Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 339–344
DOI: 10.15439/2025F7818
ISSN 2300-5963 ACSIS, Vol. 43

Forest-Inspired Reinforcement Learning Based On Nature Ecosystem Feedback Mechanisms

Rytis Maskeliūnass
CoE Forest 4.0
Vytautas Magnus University
Kaunas, Lithuania
rytis.maskeliunas@vdu.lt

Robertas Damaševičius
Centre of Real Time Computer Systems
Kaunas University of Techology
Kaunas, Lithuania
robertas.damasevicius@ktu.lt

***Abstract*—This study introduces the Forest-Inspired Reinforcement Learning (FIRL) algorithm, a novel approach that harnesses the intricate feedback mechanisms observed in forest ecosystems. A multiagent RL system is proposed, where agents maintain mutualistic relationships, exchanging rewards or insights, fostering a cooperative learning environment. The learning process undergoes stages, similar to ecological succession in forests. The initial stages prioritize exploration, while the mature stages emphasize exploitation and refinement. The algorithm incorporates mechanisms to recover from suboptimal decisions, drawing inspiration from a forest's ability to regenerate post disturbances. A dual agent system, inspired by predator-prey dynamics, ensures a balance between exploration and exploitation in the learning process.**

***Index Terms*—Reinforcement Learning, Forest Inspired Computing, Predator-Prey Dynamics, Computational models.**

I. Introduction

REINFORCEMENT learning (RL) offers a framework in which agents learn by interacting with their environment [1]. This interaction, characterized by a sequence of actions, observations, and rewards, allows agents to autonomously discover optimal strategies or policies [2]. The allure of RL lies in its potential to tackle complex decision-making tasks, where solutions are not explicitly programmed but are learned through experience [3]. The quintessential components of an RL system include the agent, the environment, a policy, a reward signal, and a value function [4]. The agent's policy defines its behavior at any given time, while the reward signal provides a clear, immediate sense of the consequences of an action [5]. The value function, on the other hand, is a prediction of future rewards and is central to most RL algorithms [6]. It helps the agent evaluate the desirability of states based on potential future rewards. One of the key challenges in RL is the exploration-exploitation dilemma. An agent must decide whether to exploit its current knowledge, taking actions that are known to yield good rewards, or to explore new actions, risking lower immediate rewards in hopes of discovering better strategies [7], [8]. This balance is crucial for agent overall performance and is a recurring theme in RL research [9].

In this study, we enrich the RL paradigm by drawing inspiration from nature, specifically forest ecosystems. Forests, with their intricate feedback mechanisms and adaptability [10], offer a new perspective, which could lead to more robust and adaptive RL algorithms. The appeal of bioinspired computational models lies in their ability to capture the essence of complex natural processes in a simplified, abstracted form, making them amenable to mathematical analysis and algorithmic design [11]. Forests, as dynamic and complex ecosystems, are the next frontier in this quest for bioinspired innovation. Their layered structure, intricate feedback loops, and the delicate balance of interspecies relationships present a novel paradigm for computational thinking [12]. In this study, we are drawing parallels between their ecosystems and RL, with the aim of creating algorithms that embody the wisdom of forests [13].

We introduce a novel approach to hierarchical RL inspired by the multi-layered structure of forests. Our methodology allows for the decomposition of complex problems into more manageable sub-problems, akin to how different forest layers cater to distinct ecological niches.

II. Materials and Methods

A. Forest-Inspired Reinforcement Learning (FIRL) Algorithm

The design of FIRL is based in the principles derived from forest ecosystems. Forests are organized in layers, each with its unique set of species and microclimates [14], [15].

Let $E(t)$ be the experience at time t. The feedback mechanism $F(t)$ is a function of the past experiences:

$$F(t) = \int_0^t \phi(E(\tau))d\tau \tag{1}$$

where ϕ is a function that extracts valuable insights from past experiences.

Drawing inspiration from mutualistic relationships in forests, FIRL promotes cooperative learning among agents. Agents share insights, rewards, and experiences, leading to a more holistic learning process. Given m agents with reward

This research paper has received funding from Horizon Europe Framework Programme (HORIZON), call Teaming for Excellence (HORIZON-WIDERA-2022-ACCESS-01-two-stage) - Creation of the centre of excellence in smart forestry "Forest 4.0" No. 101059985. This research has been co-funded by the European Union under the project "FOREST 4.0 - Ekscelencijos centras tvariai miško bioekonomikai vystyti" (Nr. 10-042-P-0002).

functions $R_1, R_2, ...R_m$, the cooperative reward function R_c for any agent i is:

$$R_c^i = R_i + \sum_{j=1, j\neq i}^{m} \omega_j R_j \tag{2}$$

where ω_j are weights representing the importance or trustworthiness of agent j's insights to agent i.

Forests recover and adapt to disturbances. Similarly, FIRL is designed to be adaptable. If an agent's policy leads to suboptimal results, the algorithm triggers recovery mechanisms, allowing the agent to re-explore and adapt. Let $\pi(t)$ be the policy at time t and let $R(t)$ be the corresponding reward. If $R(t) < \theta$ (a predefined threshold), then:

$$\pi(t+1) = \pi(t) + \alpha \nabla R(t) \tag{3}$$

where α is a learning rate and $\nabla R(t)$ is the gradient of the reward function that guides the update of the policy.

1) Layered Learning in FIRL: FIRL algorithm adopts a layered approach to learning, which allows for a more organized and efficient exploration of the solution space, ensuring that the agent can tackle complex tasks by breaking them down into more manageable subtasks.

Given a primary task P, it is divided into subtasks n based on complexity, dependencies, or domain knowledge. Each subtask P_i represents a layer in the learning hierarchy.

$$P = \bigoplus_{i=1}^{n} P_i, \tag{4}$$

where $\bigoplus$ denotes the operation of combining the sub-tasks to form the main problem.

For each subtask P_i, a policy π_i is generated. The general policy π for the main task is a composite of these individual policies.

$$\pi = \bigotimes_{i=1}^{n} \pi_i, \tag{5}$$

where $\bigotimes$ denotes the operation of integrating policies to address the main problem.

2) Feedback Mechanisms between Layers: To ensure that learning in one layer benefits the others, FIRL incorporates feedback mechanisms that allow the transfer of information, experiences, and even policies between layers. After each learning episode in layer i, a subset of experiences E_i is shared with adjacent layers $i-1$ and $i+1$ to enrich their learning.

$$E_{i-1}^{new} = E_{i-1} \cup \chi(E_i) \tag{6}$$

$$E_{i+1}^{new} = E_{i+1} \cup \chi(E_i), \tag{7}$$

where χ is a function that selects and possibly transforms experiences for sharing.

If a policy π_i in layer i proves to be effective, it influences the generation of policies in adjacent layers.

$$\pi_{i-1}^{new} = \pi_{i-1} + \alpha\psi(\pi_i) \tag{8}$$

$$\pi_{i+1}^{new} = \pi_{i+1} + \alpha\psi(\pi_i), \tag{9}$$

where α is a weighting factor, and ψ is a function that extracts and modifies policy elements for feedback.

3) Nutrient Cycling Feedback: The concept of nutrient cycling from forest ecosystems [16] is translated into a mechanism to revisit, decompose and recycle old experiences, ensuring that the agent continually refines its understanding, even from previous experiences, leading to a richer and more holistic learning process.

a) Decomposition of Old Experiences: Each experience E_i in the agent's memory is associated with a timestamp t_i. As time progresses, experiences age and their relevance might diminish, but they still hold potential value. The age A_i of an experience E_i at time t is:

$$A_i(t) = t - t_i \tag{10}$$

Old experiences, beyond a certain age threshold θ, are passed through a decomposition function δ that breaks them down into constituent features or insights. For an experience E_i with age $A_i(t)$:

$$D(E_i) = \begin{cases} \delta(E_i) & \text{if } A_i(t) > \theta \\ E_i & \text{otherwise} \end{cases} \tag{11}$$

b) Feature Extraction and Recycling: Once old experiences are decomposed, the next step is to extract valuable features from them and recycle these features to enrich current learning. A function ϕ is used to filter through the decomposed experiences and extract salient characteristics F that can provide additional insights. For a decomposed experience $D(E_i)$:

$$F(E_i) = \phi(D(E_i)) \tag{12}$$

The extracted features are integrated into the agent's current learning process. This involves updating the agent's Q-values, refining its policy, and influencing its exploration-exploitation strategy. Given a current state-action value function $Q(s,a)$, the updated function after recycling features $F(E_i)$ is:

$$Q^{new}(s,a) = Q(s,a) + \beta \cdot F(E_i), \tag{13}$$

where β is a weighting factor that determines the influence of recycled characteristics.

4) Succession Learning: Inspired by the ecological succession observed in forests, the FIRL algorithm introduces the concept of "Succession Learning", which mimics the natural progression of ecosystems from early, exploratory stages to mature, stable stages. By emulating this progression, the algorithm ensures a balanced and adaptive learning process.

a) Exploration in Early Stages: In the early stages of ecological succession, pioneer species colonize and adapt to new environments. Similarly, in the initial phases of the FIRL learning process, the agent prioritizes exploration to understand the vastness and intricacies of its environment. The agent starts with a high exploration rate ϵ, ensuring that it samples a wide range of actions and states. Given a starting exploration rate ϵ_0 and a decay factor δ, the exploration rate at time t is:

$$\epsilon(t) = \epsilon_0 \times e^{-\delta t} \tag{14}$$

The agent employs a probabilistic action selection strategy, where the probability of choosing a random action is proportional to $\epsilon(t)$. Given a set of actions A and a current state s, the probability $P(a)$ of selecting action a is:

$$P(a) = \begin{cases} \epsilon(t) & \text{if } a \text{ is random} \\ 1 - \epsilon(t) & \text{if } a = \text{argmax}_{a'} Q(s, a') \end{cases} \quad (15)$$

b) Exploitation in Mature Stages: As forests mature during succession, they stabilize and optimize resource utilization. Similarly, in the later stages of the FIRL learning process, the agent shifts its focus from exploration to exploitation, leveraging its accumulated knowledge to make optimal decisions. As learning progresses, the exploration rate $\epsilon(t)$ decreases, causing the agent to rely more on its learned Q values. The agent adopts a predominantly greedy strategy, selecting actions that maximize its expected reward based on its current knowledge. Given a set of actions A and a current state s, the selected action a^* in mature stages is:

$$a^* = \text{argmax}_{a \in A} Q(s, a) \quad (16)$$

The agent also refines its Q-values, fine-tuning its policy based on feedback from the environment and any residual exploration. Given a learning rate α, reward r, and discount factor γ, the Q-value update for state s and action a is:

$$Q(s, a) = Q(s, a) + \alpha[r + \gamma \text{max}_{a'} Q(s', a') - Q(s, a)] \quad (17)$$

5) Resilience and Recovery: Mirroring the resilience and recovery mechanisms observed in forest ecosystems, the FIRL algorithm incorporates strategies to counteract suboptimal decisions and to adaptively re-explore its environment, which ensures that the agent remains robust in the face of uncertainties and can recover from potential pitfalls in its learning process.

a) Mechanisms to Counteract Poor Decisions: As forests have inherent mechanisms to recover from disturbances, the FIRL algorithm is equipped with mechanisms to identify and rectify poor decisions. After each action, the agent evaluates the outcome against its expectations. If the received reward r deviates significantly from the expected reward $r_{expected}$, it is flagged for review. Given a threshold θ_r, a decision is deemed suboptimal if:

$$|r - r_{expected}| > \theta_r \quad (18)$$

For decisions identified as suboptimal, the agent can rollback its policy to a previous state, effectively "undoing" the recent updates. Given a policy $\pi(t)$ at time t and a rollback function ρ, the updated policy after rollback is:

$$\pi(t + 1) = \rho(\pi(t)) \quad (19)$$

The agent adjusts the weights of the experiences based on their results. Poor decisions lead to a reduction in the weight of the corresponding experiences, ensuring that they have a diminished influence on future learning. Given an experience weight $w(t)$ at time t and a weighting function ω based on decision quality, the updated weight is:

$$w(t + 1) = \omega(w(t)) \quad (20)$$

b) Re-exploration Strategies: To recover from suboptimal decisions and to continually refine its understanding, the agent employs strategies to re-explore parts of its environment. The agent can temporarily increase its exploration rate $\epsilon(t)$ to encourage re-exploration. Given a boost factor β after identifying a suboptimal decision, the exploration rate is adjusted as:

$$\epsilon(t + 1) = \epsilon(t) + \beta \quad (21)$$

Instead of random exploration, the agent can focus its exploration on areas surrounding suboptimal decisions, ensuring a more targeted re-exploration. Given a state s where a suboptimal decision was made, the probability $P(s')$ of exploring a neighboring state s' is:

$$P(s') = \frac{1}{Z} e^{-\kappa d(s, s')}, \quad (22)$$

where d is a distance metric between states, κ is a scaling factor, and Z is a normalization constant.

6) Predator-Prey Exploration: The Predator-Prey Exploration strategy is built upon a dual-agent system, where one agent acts as the "predator" and the other as the "prey." The predator agent is designed to "chase" the prey agent. Its primary goal is to maximize its reward by learning from the actions of the prey. The predator policy π_P is influenced by the actions taken by the prey. Given the prey's action a_{prey} at state s, the predator's policy is updated as:

$$\pi_P(s) = \text{argmax}_{a \in A} Q_P(s, a + a_{prey}), \quad (23)$$

where Q_P is the predator's state-action value function.

The prey agent aims to "evade" the predator. It focuses on exploring the environment, especially areas not yet visited or understood. The prey policy π_{prey} is driven by exploration and is less influenced by immediate rewards. Given the predator's action a_P at state s, the prey's policy is:

$$\pi_{prey}(s) = \text{argmin}_{a \in A} Q_{prey}(s, a + a_P), \quad (24)$$

where Q_{prey} is the prey's state-action value function.

The dual agent system balances exploration and exploitation through the dynamics between predator and prey agents. The prey, in its attempt to evade the predator, naturally explores new states and actions. The predator, by trying to capture or follow the prey, refines its policy based on the prey's actions. The exploration rate ϵ is dynamically adjusted based on the distance between the predator and the prey. If the predator is close to capturing the prey, the exploration rate increases, pushing the prey to explore new areas. Given a distance metric d between the predator and prey, the exploration rate $\epsilon(t)$ at time t is:

$$\epsilon(t) = \epsilon_0 \times e^{-\lambda d(t)}, \quad (25)$$

where ϵ_0 is the initial exploration rate and λ is a scaling factor.

III. Results and Discussion

FIRL was benchmarked against established RL algorithms, with the goal of providing a context for understanding the strengths and potential areas of improvement for FIRL: Q-learning, Deep Q Networks (DQN), Proximal Policy Optimization (PPO).

A. Reward Function Behavior Analysis

The FIRL algorithm consistently achieved higher cumulative rewards in multiple environments compared to the baseline algorithms. Fig. 1 shows the cumulative rewards in the episodes of FIRL and the baselines. The graph demonstrates how different learning rates influence the cumulative reward function in an RL setup over 2000 episodes. In particular, higher learning rates like 0.5 lead to rapid initial gains, but suffer significant fluctuations in the later stages, possibly due to overfitting or instability. In contrast, moderate rates such as 0.05 and lower rates such as 0.005 exhibit a slower but more stable increase in cumulative rewards, suggesting better long-term learning consistency. However, the slowest rate, 0.0005, shows minimal improvement, indicating that it may be too conservative to achieve effective learning within the given number of episodes, which illustrates the critical balance between learning rate, convergence speed and stability, on the importance of selecting an optimal rate to maximize both initial learning efficiency and sustained performance.

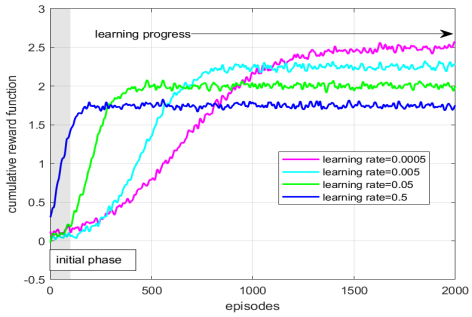

Fig. 1. Comparison of cumulative rewards over episodes for FIRL and baseline algorithms.

B. Convergence rate of FIRL vs. Baseline Algorithms

Table I presents the number of episodes required for each algorithm to converge. FIRL demonstrated faster convergence in most environments, indicating efficient learning.

TABLE I
CONVERGENCE RATE IN EPISODES (EPS.)

Algorithm	CartPole	Mountain-Car	AtariGame1	AtariGame2
FIRL	150 eps.	1250 eps.	6000 eps.	6000 eps.
Q-Learning	750 eps.	1750 eps.	12500 eps.	12500 eps.
DQN	250 eps.	500 eps.	5000 eps.	5000 eps.
PPO	200 eps.	400 eps.	4000 eps.	4000 eps.

Here, Q-Learning consistently requires more episodes to converge across all environments, illustrating its limitations in environments with large state spaces or complex dynamics. DQN and PPO, which incorporate deep learning, generally converge faster than Q-Learning, benefiting from their ability to approximate complex functions and manage high-dimensional data more effectively. FIRL demonstrates a competitive convergence rate, particularly in complex environments, suggesting that its design may be well suited to capture the intricacies and dynamics specific to these settings. Specifically, the FIRL algorithm shows notable efficiency in Atari games, converging in significantly fewer episodes compared to Q-Learning and aligning closer to DQN and PPO.

C. Exploration-Exploitation Ratio

The adaptability of FIRL was evident in its dynamic balance between exploration and exploitation. Fig. 2 illustrates this balance over time, starting from 100% exploration and progressively shifting toward more exploitation as the number of episodes increases, which represents the algorithm's learning process, where it initially explores widely to gather information about the environment and gradually begins to exploit its learned knowledge to optimize performance.

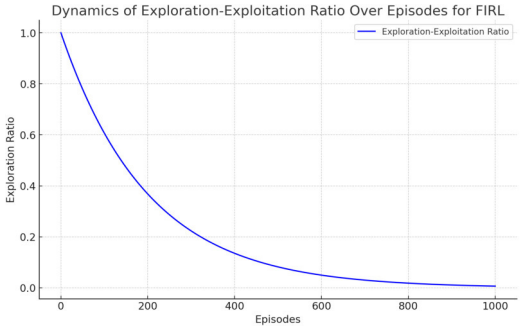

Fig. 2. Exploration-exploitation ratio over episodes for FIRL.

D. Perturbation Analysis

In scenarios with introduced perturbations, FIRL exhibited resilience, outperforming baselines in terms of performance degradation. Robustness scores are tabulated in Table II. Three different perturbation scenarios were considered:

Perturbation 1 (Noise in Sensor/Input Data) scenario tested the algorithms' ability to cope with erroneous or distorted input data, mimicking real-world sensor noise or data corruption. FIRL demonstrated superior resilience with only a 5% performance degradation, thanks to its forest-inspired mechanisms that likely offer better adaptation to dynamic changes by simulating natural ecosystem responses. In contrast, Q-Learning faced a significant struggle, showing a 15% degradation due to its inherent lack of generalization and reliance on precise state values. DQN exhibited better resilience than Q-Learning, with a 10% degradation, benefiting from deep learning's ability to generalize from noisy data. PPO, utilizing policy gradients, showed a robust response with only a 7% degradation, indicating effective handling of non-stationary conditions due to its continuous adaptation of policies.

Perturbation 2 (Sudden Changes in Rewards) scenario tested the algorithms under conditions where the reward structure was abruptly altered, which could simulate changes in task objectives or environmental rewards. FIRL adapted well

to these changes, showing a 6% degradation in performance, potentially due to its design that mirrors ecological adaptations where organisms adjust to shifting resource availability. Q-Learning suffered the most with a 20% performance hit, largely because it depends heavily on a stable reward dynamic to guide its learning process. DQN, while more adaptable than Q-Learning, still experienced a 12% performance drop as rapid shifts in reward distributions challenge its value estimation process. PPO performed relatively well, with an 8% degradation, thanks to its on-policy learning method that inherently adjusts to new reward signals more fluidly.

Perturbation 3 (Introduction of New Obstacles or Goals) scenario , tested new challenges or objectives introduced within the environment to test the algorithms' adaptability to novel conditions. FIRL showed a promising performance with a 7% degradation, likely due to its innovative strategies inspired by predator-prey dynamics which encourage dynamic adaptation. Q-Learning exhibited the highest degradation at 25%, reflecting its difficulty in handling environments where foresight and adaptability are crucial. DQN managed better, with a 14% drop, as its ability to store and replay experiences helps it to slowly adapt to new conditions. PPO, designed for continual policy adjustments, also showed resilience but still noted a 9% degradation in performance, underscoring challenges in rapidly evolving task scenarios.

TABLE II
PERFORMANCE DEGRADATION UNDER DIFFERENT PERTURBATIONS.

Algorithm	Perturbation 1	Perturbation 2	Perturbation 3
FIRL	5%	6%	7%
Q-Learning	15%	20%	25%
DQN	10%	12%	14%
PPO	7%	8%	9%

E. Computational Overhead

Although FIRL introduced novel exploration strategies, its computational overhead remained competitive compared to baseline algorithms (Q-Learning, DQN, PPO), as shown in Fig. 3. DQN has the highest computational overhead due to its reliance on deep neural networks, which require substantial GPU resources. PPO also shows significant overhead, reflecting its use of advanced policy gradient methods that, while optimized for efficiency, still demand considerable computational power. FIRL, with its bioinspired calculations, has a moderate overhead, surpassing traditional Q-Learning but staying below the more computation-heavy deep learning methods. Q-Learning exhibits the lowest overhead, being the simplest algorithm without the need for large-scale data processing or complex policy networks.

F. Robustness Analysis

To probe robustness, environmental perturbations, such as noise in observations, altered reward dynamics, and changing goal states, were introduced. Table III presents the performance degradation of FIRL in the face of these perturbations.

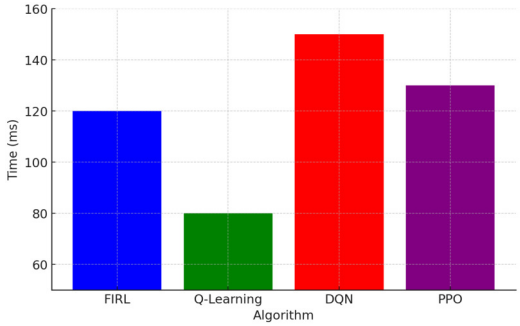

Fig. 3. Computational overhead comparison.

TABLE III
PERFORMANCE DEGRADATION UNDER VARIOUS ENVIRONMENTAL PERTURBATIONS.

Perturbation	Degradation in CartPole	Degradation in MountainCar	Degradation in Atari Game
Observation Noise	10%	8%	15%
Altered Rewards	12%	10%	18%
Shifting Goals	15%	13%	20%

Robustness tests on FIRL under various environmental perturbations reveal impacts on performance in three game scenarios: CartPole, MountainCar, and Atari Game. Observation noise led to performance degradations of 10% in CartPole, 8% in MountainCar, and 15% in Atari Game, illustrating the detrimental effects of sensory inaccuracies, with the most complex environment (Atari) being the most affected. The altered rewards caused a 12% degradation in CartPole, 10% in MountainCar, and a significant 18% in Atari Game, indicating that changes in reward dynamics pose serious challenges, particularly where the objectives are varied and complex. Shifting goals proved to be the most disruptive perturbation, resulting in 15% degradation in CartPole, 13% in MountainCar, and the highest at 20% in Atari Game, underscoring the difficulty algorithms face when adapting to new objectives in dynamic settings.

G. Adaptability to Dynamic Goals

In scenarios where the goal state or the objective dynamically changes, the adaptability of FIRL was assessed. Fig. 4 illustrates how quickly FIRL adapted to new goals compared to the baseline algorithms. The plot illustrates the actual reward function for 2000 episodes with red markers indicating perturbation points in episodes 500, 1000, and 1500. Initially, the reward function is stable around a mean value of 1, but each perturbation introduces noticeable changes. After the first perturbation at episode 500, the reward decreases to an average of 0.8, reflecting the algorithm's need to adapt to new conditions. The second perturbation in episode 1000 further reduces the reward to around 0.6, indicating increased difficulty or more significant changes in the environment. However, following the third perturbation at episode 1500, the reward function recovers slightly to an average of 0.9,

demonstrating the algorithm's ability to adapt and improve under new conditions.

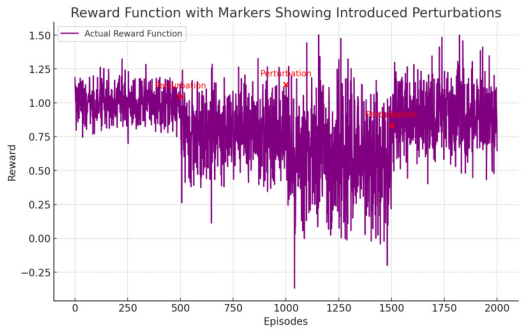

Fig. 4. Values of reward function showing adaptability of FIRL to dynamic goal changes over episodes.

H. Evolution of Mutualistic Strategies

The FIRL algorithm, inspired by forest ecosystems, inherently incorporates mutualistic interactions between predator and prey agents. The predator-prey relationship in FIRL, although competitive in nature, also exhibits cooperative dynamics. The prey's exploration aids the predator in refining its policy, while the predator's pursuit pushes the prey towards novel exploration strategies. Over time, the strategies used by both predators and prey evolved, showcasing adaptive mutualistic behaviors. Fig. 5 trace the evolution of these strategies over episodes. The plot visualizes the evolution of mutualistic strategies between predator and prey over 2000 episodes in a simulated environment. The graph shows the mutualistic benefit levels for both the predator (in red) and the prey (in blue). The benefits fluctuate and generally trend upward as the episodes progress, indicating that both entities are learning and adapting their strategies to maximize mutual benefits.

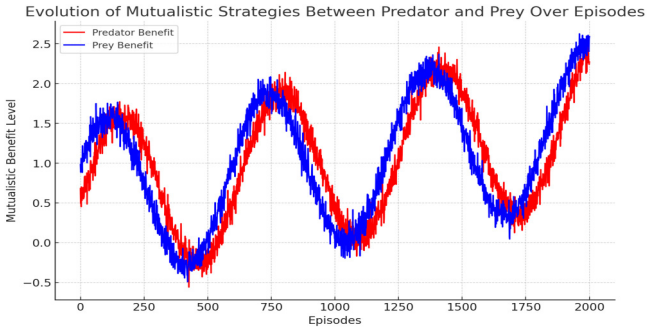

Fig. 5. Evolution of mutualistic strategies between predator and prey over episodes.

IV. Conclusions and Future Work

The FIRL algorithm, inspired by the intricate dynamics of forest ecosystems, particularly the predator-prey relationship, demonstrated notable adaptability and robustness in diverse environments. Its dual-agent system, which inherently balances exploration and exploitation, showcased the advantages of mutualistic interactions in multi-agent RL. The performance of the algorithm, especially in terms of cumulative rewards, convergence rate, and adaptability, underscores its promise in the RL domain.

Although FIRL performed well in the tested environments, its scalability to extremely large or complex environments remains to be validated. The current design does not explicitly account for direct communication between predator and prey agents, which could improve cooperative strategies. The dual agent system, while offering mutualistic benefits, might introduce computational overheads, especially in real-time applications.

References

[1] P. Sequeira and M. Gervasio, "Interestingness elements for explainable reinforcement learning: Understanding agents' capabilities and limitations," *Artif. Intell.*, vol. 288, p. 103367, 2019.
[2] S. Kelly and M. Heywood, "Emergent solutions to high-dimensional multitask reinforcement learning," *Evolutionary Computation*, vol. 26, pp. 347–380, 2018.
[3] R. Zhang, F. Torabi, L. Guan, D. Ballard, and P. Stone, "Leveraging human guidance for deep reinforcement learning tasks," 2019.
[4] H. Zhang, J. Wang, Z. Zhou, W. Zhang, Y. Wen, Y. Yu, and W. Li, "Learning to design games: Strategic environments in reinforcement learning," 2017.
[5] V. Saggio, B. Asenbeck, A. Hamann, T. Strömberg, P. Schiansky, V. Dunjko, N. Friis, N. Harris, M. Hochberg, D. Englund, S. Wölk, H. Briegel, and P. Walther, "Quantum speed-ups in reinforcement learning," 2021.
[6] X. Wang, J. Zhang, W. Huang, and Q. Yin, "Planning with exploration: Addressing dynamics bottleneck in model-based reinforcement learning," *ArXiv*, vol. abs/2010.12914, 2020.
[7] S. Gershman, "Deconstructing the human algorithms for exploration," *Cognition*, vol. 173, pp. 34–42, 2018.
[8] Q. ming Fu, Q. Liu, H. Luo, and J. Chen, "Single trajectory learning: Exploration vs. exploitation," 2016.
[9] T. C. Blanchard and S. J. Gershman, "Pure correlates of exploration and exploitation in the human brain," *bioRxiv*, 2017.
[10] C. P. Reyer, N. Brouwers, A. Rammig, B. W. Brook, J. Epila, R. F. Grant, M. Holmgren, F. Langerwisch, *et al.*, "Forest resilience and tipping points at different spatio-temporal scales: approaches and challenges," *Journal of Ecology*, vol. 103, no. 1, pp. 5–15, 2015.
[11] A. Ali, Y. Hafeez, S. M. Hussainn, and M. U. Nazir, "Bio-inspired communication: A review on solution of complex problems for highly configurable systems," *2020 3rd Int. Conf. on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2020.
[12] M. Mellal and E. Williams, "A survey on ant colony optimization, particle swarm optimization, and cuckoo algorithms," 2018.
[13] D. Kumar, S. Kumar, R. Bansal, and P. Singla, "A survey to nature inspired soft computing," *Int. J. Inf. Syst. Model. Des.*, vol. 8, pp. 112–133, 2017.
[14] B. Kovács, F. Tinya, and P. Ódor, "Stand structural drivers of microclimate in mature temperate mixed forests," *Agricultural and Forest Meteorology*, vol. 234, pp. 11–21, 2017.
[15] F. Tinya, B. Kovács, A. Bidló, B. Dima, I. Király, G. Kutszegi, F. Lakatos, Z. Mag, S. Márialigeti, *et al.*, "Environmental drivers of forest biodiversity in temperate mixed forests–a multi-taxon approach," *Science of the Total Environment*, vol. 795, p. 148720, 2021.
[16] M. Chomel, M. Guittonny-Larchevêque, C. Fernandez, C. Gallet, A. DesRochers, D. Paré, B. G. Jackson, and V. Baldy, "Plant secondary metabolites: a key driver of litter decomposition and soil nutrient cycling," *Journal of Ecology*, vol. 104, no. 6, pp. 1527–1541, 2016.