentioned, agents are restricted from moving on top of cells occupied by other agents. An action consists of nine types of movement strategies shown in Figure 2. In our implementation, agents use the discrete action space (0: up-left, 1: up, 2: up-right, 3: left, 4: remain, 5: right, 6: down-left, 7: down, 8: down-right).

0	1	2
3	4	5
6	7	8

Figure 2. Discrete action space of the policy networks.

In this paper, we assume parameter sharing with decentralized execution [42] for each class of agents. Predators and prey organisms follow their own common policies $\pi_{\theta_{predator}}$ and $\pi_{\theta_{prey}}$, respectively. All predator (or prey) agents share the same parameters $\theta_{predator}$ (or θ_{prey}) during training but condition their policies $\pi_{\theta_{predator}}(a_i|s_i)$ (or $\pi_{\theta_{prey}}(a_i|s_i)$) on agent-specific observations s_i. Note that the agent-specific observation, s_i, is the r by r restricted view of the grid-world's true underlying global state $\mathbf{s}$ around the location of the

agent i. In addition note that in this setting, if a predator (or prey) agent learns a useful new behavior in some area of the state space, then this may be available for other predator (or prey) agents by means of training with experience. In addition, note that predator (or prey) agent behaviors remain heterogeneous because they all have different observations. For simplicity, we express the parameters of predators and prey organisms collectively by θ. In addition, for convenience of presentation, we often write π_θ as π.

We make use of a deep neural network based on a multi-layer perceptron (MLP) with two hidden layers to model agent policies. The outcome features of the MLP are used to effectively control and find an approximation of the best response policy with off-policy reinforcement learning. The action of agent i at time t is sampled according to the conditional probability implemented by the network, i.e.,

$$a_{i,t} \sim \pi_\theta(a_{i,t}|s_{i,t}), i \in \mathcal{I}, \tag{2}$$

where $\mathcal{I}$ is the collection of agents. The policy networks for predators and prey organisms are implemented by deep neural networks. The output of the policy network includes a probability distribution and the corresponding logit values over actions, and the input to the network consists of the agent-specific observations.

2.3. Multi-Agent RL-Based Learning of Agents

In this section, we are concerned with a solution to the problem of understanding some important interactions that arise in predator-prey ecosystems. For the solution procedure, we utilize the framework of modern multi-agents reinforcement learning (RL) in a model-free setting, in which prior knowledge of dynamics is not available for agents' sequential-decision-making, and each class of agents (i.e., predators and prey) attempt to adaptively learn efficient policies by relying on their own previous experiences in the unknown environment. We present a multi-agent reinforcement learning procedure to perform the training of agents in a stable manner for the problem, including the learning of approximate best responses and entropy regularization.

As is well-known, the joint optimization problem posed by multi-agent reinforcement learning problems may cause challenging problems, such as non-stationarity and instability, during training [43,44]. Since multi-agent decision-making problems include the effect of other agents' behaviors, encoded through agent policies, other agents' behaviors essentially change the environment. In addition, in this predator-prey ecosystem, simultaneously training both predators and prey together may create an unstable learning landscape. To meet such challenges, we use a basic strategy [45–47] of iterating two stages for computing approximate equilibria in sequential adversarial games. In the first stage, we train prey agents, which is implemented by deep neural networks, for the purpose of computing approximate best responses, and with predator training paths detached from the backpropagation path. We also train predator agents similarly for their approximate best response with prey training paths detached from the backpropagation path. In the second stage, we perform data gathering and policy gradient (PG)-based update. In PG-based update, we perform a gradient ascent for agents, i.e., on the predator strategy to increase predators' performance against approximately best responding prey organisms, and on the prey strategy to increase preys' performance against approximately best responding predators. The use of a policy gradient in the step is due to the observation that in multi-agent problems, a policy gradient approach tends to perform better than other methods when using feed-forward neural architectures [42]. The established procedure based on the strategy is outlined in Table 1.

Note that since the policy of opponent agents is fixed, each stage deals with a standard reinforcement learning problem, in which agents iteratively explore and discover which behaviors are optimal for their objectives as defined via the reward function. As a result of the policy and behavior of opponent agents undergoing changes, the agent policy faces a non-stationary problem [43] and needs to learn adaptively in each iteration. We model the learning problem for agents in the predator-prey ecosystem as a discounted

reward reinforcement learning problem [24] with states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, discount rate $\gamma \in (0,1)$, and time steps $t \in \{0,1,\cdots\}$, in which learning agents interact with a Markov decision process (MDP) environment [48], which is defined by the tuple $(\mathcal{S}, \mathcal{A}, T, r, \gamma)$. In the MDP, the environment's dynamics are characterized by state transition probabilities $T(s,a,s') \triangleq Pr\{s_{t+1} = s' | s_t = s, a_t = a\}$ and expected rewards $r(s,a) \triangleq E[r_t | s_t = s, a_t = a]$. The learning agent takes actions following the policy described as a conditional probability $\pi(a|s) \triangleq Pr\{a_t = a | s_t = s\}$, which is parametrized as π_θ. The objective of the learning agent is to pursue a policy that can maximize the discounted expected return

$$\rho(\pi_\theta) \triangleq (1-\gamma) E_{\tau \sim P(\tau|\theta)}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)], \tag{3}$$

where $P(\tau|\theta)$ is the distribution over state-action trajectories $\tau = (s_0, a_0, s_1, a_1, \cdots)$ induced by the policy π_θ and transition probabilities $T(s,a,s')$. The left-hand side of (3), $\rho(\pi_\theta)$, is the objective of the optimization, and often referred to as the value of the policy. With the help of the related concepts, Q^π and d^π, the value of the policy can be expressed in the frameworks of optimization. For a policy π, the state-action value function $Q^\pi(s,a)$ denotes the expectation of the future discounted reward sum of following π from the initial (s,a). In addition, $d^\pi(s,a)$ denotes how likely π is to visit (s,a) when interacting with environment. Note that, with the state-action value function

$$Q^\pi(s,a) \triangleq E[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a | \pi], \tag{4}$$

and the visit occupancy $d^\pi(s,a) \triangleq \sum_{t=0}^{\infty} \gamma^t Pr\{s_t = s, a_t = a | \pi\}$, we can express the objective function of the solutions with the following primal and dual problems:

$$\begin{aligned} (P) \quad & \rho(\pi) = \min_Q (1-\gamma) E_{\mu_0,\pi}[Q(s_0,a_0)] \\ & s.t.\ Q(s,a) \geq r(s,a) + \gamma P_\pi Q(s,a), \forall s,a, \end{aligned} \tag{5}$$

$$\begin{aligned} (D) \quad & \rho(\pi) = \max_{d \geq 0} E_d[r(s,a)] \\ & s.t.\ d(s,a) = (1-\gamma)\mu_0(s)\pi(a|s) + \gamma P_\pi^* d(s,a), \forall s,a. \end{aligned} \tag{6}$$

Since both the linear nature and min-max form of the resultant Lagrangian functions may lead to numerical instability [49], the strategy of introducing additional regularization to the objective leads to a better curvature. By regularizing with the f-divergence $D_f(d\|d^D)$ [50] for solving the problem in a more stable manner, one can obtain the following modified dual version:

$$\begin{aligned} (\tilde{D}) \quad & \rho(\pi) - D_f(d\|d^D) = \max_{d \geq 0} E_d[r(s,a)] - D_f(d\|d^D) \\ & s.t.\ d(s,a) = (1-\gamma)\mu_0(s)\pi(a|s) + \gamma P_\pi^* d(s,a), \forall s,a. \end{aligned} \tag{7}$$

In addition, by taking the Fenchel-Rockafellar duality [49] of $(\tilde{D})$, one can obtain the following modified primal version:

$$\begin{aligned} (\tilde{P}) \quad & \rho(\pi) = \min_Q (1-\gamma) E_{\mu_0,\pi}[Q(s_0,a_0)] \\ & + E_{(s,a)\sim d^D}[f_*(r(s,a) + \gamma P_\pi Q(s,a) - Q(s,a))], \forall s,a. \end{aligned} \tag{8}$$

Note that real meaning about the use of (8) is the advantages coming from the convexity in unconstrained formulation. In this paper, we deal with the nested problems in Table 1 by solving $(\tilde{P})$ iteratively for finding an approximate best response policy. In the process of solving $(\tilde{P})$, we follow the strategy of AlgaeDICE [51], in which the θ of the actor π_θ is updated via a policy gradient, and the parameters related with the Q and d are fit by optimizing the objective function of $(\tilde{P})$. In regularization with f-divergence for the modified problems $(\tilde{P})$ and $(\tilde{D})$, we used $f(x) = x^2/2$. As mentioned, agent policies are

implemented within the structure of deep neural networks, the inputs and outputs of which are for observations and actions, and their weights are trained as shown in the procedure shown in Table 1. Finally, in the process of training the neural networks for agents, we use entropy regularization, which adds the policy's entropy [52,53] as an additional weighted term in the policy gradient objective. The use of the entropy regularization term, which is defined as

$$\text{Entropy}(\pi_\theta) = -E_{a\sim\pi_\theta(\cdot|s)}[\log \pi_\theta(a|s)], \tag{9}$$

promotes more effective exploration by agents when used with policy gradient (PG).

Table 1. An established procedure for co-evolution of predator-prey ecosystems by reinforcement learning (RL) agents.

Given:
- Sampling horizon h
- Off-policy single agent RL algorithm $\mathcal{A}$ (AlgaeDICE [51])
- Stopping criterion $\mathcal{C}$ (e.g., maximum number of iterations, flag indicating that the performance indices have not improved)

Goal: To find trained results for agent policy networks

Procedure:
1. Initialize policy networks
2. Reset experience buffer $\mathcal{D}$
3. Reset episode
4. For each sampling horizon, do the following:
   - Compute an approximate best response for each agent policy via RL algorithm $\mathcal{A}$
   - Data collection: For each environment step, do
     - Agents take actions based on their policies; $a_{i,t} \sim \pi(a_{i,t}|s_{i,t}), \forall i \in \mathcal{I}$
     - Environment changes via environment rule; $\mathbf{s}_{t+1} \sim EnvRule(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$, where $\mathbf{a}_t = (a_{1,t}, \cdots, a_{|\mathcal{I}|,t})$
     - Collect state transition data
   - Update experience buffer $\mathcal{D}$ by adding the collected data, and conduct gradient update step for policy networks
5. Termination check with criterion $\mathcal{C}$, and if not satisfactory, go to step 3

3. Quantitative Experiments

In this section, we perform simulations to empirically validate the proposed learning mechanism, in which reasonable behaviors of agents emerge as a result of trained policies. We provide empirical results that validate the effectiveness of the trained policies in finding stable and ecologically plausible outcomes.

At the start of each episode, a given number of agents are randomly spawned in the cells. The state of the environment is represented as a $H \times W \times C$ tensor, where H and W are the size of the grid world, C is the number of unique entities that may occupy a cell, and the value of a given element indicates that a particular entity occupies the associated location.

We use AlgaeDICE [51] and the AdamW optimizer [54] to compute policy gradients. Samples were collected using a sampling horizon of $h = 70$ time steps between policy update iterations. Our experiments were conducted on a computing environment with a CPU (Intel i5-7287U) and GPU (Nvidia Titan XP), and performing the training of a single iteration for an episode took 14.7 s on average. In the process of training the neural networks for agents, we used PyTorch [55], which provided a convenient end-to-end framework with familiar building blocks. In addition, for the data plotting and visualization, we used the plotting package Matplotlib [56].

We evaluated our method via simulations. The environment of simulations is a two-dimensional grid world with both height and width of $N = 50$. At the beginning of the simulation, $n_X = 100$ predator agents and $n_Y = 500$ prey agents are randomly located on the grid for the initial state. As time step increases occur, agents change their positions according to their policies and the environment rule. The predator and prey reproduce offspring with probabilities of $b_X = 0.2$ and $b_Y = 0.6$, respectively. The predator's starvation level and prey age are initialized at 0. In addition, the maximum starvation level of predators is $T_X = 15$, and the maximum age of prey animals is $T_Y = 30$. The parameter set used for the simulations is outlined in Table 2.

Table 2. Parameter set used in simulations.

Notation & Value	Meaning
$b_X = 0.2$	Reproduction Probability of Predator
$b_Y = 0.6$	Reproduction Probability of Prey
$T_X = 15$	Maximum Starvation Level of Predator
$T_Y = 30$	Maximum Age of Prey
$n_X = 100$	Initial Number of Predators
$n_Y = 500$	Initial Number of Preys

4. Simulation Results

In this section, we report the simulation results, and provide a performance comparison with some baseline cases. We simulate episodes for training the model, and in the simulations, agent locations are updated sequentially following the environment rule. Each episode is terminated when its time step reaches a fixed maximum length, or when one of the species becomes extinct.

In Figure 3, we report our main results with three different random seeds, in which the predator and prey policies went through the co-evolution process of Table 1. We empirically found that the maximum number of iterations is a reasonable stopping criterion for the quantitative experiments for the present paper. More specifically, we performed all simulations with a couple of tens of iterations for steps 3, 4, and 5 in the co-evolution process, which we found to be sufficient for both predators and preys to converge to relatively stable policies. As shown in the figure, there are some patterns in the predator and prey populations. In both species, the population size fluctuates around a certain value, and this fluctuation continues while maintaining a certain level of gap. This oscillating pattern may be interpreted as follows: with the population growth of predators, more prey animals would be captured, and as a result of these captures, the prey population would be reduced. Then a reduction of the predator population would follow due to the reduction in the prey population, and this cycle seems to occur continuously.

For comparison, we considered cases in which the agents interact with random policies. Figure 4 shows the changing population of predators and prey with time. From Figure 4, one can see that the random policies can induce fluctuation in the population of predators and prey, which bears some resemblance with the results of Figure 3. However, there are two critical differences in these fluctuations brought about by random policies: First, the population dominance of prey over predators is not obvious in the population fluctuations; and, second, the population values of predators and preys exhibit large fluctuations around low values, which may lead to a risk of their extinction.

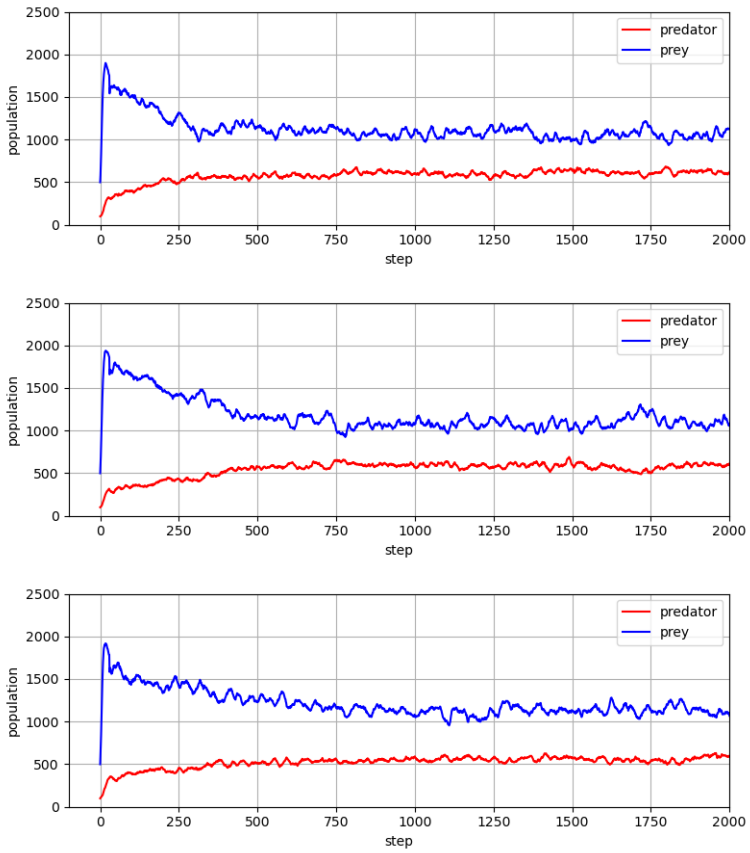

Figure 3. Population values of predator and prey, which went through the co-evolution process of Table 1.

A comparison of the simulation results in Figures 3 and 4 indicates that the co-evolution process of Table 1 brings some positive effects for the predator-prey ecological system, in that the populations of both species increased with a desirable ecological dominance and became stable.

For the final experimental issue of this section, we observe whether the trained policies have some robustness when they are started from random. More precisely, predators and prey take random actions during the first 500 steps. After 500 steps, predators and prey move according the policies on which they are trained by means of the co-evolution process as in Table 1. Simulation results of Figure 5 show that during the initial 500 steps, all three cases exhibit high fluctuation with the risk of extinction. One can see from the figure that the transition from initial high fluctuation to a stable balance occurs successfully around the change point at $t = 500$, and the magnitude of fluctuations after 500 steps becomes smaller. In addition, the average population values of predator and prey increased compared to their values in the initial 500 steps, leading to a significant reduction of extinction risk. Some more points concerning the robustness of the trained agent policies will be discussed in the next section.

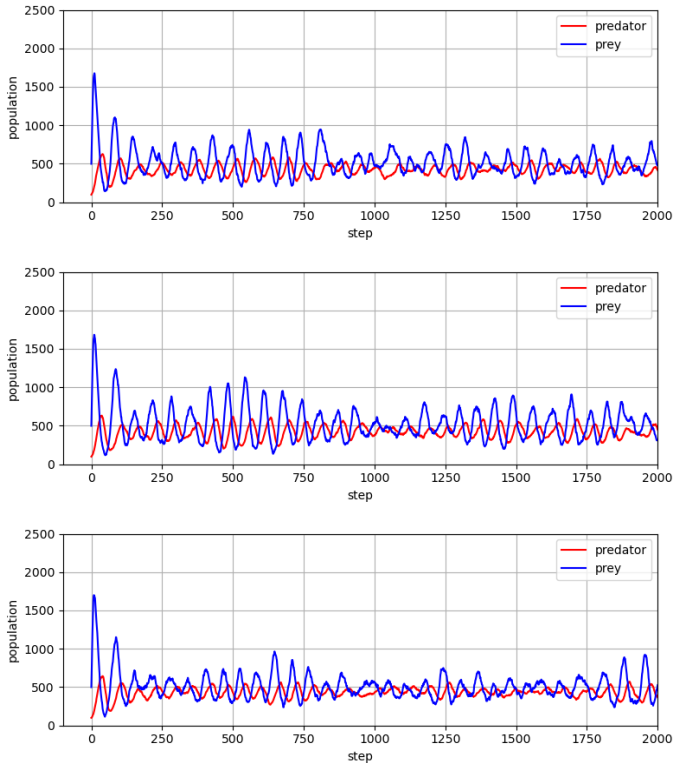

Figure 4. Population values of predator and prey, which result from random policies.

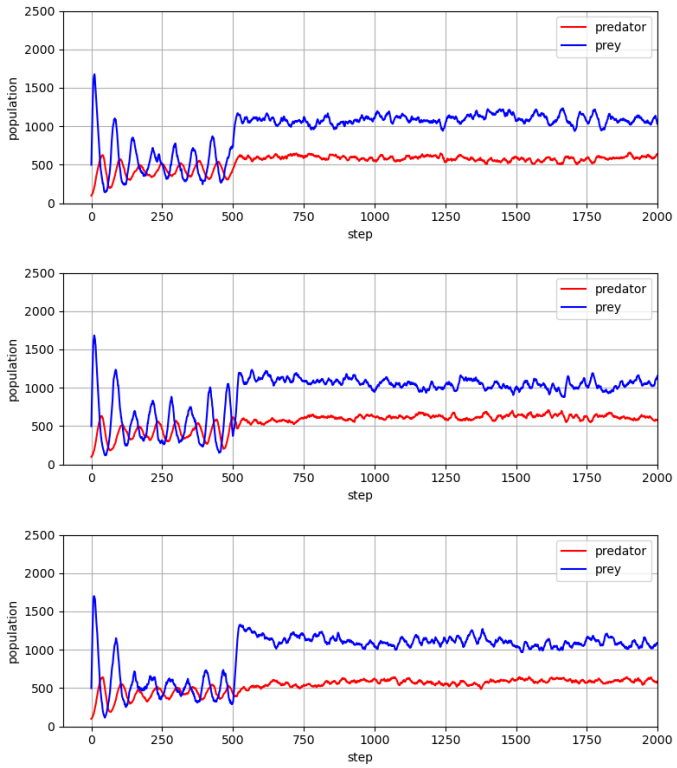

Figure 5. Population of predators and prey, which show a transition from random to trained policies at $t = 500$.

5. Discussion

In this paper, we investigated the use of multi-agent reinforcement learning for characterizing the dynamic nature of predator-prey ecosystems. Recently, reinforcement learning methods have achieved impressive advances in areas, such as games and robotics. Reinforcement learning agents focus on building strategies that lead to taking actions in an environment in order to maximize expected rewards. The key idea behind our characterization is to frame the co-evolution of predators and preys in an ecosystem as enabling learning agents to try to optimize for equilibrium policies in a manner appropriate for multi-agent reinforcement learning. As mentioned in the section of introduction, there have been important MARL efforts which are related with the present paper. These works may look somewhat similar, but details, such as what type setting (e.g., cooperative, competitive, and mixed) the problem handles, what are the assumptions for the agents, and how training proceeds, are different. For example, one of the related MARL efforts is the stable emergent policy approach called STEP of Reference [31]. The present paper and STEP share some similar features because both are multi-agent reinforcement learning efforts, and the trained polices are executed in a decentralized fashion in the sense that each agent takes action conditioned on its own observation, whereas the focus of STEP is somewhat different from our works in that they deal with cooperative tasks and rely on the planning enabled by a simulator. We believe that our final results are distinctive in explaining the considered ecological issues, and working well when planning is not available.

The problem under consideration may be described as a decentralized version of a partially observable multi-agent Markov game, and its solutions may be sought by finding Nash equilibria. A set of optimal policies form a Nash equilibrium as long as no agent wants to unilaterally deviate from its own policy. Finding exact equilibria for the ecosystems considered might be intractable and is beyond the scope of this paper. Nevertheless, we observed that a multi-agent RL-based procedure can yield reasonable co-evolution, along with emergent fluctuations somewhat similar to cycles in the Lotka-Volterra equation [39,40]. In Figure 6, the relation between the population of predators and prey is represented by a periodic pattern of fluctuations. As in Figure 5, simulations for Figure 6 began with random policies in their initial 500 steps. Then both species moved according to the policies by which they were trained via co-evolution in Table 1, which results in average population values of predators and prey increasing over the values of the initial period. During the initial 500 steps with random policies, the radius of the circle representing the magnitude of population variations of predators and prey is relatively large. After 500 steps, the cycles assume a reduced size, and their movements around the centers become more sustainable, in that they are maintained with a reduced risk of extinction. It was found that the mid-point of the circle moved to the top right after 500 steps.

As an additional pattern observed from the numerical experiments, we can see emergence of swarming in Figure 7. Self-organization in the form of swarming is a well-studied process in the field of population biology. This concerns the emergence of globally ordered population dynamics in space and time, realized from collective interactions between agents without any explicit external intervention. Figure 7 shows the emergence of swarming in the movements of the trained predators and prey organisms. The left columns of the figure shows the initial random locations of species, while the right columns show locations when their time steps reach $t = 2000$. From this figure, one can see that prey organisms exhibit a tendency of swarming for better survival from attempted capture by predators, while predators tend to swarm for convenience in hunting with fewer movements. Note that swarming here is simply a consequence of agents learning to maximize their own individual rewards, without direct intervention from the environment.

We also investigated whether the learned agent policies can be effective without additional parameter tuning for different initial conditions (e.g., Figure 8) or in environments with different sizes (e.g., Figure 9). The investigations show that the trained agent policies

resulting from the procedure of Table 1 work effectively for different initial conditions, and scales well for to a smaller or larger environment size. The positive results in this investigation suggest that by learning with principles of multi-agent reinforcement learning, agents can achieve robust performance against deviation from a nominal environment.

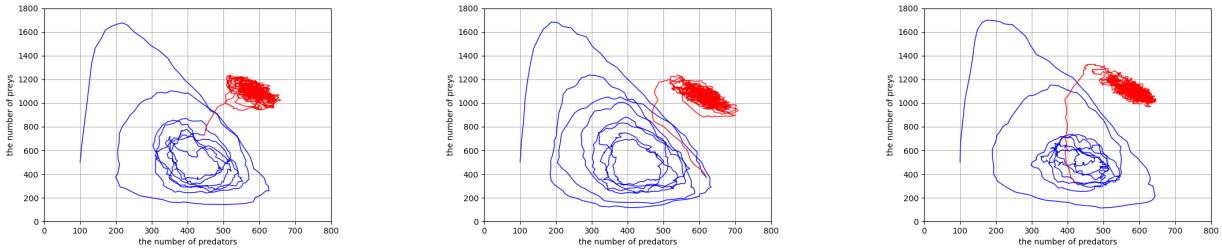

Figure 6. Emergent behaviors similar to cycles in the Lotka-Volterra equation. Initial and later locations are colored blue and red, respectively.

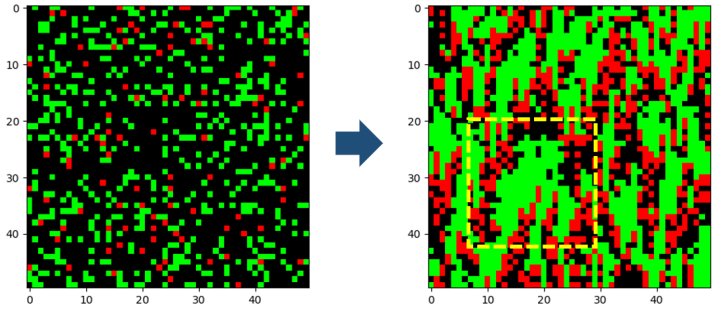

Figure 7. Left: Initial random location of predators and prey. **Right**: Emergence of swarming among predators and prey.

For enhanced realism in the model, we also conducted simulations for cases with predator age considered. As shown in Figure 10, it turns out that the simulation results remain almost the same when the maximum age of predator is older than that of prey, as is common in real-world predator-prey situations. We also observed that when the predator age is shorter than the prey age, the results become somewhat different. Detailed analysis of reasons for this difference is one of the topics to be addressed in future studies.

For performance comparison with an existing approach, we considered an approach based on deep Q-learning (DQN) [25]. As is well-known, using deep neural networks along with Q-learning [24] today is an important method for reinforcement learning, and has been utilized extensively for problems involving sequential decision-making. The problem of formalizing the co-evolution of predators and preys as a deep Q-learning-based reinforcement learning was recently introduced in Wang et al. [36]. In addition, their subsequent work [37] adopted neural networks and presented a similar deep Q-learning-based evolution mechanism for predator-prey ecosystems, with discretized features for states. These deep Q-learning-based works rely on the update mechanism of Q in the following form:

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[reward + \gamma \max_{a_{next} \in \mathcal{A}} Q(s_{next}, a_{next})].$$

The simulation results of Reference [36] show that when the predators and preys co-evolved, predators and preys updated their networks according to each other's behavior, which led to the reduced oscillations of the ecosystem. We compared the results of the proposed method to those of the DQN-based approach [36] with $\epsilon = 0.05$ for exploration, and learning rates for the predator and the prey policy networks set as 10^{-5} and 10^{-3}, respectively. Figure 11 shows the corresponding results obtained by the DQN-based

approach. One can observe that the trajectories in the center of Figure 11 looks somewhat similar to the corresponding case reported in Figure 3 of Reference [36]. A comparison shows that our results are better, in that the resulting co-evolution of predators and prey brings about more stable ecological systems.

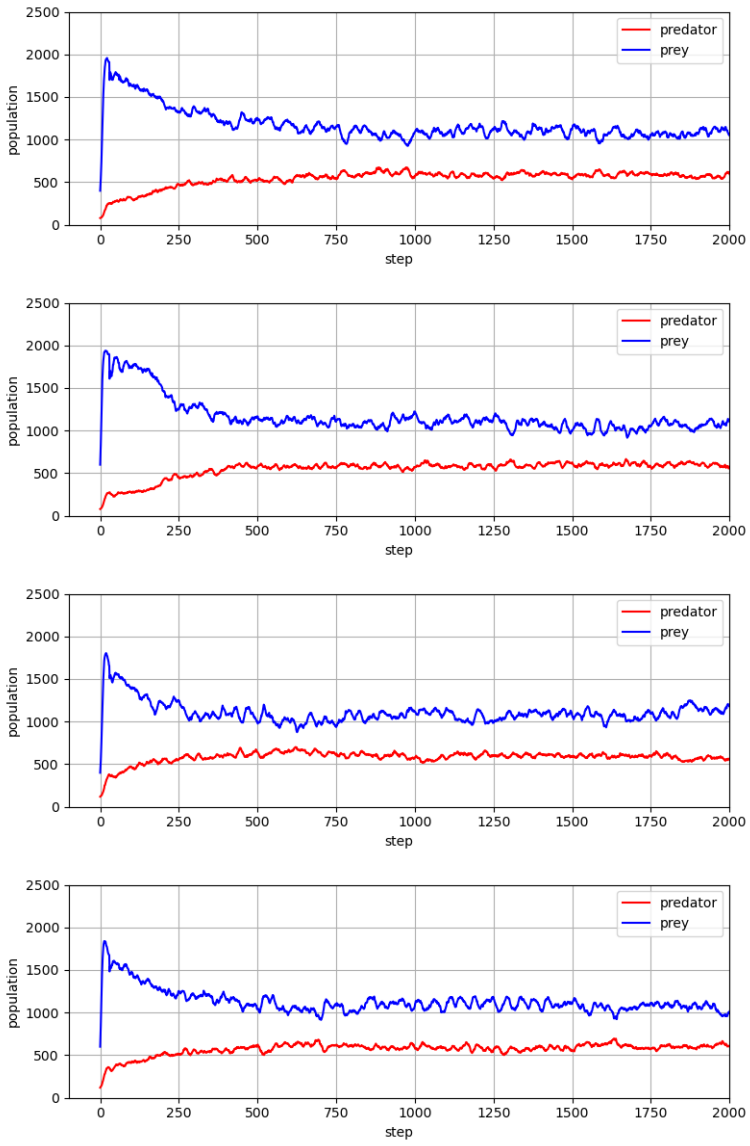

Figure 8. Population values of predator and prey, which went through the co-evolution process of Table 1. Simulation results show that learned agent policies can be effective without additional parameter tuning for different initial conditions (Initial numbers of predators and preys, n_X and n_Y, are changed by $\pm 20\%$.).

Finally, simulation studies, like ours, may naturally have some limitations. For example, they are not based on real world data on agents' behaviors and interactions, and consider a relatively small environment in order to avoid enormous computational loads. If demonstrations of predator and prey agents with successful hunting and/or survival skills are provided, along with relevant information, then, in principle, their rewards could be estimated via so-called multi-agent adversarial inverse reinforcement learning [57] and used with the procedure. Such demonstration data are currently not available to the best of authors' knowledge, and further study along these lines is a topic that warrants future research. Future simulations could further improve on this by making use of large-scale reinforcement learning methods with explicit targets conforming to real-

world observations. For improved follow-up research, one may also consider this problem on a larger scale, e.g., so that co-evolution in digital and/or business ecosystems [58] can be included.

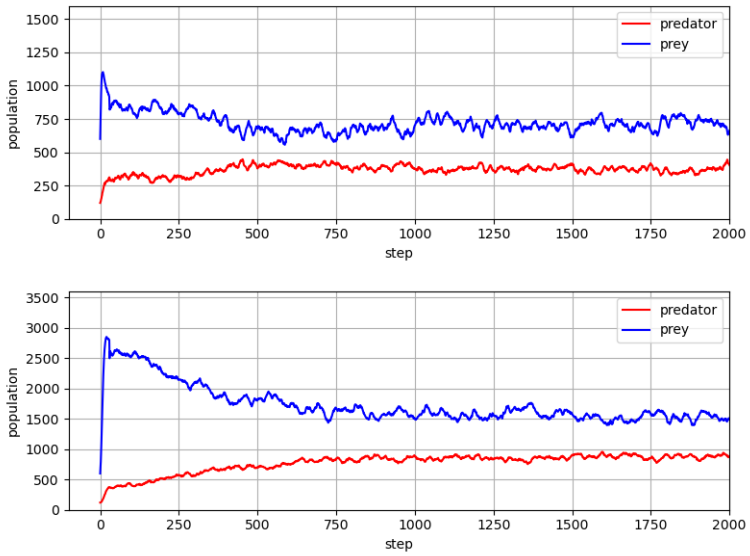

Figure 9. Population values of predator and prey, which went through the co-evolution process of Table 1. Simulation results show that learned agent policies can be effective without additional parameter tuning for different environment sizes ($N = 40$ and $N = 60$ cases).

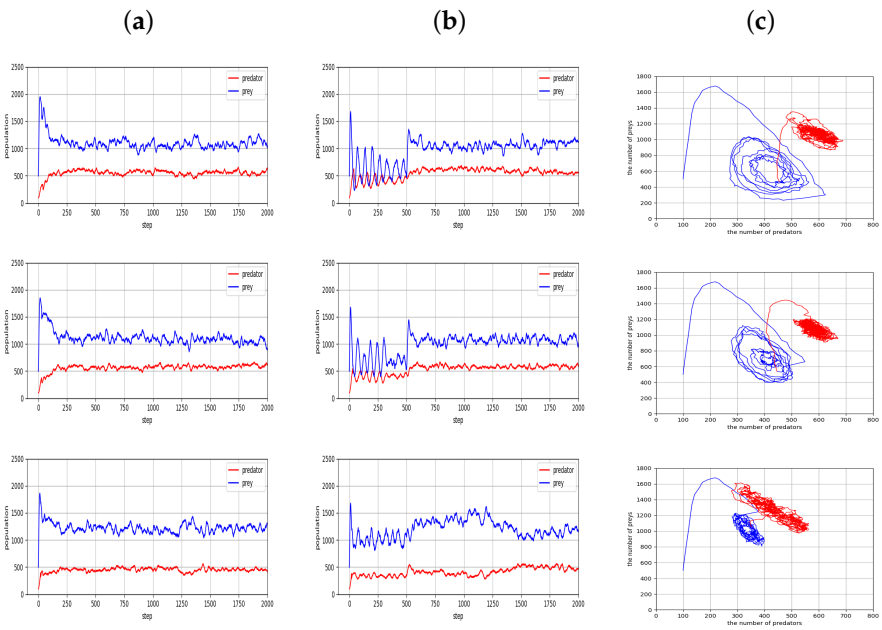

Figure 10. Simulation results for the cases trained with the maximum age of predator set as 40 (**a**), 30 (**b**), and 20 (**c**).

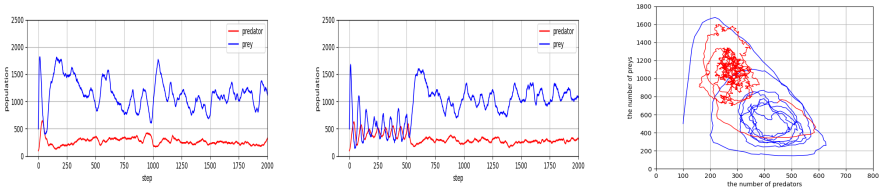

Figure 11. Simulation results for the cases trained with Deep Q-Networks (DQN)-based approach.

6. Concluding Remarks

In this paper, we use a modern deep reinforcement learning (RL) approach to explore a new avenue for understanding key population dynamics of predator-prey ecosystems. Reinforcement learning methods have achieved impressive results, and reinforcement learning agents generally focus on building strategies that lead to agents taking actions in an environment in order to maximize expected reward. In this paper, we frame the co-evolution of predators and preys in an ecosystem as building learning agents that optimize for equilibrium policies in a manner appropriate for multi-agent reinforcement learning. This novel approach leads to helpful insights on these types of complex problems.

Our simulation results show that throughout the scenarios with reinforcement learning agents, predators can find a reasonable level of sustainability, along with their prey, and co-evolution of predators and preys brings about stable ecological systems. In addition, we found that training with multi-agent and model-free reinforcement learning can yield agents with ecologically plausible behaviors, such as population fluctuations, around some constant values, and the emergence of swarming. We believe that, with a combination of a variety of ecosystems and modern reinforcement learning methods, one can find a wide range of important results. In future works, we would like to explore these combination in related fields, where agents play more general roles. We also plan to investigate the robustness of our method for environments with more general features.

Author Contributions: J.P. (Jooyoung Park), J.P. (Jeongho Park), T.K., and J.L. conceived and designed the methodology of the paper with help of I.A.; J.P. (Jeongho Park), J.L., T.K. and J.P. (Jooyoung Park) designed the simulations and wrote the computer program for the simulations; J.P. (Jooyoung Park), J.P. (Jeongho Park), and I.A. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Research Foundation of Korea, 2017R1E1A1A03070652 and 2020R1F1A1072772.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported by the National Research Foundation of Korea (NRF) grants (No. 2017R1E1A1A03070652, No. NRF-2020R1F1A1072772) funded by the Korea government (MSIT).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Averill, I.; Lam, K.Y.; Lou, Y. *The Role of Advection in a Two-Species Competition Model: A Bifurcation Approach*; American Mathematical Society: Providence, RI, USA, 2017.
2. Kuto, K.; Yamada, Y. On limit systems for some population models with cross-diffusion. *Discret. Contin. Dyn. Syst. B* **2012**, *17*, 2745.
3. Lou, Y.; Ni, W.M.; Yotsutani, S. On a limiting system in the Lotka–Volterra competition with cross-diffusion. *Discret. Contin. Dyn. Syst. A* **2004**, *10*, 435. [CrossRef]
4. Lou, Y.; Tao, Y.; Winkler, M. Nonexistence of nonconstant steady-state solutions in a triangular cross-diffusion model. *J. Differ. Equ.* **2017**, *262*, 5160–5178. [CrossRef]

5. Kim, Y.J.; Kwon, O.; Li, F. Global asymptotic stability and the ideal free distribution in a starvation driven diffusion. *J. Math. Biol.* **2014**, *68*, 1341–1370. [CrossRef] [PubMed]
6. Kim, Y.J.; Kwon, O. Evolution of dispersal with starvation measure and coexistence. *Bull. Math. Biol.* **2016**, *78*, 254–279. [CrossRef] [PubMed]
7. Choi, W.; Ahn, I. Non-uniform dispersal of logistic population models with free boundaries in a spatially heterogeneous environment. *J. Math. Anal. Appl.* **2019**, *479*, 283–314. [CrossRef]
8. Choi, W.; Baek, S.; Ahn, I. Intraguild predation with evolutionary dispersal in a spatially heterogeneous environment. *J. Math. Biol.* **2019**, *78*, 2141–2169. [CrossRef]
9. Choi, W.; Ahn, I. Strong competition model with non-uniform dispersal in a heterogeneous environment. *Appl. Math. Lett.* **2019**, *88*, 96–102. [CrossRef]
10. Choi, W.; Ahn, I. Predator-prey interaction systems with non-uniform dispersal in a spatially heterogeneous environment. *J. Math. Anal. Appl.* **2020**, *485*, 123860. [CrossRef]
11. Skellam, J.G. The formulation and interpretation of mathematical models of diffusional process in population biology. In *The Mathematical Theory of The Dynamic of Biological Populations*; Springer: Berlin/Heidelberg, Germany, 1973.
12. Okubo, A.; Levin, S.A. *Diffusion and Ecological Problems: Modern Perspectives*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
13. Cohen, D.; Levin, S.A. Dispersal in patchy environments: The effects of temporal and spatial structure. *Theor. Popul. Biol.* **1991**, *39*, 63–99. [CrossRef]
14. Johnson, M.; Gaines, M. Evolution of dispersal: theoretical models and empirical tests using birds and mammals. *Annu. Rev. Ecol. Syst.* **1990**, *21*, 449–480. [CrossRef]
15. Nagylaki, T. *Introduction to Theoretical Population Genetics*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
16. Cantrell, R.S.; Cosner, C. *Spatial Ecology Via Reaction-Diffusion Equations*; John Wiley & Sons: Hoboken, NJ, USA, 2004.
17. Choi, W.; Ahn, I. Effect of prey-taxis on predator's invasion in a spatially heterogeneous environment. *Appl. Math. Lett.* **2019**, *98*, 256–262. [CrossRef]
18. Ahn, I.; Yoon, C. Global well-posedness and stability analysis of prey-predator model with indirect prey-taxis. *J. Differ. Equ.* **2020**, *268*, 4222–4255. [CrossRef]
19. Wu, S.; Shi, J.; Wu, B. Global existence of solutions and uniform persistence of a diffusive predator-prey model with prey-taxis. *J. Differ. Equ.* **2016**, *260*, 5847–5874. [CrossRef]
20. Jin, H.Y.; Wang, Z.A. Global stability of prey-taxis systems. *J. Differ. Equ.* **2017**, *262*, 1257–1290. [CrossRef]
21. Tao, Y. Global existence of classical solutions to a predator & prey model with nonlinear prey-taxis. *Nonlinear Anal. Real World Appl.* **2010**, *11*, 2056–2064.
22. Holland, J.H.; Miller, J.H. Artificial adaptive agents in economic theory. *Am. Econ. Rev.* **1991**, *81*, 365–370.
23. Macal, C.; North, M. Introductory tutorial: Agent-based modeling and simulation. In Proceedings of the Winter Simulation Conference 2014, Savannah, GA, USA, 7–10 December 2014; pp. 6–20.
24. Sutton, R.S.; Barto, A.G. *Introduction to Reinforcement Learning*; MIT Press: Cambridge, MA, USA, 1998.
25. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjel, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
26. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef] [PubMed]
27. Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature* **2020**, *588*, 604–609. [CrossRef] [PubMed]
28. Vinyals, O.; Babuschkin, I.; Czarnecki, W.M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D.H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **2019**, *575*, 350–354. [CrossRef] [PubMed]
29. Hahn, C.; Ritz, F.; Wikidal, P.; Phan, T.; Gabor, T.; Linnhoff-Popien, C. Foraging swarms using multi-agent reinforcement learning. In *Artificial Life Conference Proceedings*; MIT Press: Cambridge, MA, USA, 2020; pp. 333–340.
30. Ritz, F.; Hohnstein, F.; Müller, R.; Phan, T.; Gabor, T.; Hahn, C.; Linnhoff-Popien, C. Towards ecosystem management from greedy reinforcement learning in a predator-prey setting. In *Artificial Life Conference Proceedings*; MIT Press: Cambridge, MA, USA, 2020; pp. 518–525.
31. Phan, T.; Belzner, L.; Schmid, K.; Gabor, T.; Ritz, F.; Feld, S.; Linnhoff-Popien, C. A Distributed Policy Iteration Scheme for Cooperative Multi-Agent Policy Approximation. Available online: https://ala2020.vub.ac.be/papers/ALA2020_paper_36.pdf (accessed on 13 April 2021).
32. Hahn, C.; Phan, T.; Gabor, T.; Belzner, L.; Linnhoff-Popien, C. Emergent escape-based flocking behavior using multi-agent reinforcement learning. In *Artificial Life Conference Proceedings*; MIT Press: Cambridge, MA, USA, 2019; pp. 598–605.
33. Gabor, T.; Sedlmeier, A.; Kiermeier, M.; Phan, T.; Henrich, M.; Pichlmair, M.; Kempter, B.; Klein, C.; Sauer, H.; Wieghardt, J. Scenario co-evolution for reinforcement learning on a grid world smart factory domain. In Proceedings of the Genetic and Evolutionary Computation Conference, New York, NY, USA, 13–17 July 2019; pp. 898–906.
34. Hüttenrauch, M.; Adrian, S.; Neumann, G. Deep reinforcement learning for swarm systems. *J. Mach. Learn. Res.* **2019**, *20*, 1–31.
35. Blasius, B.; Rudolf, L.; Weithoff, G.; Gaedke, U.; Fussmann, G.F. Long-term cyclic persistence in an experimental predator & prey system. *Nature* **2020**, *577*, 226–230.

36. Wang, X.; Cheng, J.; Wang, L. Deep-reinforcement learning-based co-evolution in a predator & prey system. *Entropy* **2019**, *21*, 773.
37. Wang, X.; Cheng, J.; Wang, L. A reinforcement learning-based predator-prey model. *Ecol. Complex.* **2020**, *42*, 100815. [CrossRef]
38. Hernandez-Leal, P.; Kartal, B.; Taylor, M.E. A survey and critique of multiagent deep reinforcement learning. *Auton. Agents -Multi-Agent Syst.* **2019**, *33*, 750–797. [CrossRef]
39. Lotka, A.J. Contribution to the theory of periodic reactions. *J. Phys. Chem.* **2002**, *14*, 271–274. [CrossRef]
40. Allman, E.S.; Allman, E.S.; Rhodes, J.A. *Mathematical Models in Biology: An Introduction*; Cambridge University Press: Cambridge, UK, 2004.
41. Carneiro, M.V.; Charret, I.C. Spontaneous emergence of spatial patterns in a predator-prey model. *Phys. Rev.* **2007**, *76*, 061902. [CrossRef] [PubMed]
42. Gupta, J.K.; Egorov, M.; Kochenderfer, M. Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*; Springer: Cham, Switzerland, 2017; pp. 66–83.
43. Papoudakis, G.; Christianos, F.; Rahman, A.; Albrecht, S.V. Dealing with non-stationarity in multi-agent deep reinforcement learning. *arXiv* **2019**, arXiv:1906.04737.
44. Zhang, Q.; Dong, H.; Pan, W. Lyapunov-based reinforcement learning for decentralized multi-agent control. In *International Conference on Distributed Artificial Intelligence*; Springer: Cham, Switzerland, 2020; pp. 55–68.
45. Lockhart, E.; Lanctot, M.; Pérolat, J.; Lespiau, J.B.; Morrill, D.; Timbers, F.; Tuyls, K. Computing approximate equilibria in sequential adversarial games by exploitability descent. *arXiv* **2019**, arXiv:1903.05614.
46. Timbers, F.; Lockhart, E.; Schmid, M.; Lanctot, M.; Bowling, M. Approximate exploitability: Learning a best response in large games. *arXiv* **2020**, arXiv:2004.09677.
47. Tang, J.; Paster, K.; Abbeel, P. Equilibrium Finding via Asymmetric Self-Play Reinforcement Learning. Available online: https://drive.google.com/file/d/0B_utB5Y8Y6D5eWJ4Vk1hSDZzZDhwMFlDYjlRVGpmWGlZVWJB/view (accessed on 13 April 2021).
48. Puterman, M.L. Markov decision processes. In *Handbooks in Operations Research and Management Science*; Elsevier: Amsterdam, The Netherlands, 1990; Volume 2, pp. 331–434.
49. Nachum, O.; Dai, B. Reinforcement learning via Fenchel-Rockafellar duality. *arXiv* **2020**, arXiv:2001.01866.
50. Belousov, B.; Peters, J. f-Divergence constrained policy improvement. *arXiv* **2017**, arXiv:1801.00056.
51. Nachum, O.; Dai, B.; Kostrikov, I.; Chow, Y.; Li, L.; Schuurmans, D. Algaedice: Policy gradient from arbitrary experience. *arXiv* **2019**, arXiv:1912.02074.
52. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv* **2018**, arXiv:1801.01290.
53. Belousov, B.; Peters, J. Entropic regularization of markov decision processes. *Entropy* **2019**, *21*, 674. [CrossRef] [PubMed]
54. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
55. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
56. Hunter, J.D. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. [CrossRef]
57. Yu, L.; Song, J.; Ermon, S. Multi-agent adversarial inverse reinforcement learning. *arXiv* **2019**, arXiv:1907.13220.
58. Riasanow, T.; Flötgen, R.J.; Greineder, M.; Möslein, D.; Böhm, M.; Krcmar, H. Co-evolution in business ecosystems: Findings from literature. In Proceedings of the 40 Years EMISA 2019, Tutzing, Germany, 15–17 May 2019.