Phantom - A RL-driven Multi-Agent Framework to Model Complex Systems

Extended Abstract

Leo Ardon
J.P. Morgan AI Research

Jared Vann
J.P. Morgan AI Research

Deepeka Garg
J.P. Morgan AI Research

Thomas Spooner*
Sutter Hill Ventures

Sumitra Ganesh
J.P. Morgan AI Research

ABSTRACT

Agent based modeling (ABM) is a computational approach to modeling complex systems by specifying the behavior of autonomous decision-making components or agents in the system and allowing the system dynamics to emerge from their interactions. Recent advances in the field of Multi-agent reinforcement learning (MARL) have made it feasible to study the equilibrium of complex environments where multiple agents learn simultaneously. However, most ABM frameworks are not RL-native, in that they do not offer concepts and interfaces that are compatible with the use of MARL to learn agent behaviors. In this paper, we introduce a new open-source framework, *Phantom*, to bridge the gap between ABM and MARL. *Phantom* is an RL-driven framework for agent-based modeling of complex multi-agent systems including, but not limited to, economic systems, markets and auctions. The framework aims to provide the tools to simplify the ABM specification in a MARL-compatible way - including features to encode dynamic partial observability, agent utility functions, heterogeneity in agent preferences or types, and constraints on the order in which agents can act (e.g. Stackelberg games, or more complex turn-taking environments). In this paper, we present the main features of *Phantom* and their design rationale.

KEYWORDS

Reinforcement Learning; Agent-Based Model; Multi-Agent; Simulation Framework

ACM Reference Format:
Leo Ardon, Jared Vann, Deepeka Garg, Thomas Spooner, and Sumitra Ganesh. 2023. Phantom - A RL-driven Multi-Agent Framework to Model Complex Systems: Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

Agent based modeling (ABM) is a paradigm to model complex systems in a bottoms-up manner by specifying the behavior of autonomous decision-making components in the system (or agents); and allowing the system dynamics to emerge from their interactions. Drawing upon their real-world counterparts they seek to model, agents assess the state of the world and make decisions that will affect the rest of the system inducing the emergence of non-trivial phenomena.

Recent advances in the field of Reinforcement Learning (RL) have brought another dimension to the study of complex multi-agent systems with the introduction of an autonomous learning component to the ABM paradigm. The Multi-Agent Reinforcement Learning (MARL) research community seeks to study the equilibrium of such non-stationary environments where multiple agents learn at the same time, by playing against or with each other.

Despite being complementary, these two fields of research have progressed in parallel. Most Agent-Based Modeling frameworks are not RL-native, in that they do not offer concepts and interfaces that are compatible with the use of MARL to learn agent behaviors in a specified ABM. We propose *Phantom*, a RL-driven framework for agent-based modeling of complex multi-agent systems to bridge the gap between ABM and MARL.

Phantom provides tools to specify the ABM in MARL-compatible terms - including features to encode dynamic partial observability, agent utility / reward functions, heterogeneity in agent preferences or types, and constraints on the order in which agents can act.

In this paper [2], we elaborate on the architecture and design of the *Phantom* framework and provide details about the main features and their rationale[1].

2 PRINCIPAL FEATURES

2.1 Partial Observability

The agents in an ABM interact by sharing information with each other, that can affect their behavior and eventually lead to uncovering interesting phenomena. However, in many real-world applications not all the information shared across the system is available for all the agents to consume. It was therefore crucial for our framework to support partially observable environments seamlessly and with the guarantee that there will be no information leakage among the agents.

Network Model: In *Phantom*, we model the relationship between agents in the system as a network or graph where each vertex / node represents an agent and each edge represents an open line of communication between two agents. One of our main desiderata for the framework was the ability to support complex and dynamic connectivity patterns between the agents. For this reason, we decided to treat the network component as a first-class citizen of

*Work done while working at J.P. Morgan AI Research

[1] Source Code: https://github.com/jpmorganchase/Phantom

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

the framework and allow the implementation of custom logic to dynamically modify the network topology. The network can be seen as the physical layer on which agents exchange information, which means that two agents will only be able to communicate if an edge exists between the two vertices representing them. This property of the framework turns out to be particularly powerful to express partial observability.

2.2 Heterogeneity and Scalability

Specifying the behaviors of agents in the system, and how they evolve, is one of the crucial tasks in specifying an ABM for a domain and often requires hand-coding of known strategies in classical ABM approaches. While *Phantom* supports taking actions from a hand-crafted (fixed, learnt from data, or evolving) policy, it is also natively geared towards supporting MARL as an approach to train the policies of the agents at scale.

Types and Supertypes: The framework provides a compact way to specify different reward formulations and their associated agent behavior. The agent reward function can be parameterized by a vector of values (*Type*) driving the learnt behavior - which in effect, implies that the agent class is associated with a space of possible reward functions rather than a single fixed one. While the *Type* construct is powerful, it is still difficult to scale for a large number of agents. We argue that it is usually easier to consider families of agents sharing the same average behavior or "persona", as it is often named in the industry. More concretely, *Phantom* allows a compact definition of a family of agents as a distribution over the *Type* parameter space - also referred to as *Supertype* [16].

Shared Policy: The framework also offers a built-in implementation of the *Shared Policy* learning technique [16], that can easily be configured via the framework's API. *Phantom* automatically augments the observation space of an agent with its *Type* parameters for each episode, making it seamless to train policies that generalize across the *Supertype*'s behavior space. It also allows the agents from the same family to share a single policy, considerably limiting the number of models to train.

2.3 Complex Environments

We build upon the standard Open-AI Gym [3] paradigm where a learning agent interacts in discrete time with the rest of the system via the intermediary of a centralized *environment*. The multi-agent setting adds a certain level of complexity to the environment component which now plays the role of orchestrator of the simulation in charge of deciding the order sequence in which the agents act.

Turn-based environment: With multiple agents at play, the complexity of the orchestration logic can rapidly increase making it harder to design complex problems. To alleviate this, *Phantom* provides a simple and modular way to implement complicated sequences of stages where only a subset of the agents act. It uses the *Finite State Machine* formalism to define the order in which the agents are required to execute their actions in the environment.

3 RELATED WORK

Despite having been around since the 70's [14], the notion of Agent Based modeling is an area that really started to grow in the 90's This sudden expansion can in part be attributed to the development of multi-agents frameworks such as *SWARM* [9], *NetLogo* [15] and others, making ABM more accessible to practitioners, reducing the barriers to entry in the field. Since then we have seen numerous applications of ABM in a variety of fields: flow simulation [**?**], markets simulation [10], organizational simulations [12].

These frameworks, built quite some time ago, have helped the research community study complex systems but are not natively geared towards leveraging MARL. *NetLogo* [15] for instance, built on top of Java to allow scalability in the number of agents requires a certain level of engineering skills to master making it less accessible to the AI research community. Although the framework was extended to include basic learning modules [8], it remains ad-hoc and does not natively supports state of the art MARL techniques.

In most recent years, we have seen an increase in the development of RL-frameworks designed for fast code iteration and rapid experimentation. In 2018, *TF-Agents* was created as an additional module to the *TensorFlow* framework to "make implementing, deploying, and testing new Bandits and RL algorithms easier" [5]. The concept of agent is introduced as a core element of the module but the design of complex multi-agent environments and all its subtleties remains at the charge of the developer. Other native ABM frameworks like *ABIDES* [4] were further extended to support RL [1].

MARL frameworks such as *WarpDrive* [6] and *MAVA* [11] are designed to enable easier and more efficient implementation of MARL algorithms. The former innovates by focusing on performance with the use of GPU and their parallelization power. *MAVA* also proposes a new distributed framework for multi-agent. Like *Phantom*, *MAVA* offers the options to specify network configuration to model the agents communication, however unlike in *Phantom* the network configuration remains fairly basic and stays static throughout the simulation and therefore does not allow the study of systems with stochastic connectivity.

Finally, *Abmarl* [13] has a similar objective to ours: to provide a library to build Agent-Based Simulations and train the agents using MARL. They integrate with OpenSpiel [7] to simulate games and provide their own implementation for GridWorld simulations. *Phantom* on the other hand does not focus on specific types of environments, but instead provides generic features to encode a variety of environments.

4 CONCLUSION

In this paper, we introduced a new framework, Phantom, that leverages the power of MARL to automatically learn multiple agent behaviors or policies, evolving in complex systems. Our framework provides the necessary tools to specify the ABM in MARL-compatible terms - including features to encode dynamic partial observability, agent utility / reward functions, heterogeneity in agent preferences (or types), and constraints on the order in which agents can act. Our hope is that Phantom enables the ABM community to conveniently leverage the power of MARL algorithms to learn complex and realistic agent behaviors at scale.

DISCLAIMER

REFERENCES

[1] Selim Amrouni, Aymeric Moulin, Jared Vann, Svitlana Vyetrenko, Tucker Balch, and Manuela Veloso. 2021. ABIDES-Gym: Gym Environments for Multi-Agent Discrete Event Simulation and Application to Financial Markets. *CoRR* abs/2110.14771 (2021). arXiv:2110.14771 https://arxiv.org/abs/2110.14771

[2] Leo Ardon, Jared Vann, Deepeka Garg, Tom Spooner, and Sumitra Ganesh. 2022. Phantom - A RL-driven multi-agent framework to model complex systems. *arXiv* (2022). https://doi.org/10.48550/ARXIV.2210.06012

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. *arXiv* (Jun 2016). https://doi.org/10.48550/arXiv.1606.01540 arXiv:1606.01540 [cs].

[4] David Byrd, Maria Hybinette, and Tucker Hybinette Balch. 2019. ABIDES: Towards High-Fidelity Market Simulation for AI Research. *CoRR* abs/1904.12066 (2019). arXiv:1904.12066 http://arxiv.org/abs/1904.12066

[5] Sergio Guadarrama, Anoop Korattikara, Oscar Ramirez, Pablo Castro, Ethan Holly, Sam Fishman, Ke Wang, Ekaterina Gonina, Neal Wu, Efi Kokiopoulou, Luciano Sbaiz, Jamie Smith, Gábor Bartók, Jesse Berent, Chris Harris, Vincent Vanhoucke, and Eugene Brevdo. 2018. TF-Agents: A library for Reinforcement Learning in TensorFlow. https://github.com/tensorflow/agents

[6] Tian Lan, Sunil Srinivasa, Huan Wang, and Stephan Zheng. 2021. WarpDrive: Extremely Fast End-to-End Deep Multi-Agent Reinforcement Learning on a GPU. https://doi.org/10.48550/ARXIV.2108.13976

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

[8] Stefano Mariani. 1997. NetLogo User Community Models: Slime RL. http://ccl.northwestern.edu/netlogo/models/community/Slime-RL

[9] M. Minar, R. Burkhart, C. Langton, and M. Askenazy. 1996. The Swarm Simulation System:A Toolkit for Building Multi-agent Simulations. http://www.santafe.edu/projects/swarm/overview/overview.html

[10] Richard G. Palmer, W. Brian Arthur, John H. Holland, Blake LeBaron, and Paul Tayler. 1994. Artificial economic life: a simple model of a stockmarket. *Physica D: Nonlinear Phenomena* 75 (1994), 264–274.

[11] Arnu Pretorius, Kale-ab Tessera, Andries P. Smit, Claude Formanek, St John Grimbly, Kevin Eloff, Siphelele Danisa, Lawrence Francis, Jonathan Shock, Herman Kamper, Willie Brink, Herman Engelbrecht, Alexandre Laterre, and Karim Beguir. 2021. Mava: a research framework for distributed multi-agent reinforcement learning. *arXiv* (7 2021). https://doi.org/10.48550/arXiv.2107.01460

[12] Michael J. Prietula, Kathleen M. Carley, and Les Gasser. 1998. Simulating organizations: computational models of institutions and groups.

[13] Edward Rusu and Ruben Glatt. 2021. Abmarl: Connecting Agent-Based Simulations with Multi-Agent Reinforcement Learning. *Journal of Open Source Software* 6, 64 (2021), 3424. https://doi.org/10.21105/joss.03424

[14] Thomas C. Schelling. 1971. Dynamic models of segregation. *The Journal of Mathematical Sociology* 1, 2 (1971), 143–186. https://doi.org/10.1080/0022250X.1971.9989794 arXiv:https://doi.org/10.1080/0022250X.1971.9989794

[15] Seth Tisue and Uri Wilensky. 2004. NetLogo: Design and implementation of a multi-agent modeling environment. In *Proceedings of agent*, Vol. 2004. Springer Cham, Switzerland, 7–9.

[16] Nelson Vadori, Sumitra Ganesh, Prashant Reddy, and Manuela Veloso. 2020. Calibration of shared equilibria in general sum partially observable Markov games. *Advances in Neural Information Processing Systems* 33 (2020), 14118–14128.