A Coevolutionary Approach to Deep Multi-Agent Reinforcement Learning

Daan Klijn Vrije Universiteit Amsterdam d.s.klijn@student.vu.nl A.E. Eiben Vrije Universiteit Amsterdam a.e.eiben@vu.nl

ABSTRACT

Traditionally, Deep Artificial Neural Networks (DNN's) are trained through gradient descent. Recent research shows that Deep Neuroevolution (DNE) is also capable of evolving multi-million-parameter DNN's, which proved to be particularly useful in the field of Reinforcement Learning (RL). This is mainly due to its excellent scalability and simplicity compared to the traditional MDP-based RL methods. So far, DNE has only been applied to complex singleagent problems. As evolutionary methods are a natural choice for multi-agent problems, the question arises whether DNE can also be applied in a complex multi-agent setting. In this paper, we describe and validate a new approach based on coevolution. To validate our approach, we benchmark two Deep coevolutionary Algorithms on a range of multi-agent Atari games and compare our results against the results of Ape-X DQN. Our results show that these Deep coevolutionary algorithms (1) can be successfully trained to play various games, (2) outperform Ape-X DQN in some of them, and therefore (3) show that coevolution can be a viable approach to solving complex multi-agent decision-making problems.

CCS CONCEPTS

• Computer systems organization \rightarrow Neural Networks; • Theory of computation \rightarrow Evolutionary algorithms.

KEYWORDS

Deep Neuroevolution, Coevolution, Evolution Strategies, Genetic Algorithm, Multi-agent Reinforcement Learning

ACM Reference Format:

Daan Klijn and A.E. Eiben. 2021. A Coevolutionary Approach to Deep Multi-Agent Reinforcement Learning. In *Proceedings of the Genetic and Evolutionary Computation Conference 2021 (GECCO '21)*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459576

1 INTRODUCTION

Recent developments have caused Evolutionary Reinforcement Learning to gain momentum. Salimans et al. [4] showed that the combination of Evolution Strategies (ES) and Neuroevolution is a perfect alternative to the MDP-based approaches that are typically used in RL. Their ES was capable of evolving DNN's that achieved

GECCO '21, July 10-14, 2021, Lille, France

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8351-6/21/07...\$15.00 https://doi.org/10.1145/3449726.3459576 state-of-the-art performance on some of the benchmarks while maintaining near-linear scalability. It also proved to be invariant to action frequency and is resistant to delayed rewards and long horizons. Such et al. [5] took the ES as an inspiration and showed that a similar thing is possible using a simple Genetic Algorithm (GA). While ES relies on approximations of the gradients, GA's are truly gradient-free. It is therefore remarkable to see that this approach is able to train networks with millions of parameters. These advances demonstrated the far-reaching capabilities of DNE and have laid the foundations for new breakthroughs in RL.

While most RL problems consist of an agent solely interacting with an environment, a subfield called Multi-Agent Reinforcement Learning (MARL) encompasses the problems where multiple agents are present. The goal of MARL is to develop agents that can successfully cooperate or compete with other agents. Similarly, Evolutionary Computation has a class of algorithms that is called Coevolutionary Algorithms. Compared to the single-agent flavor of these problems, multi-agent problems tend to be more challenging as they include another uncertainty, namely, the other agents. Due to this, the measured performance of an agent is subjective, as it depends on the other agents' performance. While MARL has been proven to be an effective solution for complex Multi-Agent problems, coevolution has never been scaled to high-complexity problems.

The main goal of this paper is to investigate whether the combination of DNE and Coevolution could open up a new door in the field of MARL. To this end, we develop two Deep coevolutionary algorithms, one based on the ES from Salimans et al. [4] and another based on the GA from Such et al. [5]. To assess our approach's viability, we train these algorithms on eight different games from the PettingZoo benchmark [7] and compare the evolved agents to the ones delivered by Ape-X DQN.

2 METHODOLOGY

In this work, we combine DNE with a coevolutionary approach. We apply this approach to two successful DNE algorithms; the ES from Salimans et al. [4] and the GA from Such et al. [5]. To prove that even a simple coevolutionary setup works in combination with DNE, we only transform the original two algorithms to their coevolutionary counterparts and limit the number of features we add to them. Below we shortly describe the specifications of the forged coevolutionary algorithms.

2.1 Coevolutionary Evolution Strategies

Salimans et al. [4] proposed an ES that is capable of evolving DNN's with millions of parameters and showed that it could solve various complex Reinforcement Learning problems. The ES proposed here is more scalable a variant of Natural Evolution Strategies [8]. We

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

altered this ES in such a way that it can be applied to multi-agent problems. While the original ES evaluates each of the mutations $\theta + \epsilon$ using the fitness function, we now need to consider that we also need to provide individuals to evaluate against. We have tried various approaches and discovered that evaluating each individual against the *k* parents of the previous populations (i.e., θ_t to θ_{t-k}) resulted in the most stable convergence. Sometimes, using only the previous parent was enough. The *k* fitnesses are then averaged for each individual and used in combination with the update rule of the original ES.

2.2 Coevolutionary Genetic Algorithm

Based on the work of Salimans et al. [4], Such et al. [5] showed that a GA is also capable of evolving similar DNN's. The GA they proposed consists of a single population, only uses the mutation operator and performs fitness-based truncation. Our coevolutionary variant is based on this GA and will stay as close to the original design as possible. Again, we replace the original fitness evaluation with an evaluator-based fitness evaluation. For the GA, we will use a Hall of Fame (HoF) that stores the best individuals of previous generations. To evaluate individuals, we select the last k elites from the HoF. This makes sure that the fitness approximation is reliable and that historical traits are not lost. Note that for both the Coevolutionary ES (CoES) and the Coevolutionary GA (CoGA), k should not be too high as that will increase the number of evaluations per individual, which can increase runtime if the fitness function is expensive to compute.

3 EXPERIMENTS

Numerous RL works benchmark algorithms using the Atari games, as they require the agent to perform a range of complex tasks that can be compared to the difficulties of some real-life tasks. While Salimans et al. [4] and Such et al. [5] use single-player Atari games to benchmark their algorithms, we will use a diverse set of multiplayer Atari games from the PettingZoo benchmark [7]. Below a short overview is given of our experimental setup. Next to that, the code that was used for these experiments can be found on GitHub¹

- **Preprocessing** We process the game frames as proposed by Mnih et al. [2]. Besides that, we also add a form of agent indication [1] as it is essential for each agent to know whether it is player one or player two.
- Network architecture We use a DNN that is similar to the large DQN that was used by Mnih et al. [2]. Furthermore, we use virtual batch normalization (VBN) [3] to ensure that the evolved agents are diverse enough. Without VBN, many evolved agents were not diverse and always performed the same actions, regardless of the input.
- **Hyperparameters** In comparison to the original ES and GA, we used a slightly higher noise standard deviation and a smaller population size. We were able to get the best results using a population size of 200 and standard deviations of 0.005 (GA) and 0.05 (ES). Furthermore, we found that using k = 3 for the GA and k = 1 for the ES was enough to ensure accurate evaluations of individuals.

We trained both the CoES and the CoGA on eight different games, whereas we start learning from scratch for each of the games. After 200 million frames, we evaluate the trained agents by playing 50 games against a random agent. The average performance against the random agent of the ES, GA and Ape-X[7] can be found in Table 1. Videos of selected gameplays of both the CoES and CoGA are available². Furthermore, a more comprehensive set of results combined with a more thorough analysis can be found in the full version of this paper. ³

.	Random	Ape-X DQN	CoES	CoGA
Game frames	-	80M	200M	200M
Training frames	-	20M	50M	50M
Generations	-	-	500	160
Wall time (96 cores)	-	-	6h	6h
Basketball Pong	0.0	~1.0	0.0	2.1
Boxing	0.0	80.0	0.7	1.0
Combat Plane	0.0	-2.0	2.5	1.8
Combat Tank	0.0	0.3	0.1	0.4
Joust	3951.0	3846.0	5362.2	4910.0
Pong	0.0	20.5	17.6	15.8
Space war	0.0	1.1	1.8	0.8
Tennis	-3.6	22.5	6.9	-20.2

Table 1: Performance of Ape-X, CoES and CoGA against a Random Policy for various games. The Ape-X results are derived from [6] and are based on the last 20 evaluations.

Our results demonstrate that the coevolutionary approach can train agents that can play most of the games to some extent while even outperforming Ape-X DQN in several of them. We hypothesize that this approach's effectiveness might be due to the EA's being more resistant to the non-stationarity that is introduced by the multi-agent aspect of these problems. Although this work only explores the capabilities of simple coevolutionary algorithms on a small range of benchmarks, we do believe that this might open up a new door in the field of MARL.

REFERENCES

- Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. 2017. Cooperative multi-agent control using deep reinforcement learning. In International Conference on Autonomous Agents and Multiagent Systems. Springer, 66–83.
- [2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.
- [3] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. (2016).
- [4] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. Evolution strategies as a scalable alternative to reinforcement learning. (2017).
- [5] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O Stanley, and Jeff Clune. 2017. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. (2017).
- [6] Justin K Terry and Benjamin Black. 2020. Multiplayer support for the arcade learning environment. (2020).
- [7] Justin K Terry, Benjamin Black, Ananth Hari, Luis Santos, Clemens Dieffendahl, Niall L Williams, Yashas Lokesh, Caroline Horsch, and Praveen Ravi. 2020. PettingZoo: Gym for Multi-Agent Reinforcement Learning. (2020).
- [8] Daan Wierstra, Tom Schaul, Jan Peters, and Juergen Schmidhuber. 2008. Natural evolution strategies. In 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). IEEE, 3381–3387.

¹https://github.com/daanklijn/neurocoevolution/

²https://youtube.com/playlist?list=PLdGO0x90WS8eCCGA1w4jOPLz9ahWSmZEw ³https://arxiv.org/pdf/2104.05610.pdf