Neuro-Symbolic World Models for Adapting to Open World Novelty

Extended Abstract

Jonathan C. Balloch
Zhiyu Lin
Xiangyu Peng
Mustafa Hussain
Aarun Srinivas
Georgia Institute of Technology
USA
balloch@gatech.edu

Robert Wright
Georgia Tech Research Institute
USA
robert.wright@gtri.gatech.edu

Julia M. Kim
Mark O. Riedl
Georgia Institute of Technology
USA
riedl@gatech.edu

ABSTRACT

Most reinforcement learning (RL) methods assume that the world is a closed, fixed process, when in reality most real world problems are *open*, changing over time. To address this, we introduce WorldCloner, an end-to-end trainable neuro-symbolic world model that learns an efficient symbolic model of transitions and uses this world model to improve novelty adaptation. We show that the symbolic world model helps WorldCloner adapt its policy more efficiently than neural-only reinforcement learning methods.

KEYWORDS

Open World Learning, Reinforcement Learning, Neuro-Symbolic

ACM Reference Format:
Jonathan C. Balloch, Zhiyu Lin, Xiangyu Peng, Mustafa Hussain, Aarun Srinivas, Robert Wright, Julia M. Kim, and Mark O. Riedl. 2023. Neuro-Symbolic World Models for Adapting to Open World Novelty: 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.

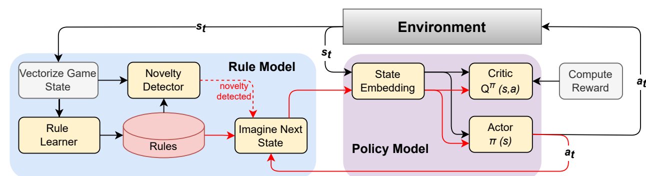

Figure 1: The WorldCloner architecture. Purple box and black arcs: conventional RL execution loop. Blue box: rule model learning and novelty detection. Red arcs: post-novelty environment using rule model "imagined" state transitions.

1 INTRODUCTION

Novelties are sudden, previously unseen changes to dataset, datastream, or environment fundamentals [3, 7, 10]. In sequential decision-making the injection of novelty, after an arbitrary and *a priori* unknown number of episodes or games t, constitutes a transformation from the original environment or MDP M to a new environment or MDP M'. Novelty adaptation is related to *transfer learning* except the adaptation must happen at deployment-time with no expectation of being able to learn the transfer offline. Novelty handling can be broken down into three challenges: novelty detection, novelty characterization, and *novelty adaptation* which is the focus of this work. When starting with some pre-novelty knowledge, attempting to adapt a model to new environments can induce catastrophic inference causing the agent to transfer little, if any, of its previous model. *World model* based reinforcement learners learn both the transition function and the policy together to drive agent performance; DreamerV2 [6] represents the state of the art in world model reinforcement learning. World-model based reinforcement learning offers possible reuse between the model and the behavior policy, but existing state-of-the-art approaches such as Dreamer [6] cannot always update rapidly in the face of sudden change.

To address this, we develop WorldCloner, an efficient *world model* based reinforcement learning system with a neural policy consisting of two online task transfer improvements to the standard deep RL execution loop: (1) A fast-updating symbolic model of the transition function that can be updated with a single post-novelty observation, allowing faster adaptation than neural world models. (2) An *imagination-based adaptation* method that improves the efficiency of deployment-time neural policy adaptation using the updated world model to simulate environment transitions in the post-novelty world. This reduces the number of real environment interactions required to update the neural policy. We build on prior world model research that used imagination to help train standard RL models [5, 8, 12] and multi-agent models [11].

2 WORLD CLONER

WorldCloner is an end-to-end trainable neuro-symbolic world model comprised of two components: (1) a neural policy and (2) a symbolic rule model that approximate the environment's latent transition function. The rule model serves two core functions. First, the rule model learns to predict state transitions pre-novelty. Rule violations thus indicate the introduction of novelty and the need to update the rule model and the policy. Second, once in a post-novelty environment, WorldCloner uses the rule model to simulate the environment, enabling rollouts for retraining the neural policy model

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.

	Adaptive Efficiency @0.95 (steps) ↓	Pre-novelty Performance ↑	Asymptotic Performance ↑	Update Efficiency (policy updates) ↓
DoorKeyChange novelty				
PPO	2.25E6	0.973	0.971	2.25E6
DreamerV2	5.3E5	0.971	0.973	3.82E8
Ours	9.8E5	0.972	0.970	**1.63E6**
LavaProof novelty				
PPO	1.39E5	0.972	0.991	1.39E5
DreamerV2	Failed to adapt	0.965	Failed to adapt	Failed to adapt
Ours	8.3E4	0.972	0.991	**1.38E5**
LavaHurts novelty				
PPO	2.08E6	0.992	0.971	2.08E6
DreamerV2	1.05E6	0.992	0.968	7.56E8
Ours	1.07E6	0.992	0.972	**1.78E6**

Table 1: Novelty metric results averaged over three runs. DreamerV2 did not adapt to the novelty on `LavaProof`.

so as to require fewer interactions with the real environment (see Figure 1). The rule model is independent of the policy implemented as an Advantage Actor-Critic (A2C) [14] neural architecture trained with Proximal-Policy Optimization (PPO) [13].

Interval-Based Symbolic World Model. The symbolic world model, which models the transition function, is represented as a set of rules $\{\rho_1 \ldots \rho_k\}$ of the form $\langle c_s, c_a, e \rangle$ such that c_s is a state precondition, c_a is the action precondition (similar to a do-calculus precondition `do(a)`), and e is an effect. The state preconditions contain a set of values corresponding to a subset of state features $\phi_1 \ldots \phi_m$. When both the state and action preconditions c_s, c_a of a rule ρ_i are satisfied, then it is applicable. Effects e are the difference between the input state and the predicted state: $e = s' - s$. This formulation has similarities to logical calculus frameworks such as ADL and PDDL [9] by encoding preconditions and effects, but our approach is designed to be learned, not engineered, similar to "game rule" learning [4]. To support learnability, preconditions are formulated as a set of *axis-aligned bounding intervals* (AABIs), also known as hyperrectangles or n-orthotopes in feature space that cover the training data. AABIs are d-dimensional convex geometries that define the minimum interval of values for each feature $\phi_1 \ldots \phi_d$.

The rule learning process constructs a compact, collision-free set of AABIs that provide maximum coverage of the state-action space while minimizing the complexity of the symbolic world model. The rule update process is as follows. After an action is taken, the rule learner receives the prior state, the action taken, and a new state. Comparing the prior state, action, and new state with the AABIs, action preconditions, and effects of existing rules, one of the following cases take effect:

(1) *No Change*: The prior state falls inside the AABI of an existing rule with a matching action and effect. (2) *Rule Creation*: There is no rule where the action precondition is satisfied or the state difference matches the effect. A "point" rule is created that exactly describes the prior state. (3) *Rule Relaxation*: A rule exists where the action precondition is satisfied and state difference matches the effect, but the prior state is not covered by the existing rule's state AABI. The rule is "relaxed" by expanding the AABI. (4) *Rule Collision Resolution*: A rule exists where the action precondition and AABI are satisfied but the effect is different. The AABI of the existing rule is split along the min-cut.

Imagination-Based Policy Adaptation. Post-novelty, an updated rule set reflects the agent's belief about the new state transition function. The agent now uses that rule model to "imagine" and update its policy without interacting or executing actions in the true environment. The agent uses the rule model to simulate state-action-state transitions that then populate the agent's update buffer—the data on which the policy will be trained. The policy training algorithm generates a loss over samples drawn from the update buffer and back-propagates loss through the policy model (Figure 1, red paths). The agent follows its policy in the imagined environment and repeatedly experiences the first rule change's consequences, receiving a reduced (or increased) expected reward, pushing the policy away from (or toward) the impacted actions. To ensure that the agent doesn't overfit to a rule model that is not completely accurate, we periodically sample state-action transitions from the real environment. We use imagination to generate 40% of state-action-state training samples. See expanded details in [1].

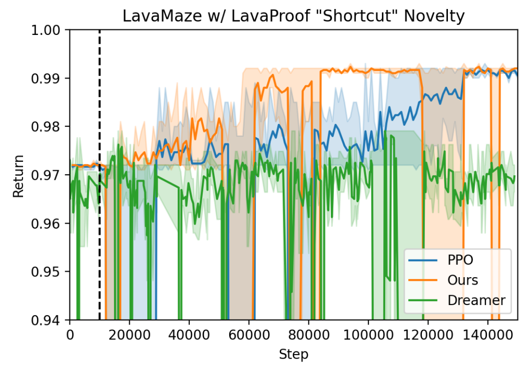

Figure 2: The adaptive performance post-novelty for the `LavaProof` "shortcut" novelty.

3 EXPERIMENTS

Experiments are performed in the NovGrid [2] environment using two 8x8 Minigrid environments as the base environments: (1) `DoorKey` a standard environment where an agent must pick up a key, unlock a door, and navigate to the goal behind that door, and (2) `LavaShortcutMaze`, a custom environment where an agent must navigate a maze that has a pool of lava lining the side of the maze nearest to the goal. Performance of our method and the baselines was evaluated on three novelty types from [2]: `LavaProof` which that makes harmful lava harmless, `DoorKeyChange` which changes the key that unlocks a door, and `LavaHurts` which makes harmless lava harmful (the inverse of `LavaProof`).

Table 1 shows that pre-novelty, as expected, all three methods converge in all three novelty scenarios to effectively the same performance. For the `DoorKeyChange` novelty, DreamerV2 slightly outperforms WorldCloner in adaptive efficiency, but WorldCloner is much more efficient in terms of environment interactions. In the `LavaProof` novelty condition, in which the agent must detect that the novelty results in a "shortcut", DreamerV2 fails to adapt to the novelty. This is illustrated in Figure 2. We attribute DreamerV2's failure to the unique way in which its policy learner depends on the accuracy of its world model, which leads to overfitting.

4 ACKNOWLEDGEMENTS

This research is sponsored in part by the Defense Advanced Research Projects Agency (DARPA), under contract number W911NF-20-2-0008. Views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of the US Department of Defense or the United States Government.

REFERENCES

[1] Jonathan Balloch, Zhiyu Lin, Robert Wright, Xiangyu Peng, Mustafa Hussain, Aarun Srinivas, Julia Kim, and Mark O. Riedl. 2023. Neuro-Symbolic World Models for Adapting to Open World Novelty. https://doi.org/10.48550/ARXIV.2301.06294

[2] Jonathan C Balloch, Zhiyu Lin, Mustafa Hussain, Aarun Srinivas, Xiangyu Peng, Julia Kim, and Mark Riedl. 2022. NovGrid: A Flexible Grid World for Evaluating Agent Response to Novelty. In *In Proceedings of AAAI Symposium, Designing Artificial Intelligence for Open Worlds*. aaai, Palo Alto, CA, 7.

[3] Terrance Boult, Przemyslaw Grabowicz, Derek Prijatelj, Roni Stern, Lawrence Holder, Joshua Alspector, Mohsen M. Jafarzadeh, Toqueer Ahmad, Akshay Dhamija, Chunchun Li, Steve Cruz, Abhinav Shrivastava, Carl Vondrick, and Scheirer Walter. 2021. Towards a Unifying Framework for Formal Theories of Novelty. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 17 (May 2021), 15047–15052. https://doi.org/10.1609/aaai.v35i17.17766

[4] Matthew Guzdial, Boyang Li, and Mark O. Riedl. 2017. Game Engine Learning from Video. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. ijcai, Melbourne, Australia, 3707–3713. https://doi.org/10.24963/ijcai.2017/518

[5] David Ha and Jürgen Schmidhuber. 2018. Recurrent World Models Facilitate Policy Evolution. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 2455–2467.

[6] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. 2021. Mastering Atari with Discrete World Models. In *International Conference on Learning Representations*. OpenReview.net, Vienna, Austria, 26. https://openreview.net/forum?id=0oabwyZbOu

[7] Pat Langley. 2020. Open-World Learning for Radically Autonomous Agents. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 09 (Apr. 2020), 13539–13543. https://doi.org/10.1609/aaai.v34i09.7078

[8] Zhixuan Lin, Yi-Fu Wu, Skand Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. 2020. Improving Generative Imagination in Object-Centric World Models. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. JMLR.org, Vienna, Austria, Article 570, 10 pages.

[9] Drew McDermott. 2000. The 1998 AI Planning Systems Competition. *AI Mag.* 21 (2000), 35–55.

[10] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. 2014. A review of novelty detection. *Signal Processing* 99 (2014), 215–249.

[11] Arnu Pretorius, Scott Cameron, Andries Petrus Smit, Elan van Biljon, Lawrence Francis, Femi Azeez, Alexandre Laterre, and Karim Beguir. 2020. Learning to communicate through imagination with model-based deep multi-agent reinforcement learning.

[12] Sébastien Racanière, Théophane Weber, David P. Reichert, Lars Buesing, Arthur Guez, Danilo Rezende, Adria Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, Razvan Pascanu, Peter Battaglia, Demis Hassabis, David Silver, and Daan Wierstra. 2017. Imagination-Augmented Agents for Deep Reinforcement Learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 5694–5705.

[13] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. https://doi.org/10.48550/ARXIV.1707.06347

[14] Lucas Willems. 2020. lcswillems/torch-ac: Recurrent and multi-process PyTorch implementation of deep reinforcement Actor-Critic algorithms A2C and PPO. https://github.com/lcswillems/torch-ac