

# Learning Simulator: A simulation software for animal and human learning

Markus Jonsson<sup>1</sup>, Stefano Ghirlanda<sup>1, 2</sup>, Johan Lind<sup>1</sup>, Vera Vinken<sup>3</sup>, and Magnus Enquist<sup>1, 4</sup>

**1** Centre for Cultural Evolution, Stockholm University, Stockholm, Sweden **2** Department of Psychology, Brooklyn College and Graduate Center, CUNY, New York, NY, USA **3** Biosciences Institute, Newcastle University, Newcastle upon Tyne, United Kingdom **4** Department of Zoology, Stockholm University, Sweden

DOI: [10.21105/joss.02891](https://doi.org/10.21105/joss.02891)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Olivia Guest](#) ↗

## Reviewers:

- [@themkar](#)
- [@CatarauCorina](#)

Submitted: 04 December 2020

Published: 24 February 2021

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

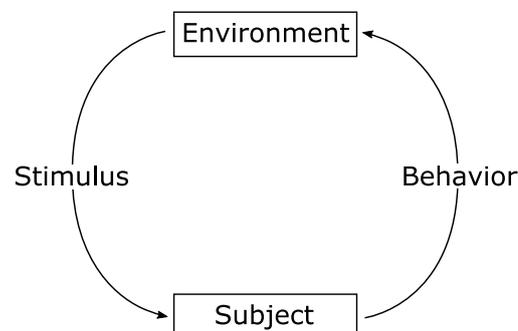
## Summary

Learning Simulator is a software for simulating learning in animals and humans, as studied for example in experimental psychology and animal cognition research. It is primarily intended for computational and behavioral biologists, ethologists, and psychologists, and it can also be useful to students and teachers in these fields.

## Introduction

Learning Simulator was developed to study learning in animals and humans. The current version implements associative learning (AL) and reinforcement learning (RL) algorithms, apt to study instrumental (operant) learning and Pavlovian (classical) learning ([Bouton, 2016](#); [Pearce, 2013](#)), including in complex situations such as social learning or maze learning. A plugin system to add more learning mechanisms is planned for a future version.

The simulator uses a common framing of learning that comprises a subject interacting with an environment. The environment presents a stimulus to the subject, and the subject responds with a behavior. As a result, the environment presents the next stimulus that the subject responds to, and so on. See [Figure 1](#).



**Figure 1:** The subject and the world can be seen as two interacting dynamical systems, where the state variables in the subject determine the probabilities for its behaviors (the subject's output), and each behavior from the subject puts the environment in a state that determines (probabilistically) its output stimulus.

The stimuli that the environment presents and the behaviors that the subject can exhibit are pre-defined by the user of the program. Each stimulus is given a reinforcement value (corresponding to genetically determined values in biological organisms). A stimulus such as food would typically have a positive value, while the perception of harm to the body would have a negative value.

As seen in [Figure 1](#), the consequence of responding with a behavior (say  $B$ ) to a stimulus (say  $S$ ) is that the subject meets the next stimulus (say  $S'$ ):

$$S \rightarrow B \rightarrow S'$$

Learning algorithms can then use the reinforcement value of  $S'$  as an indication of the quality of the response  $B$  to  $S$ . Specifically, this can be accomplished by updating one or more of the subject's memory state variables. These state variables control the probabilities of future responses: if the response  $B$  to stimulus  $S$  leads to a reward (a stimulus with high reinforcement value), the subject will be more likely to respond with  $B$  the next time it faces  $S$ .

The user of Learning Simulator specifies in a text script how the output stimulus from the environment depends on the subject's response to the previous stimulus. This script also specifies the values of all parameters used in the learning process. The simulation script, written in a simple and well-documented scripting language, is the only input to Learning Simulator, facilitating reproducible workflows. The script also specifies how to visualize the simulation data, for example how a memory state variable changes during the simulation. Learning Simulator can also export results to CSV files.

More information is available at <https://www.learningsimulator.org>.

## Applications of associative learning

Associative learning theory has a rich tradition of computational modeling, which has recently converged with RL learning models from machine learning ([Sutton & Barto, 2018](#)). During the last decade or so, AL and RL algorithms have proven increasingly powerful.

Firstly, when applied to deep neural networks, RL has been used to teach computers to find optimal play and achieve human-level skills in chess ([Silver et al., 2017](#)) and the Chinese board game Go ([Silver et al., 2016](#)).

Secondly, behaving optimally (or near-optimally) is central to animals' adaptation to their environment. AL and RL can provide explanations for a wide range of learning phenomena in both humans and animals ([Dayan & Niv, 2008](#); [Enquist et al., 2016](#); [Ghirlanda et al., 2020](#); [Niv, 2009](#)). Indeed, investigating predictions about animal and human behavior is the main motivation behind Learning Simulator, with the goal to understand biological learning and its numerous applications to animal welfare and training ([McGreevy & Boakes, 2011](#)), and to human health ([Bernstein, 1999](#); [Haselgrove & Hogarth, 2013](#); [Schachtman & Reilly, 2011](#)).

## Statement of need

As a result of a strong theoretical tradition, and of the many application areas, there are now many AL and RL models with varying properties and varying predictive power in different environments.

This situation has created a need for a general simulation software offering a convenient way to investigate different learning models. The first aim of our software is to fulfil this need.

The second aim is to provide a generic, flexible way to describe unambiguously the design of learning experiments or, more generally, the features of any learning scenario.

The main advantage of Learning Simulator is its simple scripting language that provides a way to explore and understand learning mechanisms in a multitude of learning scenarios, and investigate the effects of varying their underlying parameters.

A further strength is the simplicity with which even complex learning environments can be specified, such as psychological experiments with complex dependencies between stimuli and behaviors. The scripting language has been developed to be useful to any researcher of learning phenomena – not necessarily computer programmers.

Our software has been used in scientific publications ([Ghirlanda et al., 2020](#); [Lind, 2018](#); [Lind et al., 2019](#)) as well as in teaching at the Ethology Master's Programme at Stockholm University, at the Veterinary Programme at the Swedish University of Agricultural Sciences, and at the Department of Psychology of Brooklyn College, City University of New York.

An open source license and an accessible scripting language enable further scientific exploration of learning phenomena by students and experts alike within the fields of biology, ethology, psychology, and neuroscience.

## State of the field

Other simulating software either specialize in one specific mechanism ([Schultheis et al., 2008a, 2008b](#); [Alonso et al., 2012](#)) or only include models of classical conditioning ([George, 2019](#); [Harris & Livesey, 2010](#); [Thorwart et al., 2009](#)), or provide only one learning algorithm or environment (*Q-Learning Simulator, n.d.*).

Learning Simulator provides a common interface to several AL and RL algorithms:

- *Stimulus-response learning* ([Bush & Mosteller, 1951](#)),
- *Rescorla-Wagner* ([Rescorla & Wagner, 1972](#)),
- *Actor-critic* ([Sutton & Barto, 2018](#); [Witten, 1977](#)),
- *Q-learning* ([Watkins, 1989](#); [Watkins & Dayan, 1992](#)),
- *Expected SARSA* ([Van Seijen et al., 2009](#)), and
- *A-learning* ([Ghirlanda et al., 2020](#))

facilitating direct comparison of these mechanisms.

## Repository

The program is written in Python and its source code repository is hosted on GitHub. Its documentation is hosted on Read the Docs. The repository incorporates continuous integration with automatic build of the documentation, linting, and execution of a test suite along with code coverage measurement.

## Acknowledgements

Research and development was supported by the Knut and Alice Wallenberg Foundation (2015.0005).

## References

- Alonso, E., Mondragón, E., & Fernández, A. (2012). A Java simulator of Rescorla and Wagner's prediction error model and configural cue extensions. *Computer Methods and Programs in Biomedicine*, 108(1), 346–355. <https://doi.org/10.1016/j.cmpb.2012.02.004>
- Bernstein, I. L. (1999). Taste aversion learning: A contemporary perspective. *Nutrition*, 15(3), 229–234. [https://doi.org/10.1016/s0899-9007\(98\)00192-0](https://doi.org/10.1016/s0899-9007(98)00192-0)
- Bouton, M. (2016). *Learning and behavior: A modern synthesis* (2nd ed.). Sinauer.
- Bush, R. R., & Mosteller, F. (1951). A mathematical model for simple learning. *Psychological Review*, 58(5), 313. <https://doi.org/10.1037/h0054388>
- Dayan, P., & Niv, Y. (2008). Reinforcement learning: The good, the bad and the ugly. *Current Opinion in Neurobiology*, 18(2), 185–196.
- Enquist, M., Lind, J., & Ghirlanda, S. (2016). The power of associative learning and the ontogeny of optimal behaviour. *Royal Society Open Science*, 3(11), 160734. <https://doi.org/10.1098/rsos.160734>
- George, D. N. (2019). learnSim. In *GitHub repository*. GitHub. <https://github.com/DavidNGeorge/learnSim>
- Ghirlanda, S., Lind, J., & Enquist, M. (2020). A-learning: A new formulation of associative learning theory. *Psychonomic Bulletin & Review*, 1–29. <https://doi.org/10.3758/s13423-020-01749-0>
- Harris, J. A., & Livesey, E. J. (2010). An attention-modulated associative network. *Learning & Behavior*, 38(1), 1–26. <https://doi.org/10.3758/lb.38.1.1>
- Haselgrove, M., & Hogarth, L. (2013). *Clinical applications of learning theory*. Psychology Press. <https://doi.org/10.4324/9780203803509>
- Lind, J. (2018). What can associative learning do for planning? *Royal Society Open Science*, 5(11), 180778. <https://doi.org/10.1098/rsos.180778>
- Lind, J., Ghirlanda, S., & Enquist, M. (2019). Social learning through associative processes: A computational theory. *Royal Society Open Science*, 6(3), 181777. <https://doi.org/10.1101/446906>
- McGreevy, P., & Boakes, R. (2011). *Carrots and sticks: Principles of animal training*. Dartington Press. <https://doi.org/10.1111/j.1751-0813.2010.00653.x>
- Niv, Y. (2009). Reinforcement learning in the brain. *Journal of Mathematical Psychology*, 53(3), 139–154.
- Pearce, J. M. (2013). *Animal learning and cognition: An introduction*. Psychology Press.
- Q-Learning Simulator*. (n.d.). <https://www.mladdict.com/q-learning-simulator>.
- Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement [Incollection]. In A. H. Black & W. F. Prokasy (Eds.), *Classical conditioning: Current research and theory* (pp. 64–99). Appleton-Century-Crofts.
- Schachtman, T. R., & Reilly, S. S. (2011). *Associative learning and conditioning theory: Human and non-human applications*. OUP USA.
- Schultheis, H., Thorwart, A., & Lachnit, H. (2008a). HMS: A MATLAB simulator of the Harris model of associative learning. *Behavior Research Methods*, 40(2), 442–449. <https://doi.org/10.3758/brm.40.2.442>

- Schultheis, H., Thorwart, A., & Lachnit, H. (2008b). Rapid-REM: A MATLAB simulator of the replaced-elements model. *Behavior Research Methods*, 40(2), 435–441. <https://doi.org/10.3758/brm.40.2.435>
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., & others. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587), 484–489.
- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., & others. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv Preprint arXiv:1712.01815*.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- Thorwart, A., Schultheis, H., König, S., & Lachnit, H. (2009). ALTSim: A MATLAB simulator for current associative learning theories. *Behavior Research Methods*, 41(1), 29–34. <https://doi.org/10.3758/brm.41.1.29>
- Van Seijen, H., Van Hasselt, H., Whiteson, S., & Wiering, M. (2009). A theoretical and empirical analysis of Expected Sarsa. *2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 177–184. <https://doi.org/10.1109/adprl.2009.4927542>
- Watkins, C. J. C. H. (1989). *Learning from delayed rewards* [PhD thesis]. King's College, Cambridge.
- Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3-4), 279–292.
- Witten, I. H. (1977). An adaptive optimal controller for discrete-time Markov environments. *Information and Control*, 34(4), 286–295. [https://doi.org/10.1016/s0019-9958\(77\)90354-0](https://doi.org/10.1016/s0019-9958(77)90354-0)