PURSUIT POLICIES IN DYNAMIC ENVIRONMENTS

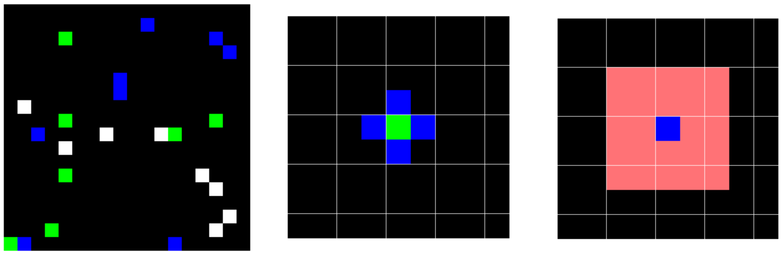
Anonymous authors
Paper under double-blind review

ABSTRACT

Cooperative pursuit is a popular multi-agent reinforcement learning (MARL) game where a team of predators target prey while avoiding obstacles. Previous literature has largely considered the impact of different predator, prey abilities on learning. Here, we investigate the impact of dynamic environments on learning predator pursuit policies from partial observations with deep Q-learning. Interestingly, we find predators are able to learn cooperative pursuit strategies that leverage moving obstacles.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) is a sub-field of reinforcement learning (RL) that focuses on multiple agents learning simultaneously in a shared environment. Within MARL, there many popular environments that are used as a test-bed for reinforcement learning research. Here, our investigation focuses on the cooperative pursuit game, played in the grid-world with two teams, predators and prey, and the ability of predators to learn with different types of obstacles.



(a) An example of a discrete pursuit environment.

(b) Successful capture by four predators (blue).

(c) Local vision (red) of a predator (blue).

Figure 1: Examples of environments rendered from experiments. Obstacles in white, predators in blue, and prey in green. Example predator vision shown with red.

There has been extensive work within MARL that has motivated this work. Cooperative pursuit has been a popular multi-agent game to investigate algorithmic performance [Parsons (1978), Chung et al. (1988), Lin (1992), Omidshafiei et al. (2017), Dibangoye & Buffet (2018), Yu et al. (2020), Vidal et al. (2002), Hollinger et al. (2010), Kehagias et al. (2009), Lowe et al. (2020), Lanctot et al. (2017)]. Moreover, MARL research has been augmented by general advances in RL [van Hasselt et al. (2015), Wang et al. (2016), Sunehag et al. (2017), Yu et al. (2015), Travnik et al. (2018), Schaul et al. (2016), Mnih et al. (2013), Svetlik et al. (2017), OpenAI et al. (2019), Jaderberg et al. (2019), Sukhbaatar et al. (2018), Leibo et al. (2019), Gronauer & Diepold (2022), Pathak et al. (2017), Bertsekas (2012)]. Further, the investigation of dynamic environments is inspired by previous experimental success in learning policies [Gottlieb & Shima (2015), Baker et al. (2020), Wang et al. (2022)]; namely, this work is inspired by the learned behavior seen in the MARL game ”Hide-and-Seek” [Baker et al. (2020)] where agents have non-local, line-of-sight vision. Experiments are programmed independently for performance with *Tensorflow* (Abadi et al. (2015)).

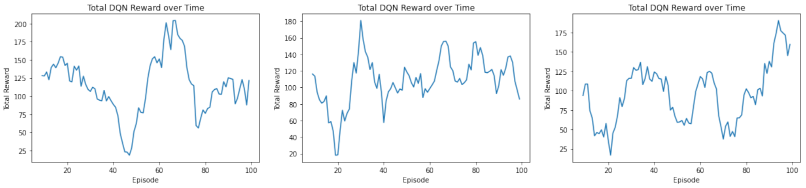
Table 1: Summary of training method used in experiments

Parameter type	Parameter value
Architecture	Deep Q-Network (DQN) Input (5x5), Hidden Layer (16), Output (4)
Episodes	100
Rounds per episode	20
Learning rate (α)	0.0001
Epsilon (ϵ)	0.1
Discount factor (γ)	0.8

2 Methods and Results

Learning of cooperative pursuit policies takes place over numerous, successive games that are simulated in our programmed environment [1]. At the start of each game, each agent is given identical networks to make decisions with. Throughout the game, each agent maintains a memory of its experiences using its two-hop vision and executes an ϵ-greedy algorithm to aid exploration. Games are played for 20 steps, then each agent's game experience replay is sampled to train the base network. The network is trained with a custom reward function that gives a reward of +10 for each agent capturing a prey, +1 for each each agent within two-hops of a prey, and -1 to all other actions. After training, agents memories are cleared and another game begins. Unlike predators, prey execute a random walk to an unoccupied position.

Our experiments' performance are measured by the collective reward of predators each game round, varying the dynamics of obstacles: static, periodic, and random. In the environment, with algorithm parameters, in Table 1. In each setting, a team of five predators move in a 15x15 grid-world environment where there are five obstacles and five prey. Predators, prey and obstacles are spawn in random positions to start. In the static case, obstacles do not move. In the periodic case, obstacles move periodically along the vertical direction. In the random case, obstacles move in a random walk. Our results are presented in Figure 2



(a) Stationary. (b) Periodic movement. (c) Random movement.

Figure 2: Reward yielded by games played with different obstacle movements. Rolling reward averaged over previous 10 games plotted.

Cooperative pursuit policies learned in dynamic environments preform as well, or better, than stationary counterparts, suggesting that observations in a dynamic environment can aid in learning.

3 Conclusion and Future Work

Here, we investigate the impact of dynamic environments on learning predator pursuit policies from partial observations with deep Q-learning. Our preliminary investigation shows that policies learned in dynamic environments can out preform the ones learning in a static environment. This work can be expanded upon in a number of directions, including, but not limited to, testing algorithmic performance with prey that have the ability to learn, leveraging local clustering to aid pursuit, and investigating different deep learning architectures.

[1] Available at Github: Removed for review.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`.

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent Tool Use From Multi-Agent Autocurricula, 2020. URL `http://arxiv.org/abs/1909.07528`.

Dimitri Bertsekas. *Dynamic Programming and Optimal Control: Volume I*. Athena Scientific, 2012.

F. R. K. Chung, Joel E. Cohen, and R. L. Graham. Pursuit—Evasion games on graphs. *Journal of Graph Theory*, 12(2):159–167, 1988. doi: 10.1002/jgt.3190120205. URL `https://onlinelibrary.wiley.com/doi/10.1002/jgt.3190120205`.

Jilles Dibangoye and Olivier Buffet. Learning to Act in Decentralized Partially Observable MDPs. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1233–1242. PMLR, 2018. URL `https://proceedings.mlr.press/v80/dibangoye18a.html`.

Yoav Gottlieb and Tal Shima. UAVs Task and Motion Planning in the Presence of Obstacles and Prioritized Targets. *Sensors*, 15(11):29734–29764, 2015. doi: 10.3390/s151129734. URL `http://www.mdpi.com/1424-8220/15/11/29734`.

Sven Gronauer and Klaus Diepold. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55(2):895–943, 2022. doi: 10.1007/s10462-021-09996-w. URL `https://doi.org/10.1007/s10462-021-09996-w`.

Geoffrey Hollinger, Sanjiv Singh, and Athanasios Kehagias. Improving the Efficiency of Clearing with Multi-agent Teams. *The International Journal of Robotics Research*, 29(8):1088–1105, 2010. doi: 10.1177/0278364910369949. URL `http://journals.sagepub.com/doi/10.1177/0278364910369949`.

Max Jaderberg, Wojciech M. Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C. Rabinowitz, Ari S. Morcos, Avraham Ruderman, Nicolas Sonnerat, Tim Green, Louise Deason, Joel Z. Leibo, David Silver, Demis Hassabis, Koray Kavukcuoglu, and Thore Graepel. Human-level performance in first-person multiplayer games with population-based deep reinforcement learning. *Science*, 364(6443):859–865, 2019. doi: 10.1126/science.aau6249. URL `http://arxiv.org/abs/1807.01281`.

Athanasios Kehagias, Geoffrey Hollinger, and Sanjiv Singh. A graph search algorithm for indoor pursuit/evasion. *Mathematical and Computer Modelling*, 50(9-10):1305–1317, 2009. doi: 10.1016/j.mcm.2009.06.011. URL `https://linkinghub.elsevier.com/retrieve/pii/S0895717709002398`.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Perolat, David Silver, and Thore Graepel. A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning, 2017. URL `http://arxiv.org/abs/1711.00832`.

Joel Z. Leibo, Edward Hughes, Marc Lanctot, and Thore Graepel. Autocurricula and the Emergence of Innovation from Social Interaction: A Manifesto for Multi-Agent Intelligence Research, 2019. URL `http://arxiv.org/abs/1903.00742`.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3):293–321, 1992. doi: 10.1007/BF00992699. URL `https://doi.org/10.1007/BF00992699`.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments, 2020. URL http://arxiv.org/abs/1706.02275.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning, 2013. URL http://arxiv.org/abs/1312.5602.

Shayegan Omidshafiei, Jason Pazis, Christopher Amato, Jonathan P. How, and John Vian. Deep Decentralized Multi-task Multi-Agent Reinforcement Learning under Partial Observability, 2017. URL http://arxiv.org/abs/1703.06182.

OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemys(l)aw D(e)biak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with Large Scale Deep Reinforcement Learning, 2019. URL http://arxiv.org/abs/1912.06680.

T. D. Parsons. Pursuit-evasion in a graph. In A. Dold, B. Eckmann, Yousef Alavi, and Don R. Lick (eds.), *Theory and Applications of Graphs*, volume 642, pp. 426–441. Springer Berlin Heidelberg, 1978. doi: 10.1007/BFb0070400. URL http://link.springer.com/10.1007/BFb0070400.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven Exploration by Self-supervised Prediction, 2017. URL http://arxiv.org/abs/1705.05363.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized Experience Replay, 2016. URL http://arxiv.org/abs/1511.05952.

Sainbayar Sukhbaatar, Zeming Lin, Ilya Kostrikov, Gabriel Synnaeve, Arthur Szlam, and Rob Fergus. Intrinsic Motivation and Automatic Curricula via Asymmetric Self-Play, 2018. URL http://arxiv.org/abs/1703.05407.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-Decomposition Networks For Cooperative Multi-Agent Learning, 2017. URL http://arxiv.org/abs/1706.05296.

Maxwell Svetlik, Matteo Leonetti, Jivko Sinapov, Rishi Shah, Nick Walker, and Peter Stone. Automatic Curriculum Graph Generation for Reinforcement Learning Agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), 2017. doi: 10.1609/aaai.v31i1.10933. URL https://ojs.aaai.org/index.php/AAAI/article/view/10933.

Jaden B. Travnik, Kory W. Mathewson, Richard S. Sutton, and Patrick M. Pilarski. Reactive Reinforcement Learning in Asynchronous Environments. *Frontiers in Robotics and AI*, 5, 2018. URL https://www.frontiersin.org/articles/10.3389/frobt.2018.00079.

Hado van Hasselt, Arthur Guez, and David Silver. Deep Reinforcement Learning with Double Q-learning, 2015. URL http://arxiv.org/abs/1509.06461.

R. Vidal, O. Shakernia, H.J. Kim, D.H. Shim, and S. Sastry. Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *IEEE Transactions on Robotics and Automation*, 18(5):662–669, 2002. doi: 10.1109/TRA.2002.804040. URL http://ieeexplore.ieee.org/document/1067989/.

Gao Wang, Trung V. Phan, Shengkai Li, Jing Wang, Yan Peng, Guo Chen, Junle Qu, Daniel I. Goldman, Simon A. Levin, Kenneth Pienta, Sarah Amend, Robert H. Austin, and Liyu Liu. Robots as models of evolving systems. *Proceedings of the National Academy of Sciences*, 119 (12):e2120019119, 2022. doi: 10.1073/pnas.2120019119. URL https://pnas.org/doi/full/10.1073/pnas.2120019119.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling Network Architectures for Deep Reinforcement Learning, 2016. URL http://arxiv.org/abs/1511.06581.

Chao Yu, Minjie Zhang, Fenghui Ren, and Guozhen Tan. Multiagent Learning of Coordination in Loosely Coupled Multiagent Systems. *IEEE Transactions on Cybernetics*, 45(12):2853–2867, 2015. doi: 10.1109/TCYB.2014.2387277. URL http://ieeexplore.ieee.org/document/7008514/.

Chao Yu, Yinzhao Dong, Yangning Li, and Yatong Chen. Distributed multi-agent deep reinforcement learning for cooperative multi-robot pursuit. *The Journal of Engineering*, 2020(13):499–504, 2020. doi: 10.1049/joe.2019.1200. URL https://onlinelibrary.wiley.com/doi/10.1049/joe.2019.1200.

4 URM STATEMENT

Acknowledgement of meeting the URM criteria.