Escape Room: A Configurable Testbed for Hierarchical Reinforcement Learning

Extended Abstract

Jacob Menashe and Peter Stone
{jmenashe,pstone}@cs.utexas.edu
The University of Texas at Austin
Austin, TX USA

ABSTRACT

Recent successes in Reinforcement Learning have encouraged a fast-growing network of RL researchers and a number of breakthroughs in RL research. As the RL community and body of work grows, so does the need for widely applicable benchmarks that can fairly and effectively evaluate a variety of RL algorithms.

In this paper we present the Escape Room Domain (ERD), a new flexible, scalable, and fully implemented testing domain for Hierarchical RL that bridges the "moderate complexity" gap left behind by existing alternatives. ERD is open-source and freely available through GitHub, and conforms to widely-used public testing interfaces for simple integration and testing with a variety of public RL agent implementations.

KEYWORDS

Reinforcement Learning; Simulation Techniques, Tools, and Platforms

ACM Reference Format:
Jacob Menashe and Peter Stone {jmenashe,pstone}@cs.utexas.edu The University of Texas at Austin Austin, TX USA . 2019. Escape Room: A Configurable Testbed for Hierarchical Reinforcement Learning. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Reinforcement Learning (RL) is an AI paradigm in which an artificial agent explores and exploits its environment through an action-observation loop. RL has seen increasing success in recent years as it has been integrated with parallel developments in AI such as Deep Learning. These successes, however, have brought to light new problems with replication and reproducibility, inspiring researchers to focus attention on the challenge of accurately gauging the efficacy of new approaches [4].

The proliferation of public source code hosting and sharing technologies has created an unprecedented potential for peer evaluation and collaboration in the applied Computer Science disciplines such as AI. Despite the availability of these tools, however, it is still common for works to exclude details and source code that are necessary for reproducibility [7]. It is reasonable to assume that when testing frameworks are open and available, researchers will be more likely to use them and will be more likely to report results that can be easily reproduced, evaluated, and integrated with parallel work.

Hierarchical Reinforcement Learning (HRL) is one specification of RL in which evaluation and reproducibility are hindered by a lack of suitable test domains. Hierarchies provide an efficient means of breaking monolithic tasks down into manageable interdependent chunks, and for this reason hierarchical learning has been an active area of RL research since RL's inception. And yet, the hierarchical domains frequently used for evaluation in the literature are generally either too simplistic or too rigid to fully evaluate the capabilities of an HRL algorithm.

In this work we present a new HRL testing framework which we call the Escape Room Domain (ERD), named after the Escape Room genre popular in both virtual- and real-world puzzle-based challenges. The ERD is a parameterized schematic for generating individual instances, where each instance is a single testbed upon which an HRL agent can be evaluated and compared against alternative HRL algorithms. The implementation is built on the OpenAI Gym [1] testbed interface using the open-source Panda3D gaming engine [2], enabling simple and fully customizable integration with existing software.

2 THE ESCAPE ROOM DOMAIN

The Escape Room Domain (ERD) is based on a series of popular video games and real-life team-building exercises. An *Escape Room* is a game consisting of an enclosed space and a series of puzzles. In order to "escape", the agent(s) inside the room must solve the available puzzles in order to unlock the exit. Often these puzzles are arranged in some sequential order and are combined with a series of clues to guide the agent toward a solution.

The ERD reconstructs the popular notions of an Escape Room as an RL testbed, and allows RL agents to solve virtual Escape Rooms using the infrastructure we present in this paper. Each ERD instance consists of a room with a single exit and a predefined puzzle that the RL agent must solve prior to exiting the room. The agent's internal state (e.g. joint configurations) and external state (e.g. world position) are concatenated into a single state vector that can be manipulated through a set of discrete actions. In the next section we describe the specific state and action spaces, as well as the transition and reward functions, which together comprise the MDP of this testbed domain.

2.1 The Escape Room MDP

Our first step in describing the ERD implementation then is to describe its theoretical foundation in the language of an MDP. It

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

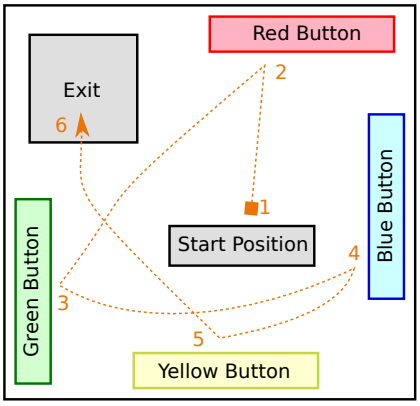

Figure 2.1: A bird's-eye view of the layout used for the ERD with embedded Button Puzzle.

is important to note that because the ERD is a "flexible" domain, it is more accurately described as a domain *schematic* than a single, static domain in itself. An *instance* of ERD may be defined based on the specific criteria and parameters laid out below; any ERD instance is therefore comprised of the following dimensions:

(1) The agent's 6-dimensional pose consisting of both 3D position (X, Y, and Z) and 3D orientation (Heading, Pitch, and Roll). These state dimensions are always continuous.

(2) A set of N_p puzzle-specific dimensions that describe the state of the puzzle (see Section 2.2) that must be solved in order to unlock the room's exit. These state dimensions may be either continuous or discrete.

(3) A set of N_j 1-DoF joint positions that describe the joints on the agent's virtual robotic arm. The arm may (optionally) be used by the agent to interact with the puzzle specific to its ERD instance.

Based on these definitions any ERD instance must have at least 6 state dimensions, however there are no restrictions on the maximum number of dimensions. Specifics related to the sizes of arm links, orientation of arm joints, or the transition dynamics of an embedded puzzle are all determined at the discretion of the researcher who has designed that particular ERD instance.

The Actions of the ERD are designed to be easily mapped to a real-world robot agent (e.g. a robot arm) with an auxiliary movement controller, and thus no explicit actions exist that are tied to a room's embedded puzzle. Instead, actions only affect the robot's position and its joint positions. The actions are defined as follows:

- Move Forward/Move Back 1 Meter
- Turn Left/Right 10 Degrees
- Strafe Left/Right 1 Meter
- Increment/Decrement Joint 10 Degress

The transition function is governed primarily by the puzzle embedded in each specific ERD instance, however movement and actuation actions each invoke the expected state transitions. For example, if the agent executes "Move Forward", it will be displaced by approximately 1 meter in whatever direction it is facing.

The reward function of ERD is intentionally simple. Whenever the agent takes an action it incurs a reward of -1. When the agent successfully exits the room, it earns a reward of 100. The purpose of using a sparse reward function is to encourage general-purpose algorithms that require minimal expert configuration prior to deployment.

2.2 Puzzles

Each ERD instance contains an embedded puzzle that integrates with the physical makeup of the room. The puzzle can be any physics-based challenge that must be solved prior to exiting the room. Figure 2.1 shows how we embedded a Button Puzzle where the buttons must be pressed in a specific order before the room can be exited. However, this Button Puzzle can be swapped out for any other puzzle that a researcher (or domain designer) wishes. In this way, the puzzle embedded in a given ERD instance dictates the overall difficulty of the instance's MDP.

Ideally, each puzzle is further customizable to a minor degree. For example, in the Button Puzzle the locations and sizes of the buttons can be adjusted in order to randomize the room and prevent overfitting. One can see how the puzzle might be extended to add more buttons, or even a variety of interactive controls to complexify the MDP and provide greater challenge and randomization to the agent.

Our public ERD implementation includes a puzzle we we refer to as the *Button Puzzle*. Although the name is new, the Button Puzzle has been used in a variety of HRL publications under different identities, such as the BitFlip Domain [3], the LightBox Domain [8], the Random Lights Domain [6], and Randomly Generated Factored Domains [5]. In each instance, the domain consists of binary variables that are causally related to one another. Only one variable can be modified at a time, and so the agent benefits greatly from learning an action hierarchy that represents the causal structure of the different variables in the domain. As shown by [6], complex versions of the Button Puzzle can be fully modeled and solved by intrinsically-motivated agents in fewer than 20,000 timesteps.

In our implementation there are up to 4 buttons that may be causally related to one another according to any Directed Acyclic Graph, meaning that a button can only be switched "on" after all of its dependent buttons have been switched "on" as well. Therefore an agent that can deduce the causal structure of the buttons can quickly determine the correct order in which to toggle them. One button is designed as the unknown goal, and when this button is toggled the agent is free to exit the room.

3 CONCLUSION

In this paper we have presented the Escape Room Domain, a new testbed domain designed for advancing RL research. We have demonstrated that the need for such testbeds exists, both as a means of bridging the "difficulty gap" left behind by existing testbeds, as well as to provide an available, flexible framework for comparison between a wide array of RL algorithms. The ERD is built on open source software and is freely available on our public GitHub repository[1].

[1] https://github.com/LARG/escape_room

REFERENCES

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. (2016). arXiv:arXiv:1606.01540

[2] Carnegie Mellon University. 2010. Panda3D - Free 3D Game Engine. (2010). http://panda3d.org

[3] Carlos Diuk, Alexander L Strehl, and Michael L Littman. 2006. A hierarchical approach to efficient reinforcement learning in deterministic domains. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. ACM, 313–319.

[4] Odd Erik Gundersen and Sigbjørn Kjensmo. 2017. State of the art: Reproducibility in artificial intelligence. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence and the Twenty-Eighth Innovative Applications of Artificial Intelligence Conference*.

[5] Assaf Hallak, Francois Schnitzler, Timothy Mann, and Shie Mannor. 2015. Off-policy model-based learning under unknown factored dynamics. In *International Conference on Machine Learning*. 711–719.

[6] Jacob Menashe and Peter Stone. 2015. Monte Carlo Hierarchical Model Learning. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. 771–779.

[7] Victoria Stodden, Peixuan Guo, and Zhaokun Ma. 2013. Toward reproducible computational research: an empirical analysis of data and code policy adoption by journals. *PloS one* 8, 6 (2013), e67111.

[8] Christopher M. Vigorito and Andrew G. Barto. 2010. Intrinsically motivated hierarchical skill learning in structured environments. *IEEE Transactions on Autonomous Mental Development* 2, 2 (jun 2010), 132–143. https://doi.org/10.1109/tamd.2010.2050205